

Современные подходы к построению методов глобальной оптимизации

Орлянская И.В. (myller@orc.ru)

МГУ, факультет ВМиК

1. Введение

Оптимизация в широком смысле слова находит применение в науке, технике и в любой другой области человеческой деятельности. Это могут быть задачи проектирования, задачи распределения ограниченных ресурсов, задачи расчета траектории полета ракеты и т.п. Подобные задачи часто встречаются в ряде прикладных областей при моделировании реальных процессов: теории управления, моделировании физических явлений, анализе данных и других областях, словом, везде, где необходимо получить наилучший результат целевой функции на множестве некоторых ограничений.

Без ограничения общности, рассмотрим задачу глобальной минимизации:

$$F(x) \rightarrow \min_{x \in X},$$

где многоэкстремальная функция цели $F(\cdot)$ является невыпуклой и для различных алгоритмов имеет различные свойства гладкости, $X \subset \mathbf{R}_k$ – допустимое множество задачи, \mathbf{R}_k – k -мерное Евклидово пространство.

Множества

$$X^{opt} = \mathop{Arg \min}_{x \in X} F(x),$$

$$X^{lopt} = \{x \in X \mid \exists \varepsilon > 0 : F(x) \leq F(x'), \forall x' \in B_\varepsilon(x) \cap X\}$$

множество оптимальных решений и множество локально-оптимальных решений соответственно и $B_\varepsilon(x) = \{x' \in \mathbf{R}_k \mid |x' - x| < \varepsilon\}$. Таким образом, задача глобальной оптимизации состоит в отыскании точки $x \in X^{opt}$, причем в случае, когда $X^{lopt} \neq X^{opt}$.

В силу широты класса многоэкстремальных функций задача глобальной оптимизации в общем случае является неразрешимой, т.е. нельзя гарантировать, что решение задачи будет получено за конечное число шагов. Даже некоторые из разрешимых задач могут попасть в класс неразрешимых, т.к. число шагов, необходимых для получения решения, может быть чрезмерно большим. Для того чтобы решить задачу, помимо непрерывности функции цели, необходимы некоторые дополнительные условия ее гладкости. Таким образом, специфика задачи глобальной оптимизации заключается в многоэкстремальности функции цели и неразрешимости в общем случае.

Существует два направления в работе с неразрешимыми задачами. Первый способ – когда формулируются а priori известные условия, наложенные на функцию цели и допустимое множество задачи, которые приводят исходную задачу к разрешимому виду или, по крайней мере, дают возможность с уверенностью говорить о том, что точное решение будет найдено. Это сужает рассматриваемый класс функций. Второй подход позволяет рассматривать более широкий класс функций цели: отказываясь от требования разрешимости задачи, необходимо получить оценку глобального решения. При таком подходе желательно иметь также некоторый критерий приемлемости полученной оценки.

В настоящей работе основное внимание уделяется «существенно» безусловным задачам глобальной оптимизации (essentially unconstrained global optimization problems), т.е. задачам, в которых глобальное решение достигается во внутренней точке допустимого множества [20], [36], [41], [46], [50]. В связи с этим предполагается простая структура допустимого множества, обычно это многомерный куб, без ограничения общности, единичный $X = \bigotimes_{n=1}^k [0,1]$.

Для получения точного решения «существенно» безусловных задач глобальной оптимизации было предложено множество методов. Точные методы либо полагаются на априорную информацию о том, насколько быстро изменяется функция (т.е. необходима информация о константе Липшица функции), либо требуют аналитически сформулировать функцию цели (например, в методе интервалов). Статистические методы, как правило, используют технику разбиения, чтобы разделить область поиска, но такие методы также пользуются априорной информацией или некоторыми предположениями о том, как функция цели может быть промоделирована. Основное предположение заключается в том, что каждая конкретная целевая функция принадлежит классу функций, которые промоделированы конкретной стохастической функцией. Информация из предыдущей выборки целевой функции может быть использована для оценки параметров стохастической функции, и эта усовершенствованная модель впоследствии может быть использована для смещения при отборе точек в исследуемой области.

