ОПРЕДЕЛЕНИЕ ПАРЕТОВСКИХ РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

И. К. Семенов, В. М. Горбунов, Е.А. Синюкова
Томский политехнический университет
iks5@tpu.ru

Введение

Эффективность использования вычислительных систем во многом определяется возможностями организации параллельной обработки. Параллелизм может возникать и обеспечиваться на разных уровнях подготовки и реализации вычислительного процесса – алгоритмов, программ, команд.

Параллельные вычисления особенно необходимы, когда задача требует высокой производительности компьютера. Распараллеливание по данным – один из главных приемов параллельного программирования, позволяющий ускорить обработку больших данных [1].

В настоящее время параллельные вычисления широко используются в многокритериальных задачах оптимизации [2]. Краеугольной задачей в многокритериальной оптимизации является поиск Парето-оптимальных (эффективных) решений, так как "…вариант проекта, по которому будет изготовляться серийная машина, обязательно должен быть Парето-оптимальным" [3, с. 3].

Сформулируем задачу многокритериальной оптимизации. Она имеет следующий вид:

$$\max F(X)$$
$$X \in D.$$

Символ maxF(X) понимается как набор символов $\max F_i(X)$, i=1,2, . . . , m. Будем предполагать, что все критерии нужно максимизировать, т.к. всегда можно перейти от $\min F_i(X)$ к $\max[-F_i(X)]$, i=1,2, . . . , m, т.е. сменой знака перед частным критерием.

Определим понятие оптимального решения в смысле Парето. Если решение $X_1 \in D$ не доминируемо никаким другим допустимым решением $X \in D$, то оно называется недоминируемым (эффективным) или оптимальным в смысле Парето.

Численные методы применяются для решения задач, которые не могут быть решены аналитически. Несколько слов о применяемом численном методе, который называют методом исследования различных вариантов. В области D генерируются N точек распределённых по равномерному закону, для них вычисляют значения частных критериев. Получаем пространство оценок (критериальное пространство), используя определение оптимальности по Парето, находим приближённо парето-оптимальные оценки, по которым находим соответствующие приближенно эффективные решения. Если количество пробных точек N возрастает, то приближённый фронт Парето в некотором смысле приближается к точному [4].

Обоснование использования параллельных вычислений

1. **Подготовка и реализация вычислительного процесса.**

Разработка параллельного алгоритма для решения данной задачи не вызывает затруднений. В параллельном варианте исходная задача разбивается на k подзадач – множество сгенерированных N точек из области D разбивается на k равных групп, в каждой их которых производится определение паретовских решений последовательным алгоритмом. После этого для получения окончательно решения задачи проводится слияние k паретовских решений групп (см. рис. 1).

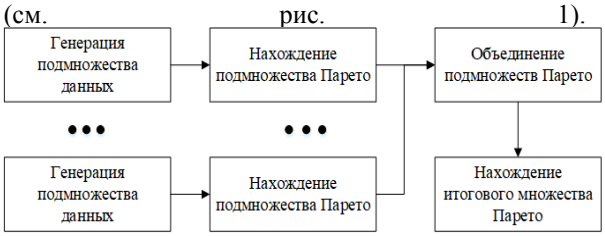

Рис. 1. Обобщенная схема алгоритма распараллеливания

2. **Теоретическое обоснование распараллеливания**

Так как определение паретовских решений определяется для каждого потока, то должно выполняться условие сумматорности транзитивности [4, с. 50, 54].

3. **Техническое обоснование**

Для эффективного использования параллельных вычислений, компьютер должен содержать несколько вычислительных ядер на центральном процессоре.

Рассмотрим применение рассмотренного алгоритма для следующей многокритериальной задачи: в области $D=[-10 \le x_1 \le 10; =[-10 \le x_2 \le 10]$ заданы два критерия $F_1(x_1, x_2)=(x_1-2)^2+(x_2-3)^2$; $F_2(x_1, x_2)=(x_1+2)^2+4x_2^2$, которые требуется минимизировать.

На начальном этапе (этапе инициализации) устанавливается количество потоков, которые будут выполнять вычисления (k), границы и объем обрабатываемой выборки (N). Следует уточнить, что вычисления проводились в двумерных пространствах параметров и оценок.

1. Генерация подмножества данных. На данном этапе каждый из k потоков заполняет соответственное подмножество параметров (парами x_1, x_2) при помощи генератора псевдослучайных чисел, распределённых по равномерному закону. При этом подмножество оценок заполняется парами F_1, F_2.

