Алчни алгоритми

Велинов Даниел, Градежен факултет-Скопје

Алгоритам е постапка или збир од правила со чија помош можеме да реализираме дадена задача. Математичките алгоритми се многу важна алатка која се користи во пресметка на финансиски ризици, регулирање на сообраќајни метежи, планирање на авионски летови, препознавање на лице со помош на компјутер, пребарување на интернет и на многу други места кои значително го олеснуваат нашиот живот.

Алчни алгоритми се алгоритми за кои најдобриот можен избор е избор на најкраток пат (најголем пат). Тие не секогаш се најкраткиот пат (најдолгиот пат) во подолг алгоритам, но често знаат да бидат многу корисни. Идејата за разгледување на екстремни елементи (кои се најголеми, најмали, најдобри или најлоши во некоја смисла) е централна во овој пристап.

Генерално алчните алгоритми имаат пет компоненти:

1. Кандидат множество, за кое се бара решение
2. Селектирачка функција, која бира најдобар кандидат за решение
3. Изводлива функција, која определува дали даден кадидат може да придонесе за наоѓање на решение
4. Објективна функција, која припишува вредност на решението или парцијалното решение
5. Фунција решение, која утврдува дали е определено вистинско решение

Алчните алгоритми даваат добри решениеја на некои математички проблеми. Повеќето математички проблеми на кои можат да се применат алчните алгоритми ги имаат следниве две својства:

- Каков и било избор да направиме кој изгледа најдобар во тој момент можеме да ги решаваме потпроблемите кои ќе се јават подоцна. Изборот направен со алчен алгоритам може да зависи од изборите направени до тој момент, но не од идните избори или од решенијата на потпроблемот. Всушност итеративно правиме алчен избор по алчен избор, упростувајќи ја почетната ситуација во попроста ситуација. Во суштина алчните алгоритми никогаш не ги преиспитуваат претходните избори. Ова е главната разлика од динамичкото прогрмирање, кое ги исцрпува сите ситуации и секогаш гарантира наоѓање на решение.
- Проблемот може да се претстави во оптимално потструктура ако оптималното решение на проблемот го содржи оптималнотот решение на потпроблемот.

Во многу други проблеми, алчните алгоритми имаат проблем да најдат оптимално решение, а некогаш дури и да дадат единствено најлошо можно решение. Еден од најпознатите примери во кои алчните алгоритми не можат да дадат одговор е проблемот на трговскиот патник, кој сеуште е еден од нерешените Милениумски проблеми. Тој се состои во следното: Трговски патник треба да посети n градови, при што се знае

оддалеченоста помеѓу секој од градовите. Како трговскиот патник треба да ја планира рутата за да помине најмало растојание, а притоа ги посети сите градови?

Со цел да го илустрираме функционирањето на алчните алгоритми, во продолжение ќе бидат дадени неколку примери од теорија на графови и некои комбинаторни проблеми.

Пример1. Во граф G со n темиња кој нема теме со степен поголем од p. Докажи дека темињата може да се обојат со најмногу $p+1$ бои така што било кои две соседни темиња не се во иста боја.

Решение. Го користиме следниот алчен алгоритам: Ги подредуваме темињата во произволен редослед. Нека боите бидат $1, 2, 3, ..., p, p+1$. Првото теме го боиме со бојата 1. Потоа во секој нареден чекор, го земаме наредното теме и го боиме со најмалиот број на боја која не е користена на некој од неговите соседи. Јасно овој алгоритам обезбедува дека било кои две соседни темиња нема да бидат обоени со иста боја. Исто така обезбедува дека највеќе $p+1$ боја ќе бидат искористени. Навистина, секое теме има најмногу p соседи, па за боењето на дадено теме a најмногу p бои ќе се искористат за боењето на неговите соседи, па најмалку една од боите од множеството $\{1, 2, 3, ..., p, p+1\}$ нема да биде искористена. На тој начин добиваме дека сите темиња од графот ќе бидат обоени со најмногу $p+1$ бои.

