Rješavanje problema 2-particije multiskupa

Ivo Ivanišević[1], *Goranka Nogo*[2]

Uvod

Pretpostavimo da košarkaški trener iz skupa od deset košarkaša, čija je kvaliteta u igri izražena pridruženim brojem, želi načiniti dvije ekipe jednake snage (ili barem minimalne razlike), tj. želi da razlika zbroja brojeva koji su pridruženi košarkašima u prvoj i drugoj ekipi bude najmanja moguća. Tako je, na primjer, za skup košarkaša

$$\{2, 10, 3, 8, 5, 7, 9, 5, 3, 2\}$$

jedan od mogućih odabira ekipa:

$$\{10, 8, 5, 2, 2\} \quad \text{i} \quad \{9, 7, 5, 3, 3\}$$

pri čemu razlika u snazi iznosi 0. No, i odabir

$$\{10, 7, 5, 3, 2\} \quad \text{i} \quad \{9, 8, 5, 3, 2\}$$

ima isto svojstvo tako da zaključujemo da rješenje problema ne mora biti jedinstveno.

Primijetimo da su u skupu košarkaša neki od njih jednako dobri. To nas dovodi do pojma multiskupa. Multiskup nalikuje skupu samo što kod ovog uzimamo u obzir ponavljanja njegovih elemenata.

U ovom radu prvo ćemo navesti jednu od ekvivalentnih definicija problema 2-particije multiskupa, a zatim ćemo opisati dva algoritma koji pronalaze jednu 2-particiju multiskupa, ne nužno minimalne razlike.

Definicija problema

Definicija problema pronalaženja 2-particije proizvoljnog multiskupa pozitivnih cijelih brojeva je izuzetno jednostavna. Za dani multiskup $A = \{a_1, a_2, \ldots, a_N\}$ pozitivnih cijelih brojeva treba pronaći particiju, to jest podskup $\mathcal{A} \subset \{1, \ldots, N\}$ takav da je razlika

$$\Delta(A) = \Big|\sum_{i \in \mathcal{A}} a_i - \sum_{i \notin \mathcal{A}} a_i\Big|$$

minimalna. Za particiju, takvu da je $\Delta = 0$ $(\Delta = 1)$ i da je suma $\sum a_j$ parna (neparna) kažemo da je savršena. Problem su prvi puta formulirali M. R. Garey i D. S. Johnson godine 1967.

Ova jednostavna formulacija problema, na prvi pogled skriva njegovu pravu težinu. Problem 2-particije multiskupa spada u klasu najteže rješivih problema, tj. spada u klasu NP-teških problema [2]. Problem je interesantan s praktičkog i teorijskog stajališta. Praktične primjene nalazimo npr. u raspoređivanju poslova u multiprocesorskom okruženju, dok je s teorijskog stajališta problem interesantan jer se koristi u dokazivanju NP-potpunosti nekih drugih (kompliciranijih) problema.

[1] Student na PMF-u, Matematičkom odsjeku Sveučilišta u Zagrebu, e-pošta: `ivo.ivanisevic@gmail.com`

[2] Docent na PMF-u, Matematičkom odsjeku Sveučilišta u Zagrebu, e-pošta: `nogo@math.hr`

Pohlepni algoritam

Od velikog broja algoritama za rješavanje ovog problema zacijelo je najjednostavniji pohlepni (*greedy*) algoritam. Naziv pohlepni dolazi od toga što algoritam "sprema" naveći preostali broj iz multiskupa u "torbu" s manjom sumom. Tako za skup $\{19, 13, 17, 6, 9\}$ imamo sljedeću tablicu koja prikazuje particioniranje našeg skupa primjenom pohlepog pristupa.

preostali brojevi	parcijalna particija	Δ
{19, 13, 17, 6, 9}	{ } { }	–
{13, 17, 6, 9}	{19} { }	19
{13, 6, 9}	{19} {17}	2
{6, 9}	{19} {17, 13}	11
{6}	{19, 9} {17, 13}	2
{ }	{19, 9, 4} {17, 13}	2

Iz primjera je jasno vidljivo da pohlepni pristup ne daje optimalno rješenje pošto je odstupanje jednako dva, dok je minimalno odstupanje jednako nuli za jedan primjer savršene particije $\{19, 13\}\{17, 9, 6\}$ traženog skupa. Ideja pohlepnog pristupa pri rješavanju problema 2-particije multiskupa je da se pri svakom koraku algoritma smanjuje odstupanje podskupova. Prilikom rada algoritma može se dogoditi da prije zadnjeg koraka dva podskupa imaju jednake sume. U tom slučaju algoritam završava s razlikom jednakoj najmanjem elementu multiskupa. Budući da algoritam u svakom koraku odabire najveći broj, prilikom implementacije algoritma je potrebno ulazne podatke sortirati silazno da bi se ubrzalo izvođenje.