2. Нахождение подмножества Парето. После завершения первого этапа в сформировавшихся подмножествах находятся недоминируемые решения. Данные решения выделяются в отдельные подмножества.
3. Решения, полученные на предыдущем этапе (подмножества Парето-оптимальных решений) объединяются.
4. Нахождение множства Парето для множества, полученного при объединении подмножеств Парето-оптимальных решений. Данный шаг необходим для отсеивания решений, доминируемых решениями из других подмножеств.

Данное приложение было написано на языке программирования Visual C#. Для реализации параллельных вычислений использовались средства Microsoft .NET Framework, в частности, библиотека TPL (Task Parallel Library). Данная библиотека позволяет выполнять код параллельно с помощью классов Task (задача) и Thread (поток). В конкретной реализации параллельные вычисления осуществлялись за счет использования массивов объектов типа Task.

Эффективность параллельных вычислений.

Для оценки эффективности было проведено автоматизированное тестирование: подсчитано среднее время работы программы для разного количества потоков и разных объемов выборки после 100 выполнений алгоритма. Результаты тестирования представлены в таблице 1 и рисунке 2.

Таблица 1. Среднее время работы алгоритма (в секундах)

Число потоков	Число точек				
	100	1000	10000	50000	100000
1	0,000559	0,00177	0,073454	2,264257	10,64915
2	0,000423	0,001096	0,026671	0,710113	3,427886
4	0,000438	0,000906	0,013927	0,267502	1,348221
8	0,000429	0,000837	0,008999	0,158871	0,655979
16	0,000714	0,000851	0,006991	0,110031	0,449907
32	0,000656	0,000946	0,006094	0,064532	0,247039

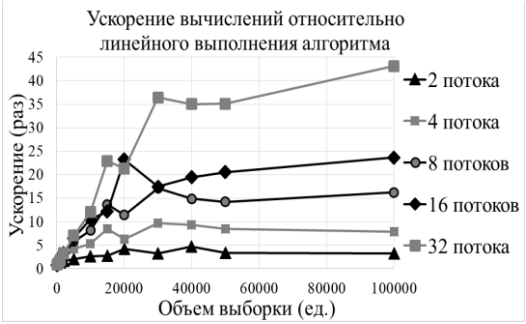

Рис. 2. Зависимость ускорения вычислений относительно линейного выполнения алгоритма от количества потоков и объема выборки

По данным из рисунка 2 можно заметить, что на малых объемах выборки (100 – 2000) ускорение изменяется незначительно для разного количества потоков, причем эффективность выполнения алгоритма на 16 и 32 потоках при выборке до 2000 элементов ниже, чем эффективность выполнения на 2, 4 и 8 потоках. Также видно, что ускорение работы алгоритма на 8 потоках больше, чем ускорение на 16 потоках для 15000 элементов и приблизительно равно ему для 30000 элементов. В свою очередь, выполнение алгоритма на 16 потоках для выборки объемом 20000 элементов дает большее ускорение, чем выполнение на 32 потоках.

Заключение

Дано обоснование применения параллельных вычислений для определения эффективных решений.

Проведенные расчеты показали, что использование параллельных вычислений позволяет существенно сократить время решения задачи (см. табл. 1).

Разработанная программа, например, позволяет увеличить количество исходных точек без потери времени, что позволяет, например, уточнить границы фронта Парето [5, с. 60].

Список использованных источников

1. Биллиг В. А. Параллельные вычисления на C#. Факты и гипотезы. [Электронный курс] URL: http://www.keldysh.ru/abrau/2015/25.pdf (Дата обращения 25.09. 2017).
2. Казаков В.Ю., Тимченко С.В. Параллельный алгоритм для решения задач многокритериальной оптимизации // Вторая Сибирская школа-семинар по параллельным вычислениям. – Томск: Изд-во Том. ун-та, 2004. – С. 32-37.
3. Статников Р.Б., Матусов И.Б.. Многокритериальное проектирование машин. – М.: Знание, 1989. – 48 с. /Новое в жизни, науке, технике. Сер. «Математика, кибернетика»; №5.
4. Соболь И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями: учеб. пособие для вузов. – М.: Дрофа, 2006. – 182 с.
5. Теория выбора и принятия решений: Учебное пособие /И.М. Макаров и др. - М.: Наука, 1982. - 388 с.