Да забележиме дека "алчноста" на алгоритамот лежи во тоа што секогаш ја избираме бојата со најмал број. Интуитивно, ние користиме поголем број на боја доколку е потребно.

Пример2. (Задача од India Team Selection Test) Во $2 \times n$ табела се запишани позитивни реални броеви така што во секоја од колоните збирот на реалните броеви запишани во неа е 1. Докажи дека може да избереме по еден реален број од секоја од колоните така што збирот на избраните броеви е најмногу $\frac{n+1}{4}$. Во продолжение е даден пример на табела 2×6 која ги задоволува условите на задачата.

0,3	0,4	0,1	0,8	0,5	0,3
0,7	0,6	0,9	0,2	0,5	0,7

Решение. Многу едноставен алчен алгоритам би бил да го избираме најмалиот број од секоја колона. Но, овој алгоритам нема секогаш да фунцкионира. На пример нека првата редица сите реални броеви се $0,4$. Лесно може да се провери дека алгоритамот не функционира. Затоа мораме да избереме попаметна стратегија. Нека броевите во првата редица се наредени во монотоно растечки редослед $a_1, a_2, ..., a_n$ и соодветните броеви во втората редица $b_1, b_2, ..., b_n$ во монотоно опаѓачки редослед. Понатаму, без губење на општоста може да претпоставиме дека сумата на броевите во првата редица е поголема од сумата на броевите во втората редица. Да забележиме дека идејата за подредување на

елементите се користи многу често, бидејќи ни обезбедува структура со која полесно може да работиме.

Нашиот алгоритам ќе биде следниот: Почнувајќи од a_1 го избираме најмалиот останат елемент од горната редица се додека е тоа можно. Со други зборови, ние ги избираме сите $a_1, a_2, ..., a_k$ се додека $a_1 + a_2 + ... + a_k \le \frac{n+1}{4}$ и $a_1 + a_2 + ... + a_k + a_{k+1} > \frac{n+1}{4}$. Сега не може да избираме веќе елементи од првата редица бидејќи нема да бидат задоволени условите на задачата, па во остананите колони ќе ги избереме елементите од втората редица.

Сега само треба да покажеме дека збирот на избраните елементи од втората редица не надминува $\frac{n+1}{4}$. Да забележиме дека a_{k+1} е најмалку аритметичката средина на $a_1, a_2, a_3, ..., a_k, a_{k+1}$ што е повеќе од $\frac{n+1}{4(k+1)}$.

Следува $b_{k+1} = 1 - a_{k+1} < 1 - \frac{n+1}{4(k+1)}$. Уште b_{k+1} е најголем од избраните елементи од втората редица, па сумата на избранит елементи не може да надмине $\left(1 - \frac{n+1}{4(k+1)}\right)(n-k)$. Сега со едноставна проверка може да утврдиме дека начинот на кој ги избравме реалните броеви, по еден од секоја колона, го задоволува условот на задачата.

Пример 3. Во граф G со V темиња и E ребра, докажи дека постои подграф H, таков што секое теме има најмалку степен $\frac{E}{V}$.

Решение. Да забележиме дека средната вредност на степенот на некое од темињата на графот G е $\frac{2E}{V}$. Интуитивно гледано, ние би требало да се ослободиме од "лошите" темиња: темињата кои имаат степен помал од $\frac{E}{V}$. Па природно би било да го направиме следното: Почнуваме со графот G и се додека постои теме со степен помал од $\frac{E}{V}$ го бришеме. Да забележиме дека додека ги бришеме темињата со степен помал од $\frac{E}{V}$ ги бришеме и ребрата кои се побрзани со нив, па темиња кои на почетокот не се "лоши" може да станат "лоши" во новодобиениот подграф. Оваа постапка ја повторуваме се додека не добиеме подграф H со кој условот на задачата е задоволен. Останува уште да одговориме на прашањето дали може да се случи да завршиме со подграф во кој сите темиња се "лоши"? Ова не може да се случи. Да забележиме дека односот помеѓу ребрата и темињата е строго растечки (ако почнеме со $\frac{E}{V}$ и секој пат кога избришеме едно теме,

помалку од $\frac{E}{V}$ ребра се избришани од условот на нашиот алгоритам). Оттука, невозможно е да остане едно теме, бидејќи во тој случај односот ќе биде нула. Па, во даден момент алгоритамот мора да заврши, што ќе резултира со подграф H со повеќе од едно теме, во кој сите темиња имаат степен најмалку $\frac{E}{V}$.

