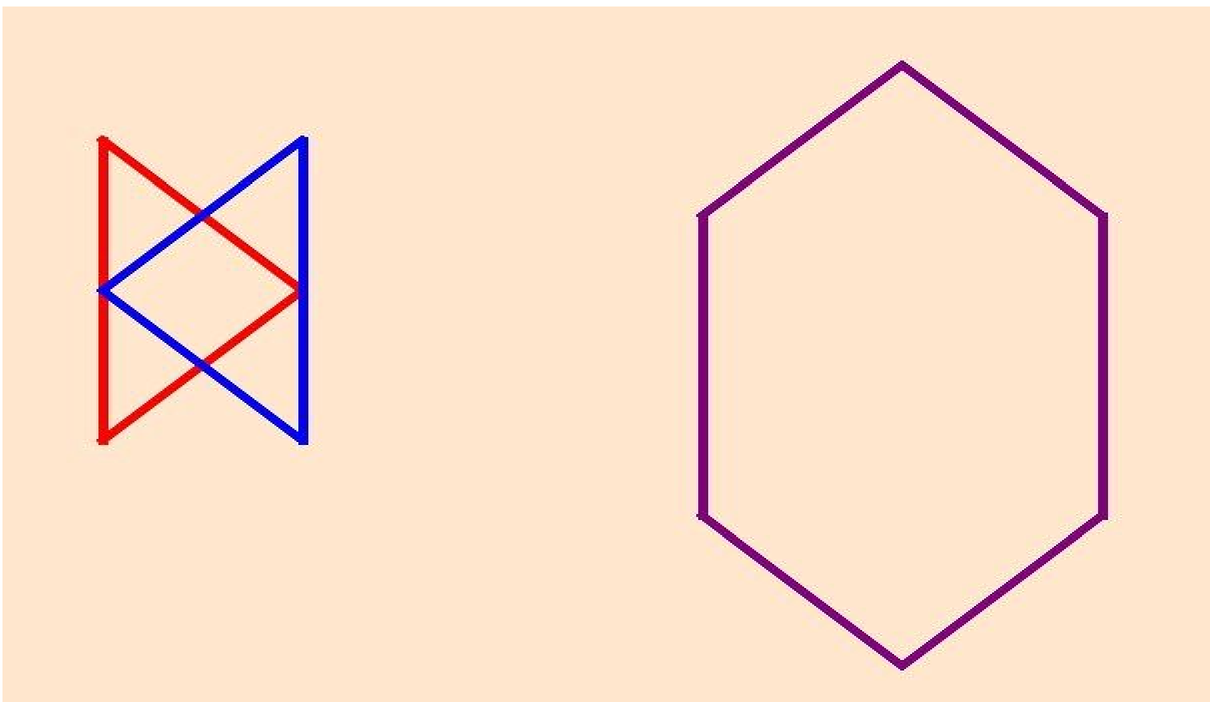
Algoritam za određivanje sume Minkovskog dva konveksna poligona u složenosti O(n+m)

Vojislav Stanković



Opis problema

U geometriji suma Minkovskog dva skupa pozicionih vektora N i M u Euklidskom prostoru formira se dodavanjem svakog vektora u N svakom vektoru u M.

Sabiranje Minkovskog ponaša se dobro u slučaju sabiranja konveksnih omotača što je prikazano u sledećoj tvrdnji:

Za sve podskupove S_1 i S_2 realnog vektorskog prostora, konveksni omotač njihovih zbirova predstavlja zbir njihovih konveksnih omotača $Conv(S_1 + S_2) = Conv(S_1) + Conv(S_2)$.

Uopštenje za svaki konačni niz skupova koji nisu prazni: $Conv(\sum S_n) = \sum Conv(S_n)$.

Primene

Sume Minkovskog su korisne geometrijske operacije koje se mogu koristiti na odgovarajući način za “podebljavanje” objekata. Na primer, popularan pristup pri planiranju kretanja robota predstavljenog poligonom u prostoriji sa poligonalnim preprekama podebljava svaku od prepreka tako što postavlja poligonalnu prepreku u sumu Minkovskog sebe i robota. Ukoliko pri kretanju robot seče konveksni omotač sume Minkovskog to znači da će ukoliko nastavi da se kreće u istom pravcu doći do kolizije sa poligonalnom preprekom i da treba da se kreće drugim putem. Ovo svodi problem kretanja robota od početka do cilja na problem nalaženja najkraće putanje.

