Równoległe i dokładne algorytmy dla problemów ścieżkowych w grafach skierowanych

Jak teoretycy zwykle myślą o obliczeniach równoległych? Można patrzeć na obliczenie jak na obwód składający się z bramek wykonujących elementarne operacje logiczne bądź arytmetyczne – albo na danych wejściowych, albo wartościach na wyjściach innych bramek obwodu – lecz bez zależności cyklicznych. Podczas gdy, w tym spojrzeniu, projektowanie efektywnych (w sensie czasu działania) algorytmów sekwencyjnych to nic innego niż znajdowanie obwodu o możliwie małej liczbie bramek (wielkość tę nazywamy *czasem*) realizującego dane zadanie obliczeniowe, w projektowaniu algorytmów równoległych cel jest dwojaki. Znów, szukamy obwodu o możliwie małej liczbie bramek, nazywanej teraz *pracą*, ale również o możliwie krótkim najdłuższym łańcuchu zależności sekwencyjnych pomiędzy bramkami, nazywanym *głębokością*.

Głębokość określa, dość dokładnie, w jakim stopniu obliczenie może zostać zrównoleglone. Choć rzeczą niewyobrażalną wydaje się wykonanie obliczenia o pracy W na maszynie równoległej (o dowolnie dużej liczbie procesorów) szybciej niż w czasie proporcjonalnym do jego głębokości D, tzw. prawo Brenta mówi, że takie obliczenie istotnie da się zaplanować używając P procesorów tak, aby zakończyło sie w optymalnym czasie proporcjonalnym do $W/P+D$.

Oczywiście, idealny algorytm równoległy ma pracę porównywalną z najszybszym znanym algorytmem sekwencyjnym rozwiązującym ten sam problem, i do tego ma bardzo małą głębokość, np. ograniczoną przez jakąś stałą. Niestety, niewiele jest problemów obliczeniowych, dla których pokazano, że taki algorytm jest możliwy. Osiągnięcie tego wyidealizowanego celu wydaje się bardzo trudne na przykład dla większości klasycznych problemów na grafach skierowanych, takich jak obliczanie najkrótszych ścieżek. Wczesne badania naukowe nad algorytmami równoległymi skupiały się na optymalizacji głębokości, byle tylko praca pozostawała wielomianowa. Jednakże, równie interesującym jest badanie jakie *kompromisy pomiędzy pracą a głębokością* są możliwe do osiągnięcia. Na przykład, w praktyce szuka się tzw. algorytmów *efektywnych ze względu na pracę*, tzn. algorytmów równoległych o pracy porównywalnej z najszybszym znanym algorytmem sekwencyjnym i możliwie małą głębokością.

Celem tego projektu, ogólnie rzecz biorąc, jest badanie równoległych aspektów kilku spośród najbardziej fundamentalnych problemów obliczeniowych na grafach skierowanych – osiągalności (*„czy istnieje ścieżka pomiędzy dwoma wierzchołkami s i t grafu?”*), najkrótszych ścieżek (*„która droga z s to t ma najmniejszy sumaryczny koszt?”*) oraz maksymalnego przepływu (*“ile maksymalnie ścieżek z s to t można wybrać naraz tak, aby żadna krawędź e nie została użyta więcej niż $c(e)$ razy?”*).

Konkretniej, postaramy się znaleźć odpowiedź na następujące pytania:

- Jakie kompromisy pomiędzy pracą a głębokością są możliwe dla tych problemów?
- Jakie kompromisy można osiągnąć deterministycznie, tzn. bez używania losowości?
- Co to znaczy rozwiązywać problemy optymalizacyjne na grafach *dokładnie*?
- Jaki jest związek pomiędzy projektowaniem algorytmów równoległych dla tych problemów a projektowaniem algorytmów dokładnych?