

Аналіз складності алгоритмів в середньому

При вирішенні NP-складних задач виділяють два основних підходи. Один полягає в побудові наближених алгоритмів з гарантованими оцінками точності одержуваного рішення, другий у відмові від аналізу складності алгоритмів за найгіршим варіантом і переході до аналізу складності в середньому.

Більшість задач дискретної оптимізації є NP-складними, для них існування поліноміальних алгоритмів малоімовірно. Проте для ряду вихідних даних такі задачі легко розв'язуються на практиці. Вся складність задачі може полягати в невеликій підмножині входів. Концепція складності в середньому для таких задач представляється більш адекватною, ніж концепція складності в гіршому випадку. Хай для даного алгоритму A час його роботи на вході I позначається через $TA(I)$ і для кожного натурального n заданий розподіл ймовірності на входах I довжини n , що позначається $P_n(I)$. Оскільки T є випадковою величиною, природним здається визначити середній час роботи алгоритму A як математичне очікування часу роботи алгоритму $E_n TA$ на входах довжини n . Алгоритм A є поліноміальним в середньому (точно), якщо середній час його роботи обмежений поліномом від довжини входу, тобто $\exists c > 0: E_n T_A = O(n^c)$.

Проте інколи поліноміальність в середньому прийнято визначати по-іншому, тому що наведене визначення дуже чутливе: зміна T на T^2 (типова оцінка при симуляції однієї обчислювальної моделі на іншій) може привести до того, що алгоритм, поліноміальний в середньому, скажімо, на моделі RAM, не є поліноміальним в середньому на машині Тюрінга. Нічого подібного не відбувається при аналізі складності по гіршому випадку: визначення поліноміального алгоритму не залежить від вибраної моделі обчислень. Тому часто використовують інше визначення: алгоритм називається поліноміальним в середньому, якщо для часу роботи алгоритму T виконується: $\exists \epsilon > 0: E_n T^\epsilon = O(n)$. Та при дослідженні складності в середньому конкретних алгоритмів використовується поняття “поліноміальний в середньому (точно)” з зазначенням конкретної обчислювальної моделі, для якої проводиться аналіз.

Отже спроби послабити вимоги у визначенні ефективності привели до поняття складності в середньому, під яким розуміється математичне очікування часу роботи алгоритму при заданому ймовірнісному розподілі на вихідних даних. Користь переходу від аналізу за гіршим випадком до аналізу складності в середньому полягає в тому, що деякі практично ефективні алгоритми не є ефективними при аналізі за гіршим випадком. Найбільш відомим прикладом такого алгоритму є симплекс-метод рішення задач лінійного програмування.

¹ студент 3^{го} курсу навчання за напрямом “Комп’ютерна інженерія”;

² викладач кафедри програмного забезпечення