Развитие методов глобальной оптимизации стимулируется не только актуальностью или сложностью этих задач, но и развитием электронно-вычислительных средств. В настоящее время параллельные и векторные суперкомпьютеры рассматриваются как один из основных инструментов для проведения исследований в различных научных и прикладных дисциплинах. Всего лишь несколько лет назад параллелизм рассматривался как редкая и экзотическая область вычислений, интересная, но не предназначенная для «простых» пользователей. Анализ тенденций развития прикладных приложений, архитектур вычислительных систем, сетей показывает, что эта точка зрения уже неверна. Параллельные компьютеры перестали быть редкостью, и последние исследования в оптимизации принимают во внимание архитектурные особенности современных компьютеров, на которых эти алгоритмы предполагается реализовать. Проблемам параллельных вычислений в задачах глобальной оптимизации, а также различным подходам к распараллеливанию алгоритмов глобальной оптимизации в настоящее время придается большое значение [6], [8], [18], [28], [29], [39], [40], [48].

Одновременно с возникающими задачами и потребностями в их решении ведутся работы по разработке новых и усовершенствованию существующих специальных

инструментов, позволяющих решать эти проблемы, используя весь накопленный опыт и современные вычислительные средства. Проблемам глобальной оптимизации, разработке новых методов и другим важным исследованиям в глобальной оптимизации посвящено немало работ. Вот некоторые сведения о публикациях по глобальной оптимизации: первые монографии опубликованы в [25], [38], [32], [41]; первые книги со статьями – [11], [12]; с 1991 выходит собственный журнал *Journal of Global Optimization*; подробное руководство – [20], [30], [40], [42]; Интернет-сайт по проблемам глобальной оптимизации – <http://www.mat.univie.ac.at/~neum/glopt.html>.

2. Свойства методов глобальной оптимизации

Процедуру (или алгоритм) решения задачи оптимизации можно представить в виде итеративного процесса, который порождает последовательность точек в соответствии с предписанным набором правил, включающим критерий окончания счета. Обычно, глобальное решение задачи оптимизации предлагается найти, перебрав все ее локальные решения. Такая задача, как правило, оказывается трудоемкой. Другой подход – перебрать часть локальных решений и показать, что оставшиеся локальные минимумы не влияют на точность решения. Таким образом, идея всех методов глобальной оптимизации – оценить значения целевой функции $F(\cdot)$ на некотором множестве точек x_1, \dots, x_N из допустимого множества X , и различие методов заключается в способах выбора этих точек.

Поскольку мы не знаем, где именно в X можно найти глобальные минимумы, то необходимо применить некоторую стратегию, чтобы «разбросать» эти точки по множеству X . Любую такую стратегию называют глобальной техникой (global technique). Далее, вполне возможно, что в окрестности выбранной точки x_n существует лучшее значение функции цели $F(\cdot)$, чем $F(x_n)$. Для получения лучшего значения функции $F(\cdot)$ к полученной точке x_n применяют алгоритм локального спуска. Метод, осуществляющий такой локальный спуск, называют локальной техникой (local technique). Почти все методы глобальной оптимизации используют методы локальной оптимизации, по крайней мере, для того чтобы улучшить уже найденную оценку глобального решения, поэтому независимо от применяемой глобальной техники, использование локальной техники является важной частью любого метода глобальной оптимизации.

Другим важным свойством алгоритма оптимизации является сходимость генерируемой им последовательности точек к глобальному оптимальному решению. Однако в большинстве случаев получаются менее благоприятные результаты: невыпуклость функций, большая размерность задачи или другие трудности вынуждают останавливать алгоритм, если получена точка, принадлежащая некоторому множеству приближенных решений. Другими словами, для любого численного алгоритма необходимы условия остановки. Это ключевой момент, т.к. без какой-либо дополнительной информации или предположений о задаче невозможно сделать выводы о точности решения, полученного за некоторое фиксированное число шагов алгоритма. Таким образом, для четкого определения условий остановки алгоритма необходима дополнительная информация или предположения, или же условия остановки должны быть эвристическими. Это подтверждает высказанный ранее факт, что задача глобальной оптимизации в общем случае неразрешима, и мы должны быть готовы принять полученное с помощью численного метода

приближение за решение задачи. Таким образом, получить численное решение задачи глобальной оптимизации означает получить некую точку $x \in X^{opt}(\varepsilon) = \{x \in X \mid F(x) \leq F_{opt} + \varepsilon\}$, где $F_{opt} = \min_{x \in X} F(x)$, а величина ε неизвестна.

