Модифицированный метод половинных делений

Постановка задачи и основные определения

Без ограничения общности, рассмотрим задачу безусловной глобальной оптимизации непрерывной функции , заданной на допустимом множестве  в следующей постановке:

. (1)

Положим, что глобальный минимум  принадлежит множеству , причем , а , является многомерным единичным гиперкубом. Далее используются условные обозначения и основные определения, заимствованные из работы [1].

Определим множество -решений задачи (1) следующим образом:

. (2)

Нахождение приближенного решения задачи (1) заключается в поиске хотя бы одной точки множества .

Впервые метод неравномерных покрытий был предложен Ю.Г. Евтушенко в 1971 году [2]. Кратко опишем его основные положения, поскольку предлагаемый в данной работе модифицированный метод половинных делений во многом основывается на идеях этого метода.

В методе неравномерных покрытий строится последовательность  подмножеств множества  и точек  (). В каждой точке  вычисляется значение функции  и определяется её рекордное значение :

. (3)

На каждом из подмножеств  вводятся *миноранты*, т.е. такие функции , что . Тогда последовательности подмножеств  можно поставить в соответствие совокупность покрывающих подмножеств , определяемых из условия:

. (4)

Множество  строится таким образом, что  выполняется неравенство , из чего следует, что глобальный минимум функции  на множестве  не может быть меньше рекорда  более чем на . Таким образом, в процессе поиска глобального минимума подмножество  можно исключить из рассмотрения и продолжить оптимизацию на множестве . Алгоритм оптимизации останавливается, когда допустимое множество оказывается покрытым подмножествами , т.е.

. (5)

Алгоритм неравномерных покрытий, предложенный в работах [6, 40], гарантирует нахождение -оптимального решения, что подтверждается основополагающей теоремой [40].

Метод половинных делений

Идеяметода половинных делений [2] заключается в организации непропорционального деление исходной области поиска (обычно – единичного гиперкуба) на гиперпараллелепипеды, меньшей размерности. Чаще всего реализуется дихотомическое деление параллелепипедов по наибольшему ребру [1, 3, 4]. При этом (для двухмерного пространства) единичный куб делится на две прямоугольные области, которые затем делятся на квадраты четвертной размерности т.д. как показано на рисунке 1. В результате в пространстве поиска формируется нерегулярная сетка испытаний исследуемой функции. Возможен вариант непропорционального деления параллелепипеда по его главной диагонали [5].

Пусть допустимое множество  делится на прямоугольные параллелепипеды  с центрами  и главными диагоналями , где  - радиус параллелепипеда  ().Метод половинных делений [2] сводится к построению последовательности  параллелепипедов подмножеств множества  и точек  (). В каждой точке  вычисляется значение функции  и определяется её рекордное значение  по формуле (3).

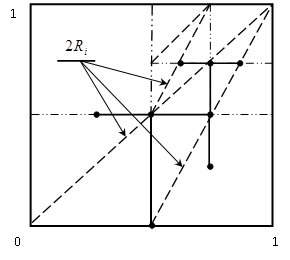


Рисунок 1 - Деление допустимого множества

Для функций, удовлетворяющих условию Липшица, на каждом шаге алгоритма выбирается «критический» параллелепипед, для которого

.

Именно он подвергается дальнейшему делению на две части.

Используя условие Липшица, параллелепипеды, для которых  исключаются из дальнейшего рассмотрения. Здесь *fr* - текущий рекорд при поиске минимума функции.

Для остановки алгоритма можно использовать условие:

.

Выбор критического параллелепипеда

Сохранив схему двоичного деления параллелепипедов [3], изменим правило выбора «критического» параллелепипеда, подвергающегося дальнейшему двоичному делению. Произвольный выбор «критического» параллелепипеда [3], или ориентация на параллелепипед, имеющий минимальное текущее значение функции в точке  [4], может привести к чрезмерно детальному «изучению» окрестностей локальных минимумов функции и «отсрочить» нахождение её нового рекорда , т.е. увеличить трудоемкость алгоритма.

