

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИНТЕРВАЛЬНОГО
МЕТОДА ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ,
ОРИЕНТИРОВАННОГО НА ВЫПОЛНЕНИЕ НА
ГРАФИЧЕСКИХ УСКОРИТЕЛЯХ**

Ткаченко Леонид Юрьевич

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: tkacleonid@yandex.ru

При решении разнообразных практических задач, например разработка новых материалов или расчет конструкции робототехнических систем, часто возникает необходимость глобальной оптимизации исследуемой функции. Как и множество других NP-трудных задач, часто поиск глобального минимума исследуемой функции выполняют методом ветвей и границ [1]. Особый интерес представляет интервальный метод ветвей и границ [2], который имеет множество практических применений, в том числе и для поиска глобального минимума целевой функции с заданной точностью.

В работе приводятся результаты исследования и разработки интервального метода глобальной оптимизации, который ориентирован на выполнение на графических ускорителях (GPU). Для реализации алгоритмов использовался язык C/C++ и технология NVIDIA CUDA. С целью повышения эффективности использования графических ускорителей были исследованы различные способы организации хранения брусков и интервальных оценок в памяти. Высокую эффективность показал способ хранения брусков в виде одномерного массива, содержащего дополнительное свободное место, которое использовалось графическим ускорителем при разбиении брусков на подбрусы. Такой способ хранения является эффективным при горизонтальном масштабировании на несколько графических ускорителей, а также при построении гибридных систем, в которых задействованы все доступные вычислительные мощности, CPU и GPU. Каждый поток на GPU при выполнении задачи интервальной оценки и генерации рабочего пула брусков реализовывал метод поиска в глубину и дихотомическое разбиение. Было показано, что использование других типов разбиений при таком способе организации хранения замедляло работу алгоритма из-за быстрого возникновения существенной разницы между числом брусков в рабочих пулах различных потоков исполнения на GPU. Для достижения максимальной эффективности использования вычислительных ресурсов GPU,

отдельное внимание в работе посвящено задаче балансировки. Были исследованы различные алгоритмы балансировки с учетом особенностей архитектуры CUDA. Балансировка выполнялась как на GPU, так и на CPU. На CPU балансировка осуществлялась для равномерной загрузки брусами нескольких графических ускорителей. При этом исследовалось влияние периодичности запуска балансировки на итоговое время решения задачи глобальной оптимизации.

Для осуществления вычислительных экспериментов использовались следующие графические ускорители: Tesla K40, Tesla K20c, Tesla C2075 и Tesla P100. Реализованный алгоритм позволяет эффективно распределять нагрузку между несколькими графическими ускорителями с различным количеством CUDA ядер.

Литература

1. Jens Clausen. Branch and Bound Algorithms - Principles and Examples (Technical report). Denmark, University of Copenhagen, 1999
2. Hansen E. R. and Walster G. W., Global Optimization using Interval Analysis, Revised and Expanded. Marcel Dekker, 2nd ed., 2004