Любой численный метод имеет свои преимущества и недостатки, в связи с чем возникает вопрос о сравнении различных методов. Существует ряд факторов, которые следует учитывать при оценке эффективности алгоритмов и их сравнении. Так, универсальность алгоритма определяется тем классом задач, для решения которых он предназначен, а также рамками требований, предъявляемых алгоритмом к задачам данного класса. Другими важными характеристиками алгоритма является его надежность (или устойчивость), точность, чувствительность к параметрам и исходным данным, затраты на предварительную обработку и вычисления.

Кроме того, с появлением параллельных вычислительных систем возникает другой интересный вопрос: как сравнивать параллельный и последовательный алгоритмы. Не вызывает сомнения факт, что сравнивать параллельные алгоритмы необходимо по-своему [6]. Для оценки эффективности параллельных алгоритмов используют различные подходы, наиболее распространенным из которых является показатель ускорения (speedup). Ускорение (speedup), получаемое при работе алгоритма на p процессорах – это отношение времени работы алгоритма на одном процессоре ко времени работы того же алгоритма на p процессорах. Линейное ускорение (linear speedup) наблюдается, когда параллельный алгоритм на p процессорах работает в p раз быстрее, чем на одном процессоре. Сублинейное ускорение (sub-linear speedup) достигается, когда улучшение в скорости счета меньше p . Суперлинейное ускорение (super-linear speedup) достигается, когда улучшение в скорости счета больше p . Закон Амдала [5] позволяет вычислить верхнюю границу ускорения, которое можно ожидать от параллельной реализации алгоритма.

3. Методы глобальной оптимизации

К сожалению, для решения задачи глобальной оптимизации не существует универсального по эффективности алгоритма. Поэтому при разработке специфических методов глобальной оптимизации в первую очередь учитывают свойства целевой функции $F(\cdot)$ и допустимого множества X рассматриваемого класса задач, для которых разрабатывается метод.

Все известные методы глобальной оптимизации можно разделить на две категории: детерминированные [44] и стохастические [3], [26], [35]. Детерминированные методы получают глобальное решение посредством исчерпывающего поиска на всем допустимом множестве. Поэтому большинство детерминированных методов теряют эффективность и надежность с возрастанием размерности задачи. Более того, чтобы гарантировать успех, такие методы требуют выполнения дополнительных предположений, наложенных на функцию цели. Детерминированные методы не используют стохастики.

Стохастические алгоритмы позволяют уйти от проблем детерминированных алгоритмов. Здесь стохастический подход присутствует не только в разработке и анализе алгоритма, но и используется в решении базовых проблем, например, при определении условия останова. Большинство стохастических методов оценивают

значение функции цели в случайных точках допустимого множества с последующей обработкой выборки. Как следствие, стохастические методы не гарантируют успех.

Ниже приведены некоторые примеры современных методов глобальной оптимизации. Описание методов оптимизации также можно найти в [7], [12], [13], [14], [21], [22], [30], [31], [33], [36], [41], [43], [44], [46], [48], [49], [50].

Исторически первым методом глобальной оптимизации является метод Монте-Карло, на базе которого был создан метод мультистарта (multistart method) [35]. В методе мультистарта из множества X случайно или детерминировано выбирается некоторое подмножество из N точек. Последовательно из каждой точки запускается алгоритм локального спуска, и из полученного множества локальных решений выбирается наилучшее. В чистом виде метод мультистарта не является эффективным, т.к. одно и то же локальное решение может быть найдено не один раз. Мультистарт – это обобщенный подход: большинство эффективных методов глобальной оптимизации основано на идее метода мультистарта – запуска стандартных локальных алгоритмов из множества точек, равномерно распределенных на множестве X . Таким образом, метод мультистарта можно назвать прототипом таких методов.