В работе [6] для одномерных функций предложена одна из самых эффективных стратегий многоэкстремальной оптимизации, основанная на использовании приближенного апостериорного распределения вероятностей расположения глобального экстремума, формируемого в процессе испытаний функции, что реализует более сбалансированную организацию поиска глобального минимума функции. Данная стратегия настолько эффективна, что её часто переносят с одномерного случая на случай оптимизации функций многих переменных. Например, в работе [7] многомерная задача оптимизации, с помощью кривых Пеано, редуцируется к одномерной задаче глобальной оптимизации, в которой используется характеристическая схема оптимизации Стронгина Р. Г.. Аналогичные идеи используются в диагональном методе глобальной оптимизации Я.Д. Сергеева [5]. Попытаемся использовать эту стратегию и для метода половинных делений.

Если в процессе двоичных делений соединять центры параллелепипедов предшественников и потомков, то получится семейство непрерывных непересекающихся линий – траекторий перемещения центров параллелепипедов (рисунок 8), для которых можно вычислять характеристики соответствующих участков кривых по аналогии с одномерным случаем [6]. Оценки характеристик, полученных для участков траекторий перемещения центров параллелепипедов, припишем соответствующим параллелепипедам  с центрами в точках .

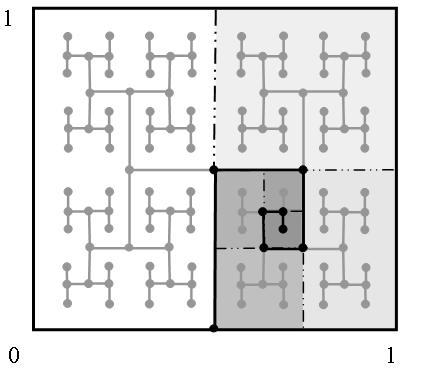


Рисунок 8 – Траектория перемещения центров параллелепипедов

Таким образом, каждый параллелепипед  свяжем с набором параметров . Здесь  - центр параллелепипеда,  - значение функции в точке ,  - радиус параллелепипеда, - расстояние между текущим параллелепипедом и его предшественником. – характеристика i-го параллелепипеда, которую в соответствии с [43] можно вычислить по формуле:

,

где – значение функции, вычисленное в предшествующем параллелепипеде в точке ; *w* – оценка константы Липшица. Последняя, адаптивно вычисляется в процессе вычислений функции:

 ,

где – коэффициент.

Из наборов характеристик  сформируем упорядоченный в направлении убывания характеристики  список , где .

Теперь в методе половинных делений в качестве «критического» параллелепипеда можно выбрать первый параллелепипед из списка , т.е. параллелепипед, для которого выполняется условие: .

Следует отметить, что в списке  на любом этапе работы модифицированного алгоритма половинных делений (МАПД) содержатся параллелепипеды, ещё не подвергнутые двоичному делению. Естественно, что «критические» параллелепипеды каждые раз удаляются из списка. Работа МАПД заканчивается, когда список  пуст.

Двухфазный алгоритм метода половинных делений (ДАМПД)

Как любой алгоритм ГО, алгоритм половинных делений имеет экспоненциальный характер роста сложности в зависимости от числа переменных. Одной из эффективных стратегий совершенствования алгоритмов глобальной оптимизации является использование *локальной техники* [8], когда стратегия глобального поиска удачно сочетается с методами локальной оптимизации.

В этом случае основной задачей на этапе глобальной оптимизации является определение областей притяжения локальных экстремумов функции, из которых можно запустить алгоритмы локальной оптимизации, поэтому его можно производить достаточно грубо. При ограниченном количестве минимумов функции  области притяжения  имеют значительные размеры. Предположим, что относительно оптимизируемой функции известно количество локальных минимумов –  (включая глобальный) и размеры областей притяжения: . Введем понятие гарантированного радиуса области притяжения минимумов функции, под которым будем понимать .

Дополнительная информация о размерах областей притяжения локальных минимумов функции позволяет построить двухфазную схему алгоритма глобальной оптимизации, в которой выделим фазы глобальной и локальной оптимизации (рисунок 9).

В ДАМПД фаза глобальной оптимизации реализуется с помощью двоичного деления, с той лишь разницей, что деление параллелепипедов в МАПД происходит до достижения заданных размеров их радиусов . Величина  определяется размерами гарантированного радиуса  области притяжения глобального минимума. На рисунке 9 пунктирными линиями отмечены гарантированные области притяжения минимумов функции. Заштрихованные прямоугольники являются прямоугольниками, «отбракованными» с использованием константы Липшица. В итоге, исходя из условия , фаза глобальной оптимизации завершилась, породив, для приведенного на рисунке 2 примера, 24 прямоугольника заданного радиуса.

