

Способ распараллеливания эволюционного алгоритма решения задачи Штейнера на графе для системы с общей памятью

И. В. Аппель, А. В. Панюков

В докладе рассмотрена задача Штейнера на графе. Для решения данной задачи используется эволюционный алгоритм, для которого был разработан набор операций локальной оптимизации. Операции оптимизации имеют вид $\Delta(P, (W, F)) \rightarrow (X, G)$, где

- $P = \langle (V, E), d, Q \rangle$ - заданное условие задачи;
- (W, F) - исходное частное решение;
- (X, G) - производное частное решение.

Все используемые операции оптимизации являются неухудшающими: $\Psi(X, G) \leq \Psi(W, F)$ (где Ψ - целевой функционал для оптимизации), причем $\Psi(X, G) = \Psi(W, F) \Leftrightarrow (X, G) \equiv (W, F)$. Для поиска решения строится дерево частных решений, где дочерними элементами каждого узла являются его производные решения, которые ему нетождественны, причем указанное условие неухудшения решения позволяет гарантировать, что дерево решений будет иметь конечный размер. Оптимальное из найденных частных решений считается окончательным.. В рекуррентном виде данный алгоритм можно представить следующим образом:

$$\begin{aligned} solve(P, S) &= \begin{cases} best(P, siblings(P, S)), & siblings(P, S) \neq \emptyset, \\ S & siblings(P, S) = \emptyset, \end{cases} \\ children(P, S) &= \{\Delta_i(P, S) | i = 1, 2, \dots, n\}, \\ siblings(P, S_0) &= \{solve(P, S) | S \in children(P, S_0), d(S) \neq d(S_0)\}. \end{aligned}$$

Для приведенного итерационного алгоритма решения можно предложить очень простой подход к распараллеливанию для системы с общей памятью. Подход таков:

1. Стек решений выносится из функции поиска и доступ к нему синхронизируется для предотвращения проблем с одновременным доступом.
2. Переменная с лучшим найденным решением также выносится из функции поиска и доступ к ней так же синхронизируется.
3. Синхронизируется доступ к построенному при помощи алгоритма Дейкстры массиву данных о расстояниях между вершинами базового графа и кратчайших маршрутах между ними.
4. Функции поиска решения запускаются в нескольких параллельных потоках.

Поскольку операции оптимизации не используют никаких разделяемых данных, кроме массива маршрутов, можно ожидать эффективность распараллеливания, близкую к единице. Также был рассмотрен вариант распараллеливания по поддеревьям дерева решений. Анализ такой схемы параллелизма показал, что при ее использовании, задача балансировки нагрузки между потоками значительно усложняется, поскольку в данном случае эта задача сводится к задаче точно детерминированной оценки относительных размеров поддеревьев, сделать которую достаточно сложно. В то же время, нет причин считать, что такая организация параллельного алгоритма, будучи реализованной, приведет к заметному ускорению по сравнению с используемой организацией.

Был произведен ряд нагрузочных тестов, для тестирования использовался ПК с процессором Intel Core 2 Duo, работающий под управлением операционной системы Microsoft Windows XP. Программа для решения написана на Java и для ее запуска использовалась Java Runtime Edition версии 1.5.