Методы группировок (clustering methods) [38] являются одной из модификаций метода мультистарта. Здесь предпринята попытка устранения главного недостатка мультистарта путем тщательного отбора точек, из которых запускается локальный поиск. Рассматривается некоторая выборка точек, например, равномерно распределенных на X . Затем, пока не выполнится некоторое условие остановки, последовательно выполняются следующие три шага:

1. из каждой точки запускается алгоритм локального спуска, в результате будет получен набор локальных решений;
2. используя специальную технику группировки, определяются группы точек;
3. в качестве новой выборки точек рассматривается каждая m -тая точка из группы, и осуществляется переход к первому шагу.

Таким образом, решения находятся посредством локальных алгоритмов спуска из наилучших точек каждой группы.

Существуют различные модификации оригинального метода группировок, разработанные многими авторами. Одна из таких модификаций, где наилучший элемент группы определяется непосредственно, без применения техники группировки, носит название топографического метода (topographical method) [39], [40]. Идея топографического метода состоит из трех концептуальных шагов. На первом шаге рассматривается произвольная выборка точек, равномерно распределенных на допустимом множестве X . На втором шаге необходимо определить некоторое множество S допустимых точек. Это делается с помощью построения так называемой топографии. Топография представляет собой направленный граф из точек выборки первого шага, в вершинах которого хранится информация о целевой функции в данной точке. Каждая вершина графа соединяется направленными дугами с соседними вершинами, в которых значения целевой функции лучше. Минимумы, т.е. вершины, из которых не выходит ни одной дуги, в графе образуют множество S . Третий шаг определяет локальные минимумы, представленные множеством S : из каждой точки этого множества запускается локальный алгоритм. Наилучшая полученная точка будет решением исходной задачи. Используемая топография дает возможность реализации топографического алгоритма на машинах с параллельной

архитектурой. Результаты работы последовательных и параллельных версий топографического метода представлены в [39].

Методы деления пополам (bisection methods) и методы интервалов (interval methods) [13], [19], [27], [28], [34] гарантируют, что решение будет получено с заданной точностью. Эти методы в глобальной оптимизации также носят название методов покрытий (covering methods) или методов ветвей и границ (branch-and-bound methods). Цена предлагаемой методами гарантии – некоторая а priori известная информация о функции цели. Так, для метода деления пополам необходимо знание константы Липшица функции цели, а для метода интервалов функция цели обязана быть дважды непрерывно дифференцируемой, и первая и вторая производные должны иметь конечное число нулей. Большинство методов интервалов используют стратегию ветвей и границ (branch-and-bound strategy) [28]. Такие алгоритмы разделяют область поиска на набор многомерных кубиков, на которых нижняя граница функции цели вычисляется с помощью интервальной техники. Используя интервальную арифметику на каждом шаге, получаем набор последовательно уменьшающихся интервалов, который содержит глобальное решение исходной задачи. Алгоритм останавливается, когда размер интервалов достигает заранее заданного значения [27]. Методы интервалов требуют, чтобы функция цели была задана явно, т.к. это выражение используется интервальной техникой для вычисления границ. Методы, использующие технику ветвей и границ, в силу древовидной структуры являются объектом исследований в контексте параллельных вычислений. Некоторые разработки параллельных алгоритмов ветвей и границ приводятся в [28].

Каждый шаг метода туннелей (tunneling method) [15] состоит из двух фаз: фаза минимизации улучшает значение текущего рекорда; фаза туннелирования находит точку из допустимого множества, отличную от последней, где найдено значение минимума. Полученная на фазе туннелирования точка рассматривается как стартовая для очередного цикла. Главный недостаток этого алгоритма – необходимость решения сложных нелинейных дифференциальных уравнений.