В фазе локальной оптимизации из точек, принадлежащих областям притяжения локальных минимумов, организуется поиск минимумов функции с помощью локальных алгоритмов оптимизации. В нашем случае использовался метод деформированных многогранников [9] [10].

Такая схема организации процедуры поиска глобального минимума функции существенно сокращает трудоемкость фазы глобальной оптимизации.

Алгоритм построения множества точек начальных приближений для алгоритма локальной оптимизации.

Для больших размерностей вектора независимых переменных оптимизируемой функции в фазе глобальной оптимизации порождается значительное количество параллелепипедов заданного размера. В этом случае объем вычислений в фазе локальной оптимизации становится чрезмерно большим. Рассмотрим следующий адаптивный алгоритм формирования списка точек начальных приближений областей притяжения локальных минимумов оптимизируемой функции, осуществляющий сжатие количества точек начальных приближений, используемых во второй фазе алгоритма.

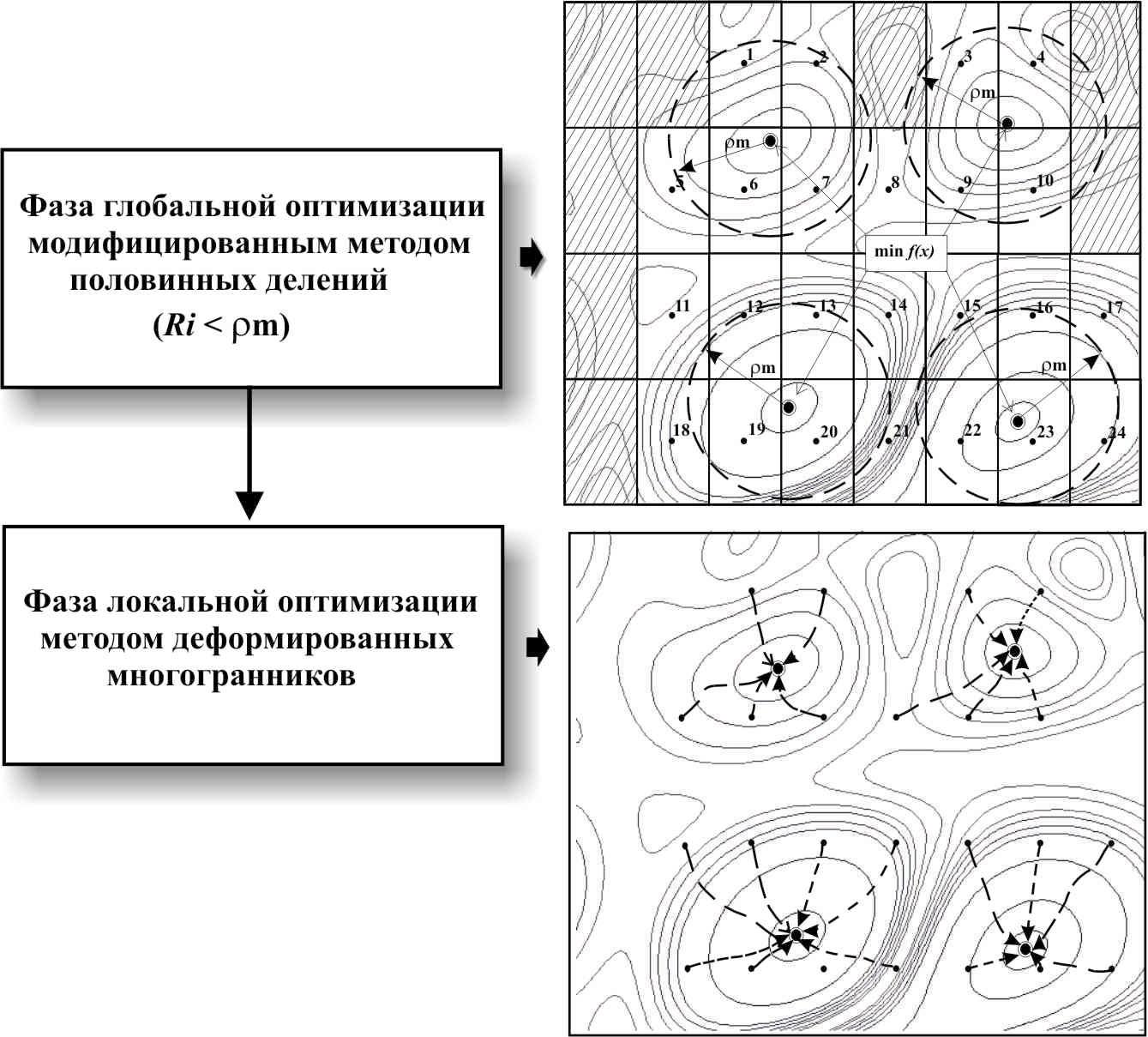


Рисунок 9 – Двухфазный алгоритм метода половинных делений

Будем считать, что область притяжения  определена, если она содержит, по крайней мере, одну точку из множества .

Пусть *r* – количество зон притяжения минимума функции;  – гарантированный радиус области притяжения минимума функции, обеспечивающий нахождение всех локальных минимумов.

Сформируем список областей притяжения локальных минимумов , элементами которого являются структуры , где  - координаты «представителя» *i*-й области притяжения, имеющей наилучшую достигнутую для этой области оценку . Вектор  условно считается центром *i*-ой области притяжения.

