Параллельный метод вариаций в пространстве управлений

А.И. Карамова[a], Д.А. Петров[a]

[a] *Стерлитамакский филиал ФГБОУ ВО "Башкирский государственный университет", 453100, Стерлитамак, Россия*

Аннотация

В работе представлен алгоритм параллельного метода вариаций в пространстве управлений для решения задач оптимального управления. Приводится сравнение работы параллельного и последовательного алгоритма на примере с известным аналитическим решением.

Ключевые слова: оптимальное управление; параллельный алгоритм; вариации в пространстве управлений

1. Введение

Решение задач оптимального управления численными методами, сопровождается большим количеством вычислений, что соответственно влияет на время ожидания результата. При проведении большинства расчетов заказчики требуют быстрые и точные ответы от используемых алгоритмов. К счастью научный прогресс не стоит на месте. Старые одноядерные компьютеры вытесняются более производительными многоядерными, но и эта тенденция не проживет долго. Уже сегодня на первый план выходят кластерные системы, основанные на большом количестве процессоров. Сила таких систем заключена в том, что алгоритмы выполняются параллельно, тем самым давая колоссальный прирост производительности. Многие последовательны алгоритмы решения прикладных задач, располагают возможностью изменения своей структуры и хорошо адаптируются на параллельные архитектуры.

В данной работе представлена разработка алгоритма метода вариаций для решения задачи оптимального управления на системах с большим количеством вычислительных ядер.

2. Постановка задачи

Запишем формулировку задачи оптимального управления. Дана система (объект, процесс), состояние которой описывается дифференциальным уравнением:

$$\dot{x} = f(t, x, u),$$

где x – вектор фазовых координат, u – вектор управления, t – время. На вектора x и u могут быть наложены ограничения $x \in X$, $u \in U$. Система рассматривается на интервале $t \in [0, T]$.

Требуется определить вектор-функции $u(t)$, $x(t)$ доставляющие минимум функционалу $J=J(x,u)$ при переводе из начального состояния $(x(0), 0)$ в конечное состояние $(x(T), T)$.

Задачи оптимального управления классифицируются по способу задания функционала, по способу задания ограничений вдоль траектории и по способу задания краевых условий [1]. Не останавливаясь подробно на изложением общего материала, перейдем к частному методу численного решения задачи оптимального управления, а именно методу вариаций в пространстве управлений. Алгоритм последовательного метода представлен на рис. 1.

Суть метода распараллеливания довольна проста – «разделяй и властвуй». Элементарная операция выполняется для каждой точки начального оптимального управления. Тем самым для каждой точки решается система дифференциальных уравнений одним из численных методов [2]. По мере роста размера разбиений оптимального управления, растет количество однотипных вычислений. Результатом работы является сведение всех однотипных вычислений в блок выполняемый за одну итерацию. То есть элементарные операции выполняются не друг за другом, а параллельно независимо друг от друга. Технология написания программ на графических процессорах представлена в работе [3], а блок-схема параллельного алгоритма метода вариаций в пространстве управлений представлена на рис. 2.

Сравнение производительности двух вариантов алгоритмов поиска оптимального управления было произведено на тестовом примере с известным аналитическим решением, результаты сравнения представлены в таблице 1 и на рис. 3.

Тестовый пример. Даны модель объекта управления:

$$\dot{x}_1(t) = x_2(t),$$
$$\dot{x}_2(t) = -x_1(t) + u(t), \quad |u| \le 1,$$

с начальными условиями $x_1(t) = 0, x_2(t) = 0$, и функционал $J = x_2(2\pi) \to min$.

Требуется найти оптимальное программное управление $u^*(.)$ и соответствующую ему траекторию $x^*(.)$, доставляющие минимум функционалу J.

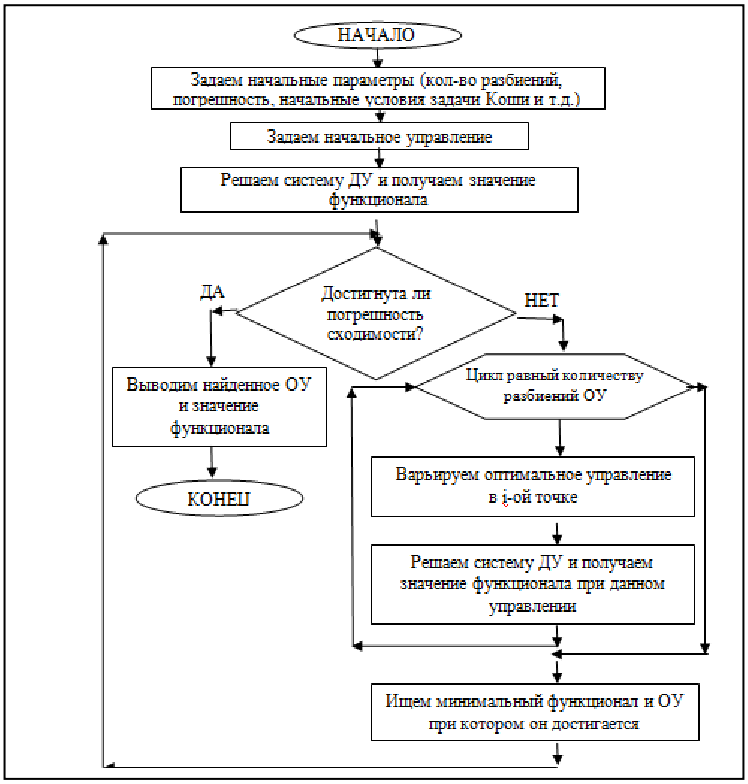

Рис. 1. Блок-схема последовательного алгоритма (CPU) поиска оптимального управления.