Algoritam

Ulaz: *konveksni omotač poligona N i poligona M*

Izlaz: *konveksni omotač zbira poligona N i M*

Naivno rešenje problema

Naivno rešenje datog problema je prilično intuitivno. Ako se konveksni omotači predstave kao skupovi vektora, potrebno je izračunati zbir svakog para vektora iz skupa N i skupa M. Složenost ovog pristupa je ***O(nm)*** gde su ***n*** i ***m*** broj temena poligona N i M, respektivno.

U pseudokodu suma Minkovskog skupova ***N*** i ***M*** predstavljena je nizom ***sum***. Računa se suma svakog para vektora i dodaje u ***sum***.

Algorithm 1 Suma Minkovskog dva konveksna poligona

```
procedure MinkowskiSum(N, M)
    sum = {}
    for each vector n in N do
        for each vector m in M do
            sum.push_back(n + m)
        end for
    end for
end procedure
```

Napredni algoritam

Pretpostavimo da su date ivice poligona N i poligona M usmereni vektori a smer je npr. u smeru kazaljke na satu duž granice poligona. Kod konveksnog poligona ivice su već sortirane po polarnom uglu u odnosu na centar poligona. Zbog toga možemo da objedinimo uređenu sekvencu ivica poligona N i uređenu sekvencu ivica poligona M u uređenu sekvencu S u linearnom vremenu po broju temena poligona N i M. Time dobijamo zbir konveksnih omotača poligona N i M. Potrebno je još samo translirati ivice tako što ćemo početak jedne ivice nadovezivati na kraj prve ivice. Složenost ovog pristupa je O(n+m) gde su n i m broj temena poligona N i M.

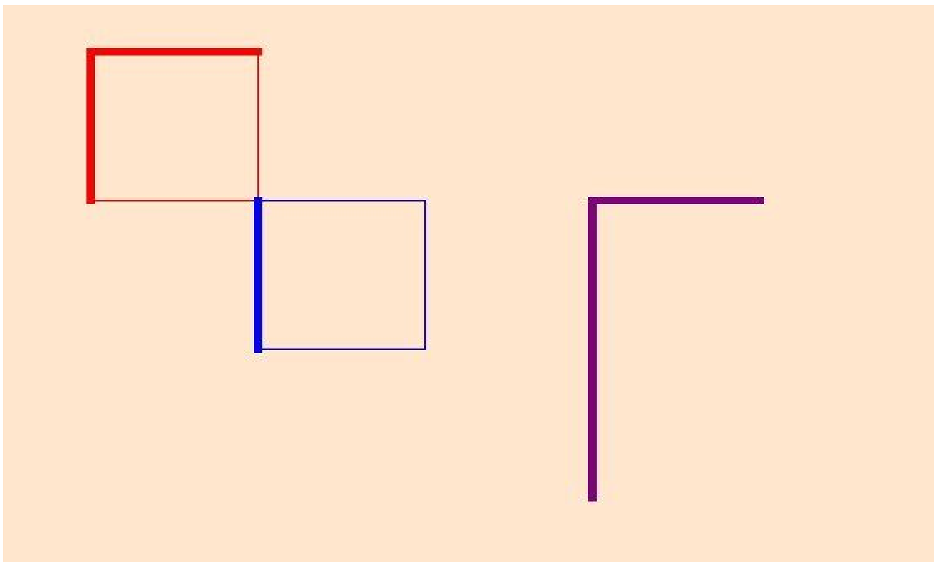
U pseudokodu vidimo da je prvo potrebno samo preurediti nizove N i M funkcijom ***shift*** koja pomera sve elemente niza u levo dok ne postavi vektor sa najmanjim polarnim uglom na početak niza tako da niz bude sortiran (nije potrebno sortiranje ***(2, 3, -1, 1) << (-1, 1, 2, 3)***) Nakon toga se u linearnom vremenu objedinjavaju ova dva sortirana niza funkcijom ***merge***. Potom se svi vektori transliraju na neku fiksnu poziciju da bi se iscrtao konveksni omotač.