Основная идея метода моделирования обжига (simulated annealing) [1], [4], [23], [18] исходит из физики процесса замерзания жидкостей или рекристаллизации металлов в процессе обжига. Целевая функция здесь является аналогом равновесия термодинамической системы и видоизменяется путем добавления случайных величин (условий температурного режима) [23]. Процесс повторяется достаточное число раз для каждой температуры, после чего температура понижается и весь процесс происходит снова до состояния полной заморозки. Избегание попадания в незначительные локальные минимумы (замерзание) зависит от «схемы обжига», выбора начальной температуры, количества итераций для каждой температуры и насколько уменьшается температура на каждом шаге процесса «охлаждения».

Эволюционные алгоритмы (evolutionary algorithms) [24] являются поисковыми методами, основная идея которых заимствована из биологического процесса естественного отбора и процесса выживания. Такие алгоритмы отличаются от традиционных методов оптимизации тем, что поиск производится из «популяции» решений, а не из одной точки. Каждая итерация метода производит «естественный отбор», который отсеивает неподходящие решения. Решения с высокой пригодностью («биологической реакцией на естественный отбор») «скрещиваются» с другими решениями путем обмена частями одних решений на другие. Решения могут «мутировать» из-за небольших замен одного элемента решения. Скрещивания и мутации генерируют новые решения, которые «генетически» настроены на области

допустимого множества, для которых уже было обнаружено хорошее решение. Существует несколько различных типов эволюционных поисковых алгоритмов: алгоритмы генетического программирования (genetic programming); алгоритмы эволюционного программирования (evolutionary programming); алгоритмы эволюционных стратегий (evolutionary strategies); генетические алгоритмы (genetic algorithms). Эволюционные алгоритмы обладают слабой сходимостью к глобальному решению, но вместе с тем, хорошо обрабатывают сильно зашумленные функции с большим числом незначительных локальных решений, не «прилипают» к локальным экстремумам и способны получить глобальное решение.

Алгоритм контролируемого случайного поиска (controlled random search algorithm) [32] является методом прямого поиска. Алгоритм начинает работу на множестве S начальных точек, равномерно распределенных на допустимом множестве задачи. Многократно образуются новые испытываемые точки, которые замещают подмножество W худших точек из множества S в случае, если они лучше. Алгоритм останавливается, когда все полученные значения функции достаточно близки. Существует множество модификаций оригинального метода: различия заключаются в способе выбора испытываемых точек и использовании локальной минимизации. Последние модификации для генерации испытываемых точек используют квадратичную аппроксимацию вместо симплексного метода [2]. Хотя алгоритмы прямого поиска являются устойчивыми к помехам различного рода, они требуют намного больше вычислений функции, чем алгоритмы, использующие градиент. Методы прямого поиска хорошо подходят для параллельных вычислений [8].

Траекторные методы (trajectory methods) [9], [10], [37], [42] и, в частности, метод продолжения (continuation method) [45], [47] занимают важное место в глобальной оптимизации. Траекторные методы на допустимом множестве исходной задачи строят множество кривых (траекторий). Основная идея траекторных методов заключается в том, что все решения исходной задачи а priori лежат на этих кривых. Во многих случаях эти траектории являются решением систем обыкновенных дифференциальных уравнений первого или второго порядка. Подробнее остановимся на методе продолжения.

4. Метод продолжения

Некоторые результаты разработки методов продолжения были предложены Guddat и Jongen (см. [16], [17]). Однако в указанных работах алгоритм продолжения строится для широкого класса задач, что, естественно, отрицательно сказывается на эффективности построенных алгоритмов. В работе [47] рассматривается весьма узкий класс задач глобальной оптимизации и строится алгоритм продолжения, существенно использующий специфику рассматриваемого класса задач.