Первоначально список *V* – пуст. По мере вычислений функции он наполняется элементами, но в конце этапа глобального поиска не может содержать больше *m* элементов (*m* – заданный размер списка, ). Размер списка является эмпирическим параметром *m* и зависит от свойств оптимизируемой функции и выбирается из соображений попадания в него зоны притяжения глобального минимума функции. Элементы списка *V* упорядочены таким образом, что .

Пусть в процессе работы алгоритма глобальной оптимизации произведено очередное испытание . Эволюция содержимого списка *V* происходит по следующим простым правилам:

1. Проверяется принадлежность центра очередного параллелепипеда  окрестностям одной из имеющихся областей притяжения  (*l* – текущий размер списка).
   1. Если выполняется условие

. (12)

1.1.1. То при  содержимое элемента  обновляется:

, , где  - центры параллелепипедов «попавших» в область . Далее выполняется действие 1.1.3.

1.1.2. Иначе уточняется только значение . Выполняется действие 1.1.3.

1.1.3. Список *V* упорядочивается в порядке возрастания .

1.2. Если условие (12) не выполняется, то проверяем правило 2.

1. Определяется возможность включения нового элемента в список *V*.

2.1. Если , то элемент  записывается в голову списка *V*.

2.2. Если , то элемент  записывается между  и  элементами списка.

2.3. Если , то элемент  записывается в конец списка *V*.

1. При превышении предельного числа элементов списка, из списка исключается последний элемент списка.

Предложенный алгоритм является эвристическим, однако, вычислительные эксперименты с тестовыми функциями показали его высокую эффективность. Более того, как правило, в первой сотне элементов списка содержится начальное приближение глобального минимума функции. На рисунке 10 представлены элементы списка областей притяжения. Нумерация соответствует весу элемента. Из рисунка видно, что каждый элемент списка сформировался, агрегируя большое количество точек вычисления функции. Например, элемент  агрегировал 9 точек,  - 8, и т.д.

