

Вопрос № 1

Целочисленные оптимизационные задачи и методы их решения

Найти x_1^*, \dots, x_n^* , при которых $f_0(x_1, \dots, x_n) \rightarrow \min$, при ограничениях:

$$f_i(x_1, \dots, x_n) \geq 0, i = 1 \dots m$$

Дополнительное условие:

- 1) x_1, \dots, x_k , где $k < n$ - целые (задача частично целочисленного программирования)
- 2) $x_i, i = 1 \dots n$ - целые и могут принимать только значения 0 и 1 (задача бивалентного программирования)

Целевая функция может быть любой.

Метод, заключающийся в решении данной задачи без учета целочисленности x_i с последующим округлением до ближайшего целого не применим.

Методы решения:

- 1) Метод полного перебора. Преимущества: простота и гарантированное нахождение экстремума. Недостатки: большой объем вычислений; применение метода в случае незамкнутой области изменения независимых переменных.
- 2) Метод случайного поиска. Метод может быть применим только для замкнутой области изменений независимых переменных. Генератором случайных чисел получаем случайное число $0 \leq z_i \leq 1$. Задаем $x_i = \{z_i * (x_{i\max} - x_{i\min})\}$. $\{ \}$ – это операция выделения целой части. Проверяем для найденной точки выполнение условий попадания в разрешенную область. Если точка находится в разрешенной области – для нее вычисляется и запоминается целевая функция.
- 3) Метод ветвей и границ. Этот метод эвристический. Он заключается в анализе перспективных и не перспективных вариантов и отбрасывания

не перспективных ветвей. Недостаток алгоритма – возможность отсекаания оптимальной ветви, поэтому решения, найденные методом ветвей и границ, называют оптимистичными.

4) Алгоритм Гомори.

1. Отменяется требование целочисленности и решается задача обычными методами нелинейного программирования.
2. Полученное решение округляется до ближайшего целого.
3. Полученное целое значение проверяется на принадлежность к разрешенной области. Если нет, то по специальной методике, разработанной Гомори строится дополнительное ограничение, которое отсекает найденное нецелочисленное решение.
4. После добавления дополнительного ограничения процедура повторяется.

Вопрос № 2

Параллельные вычислительные системы. Примеры использования при решении конкретных задач

Архитектуры параллельных компьютеров могут значительно отличаться друг от друга. Рассмотрим некоторые существенные понятия и компоненты параллельных компьютеров. Параллельные компьютеры состоят из трех основных компонент:

- процессоры;
- модули памяти;
- коммутирующая сеть.

Можно рассмотреть и более изощренное разбиение параллельного компьютера на компоненты, однако, данные три компонента лучше всего отличают один параллельный компьютер от другого. Процессоры осуществляют обработку информации, работая параллельно. Модули памяти хранят данные для обработки, промежуточные и конечные результаты. Коммутирующая сеть обеспечивает связь процессоров с другими процессорами или памятью. Коммутирующая сеть собственно и создает параллельный компьютер, без нее речь могла бы идти лишь о совокупности компьютеров.

Классификация по числу потоков команд/данных.

Это одна из первых классификаций, ссылки на которую наиболее часто встречаются в литературе, была предложена М. Флинном в конце 60-х годов прошлого века. Она базируется на понятиях двух потоков: команд и данных. (Single Instruction Single Data). Такую архитектуру имеют все однопроцессорные системы.

MISD (Multiple Instruction Single Data). Эта архитектура не получила распространения, хотя формально может существовать. М. Флинн не смог привести ни одного примера реально

существующей системы, работающей на этом принципе. Некоторые авторы в качестве представителей такой архитектуры называют векторно-конвейерные компьютеры, однако такая точка зрения не получила широкой поддержки. Считается, что есть только один представитель данного семейства - мультипроцессор C. mpp университета Карнеги-Мелон. Но он может работать как в MISD, так и в MIMD режимах, причем, второй считается штатным.

SIMD (Single Instruction Multiple Data). SIMD компьютер (рисунок 1) имеет N идентичных синхронно работающих процессоров, N потоков данных и один поток команд.

Каждый процессор обладает собственной локальной памятью. Сеть, соединяющая процессоры, обычно имеет регулярную топологию.