Пример 4. (IMO shortlist 2001) Множество од три природни броеви $\{x, y, z\}$, такви што $x < y < z$ го задоволуваат условот $\{z-y, y-x\} = \{1776, 2001\}$ се нарекува историско множество. Докажи дека множеството од природни броеви може дас е запише како дисјунктна унија од историски множества.

Решение. Нека $a = 1776$ и $b = 2001$. Едно историско множество од условот на задачата е во облик $\{x, x+a, x+a+b\}$ или $\{x, x+b, x+a+b\}$. Ќе ги нарекуваме мали и големи множества соодветно. За да го "покриеме" множеството од природни броеви. Да го претставиме проблемот на следниов начин: Нека множеството од природни броеви е запишано на бројна права. Со секој чекор, избираме историско множество и ги покриваме тие броеви на правата. Секој број мора да биде покриен на "крајот" од нашата бесконечна постапка и ниту еден број не треба да биде покриен два или повеќе пати.

Да го забележиме следново: Во секоја ситуација, најмалиот број кој сеуште не е покриен е "најпроблематичен", бидејќи можеме да не успееме да го покриеме, па така целата постапка да не успее. Ако x е најмалиот број кој сеуше не е покриен, а $x+a+b$ е покриен, никогаш нема да успееме да го покриеме x. Така во секој чекор ќе мора да го покриваме најмалиот непокриен број x. На овој начин добиваме дека треба да ги преферираме малите множества пред големите множества.

Врз база на овие забелешки, ќе конструираме алчен алгоритам: Во било кој чекор, го избираме x да биде најмалот број кој не е покриен. Потоа користиме мало множество ако е можно, во спротивно користиме големо множество. Во продолжение ќе покажеме дека овој алгоритам функционира.

Нека претпоставиме дека алгоритамот не функционира, односно користејќи мало или големо множество ќе покрие некој од веќе покриените броеви во $(n+1)-$от чекор. Нека x_i е вредноста за x во $i-$от чекор. Пред $(n+1)-$от чекор, x_{n+1} не е покриено од тоа како функционира алгоритамот. Уште и $x_{n+1}+a+b$ не е покриено бидејќи е поголемо од сите покриени елементи (во нашиот алгоритам $x_i < x_{i+1} < x_{n+1}$). Па проблемот мора да биде предизвикан од $x_{n+1}+a$ и $x_{n+1}+b$, односно $x_{n+1}+a$ и $x_{n+1}+b$ мора да бидат веќе покриени. Понатаму, $x_{n+1}+b$ мора да бдие најголем елемент во тоа множество. Па најмалиот елемент во тоа множевто би бил $x_{n+1}+b-(a+b) = x_{n+1}-a$. Но, во оваа ситуација x_{n+1} не е покриен, па мало множество треба да се искористи и уште x_{n+1} треба да биде во него. Но

ова е контрадикција, па нашата претпоставка е лоша, па алгоритамот навистина фунцкионира.

Користена литература

1. G. Brendall, F. Margot, Greedy type Resistance of Combinatorial Problems, Discrete optimization 3 (2006), 288-298;
2. J. Bang-Jensen, G. Gutin, A. Yeo, When the greedy algorithm fails, Discrete Optimization 1 (2004), 121-127;
3. Pranav A. Sriram, Olympiad Combinatorics, 2014;
4. G. Gutin, A. Yeo, A. Zverovich, Traveling salesman should not be greedy: dominant analysis of greedy-type heuristics for the TST. Discrete applies Mathematics 117 (2002), 81-86.