



Решения задач векторной оптимизации с использованием методов параллельного программирования

П.К. Казаков

Тульский государственный университет, Тула

Аннотация: В данной статье рассматривается подход к решению задач векторной оптимизации с применением современных технологий, а именно – возможностей параллельного программирования. Решение задач большой размерности занимает довольно длительное время даже с использованием мощных вычислительных систем. В связи с этим, была предпринята попытка использовать все доступные логические ядра процессора персонального компьютера, получив таким образом доступ ко всей его вычислительной мощности. Написан более оптимальный алгоритм распределения нагрузки по потокам, позволяющий запускать несколько независимых друг от друга действий. Переработан метод решения задач векторной оптимизации, с применением алгоритма. Выделены преимущества разработанной программной реализации.

Ключевые слова: векторная оптимизация, экономическая деятельность, параллельное программирование, .NET, библиотека TPL, метод главного критерия, метод последовательных уступок, методы целевого программирования, многопоточность.

В настоящий момент в различных областях экономической деятельности большая часть задач управления и планирования, а также множество прикладных задач в основном решаются различными методами математического программирования. В разделе решения задач оптимизации наиболее развитыми являются методы векторной оптимизации, так же называемые многокритериальной оптимизацией или же просто программированием [1,2]. Эти методы позволяют решать большой круг задач экономической деятельности с достаточной точностью.

Некоторые из методов векторной оптимизации уже были реализованы в различных математических пакетах на компьютерах, но на решение задач больших размерностей требуется достаточно большое количество времени [3]. Для его сокращения было принято решение использовать все доступные аппаратные ресурсы компьютера, используя для этого параллельное программирование в среде .NET, которое позволяет задействовать все имеющиеся логические ядра процессора.

Постановку задачи экономической деятельности можно представить в виде математической модели программирования, когда целевую функцию записывают в векторной форме, а связь с ограниченными ресурсами – с помощью линейных уравнений или неравенств. А также, добавляется еще ограничение – переменные не должны быть отрицательными, так как они являются такими величинами как расходы, время работы, товарооборот и прочие финансовые показатели.

Величины x_1, x_2, \dots, x_n , полностью характеризующие описанный в задаче экономический процесс, называются переменными задачи, а система уравнений и неравенств, которая описывает ограниченность ресурсов в рассматриваемой задаче, называется системой ограничений. Математическая модель задачи имеет следующий вид: Найти числа x_1, x_2, \dots, x_n для которых функции

$$z_k = f_k(x_1, x_2, \dots, x_n), k = 1, 2, \dots, K \quad (1)$$

достигают максимального значения, и которые удовлетворяют системе ограничений:

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \geq b_1 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \geq b_m \end{cases} \quad (2)$$

и условиям неотрицательности $x_i \geq 0 (i = \overline{1, n})$ [4].

Идеальным решением задачи многокритериальной оптимизации (1) и (2) является вектор $F^* = (F_1^*, F_2^*, \dots, F_m^*)$,

где $F_k^* = \min_{X \in D_X} F_k(X), k = \overline{1, m}$ – минимальное значение частного критерия оптимальности $F_k(X)$ во множестве D_X [5].

Многопоточность или распределение выполнения некоторых участков программы по нескольким независимым друг от друга потокам, дает возможность приложениям разделять задачи и работать над каждой из них

автономно, для того чтобы максимально результативно задействовать ресурсы процессора и благодаря этому существенно сэкономить время пользователя. Для распараллеливания задач в язык программирования C# среды NET была внедрена специальная библиотека Task Parallel Library (далее TPL) [6].

Для того чтобы использовать возможности параллельного программирования был использован метод `Parallel.Invoke()` из встроенной библиотеки TPL, который позволяет выполнять несколько действий, не зависящих от результатов друг друга, а также реализован специальный доработанный цикл перебирающих значений "for" для его параллельного выполнения:

```
void ParallelFor(int fromIndex, int toIndex, Action<int> action) {
    int nextIteration = fromIndex;
    int procCount = Environment.ProcessorCount;
    //объект синхронизации и определения завершения всех заданий
    using (var resetEvent = new ManualResetEvent(false)) {
        for (var event = 0; event < Environment.ProcessorCount; event++) { //создание события
            ThreadPool.QueueUserWorkItem(delegate {
                int index; //передача событий на выполнение по очереди
                while ((index = Interlocked.Increment(ref nextIteration) - 1) < toIndex) {
                    action(index);
                }
                //проверка последнего события
                if (Interlocked.Decrement(ref procCount) == 0) {
                    resetEvent.Set();
                }
            });
        }
        resetEvent.WaitOne(); //ожидание завершения обработки событий
    }
}
```

Для решения задачи векторной оптимизации было рассмотрено большинство известных методов, и выбраны наиболее подходящие для программной реализации, а именно: метод главного критерия, метод последовательных уступок, а также метод целевого программирования. Наиболее предпочтительным является последний, так как он исключает в себе человеческий фактор, тогда как остальные требуют отдать предпочтение какому-либо из критериев или же указать порядок убывания их важности. Но стоит отметить, что любой из выбранных методов даст оптимальное решение из множества Парето, хотя оно может и не оказаться эффективным [7,8].