MIMD (Multiple Instruction Multiple Data). MIMD компьютер имеет N процессоров, N потоков команд и N потоков данных.

Примеры использования параллельных вычислительных систем или суперкомпьютеров можно найти в разных областях промышленности, медицины, образования. Вот лишь небольшой список областей человеческой деятельности, где использование суперкомпьютеров действительно необходимо:

- автомобилестроение;
- нефте - и газодобыча;
- фармакология;
- прогноз погоды и моделирование изменения климата;
- сейсморазведка;
- проектирование электронных устройств;
- синтез новых материалов и многие, многие другие.

В 1995 году корпус автомобиля Nissan Maxima удалось сделать на 10% прочнее благодаря использованию суперкомпьютера фирмы Cray (The Atlanta Journal, 28 мая, 1995г). С помощью него были найдены не только слабые точки кузова, но и наиболее эффективный способ их удаления.

По данным Марка Миллера (Mark Miller, Ford Motor Company), для выполнения crash-тестов, при которых реальные автомобили разбиваются о бетонную стену с одновременным замером необходимых параметров, съемкой и последующей обработкой результатов, компании Форд понадобилось бы от 10 до 150 прототипов новых моделей при общих затратах от 4 до 60 миллионов долларов. Использование суперкомпьютеров позволило сократить число прототипов на одну треть.

Ещё один пример - это развитие одной из крупнейших мировых систем резервирования Amadeus, используемой тысячами агентств со 180000 терминалов в более чем ста странах. Установка двух серверов Hewlett-Packard T600 по 12 процессоров в каждом позволила довести степень оперативной доступности центральной системы до 99.85% при текущей загрузке около 60 миллионов запросов в сутки.

И подобные примеры можно найти повсюду. В свое время исследователи фирмы DuPont искали замену хлорофлюорокарбону. Нужно было найти материал, имеющий те же положительные качества: невоспламеняемость, стойкость к коррозии и низкую токсичность, но без вредного воздействия на озоновый слой Земли. За одну неделю были проведены необходимые расчеты на суперкомпьютере с общими затратами около 5 тысяч долларов. По оценкам специалистов DuPont, использование традиционных экспериментальных методов исследований потребовало бы около трех месяцев и 50 тысяч долларов и это без учета времени, необходимого на синтез и очистку необходимого количества вещества.

Вопрос № 3

Записать алгоритм решения системы линейных алгебраических уравнений методом итераций.

Пусть дано:

$$y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n + b_1$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n + b_2$$

...

$$y_n = a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n + b_n$$

В общем виде система записывается так: $y_i = \sum_{j=1}^n a_{ij} * x_j + b_i$

необходимо определить применимость метода простых итераций к решению нашей системы линейных уравнений, для этого проверяем условие $\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 < 1$

, если условие выполняется, то метод применим, иначе нет, сам метод

итераций выглядит следующим образом: $x_i^{(k+1)} = \sum_{j=1}^n a_{ij} * x_j^{(k)} + b_i$, приближения k,

k+1, k+2, k+3.... вычисляем до тех пор пока не выполнится условие:

$$\sqrt{\sum_{i=1}^n (x_i^{(k)} - x_i^{(k+1)})^2} < \varepsilon, \text{ где } \varepsilon - \text{точность.}$$

Алгоритм:

1. НАЧАЛО
2. Sum := 0
3. цикл i от 1 до n
4. цикл j от 1 до n
5. Sum := Sum + a_{ij}
6. если Sum < 1 то переход на п.8 иначе переход на п.20

7. задаём начальное приближение $x_i, i = 1 \dots n$
8. цикл i от 1 до n
9. $\text{SumA}_j := 0$
10. цикл j от 1 до n
11. $\text{SumA}_j := \text{SumA}_j + a_{ij} * x_{ij}$
12. конец цикла по j
13. $x_{i\text{old}} = x_i$
14. $x_i = \text{SumA}_j + b_i$
15. конец цикла по i
16. $\text{Er} := 0$
17. цикл i от 1 до n
18. $\text{Er} := \text{Er} + (x_i - x_{i\text{old}})^2$
19. если $\sqrt{\text{Er}} > \varepsilon$ то переходим на п.8 иначе переход на п.20
20. КОНЕЦ