Рассматривается следующий класс задач глобальной оптимизации:

$$P: F(x) \rightarrow \min_{x \in X},$$

где $F(\cdot)$ предполагается дважды непрерывно дифференцируемой на X ,

$$X = \Pi^{(k)} = \{x = (x_1, \dots, x_k) \in \mathbf{R}_k \mid 0 \leq x_n \leq 1, n = 1, \dots, k\} = \bigotimes_{n=1}^k [0, 1].$$

Метод продолжения предлагает строить квази-оптимальные траектории, соединяя все локальные минимумы функции цели, не упуская ни одного из них. При определенных условиях, наложенных на функцию цели, множество Куна-Таккера задачи P имеет простую топологическую структуру. Метод продолжения основан на использовании глобальных свойств структуры множеств Куна-Таккера и заключается в последовательном решении ряда задач оптимизации размерности n , для $n = 1, 2, \dots, k-1$ по следующей схеме:

Шаг 1. Пусть $n = 1$. Решается одномерная задача локальной минимизации:

$$P_1(0): F(x_1, 0, K, 0) \rightarrow \min_{0 \leq x_1 \leq 1},$$

получается $S_1 = \{x_1, K, x_{s_1}\}$ -- множество локальных решений задачи $P_1(0)$.

Шаг 2. $n := n + 1$.

Шаг 3. Из каждой точки полученного ранее множества $S_{n-1} = \{x_i, 1 \leq i \leq s_{n-1}\}$ для построения квази-оптимальных траекторий $x_{i,n}(t)$, $t \in [0, 1]$ запускается алгоритм продолжения. В зависимости от условий, наложенных на функцию цели, и от доступных вычислительных ресурсов можно использовать различные подходы при построении квази-оптимальных траекторий (например, решение систем обыкновенных дифференциальных уравнений второго порядка, использование разностных схем, использование различных методов локального спуска). Квази-оптимальные траектории обладают следующим свойством: для каждого значения $t \in [0, 1]$ соответствующая точка траектории $x_{i,n}(t) \in \Pi^{(n)}$ является локальным решением задачи

$$F(x_1, K, x_n, t, 0, K, 0) \rightarrow \min_{x \in \Pi^{(n)}, 0 \leq t \leq 1}.$$

Далее, если $n < k - 1$, то происходит переход к шагу 4, иначе – к шагу 5.

Шаг 4. Для всех траекторий решается вспомогательная задача оптимизации

$$F(x_{i,n}(t), t, 0, K, 0) \rightarrow \min_{0 \leq t \leq 1},$$

получается $S_n = \{x_1, K, x_{s_n}\}$ -- множество локальных решений этой задачи и осуществляется переход к шагу 2.

Шаг 5. Для всех траекторий решается вспомогательная задача оптимизации

$$F(x_{i,k-1}(t), t) \rightarrow \min_{0 \leq t \leq 1},$$

и из всех решений выбирается наилучшее – $x^* \in \Pi^{(k)}$. Полученная точка x^* является решением задачи P . Алгоритм останавливается.

В предложенном алгоритме продолжения легко видеть, что на этапах построения квази-оптимальных траекторий и поиска локальных минимумов на них, происходят идентичные действия. Очевидно, что расчеты таких траекторий можно выполнять параллельно, и можно рассчитывать на некоторое ускорение счета. Один из подходов к распараллеливанию алгоритма продолжения был реализован и протестирован на

машине с параллельной архитектурой PARSYTEC (T805) GC-1/64 в ВЦ РАН [29]. Ключевыми моментами в реализации параллельного алгоритма являются использование виртуальной топологии «кольцо» OS PARIX и независимость алгоритма от количества доступных транспьютеров. Наилучшее ускорение алгоритма продолжения достигается на задачах небольшой размерности с большим количеством квази-оптимальных траекторий.

5. Заключение

Задача разработки новых и усовершенствования существующих методов глобальной оптимизации остается открытой. Для решения задачи глобальной оптимизации не существует универсального по эффективности алгоритма. Поэтому при разработке специфических методов глобальной оптимизации в первую очередь учитывают свойства конкретного класса задач, для которых разрабатывается метод. Методам глобальной оптимизации посвящено много работ. В настоящей статье рассмотрены современные методы глобальной оптимизации и, в частности, метод продолжения. Метод продолжения является точным методом глобальной оптимизации и гарантирует глобальное приближение к решению задачи.