Метод главного критерия является наиболее употребляемым в инженерной практике, и ввиду его простоты для примера решено было рассмотреть именно его.

Параллельно будут выполняться построение условий задачи на основе входных данных, а также участки алгоритма решения симплекс методом задачи, сведенной к одному главному критерию, т.е. нахождение ведущих столбца и строки, перерасчет новой симплекс-таблицы методом прямоугольника и проверка решения на оптимальность [9,10]. Все эти выполняемые операции являются относительно независимыми.

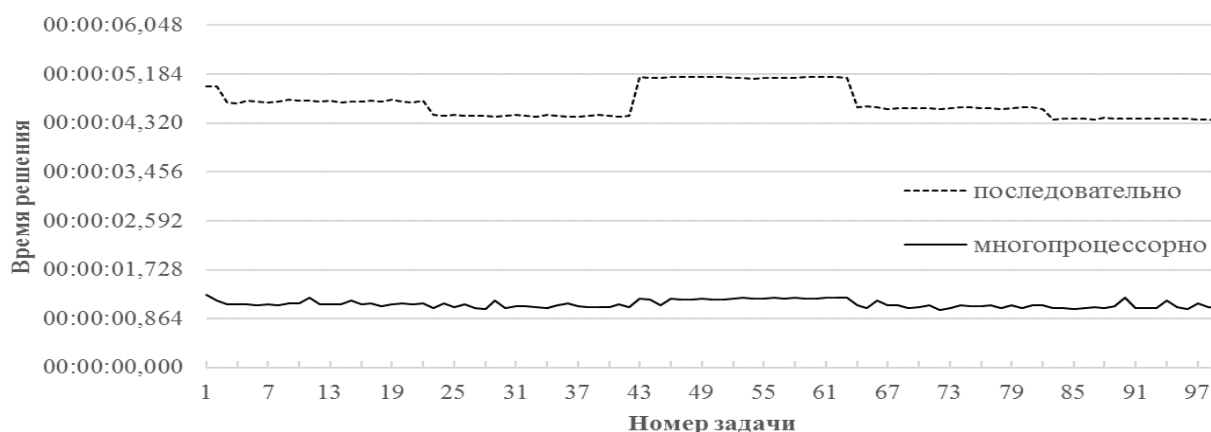


Рис. 1 – Время решения задач с 50 переменными и 200 ограничениями

На рис.Рис. 1 приведены результаты времени решения 100 различных задач с 50-ю переменными и 200-ми ограничениями. По данным графика

отчетливо видно, что нахождение оптимального решения методом главного критерия в многопоточном варианте выполняется намного быстрее, чем в последовательном.

Неравномерность графика объясняется различной сложностью решения задач с одинаковой размерностью, но различными данными, а также степенью загруженностью компьютера в процессе выполнения программы.