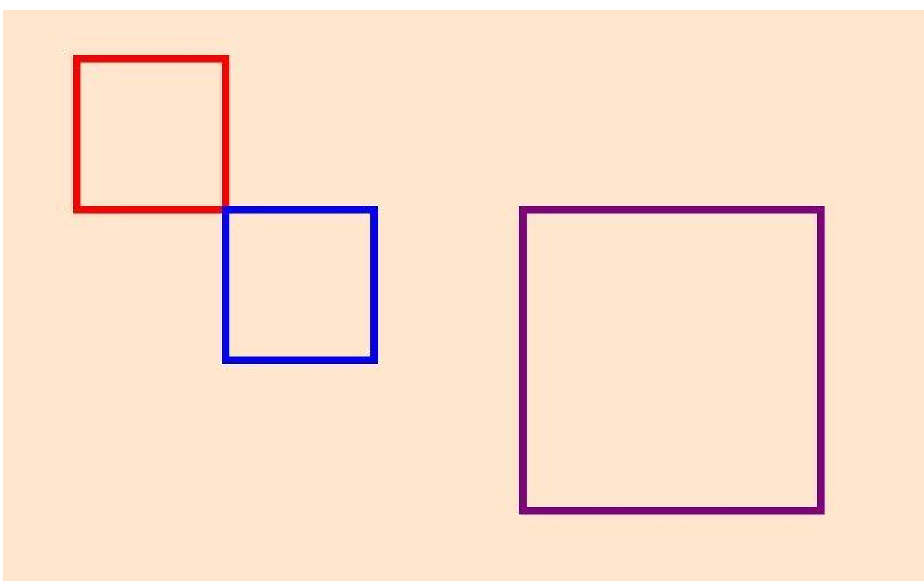
Algorithm 2 Suma Minkovskog dva konveksna poligona

procedure MinkowskiSumOptimal(N, M)
    $sum = \{\}$
    $shift(N)$
    $shift(M)$
    $sum = merge(N, M)$
    $sum = translate(sum)$
end procedure

Vizuelizacija algoritma

Na prvoj slici je prikazano transliranje trenutne (tamno crveno) ivice i nadovezivanje početka translirane ivice na kraj prethodne (ljubičasto) ivice. Druga slika prikazuje rezultujući konveksni omotač N+M (ljubičasto).





Poredjenje efikasnosti naivnog i naprednog algoritma

U tabeli je prikazano vreme izvršavanja u sekundama naivnog i naprednog algoritma za različite veličine ulaza.

alg./dim ulaza	100	1000	3000	5000
napredni	0.000006	0.000003	0.000004	0.000005
naivni	0.001716	0.142202	1.4665	3.96007

Testiranje ispravnosti algoritma

Naziv testa	Opis testa	Ulaz	Očekivani izlaz
noPolygons	Zadavanje oba poligona dimenzije 0. Program neće biti izvršen.	N=0, M=0,	INVALID_INPUT
emptyFirstPolygon	Zadavanje prvog poligona dimenzije 0. Program neće biti izvršen.	N=0, M=3	INVALID_INPUT
emptySecondPolygon	Zadavanje drugog poligona dimenzije 0. Program neće biti izvršen.	N=3, M=0	INVALID_INPUT
lessThanThree	Zadavanje bilo kog poligona dimenzije manje od 3. Program neće biti izvršen.	N=2, M=2	INVALID_INPUT
randomTest1	Veličina prvog poligona je 10 a drugog 5.	N=10, M=5	Konveksni omotač veličine N+M
premadeTest	Unapred poznati elementi niza ulaznih poligona.	N = [{100, 300, 100, 100}, {100, 100, 200, 200}, {200, 200, 100, 300}]	[{200, 300, 100, 200}, {100, 300, 100, 100}, {100, 200, 200, 100},

		M = [{100, 200, 200, 100}, {200, 100, 200, 300}, {200, 300, 100, 200}]	{100, 100, 200, 200}, {200, 100, 200, 300}, {200, 200, 100, 300}]