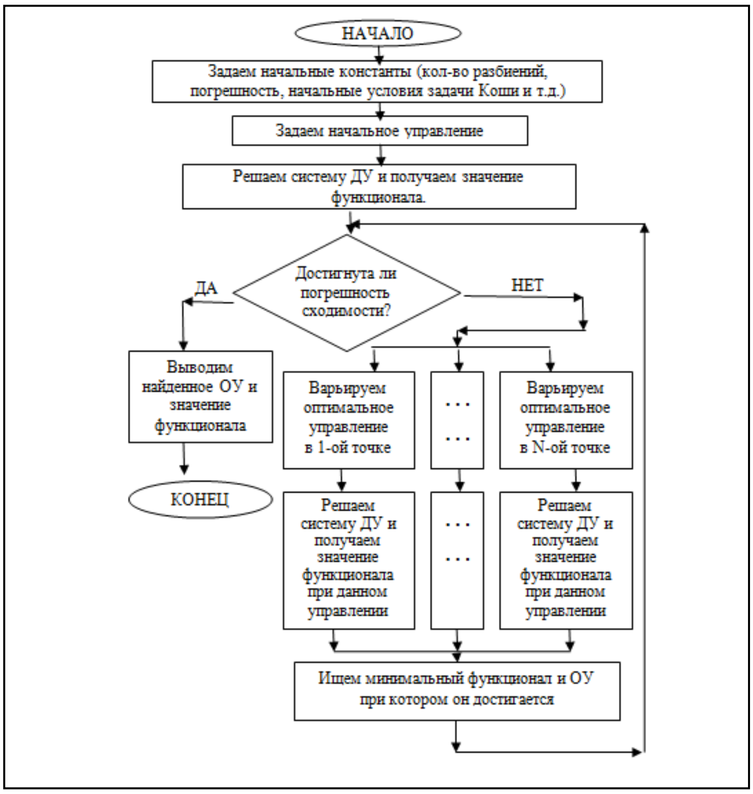

Рис. 2. Блок-схема параллельного алгоритма (GPU) поиска оптимального управления.

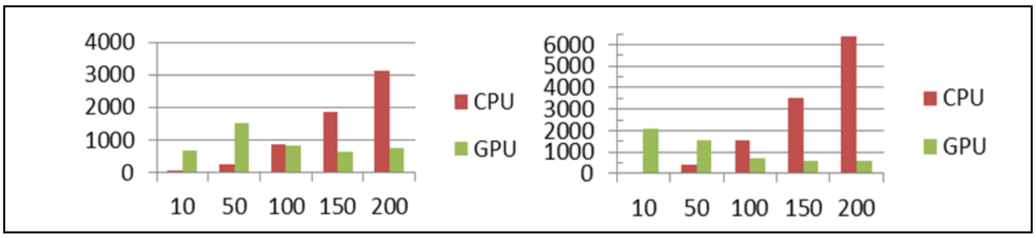

Рис. 3. Графическое сравнение производительности двух алгоритмов.

Таблица 1. Сравнение производительности последовательного и параллельного алгоритмов

Число итераций	Число разбиений ОУ	Время работы в миллисекундах	
		CPU	GPU
100	10	16	686
	50	234	1498
	100	858	812
	150	1872	624
	200	3135	734
200	10	31	2106
	50	405	1560
	100	1560	702
	150	3500	608
	200	6361	608

3. Заключение

Результаты расчетов подтверждают, что при малом количестве разбиений, последовательный алгоритм выполняется намного быстрее параллельного, но имеет прямую зависимость от количества итераций. При увеличении числа итераций значительно меняется время выполнения алгоритма. А при работе параллельного алгоритма наблюдается, что на время счета сильно не влияет количество итераций, но при малом количестве разбиений параллельный алгоритм просто не успевает разогнать мультипроцессор и поэтому при малых количествах разбиений он проигрывает своему конкуренту. Таким образом при большом количестве итераций и малом шаге разбиений параллельный алгоритм работает более стабильно и значительно быстрее, что подтверждает эффективность использования подобных алгоритмов при решении конкретных прикладных задач.

Литература

[1] Киреев, В.И. Численные методы в примерах и задачах / В.И. Киреев, А.В. Пантелеев. - М.: Высшая школа, 2008. – 480 с.

[2] Карамова, А.И. Параллельный алгоритм метода локальных вариаций для задачи оптимального управления / А.И. Карамова, Д.А. Петров // Международный научно-исследовательский журнал. – 2015. –№4-1 (35). – С. 4-8.

[3] Перепёлкин, Е.Е. Вычисления на графических процессорах (GPU) в задачах математической и теоретической физики / Е.Е. Перепёлкин, Б.И. Садовников, Н.Г. Иноземцева. – М.: URSS, 2014. – 176 с.