Nr rejestracyjny: 2022/45/N/ST6/04232; Kierownik projektu: mgr Jana Novotná

W poszukiwaniu szybszych algorytmów: Wykluczanie podpodziałów i grafów krawędziowych

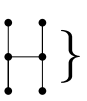
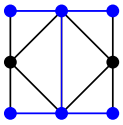
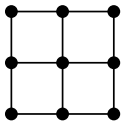
Jana Novotná

Wyobraźmy sobie mapę połączeń między miastami, elektryczną sieć przesyłową, stronę internetową lub zbiór zadań do wykonania. Wszystkie te rzeczy możemy badać pod kątem interesujących nas własności, jednak z matematycznego punktu widzenia wygodnie jest zapomnieć o większości szczegółów i skupić się przede wszystkim na strukturze, która opisuje relacje pomiędzy interesującymi nas obiektami. Taką strukturę nazywamy *grafem*, obiekty *wierzchołkami*, a związki pomiędzy dwoma obiektami *krawędziami*. Przykładowo, wierzchołki mogą reprezentować zadania, które trzeba wykonać, a krawędź pomiędzy dwoma zadaniami oznacza, że nie można wykonać ich w tym samym czasie.

Jednym z problemów grafowych, który od dawna cieszy się niesłabnącą popularnością, jest problem kolorowania grafów. Chcemy przypisać wierzchołkom grafu kolory w taki sposób, że żadne dwa sąsiadujące (połączone krawędzią) wierzchołki nie dostaną tego samego koloru. Dla danego grafu, jaka jest najmniejsza możliwa liczba kolorów, gwarantująca istnienie takiego „bezkonfliktowego" kolorowania? Taki problem, i wiele jego wariantów, znajduje zastosowanie w świecie rzeczywistym, w tym także przy planowaniu zadań: w naszym przykładzie wierzchołki w tym samym kolorze reprezentują zadania, które można wykonywać w tym samym czasie. Najmniejsza liczba kolorów, którą możemy „bezkonfliktowo" pokolorować graf odpowiada najkrótszemu czasowi potrzebnemu na wykonanie wszystkich zadań.

Niestety, problem kolorowania grafów jest NP-zupełny, co oznacza, że nie istnieje algorytm, który szybko znajdowałby rozwiązanie dla dowolnego grafu (zakładając słynną hipotezę P $\neq$ NP). Co więcej, okazuje się, że problem 3-kolorowania, w którym pytamy, czy dany graf da się bezkonfliktowo pokolorować używając tylko trzech kolorów, choć wydaje się łatwiejszy, również jest NP-zupełny. Z drugiej strony, wiele trudnych obliczeniowo problemów staje się istotnie łatwiejszych, jeśli założymy, że wejściowy graf należy do pewnej ustalonej klasy (czyli posiada dodatkowe własności, z których możemy skorzystać przy projektowaniu algorytmu). Tak więc pytanie, które można sobie zadać, brzmi: dla których klas grafów możemy zaprojektować efektywny algorytm rozwiązujący nasz problem, a dla których pozostaje on trudny obliczeniowo?

W projekcie skupimy się na klasach grafów, które mogą być zdefiniowane przez zabranianie pewnych struktur – jeśli $\mathcal{F}$ jest ustalonym (niekoniecznie skończonym) zbiorem grafów, możemy rozważać grafy, które nie zawierają w swojej strukturze żadnej kopii grafu z $\mathcal{F}$. Takie grafy nazywamy $\mathcal{F}$-wolnymi. W przykładzie poniżej zbiór $\mathcal{F}$ składa się z jednego grafu (z krawędziami jak w literze „H"). Graf na środkowym rysunku nie jest $\mathcal{F}$-wolny, ale graf na prawym już tak (zawiera „dodatkowe" krawędzie).

$\mathcal{F} = \{$  $\}$  

Projektowanie algorytmów lub dowodzenie trudności problemu w oparciu o charakteryzację przez zabronione struktury zyskuje coraz większą popularność na przestrzeni ostatnich lat. Największe zainteresowanie budzą klasy grafów, które mogą być zdefiniowane przez zabronienie jednego lub kilku grafów, jak również całej rodziny o specyficznych własnościach. W projekcie największą uwagę poświęcimy klasom grafów scharakteryzowanych przez nieskończone rodziny zabronionych struktur, które powstają z jednego grafu F przy pomocy dwóch operacji: wzięcia podpodziału (każda krawędź F zostaje zastąpiona dowolnie długą ścieżką) i grafu krawędziowego (który, intuicyjnie, reprezentuje sąsiedztwa pomiędzy krawędziami F).

Celem projektu jest zbadanie, jak nieobecność poszczególnych struktur w grafie wejściowym wpływa na złożoność problemu, oraz, w szczególności, uzyskanie nowych, szybszych algorytmów w rozważanych klasach grafów. Chcemy podejść do zagadnienia na dwa sposoby: z jednej strony, skupimy się na analizowaniu strukturalnych własności grafów $\mathcal{F}$-wolnych (dla ustalonego $\mathcal{F}$), aby użyć ich do projektowania efektywnych algorytmów dla wielu problemów. Z drugiej strony, będziemy szczegółowo badać strukturę grafów wejściowych oraz potencjalnych rozwiązań konkretnych problemów. Szczególny nacisk położymy na problem 3-kolorowania grafów.