Karmarkar-Karpov diferencijski algoritam

Diferencijski algoritam Karmarkara i Karpa (1982.) smatra se jednim od najboljih algoritama za rješavanje problema 2-particije multiskupa. Razlog za to ponajviše leži u činjenici da daje rješenja jako blizu optimalnog, a uz to je i jako brz. Glavna ideja je da se smanji veličina brojeva u multiskupu i to na način da se dva najveća broja zamijene njihovom razlikom. To znači da će svaka dva najveća broja ići u različit podskup, no bez odluke u koji će točno podskup svaki od brojeva ići. Na primjer, pogledajmo kako izgleda particioniranje skupa $\{19, 17, 13, 9, 6\}$. Ukoliko stavimo broj 19 u prvi podskup i broj 17 u drugi, tada je to ekvivalentno tome da stavimo broj 2 u prvi podskup. Nadalje, ukoliko odlučimo 19 staviti u drugi podskup a 17 u prvi, tada je to ekvivalentno stavljanju broja 2 u drugi podskup. Zaključujemo da ukoliko odlučimo da 19 i 17 idu u različite podskupove, tada ih je dovoljno zamijeniti njihovom razlikom. Zatim, na isti način, particioniramo skup $\{13, 9, 6, 2\}$.

Algoritam zasnovan na diferencijama se izvodi u dvije faze, pod uvjetom da je multiskup sortiran silazno. U prvoj se fazi dva najveća broja (prva dva s lijeve strane) zamjenjuju njihovom razlikom. Postupak ponavljamo dok ne dobijemo samo jedan broj, koji zapravo određuje razliku suma podskupova. U drugoj fazi se vraćamo unatrag te iz niza diferencijskih odluka generiramo particiju zamjenjujući broj koji predstavlja razliku dva broja s pripadnim brojevima koji generiraju tu razliku. Ukoliko je pročitani broj element početnog multiskupa on se fiksira u jedan od podskupova. Pri tome se veći broj pridjeljuje onom podskupu s manjom trenutnom sumom. Hod algoritma za skup $\{19, 17, 13, 9, 6\}$ vidimo u sljedećoj tablici.

skup za particioniranje	razlika	prvi podskup	drugi podskup	odstupanje
19 17 13 9 6	**2**	13 **19**	**17** 9 6	0
13 9 6 2	**4**	**13** 2	**9** 6	0
6 4 2	**2**	**4** 2	**6**	0
2 2	**0**	**2**	**2**	0
0			**0**	0

Vidimo da algoritam za ovaj skup generira savršenu particiju, no lako se pokaže da diferencijski algoritam ne daje uvjek optimalna rješenja. Jasno je da za skup čija savršena particija u jednom podskupu sadrži oba najveća elementa ovaj algoritam ne generira optimalno rješenje, jer ih on "sprema" u različite podskupove.

Modificirani Karmakar-Karpov diferencijski algoritam je zasnovan na modifikaciji diferencijskog algoritma na način da se u svakom koraku algoritma redom zamjenjuju po dva elementa njihovom razlikom. Time se u svakom koraku algoritma dobiva smanjenje veličine multiskupa za faktor dva i veličine brojeva skupa. Hod algoritma na već poznatom skupu $\{19, 17, 13, 9, 6\}$ opisuje sljedeća tablica.

skup za particioniranje	razlika	prvi podskup	drugi podskup	odstupanje
19 17 *13 9* 6	**2** *4*	**19** *13*	**17** *9* 6	0
6 *4* **2**	2	*4* **2**	6	0
2 2	**0**	**2**	2	0
0			**0**	0

O težini problema

Opisani algoritmi daju optimalno rješenje u mnogim slučajevima. Zbog toga i zbog jednostavne formulacije, za problem 2-particije multiskupa se koristi i naziv *najlakši teški problem*.

Kao što smo prilikom definiranja naveli, problem 2-particije multiskupa je težak. No, nije sasvim jasno pod kojim uvjetima opisani problem postaje uistinu težak. Za multiskupove od stotinjak tisuća prirodnih brojeva ne većih od 100 algoritmi gotovo uvijek pronalaze optimalno tješenje, dok je za multiskupove od 10-ak prirodnih brojeva većih od 2^{10} gotovo nemoguće pronaći savršenu particiju ukoliko ona postoji.

Dva faktora utječu na veličinu problema: koliko imamo brojeva u multiskupu i kolika je njihova veličina (broj bitova u binarnom prikazu). Preciznije rečeno, veličina problema zavisi o broju bitova potrebnih za njegovu reprezentaciju, a to pak zavisi o broju n pozitivnih cijelih brojeva u multiskupu i o broju m bitova u binarnom prikazu najvećeg od njih. Moglo bi se pretpostaviti da i težina problema raste kako rastu brojevi n i m, no pokazuje se da pravu informaciju o težini problema daje omjer $\frac{m}{n}$.

Ako je $n \gg m$ neće biti problematično naći savršenu particiju. No, ako je $m \gg n$ rješavanje problema postaje izuzetno nezahvalna zadaća. Prijelaz između lako i teško rješivih instanci problema događa se kada je $\frac{m}{n} \approx 1$. Otvoreno je pitanje točnog određivanja točke prijelaza.

Literatura

[1] THOMAS H. CORMEN, CHARLES E. LEIESERSON, RONALD L. RIVEST, CLIFORD STEIN, *Introduction to Algorithms*, The MIT press, 2009.

[2] BRIAN HAYES, *The Easiest Hard Problem*, (2002), `http://www.americanscientist.org/issues/pub/the-easiest-hard-problem/5`.

[3] STEPHAN MARTENS, *The Easiest Hard Problem: Number Partitioning*, (2003), `http://arxiv.org/ftp/cond-mat/papers/0310/0310317.pdf`.