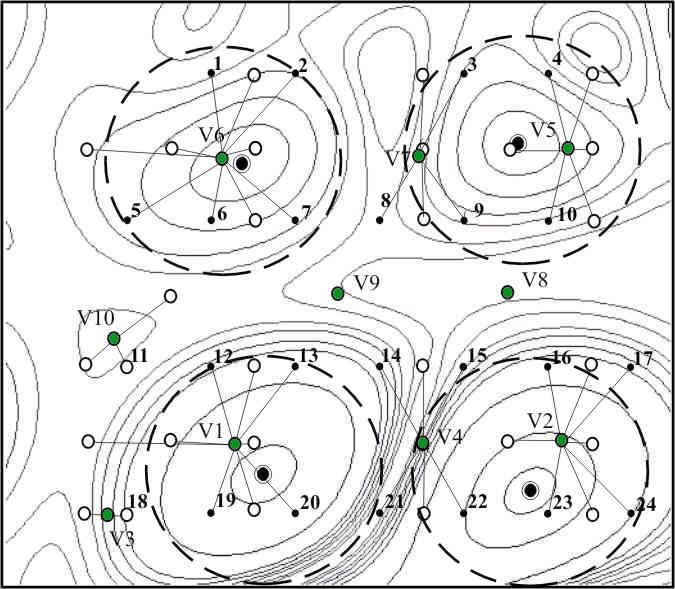


Рисунок 10 – Формирование списка областей притяжения

Параллельные версии двухфазного алгоритма метода половинных делений

Очевидно, что **з**атратывремени на решение задачи глобальной оптимизации можно, в известной степени, сократить за счет распараллеливания вычислительного процесса. Однако архитектура современных кластеров с распределенной памятью значительно затрудняет распараллеливание алгоритмов глобальной оптимизации, поскольку вынужденно приходится заниматься организацией эффективной передачи данных между процессорами, а не непосредственным распараллеливанием вычислений. При этом обычно используется «тяжеловесный», но естественный для подобных систем стандарт MPI. Как показала практика, наилучшие результаты по критериям оценки качества распараллеливания вычислений (ускорению и эффективности) получаются, когда удается свести к минимуму обмен информацией между процессорами.

Для двухфазного алгоритма метода половинных делений для первой фазы глобальной оптимизации за каждым процессором можно закрепить один параллелепипед из списка , построенного МАПД. Очевидно, что параллельные вычисления будут эффективны, если параллелепипеды будут иметь одинаковый радиус. В этом случае, если не учитывать «отбраковку» параллелепипедов по критерию Липшица, на каждом процессоре будет произведено одинаковое количество вычислений до достижения заданного уровня разбиения параллелепипедов на части. Из свойств самого алгоритма очевидно, что этого можно добиться, если допустимую область разбивать на  равных частей.

Первая версия алгоритма

Для реализации алгоритма ДАМПД использовался программный комплекс моделирования параллельных алгоритмов PGRAPH. Реализация алгоритма в рамках нотации ГСП представлена на рисунке 10. Для удобства описания модели все вершины пронумерованы.

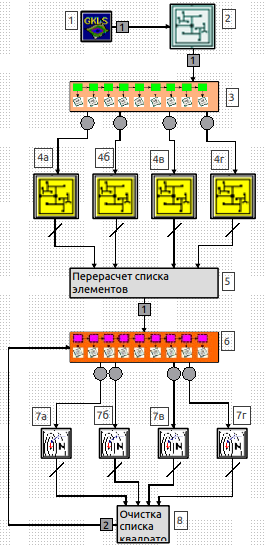


Рисунок 10 – Граф-модель алгоритма ГО ММПД

В вершине 1 устанавливаются значения параметров теста GKLS и его инициализация [10]. Как было писано ранее весь алгоритм состоит из двух фаз. В первую фазу алгоритма входят вершины 2-5. В вершине 2 вычисляется первое значение функции в точке x[i]=0,5, i=0..N, N - размерность пространства и в характеристический список *D* добавляется первый гиперкуб, охватывающий все пространство. В этой же вершине с помощью последовательного варианта алгоритма двоичного деления, описанного выше, формируется начальный характеристический список *D* с числом элементов не менее числа процессоров. При начальном делении областей прореживание выключено и на каждом шаге мы точно можем подсчитать количество элементов и состав списка. Очевидно из самого алгоритма половинного деления, что через 2n шагов список будет состоять из областей равного размера. Это уравнивает начальные условия для последующего параллельного деления.

Первый этап алгоритма осуществляет параллельное половинное деление и глобальный поиск в вершинах 4а-4г. На рисунке 10 приведён вариант алгоритма для 4х процессоров, но, с использованием соответствующей вершины-паттерна, алгоритм легко масштабируется на произвольное число процессоров. В вершине 3 происходит раздача заданий по процессорам. Каждый процесс получает один параллелепипед. Вершины 4а-4г имеют одинаковые агрегаты и запускаются каждая на своём процессоре. На каждом из процессоров с помощью МАПД реализуется покрытие частных параллелепипедов (в нашем примере ), причем их дробление производится до достижения определенных размеров. При этом формируются списки точек лежащих в областях притяжения локальных минимумов. В вершине 5 списки, сформировавшиеся отдельно на каждом процессоре, объединяются.