В настоящее время разработка методов оптимизации ведется не только в соответствии с существующими потребностями в решении конкретных задач, но и учитывается современное состояние вычислительной техники. Параллельные вычисления представляют собой мощное средство для проведения научных экспериментов. Рассматриваемый метод продолжения – это не только еще один способ решения задачи глобальной оптимизации, это еще и возможность усовершенствования методов глобальной оптимизации в параллельных компьютерных средах.

Литература

1. Aarts E.H.L., van Laarhoven P.J.M. *Simulated Annealing: Theory and Applications* // London, Kluwer, 1987.
2. Ali M., Törn A., Viitanen S. *A Numerical Comparison of Some Modified Controlled Random Search Algorithms* // J. of Global Optimization 11, 377-385, 1997.
3. Ali M., Törn A., Viitanen S. *Stochastic Global Optimization: Problem, Classes and Solution Techniques* // J. of Global Optimization 14, 437-447, 1999.
4. Ali M., Storey C. *Aspiration Based Simulated Annealing Algorithm* // J. of Global Optimization 11, 181-191, 1996.
5. Amdahl G. *The Validity of Single Processor Approach to Achieving Large Scale Computing Capabilities* // AFIPS Proc. Vol. 30, 483-485, 1967.
6. Bertsekas D.P., Tsitsiklis J.N. *Parallel and Distributed Computation: Numerical Methods* // N.J., Prentice Hall, 1989.
7. Bomze I.M., Csendes T., Horst R., Pardalos P.M., eds. *Developments in Global Optimization* // London, Kluwer, 1997.
8. Dennis J.E.Jr., Torczon V. *Direct Search Methods on Parallel Machines* // SIAM J. Optimization, 1, 4, 448-474, 1991.

9. Diener I. *On the Global Convergence of Path-following Methods to Determine All Solutions to a System of Nonlinear Equations* // Math. Programming, 39, 181-188, 1987.
10. Diener I. *Trajectory Methods in Global Optimization* // Horst R., Pardalos P.M. (eds): *Handbook of Global Optimization*. Dordrecht, Kluwer, 649-668, 1995.
11. Dixon L.C.W., Szegö G.P. (eds). *Towards Global Optimization* // North-Holland, 1975.
12. Dixon L.C.W., Szegö G.P. (eds). *Towards Global Optimization 2* // North-Holland, 1978.
13. Evtushenko Yu. G., Potapov M. A., Korotkich V. V. *Recent Advances in Global Optimization* // Princeton, Princeton University Press, 274-297, 1992.
14. Floudas C.A., Pardalos P.M. (eds). *Recent Advances in Global Optimization* // Princeton, USA, Princeton University Press, 1992.
15. Gomez S., Levy A.V. *The Tunneling Method Applied to Global Optimization* // SIAM, Numerical Optimization (Boggs, P.T., ed.), 213-244, 1985.
16. Guddat J., Guerra Vasquez F., Jongen H.Th. *Parametric Optimization: Singularities, Pathfollowing and Jumps* // Chichester, Willey, 1990.
17. Guddat J., Jongen H.Th. *On Global Optimization Based on Parametric Optimization* // Advances in Math. Optimization. Math. Res. №15, 1988.
18. Hamma B., Viitanen S., A. Törn. *Parallel Continuous Simulated Annealing for Global Optimization* // Optimization Methods and Software, Vol. 13, №2, 93-116, 2000.
19. Hansen E.R. *Global Optimization Using Interval Analysis* // New York, Marcel Dekker, 1992.
20. Horst R., Pardalos P.M. (eds) *Handbook of Global Optimization* // Dordrecht, Kluwer, 1995.
21. Horst R., Tuy H. *Global Optimization, Deterministic Approaches* // Berlin, Springer-Verlag, 1990.
22. Huyer W., A. Neumaier. *Global Optimization by Multilevel Coordinate Search* // J. of Global Optimization 14, 331-355, 1999.
23. Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E. *Equation of State Calculations by Fast Computing Machines* // J. Chem. Phys. Vol. 21, No. 6, 1087-1092, 1953.
24. Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Edn.)* // New York, Springer-Verlag, 1996.
25. Moccus J. *Multiextremal Problems in Design* // M.: Nauka, 1966.
26. Moccus J. *Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization* // J. Global Optimization, Vol. 4, No. 4, 347-356, 1994.
27. Moore R. *Interval Analysis* // New Jersey, Prentice-Hall, 1966.
28. Nemhauser G.L., Pruul E.A., Rushmeier R.A. *Branch-and-bound and Parallel Computation: a Historical Note* // Oper. Res. Let., 7, 65-69, 1988.

