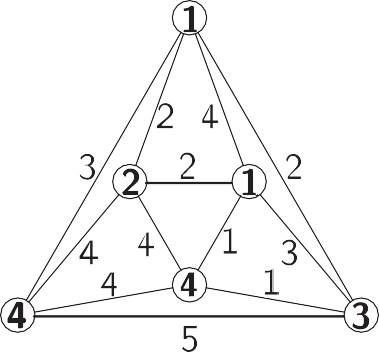
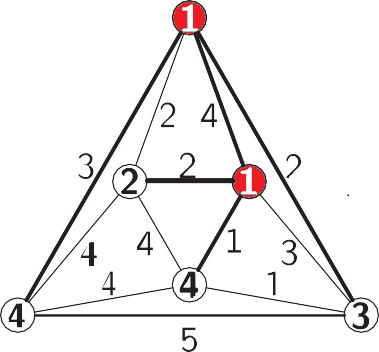
Opis problema

Težinski problem potpune dominacije je problem iz kombinatorne optimizacije. Dat je neusmjeren graf , gdje predstavlja skup čvorova, a skup ivica grafa . Ivica koja spaja čvorove se označava sa ili sa . Susjedstvo čvorova se definiše kao , a skup ivica identičnih sa čvorom je definisano kao . Za dati graf podskup čvorova se naziva dominantan skup ako je svaki čvorsusjedan bar jednom čvoru iz D, odnosno, ako za svaki čvor postoji bar jedan čvortakav da je . Drugim riječima, skup koji dobijemo kao rezultat ustvari predstavlja podskup čvorova tako da je svaki čvor u grafu, uključujući i čvorove u , susjedan čvoru u .

**Slika1.** Primjer težinskog problema potpune dominacije. Slika lijevo predstavlja težinski graf. Slika desno predstavlja optimalan totalni skup težinskog grafa

Matematička formulacija

gdje predstavlja skup ivica unutar skupa , a skup susjeda od .  
Dakle, data funkcija se sastoji od 3 dijela:  
1. Ukupne težine tjemena u   
2. Ukupne težine ivica u podgrafu   
3. Ukupna težina ivica minimalne težine koje povezuju svaki vrh sa vrhovima koji se nalaze u .

# Matematički model

Neka ja A matrica susjednosti grafa , pri čemu je . Koristimo binarnu promjenljivu xi  za označavanje da li je čvor izabran za rješenje ili ne.   
Neka je i . Tada minimalnu veličinu dominantnog skupa možemo dobiti na sledeći način:

s.t.

Data ograničenja obezbijeđuju da svaki čvor u grafu G ima susjeda koji je izabran, i predstavljaju ograničenja ukupnog dominantnog skupa.   
  
**ILP model za WTD problem :**   
Neka su {xv}, {ye} i {ze} skupovi binarnih promjenljivih. Za svaki čvor promjenljiva xi pokazuje da li je v izabrano kao rješenje ili ne. Za svako promjenljiva ye pokazuje da li je e izrabrano kao rješenje ili ne, dok promjenljiva ze predstavlja minimalnu vrijednost xu i xv. Tada funkciju cilja možemo formulisati na sledeći način:

s.t.

gdje je skup svih ivica iz čvora

Pohlepni algoritam

**Algoritam 1** Pohlepni algoritam

1: ***ulaz:*** *neusmjeren povezan težinski graf*

2:

3: *U D dodajemo čvor za koji ima*

4: ***dok*** *postoji čvor iz koji nije susjed ni jednom čvoru iz* ***radi***

5: *čvor v i granu (u,v) sa*

*dodajemo u podgraf D*

6: **kraj**

7: **izlaz:** Totalan dominantan skup grafa

Za ulazni graf , algoritam počinje sa praznim skupom , odnosno grafom koji ne sadrži ni jedan čvor. Prvo biramo prvi čvor koji dodajemo u graf S i to radimo tako što tražimo minimalan odnos stepena čvora i težine tog čvora za svaki čvor grafa. Ukoliko ovo nije totalni dominirajuć skup, nastavljamo da dodajemo čvorove dok se to ne desi. U svakom narednom koraku biramo čvor sa najmanjim zbirom odnosa stepena čvora i težine čvora i težine grane . Čvor biramo iz susjednih čvorova čvorevima koji su već u , tako da se taj čvor ne nalazi u . Novoizabrani čvor i granu dodajemo u naš graf . Poslije izvijesnog broja koraka dobijamo totalan dominantan skup početnog grafa .