Второй этап осуществляет поиск локального минимума из точек определённых на первом этапе. Этап состоит из вершин 6-8. В вершине 6 мастер ветвь раздает начальные точки для локального поиска. В вершинах 7а-7г запускается локальный поиск максимального значения функции с помощью метода деформированных многогранников. В вершине 9 происходит подсчёт количества найденных минимумов. Выходом из второго этапа служит опустошение списка начальных областей или выполнение необходимого минимума запусков локального поиска. Если имеются непроверенные области локальных минимумов, то перед завершением программы они удаляются.

Вторая версия алгоритма

Во второй версии параллельного алгоритма ГО с целью повышения эффективности вычислений, каждому из процессоров доступна информация о найденном текущем максимальном значении функции. Совместно с локальной оценкой константы Липшица, знание глобального значения максимума функции значительно сокращает количество проверяемых параллелепипедов по критерию прореживания, описанному в главе 3. С помощью технологии ГСП, модификация алгоритма из первой версии во вторую сводится к замене типа переменной, хранящей значение глобального минимума, с локальной на общую.

Третья версия алгоритма

В третьей версии алгоритма синхронный режим работы процессоров на фазах ГО и ЛО изменен на асинхронный.

Для реализации данного подхода требуется либо написания более низкоуровневого актора, использующего функции MPI для ручного управления процессами, либо введение дуг синхронизации в модель алгоритма. Изначально модели алгоритмов параллельных вычислений, построенные в технологии ГСП, являются синхронными, но с помощью дуг синхронизации возможно моделирование асинхронной работы. Пример реализации для 4х процессоров с использованием дуг синхронизации показан на рисунке 15. Данная схема асинхронного управления часто называется менеджер-исполнитель, клиент-сервер и т.д.

Рассмотрим подробнее схему работы алгоритма на рисунке 15. Изначально готовится список локальных областей притяжения, больший, чем число процессоров. Все процессоры одновременно запускаются на поиск локального минимума каждый из своей точки. После завершения своего поиска ветвь-менеджер переходит в режим диспетчера и ждет сообщения о завершении от остальных ветвей. После приёма сообщения от любой параллельной ветви, ветвь-менеджер выдает новую точку отработавшей ветви и запускает её. Как только список локальных областей притяжения заканчивается, или проверено необходимое минимальное количество областей, ветвь-менеджер выходит из цикла приёма сообщений и все ветви просто завершают работу.

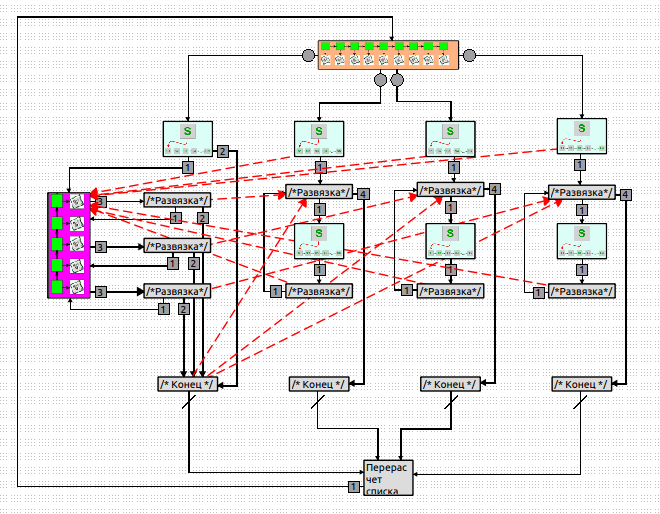


Рисунок 15 – Этап локальной оптимизации в режиме менеджер-исполнитель с синхронизацией

Для фазы ГО асинхронное управление процессорами строиться по аналогичной схеме. Необходимо лишь на этапе подготовки начального списка параллелепипедов подготовить параллелепипедов больше, чем число процессоров, таким образом обеспечить запас заданий для «быстрых» процессоров. Затем с использованием аналогичной схемы вычислений — менеджер-исполнитель — выполнять раздачу новых параллелепипедов уже закончившим работу процессам.

Выводы

Разработка новых алгоритмов и исследование их эффективности подразумевает создание большого числа версий и проведения большого числа экспериментов. Особенно остро данная проблема стоит при разработке параллельных алгоритмов, где кроме основной задачи эффективного распараллеливания, стоит не менее сложная, но рутинная (в техническом плане) задача управления данными. Как показала практика, применение технологии ГСП, позволяет сконцентрироваться на основной задаче.