29. Orlyanskaya I.V., Perunova Y.N., Zavriev S.K. *An Implementation of the Parallel Continuation Algorithm for Global Optimization* // Сборник трудов 3-й Московской международной конференции по исследованию операций, 91-92, 2001.
30. Pardalos P.M., Rosen J.B. *Constrained Global Optimization: Algorithms and Applications* // Berlin, Springer Verlag, Lecture Notes in Computer Science vol. 268, 1987.
31. Pintér J.D. *Global Optimization in Action* // London, Kluwer, 1996.
32. Price W.L. *A Controlled Random Search Procedure for Global Optimization* // The Computer Journal, 20, 367-370, 1977.
33. Ratschek H., Rokne J. *New Computer Methods for Global Optimization* // Chichester, Ellis Horwood, 1988.
34. Ratschek H., Rokne J.G. *Interval Methods* // Horst R., Pardalos P.M. (eds): Handbook of Global Optimization. Dordrecht, Kluwer, 751-828, 1995.
35. Rinnoy Kan A.H.G., Timmer G.T. *Stochastic Global Optimization Methods* // Mathematical programming, 39, 27-78, 1987.
36. Gray P., Hart W.E., Painton L., Phillips C., Trahan M., Wagner J. *A Survey of Global Optimization Methods* // Sandia National Laboratories, 1998.
37. Sturua E.G., Zavriev S.K. *A Trajectory Algorithm Based on the Gradient Method. The Search on Quasioptimal trajectories* // J. of Global Optimization, №1, 1991.
38. Törn A. *Global Optimization as a Combination of Global and Local Search* // Turku, Abo Akademi University, HHÅA A:13, 1974.
39. Törn A., Viitanen S. *Topographical Global Optimization* // Floudas C.A., Pardalos P.M. (eds): Recent Advances in Global Optimization. Princeton University Press, 384-398, 1992.
40. Törn A., Viitanen S. *Topographical Global Optimization Using Pre-Sampled Points* // Journal of Global Optimization, Vol 5, No 3, 267-276, 1994.
41. Törn A., Zilinskas A. *Global Optimization* // Berlin, Springer Verlag, 1989.
42. Zavriev S.K. *On the Global Optimization Properties of Finite-difference Local Descent Algorithms* // J. of Global Optimization, 3, 63-78, 1993.
43. Васильев Ф.П. Численные методы решения экстремальных задач // М.: Наука, 1980.
44. Евтушенко Ю.Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Ж. вычисл. матем. и матем. физики, т. 11, №6, 1971.
45. Завриев С.К. Конечные алгоритмы метода продолжения в задачах выпуклой параметрической оптимизации // Обратные задачи математического программирования, ВЦ РАН, Москва, 1992.
46. Жиглявский А.А., Жилинскас А.Г. Методы поиска глобального экстремума // М.: Наука, 1991.

47. Завриев С.К., Орлянская И.В., Перунова Ю.Н. *Об одном подходе к конструированию алгоритмов продолжения в глобальной оптимизации* // Вестн. МГУ, Вычисл. математика и кибернетика, №2, 2000.
48. Завриев С.К., Перунова Ю.Н. *Параллельные версии модифицированных методов покоординатного и градиентного спуска и их применение для решения некоторого класса задач глобальной оптимизации* // Прикладная математика и информатика №10, М.: Диалог-МГУ, 2002.
49. Поляк Б.Т. *Введение в оптимизацию* // М.: Наука, 1983.
50. Сухарев А.Г. *Глобальный экстремум и методы его отыскания* // М.: Изд. МГУ, 1981.