

Маниашка писта

Анализ на Недялко Присадников

Поради ограниченията, задачата не може да се реши оптимално и трябва да се използват различни техники за намиране на колкото се може по-добър резултат. Аз използвах алчен подход за начално решение, което след това опитвах да подобра, отново с алчен подход.

Първоначално представям структурата от мазета и улеи, които ги свързват, в граф, като в този граф всяко мазе е връх и всеки улей е ребро. По време на изпълнение на програмата често се налага да изброяват всички ребра, които са съседни на дадено ребро, т.е. имат общ връх. Това не е много удобно и поради тази причина използвах друг граф, чиито върхове са ребрата на първоначалния и ако две ребра (вече върхове) са съседни има ребро, което ги свързва. Така прост списък на съседство е достатъчен за манипулиране на графа. Оттук в анализа на задачата, винаги ще имам предвид модифицираният граф.

При много малки графи (с по-малко от 10 върха) използвам Монте Карло алгоритъм, т.е. за 5 секунди постоянно генерирам произволни разполагания на медове по улеите (оцветявам графа по върхове) и проверявам дали текущото разположение е по-добро от най-доброто до този момент. Предвид размера на входа за този алгоритъм, вероятността да даде слаб резултат е доста малка. При по-голям вход използвам различна стратегия. Тя се състои в два етапа. В първия етап се генерира seed – начално решение, което след това да се подобрява. По време на втория етап използвам алчен подход, който да подобри решението. Първо ще обясня алчния алгоритъм от втория етап.

Трябва да имаме някакво базисно решение, т.е. графът да е оцветен по върхове. На всеки връх (улей) сме съпоставили някакъв мед. Сега хващаме един връх и проверяваме как би се изменил резултата, като му поставяме всички възможни медове. Избираме най-добрия такъв мед и го поставяме на този връх. Очевидно, макар че този алгоритъм е алчен и не може да гарантира оптималност, то той не може да влоши резултата, с който започваме. Тази процедура се изпълнява за всички върхове многократно. След като сме подобрили всеки връх, графът като цяло може да се измени доста и ако приложим метода за първия връх отново е възможно да трябва да сменим меда му и да подобрим резултата още. Поради това е хубаво този алгоритъм да се изпълнява колкото се може повече пъти. Аз го правя да се изпълнява докато има право да работи програмата. Подобряването на върхове се използва многократно и заради това трябва да е максимално ефективно. Аз използвам масив $as[i][j]$, който показва колко съседа със мед j има

върхът i . С него лесно се изчислява с колко допринася даден връх за резултата. Ако сменим меда на някой връх е нужно да редактираме стойностите в масива за „новия“ и „стария“ мед за всички съседи на върха. Макар, че броят на върховете може да е голям (до 10000), съседите на един връх са не повече от $2 \cdot N - 3$, където N е най-много 400. Този масив е удобен и за изчисляването на резултата, като сменяме меда на някой връх, защото тази промяна е локална и резултатът се изменя само за улея, чийто мед сменяме, и неговите съседи.

Преди алчния алгоритъм, трябва да генерираме базисно решение. Този етап се оказва доста важен за крайния резултат на програмата. Различни начални оцветявания с мед водят до доста различни резултати. Първоначално, оцветявах графа произволно или поставях най-добрия мед навсякъде. Тези идеи работеха добре на някои по-малки тестове, но в повечето случаи вадиха слаб резултат. Заради това използвах друга идея за оцветяване с мед. За всеки тип мед избирам произволен (неоцветен) връх и с BFS (Breadth-First Search) правя „група“ от върхове, оцветени с този мед, като се опитвам да „разпространявам“ меда така, че маниакалността на всички върхове в тази група да е максимална. Това става, като за всеки връх избирам колко още негови съседи трябва да са със същия мед, за да се максимизира интереса. Тази процедура се изпълнява за всички медове. След това, може да има върхове, които да не са оцветени. За тях използвам алчния алгоритъм описан по-горе. Изборът на върхове за началото на BFS-ите може да окаже влияние върху резултата от разполагането, затова аз пусках алгоритъм няколко пъти и след това вземах най-доброто оцветяване за базисно решение.

Недялко Присадников