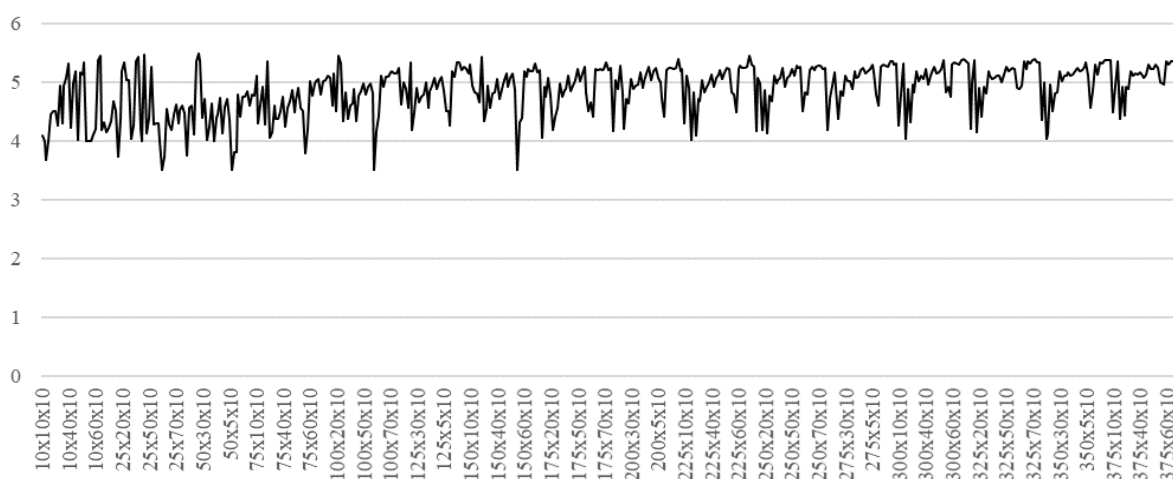


Рис. 2 – Время решения задач с 10-70 переменными и 10-375 ограничениями

На рис.Рис. 2 приведены результаты полученного коэффициента прироста решения задачи в параллельном режиме в сравнении с последовательным на 8 ядерном процессоре. Полученный выигрыш во времени составляет в среднем $0.6N$, где N – количество логических ядер процессора.

Следует отметить, что при возрастании количества ограничений увеличивается отрыв во времени параллельного решения задачи. Чуть меньший прирост мы получаем при количестве ограничений от 10 до 20, ограничений от 10 до 50. При еще меньших значениях использовать параллельный метод решения нецелесообразно, так как выигрыша по времени не будет – задача будет решаться медленнее последовательного режима. Так что при реализации программы этот момент был учтен, и в случаях небольшого количества итераций в цикле параллельный режим

работы не применялся. Таким образом получена гибридная версия программы, обеспечивающая наибо́льшее нахождение оптимального решения, применяя параллельную или последовательную версию в зависимости от размерности задачи.

Реализованная программа дает прирост производительности приблизительно в $0.6N$ раз, где N – количество логических ядер процессора, т.е. задача решается в $0.6n$ раз быстрее. Данная программа может быть использована для решения различных задач коммерческой деятельности после определенной доработки.

Литература

1. Нуркаева А.В. Методы многокритериальной оптимизации транспортной задачи // Инженерный вестник Дона, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4705
2. Веремеенко Т.В., Третьякова Л.Г. Высшая математика - Минск: ГИУСТ БГУ, 2010, 158 с.
3. Аль-хулайди А.А., Чернышев Ю.О. Разработка параллельного алгоритма нахождения оптимального решения транспортной задачи на кластере // Инженерный вестник Дона, 2011, №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445
4. Гасс С. Линейное программирование - М.: Государственное издательство физико-математической литературы, 2015, 304 с.
5. Хоменюк В.В. Элементы теории многокритериальной оптимизации - М.: Наука, 1983, 124 с.
6. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления - СПб.: БХВ-Петербург, 2002, 608 с.
7. Габасов Р. Методы оптимизации - Минск: Изд-во «Четыре четверти», 2011, 472 с.



8. Штойер Р. Многокритериальная оптимизация. Теория, вычисления и приложения - М.: Радио и связь, 1992, 504 с.

9. Dantzig, G.B, Minimization of a Linear Function of Variables Subject to Linear Inequalities. John Wiley, New York. 1951.

10. Bland, R.G, New Finite Pivoting Rules for the Simplex Method. Mathematics of Operations Research, 1977. pp. 103-107.

References

1. Nurkaeva A.V. Inzhenernyj vestnik Dona, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4705

2. Veremeenko T.V., Tret'jakova L.G. Vysshaja matematika [Advanced mathematics]. Minsk, 2010. 158p.

3. Al'-khulaydi A.A., Chernyshev Yu.O. Inzhenernyj vestnik Dona, 2011, №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445

4. Gass S. Linejnoe programmirovaniye [Linear programming]. Moscow, 2015. 304 p.

5. Homenjuk V.V. Jelementy teorii mnogokriterial'noj optimizacii [Elements of the theory of multi-criteria optimization]. Moscow, 1983. 124 p.

6. Voevodin V.V., Voevodin VI.V. Parallel'nye vychisleniya [Parallel computation]. St.Petersburg, 2002. 608 p.

7. Gabasov R. Metody optimizatsii [Optimization methods]. Minsk, 2011. 472 p.

8. Shtojer R. Mnogokriterial'naja optimizacija. Teorija, vychislenija i prilozhenija [Multi-criteria optimization. Theory, calculations, and applications]. Moscow, 1992. 504 p.

9. Dantzig, G.B. Minimization of a Linear Function of Variables Subject to Linear Inequalities. John Wiley, New York. 1951.

10. Bland, R.G, New Finite Pivoting Rules for the Simplex Method. Mathematics of Operations Research, 1977. pp. 103-107.