

Grafy planarne

Głównym celem naukowym rozprawy doktorskiej jest konstrukcja nowych efektywnych algorytmów działających na grafach planarnych. Graf nazywamy *planarnym*, jeśli da się go narysować na płaszczyźnie tak, aby jego krawędzie się nie przecinały.

Algorytmika grafów planarnych jest aktywnie badana w ostatnich latach. Dzieje się tak z dwóch głównych powodów. Po pierwsze, grafy planarne są szczególnie ważne, ponieważ sieci występujące w rzeczywistych zastosowaniach często faktycznie można narysować bez lub z małą liczbą przecięć. Przykładem tu mogą być sieci drogowe, linie metra, czy też sieci połączeń w układach scalonych. Szczególne przypadki grafów planarnych, takie jak kraty, występują często w przetwarzaniu obrazów.

Drugim powodem popularności badań nad grafami planarnymi jest ich bogata kombinatoryczna struktura, która pozwala na konstruowanie bardzo eleganckich, ale i bardzo nietrywialnych algorytmów, o wiele szybszych niż te dla grafów ogólnych. W przypadku wielu spośród najbardziej podstawowych problemów grafowych, najlepsze znane algorytmy dla grafów planarnych są już bliskie optymalnym. Metody używane do uzyskania tych algorytmów leżą na styku takich dziedzin jak teoria grafów, geometria obliczeniowa i projektowanie struktur danych.

Struktury danych

Na grafy planarne patrzymy właśnie z punktu widzenia *struktur danych* użytecznych do konstrukcji algorytmów grafowych. Jako strukturę danych rozumiemy tutaj całościowo reprezentację pewnego podstawowego obiektu (przykładowo, grafu) w pamięci komputera wraz z podstawowymi operacjami na tej reprezentacji (np. dodawaniem/usuwaniem krawędzi, bądź sprawdzaniem, czy dwa wierzchołki są połączone krawędzią). Algorytmy podaje się zwykle jako ciągi operacji na odpowiednich strukturach danych; struktura danych jest więc pojęciem niższego poziomu niż algorytm. Przykładowo, kluczem do szybkiego działania słynnego *algorytmu Dijkstry*, obliczającego najkrótsze ścieżki w grafie, jest efektywna implementacja struktury danych zwanej *kolejką priorytetową*.

Obranie punktu widzenia struktur danych powoduje, że zamiast zajmować się konkretnymi problemami grafowymi (jak np. obliczanie maksymalnego przepływu, znajdowanie najkrótszych ścieżek) skupiamy się głównie na abstrakcyjnych podproblemach, powtarzających się w wielu znanych konstrukcjach algorytmicznych i będących zwykle wąskim gardłem obliczeń. Badając struktury danych, zależy nam więc na możliwie precyzyjnym określeniu ich parametrów (złożoności pamięciowej oraz złożoności czasowej indywidualnych operacji), gdyż ich efektywność wpływa na efektywność algorytmów dla wielu różnych problemów jednocześnie.

W rozprawie ulepszymy niektóre z dotychczas znanych podstawowych struktur danych używanych w algorytmach na grafach planarnych. Identyfikujemy także nowe problemy abstrakcyjne i podajemy motywowane nimi nowe struktury danych, które pozwoliłyby uprościć wcześniej znane algorytmy poprzez opisanie nowego abstrakcyjnego podproblemu.

Algorytmy dynamiczne

Badamy także *algorytmy dynamiczne* dla grafów planarnych. W pojęciu dynamicznego algorytmu grafowego zacierą się w pewnym sensie granica między algorytmem a strukturą danych. Mamy za zadanie utrzymywać graf w taki sposób, aby po małej zmianie, np. dodaniu bądź usunięciu krawędzi, można było zaktualizować interesującą nas informację – przykładowo długość najkrótszej ścieżki pomiędzy dwoma wskazanymi wierzchołkami – w sposób bardziej efektywny niż ponowne uruchomienie *statycznego* algorytmu (np. algorytmu Dijkstry) na zmodyfikowanym grafie.

Algorytmy dynamiczne zwykle rozważa się *per se*, ale często zdarza się, że istnienie efektywnego algorytmu dynamicznego dla pewnego problemu (nawet przy ograniczonych założeniach, np. że wolno nam tylko usuwać krawędzie początkowego grafu) natychmiast implikuje efektywny algorytm dla innego problemu *statycznego*. W takich sytuacjach algorytm dynamiczny jest wykorzystywany właśnie w roli struktury danych.