Problem trgovskega potnika

1 Motivacija in opis problema

Problem trgovskega potnika (v nadaljevanju PTP, angl. *Traveling salesman problem – TSP*) je v 19. stoletju prvi matematično obravnaval irski matematik W. Hamilton, danes pa je verjetno najbolj proučevan problem kombinatorične optimizacije. Imamo n mest in množico povezav $i \to j$ z neko utežjo (dolžno, ceno) $d_{i,j}$. Zanima nas, po kateri poti moramo iti, da bomo obiskali vsako mesto natanko enkrat in se vrnili v izhodišče. Cilj je, da bo skupna cena (razdalja) poti najmanjša (najkrajša).

Lahko predpostavimo, da imamo opravka s polnim grafom. To pomeni, da med poljubnima mestoma obstaja usmerjena povezava. V primeru, da med mestoma i in j ni usmerjene povezave nastavimo $d_{i,j} = \infty$. Iščemo najkrajši Hamiltonov obhod tega grafa. Hamiltonov obhod je pot v grafu, ki gre skozi vse točke grafa natanko enkrat in se na koncu vrne v začetno.

Problem spada med NP - polne probleme, kar med drugim pomeni, da zanj še vedno ne poznamo algoritma, rešljivega v polinomskem času. Oznaka NP prihaja iz angleščine (angl. *nondeterministic polynomial*). V splošnem se časovna zahtevnost za ta problem veča eksponentno; če imamo n mest, potem je trajanje izvajanja algoritma sorazmerno s C^n , kjer je C neka konstanta.

2 ILP formulacija

Rešujemo problem z n mesti. Definirajmo spremenljivko

$$x_{i,j} = \begin{cases} 1, & \text{uporabimo pot } i \to j \\ 0, & \text{sicer} \end{cases}$$

in naj $d_{i,j}$ označuje ceno poti $i \to j$ in naj bo $d_{i,i} = \infty$ za $i = 1, \ldots, n$. Zapišimo problem v jeziku celoštevilskega linearnega programa.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j} d_{i,j}$$

p.p

(1) $\displaystyle\sum_{j=1}^{n} x_{i,j} = 1 \quad \text{za } i = 1, \ldots, n$

(2) $\displaystyle\sum_{i=1}^{n} x_{i,j} = 1 \quad \text{za } j = 1, \ldots, n$

(3) $x_{i,j} = \{0, 1\} \quad \text{za } i, j = 1, \ldots, n$

(4) Rešitev je natanko en cikel

Upoštevajoč le omejitve (1),(2) in (3) se problem prevede na problem najmnajšega prirejanja opravil, ki ga lahko rešimo npr. z madžarsko metodo. Zavedati se moramo, da ob neupoštevanju omejitve (4) lahko dobimo več manjših ciklov, ki zadoščajo prvim trem pogojem, ne pa tudi problemu, ki ga obravnavamo.

3 Posebni tipi TSP

- **Simetrični in asimetrični problem trgovskega potnika**

  Problem trgovskega potnika na grafu je simetrični PTP, saj se cene potovanje med točkami v obe smeri enake. Asimetrični PTP pa definiramo na digrafu, kar pomeni, da sta lahko ceni potovanja med dvema točkama različni, odvisno v katero smer potujemo. Prometne nesreče, enosmerne ulice, cene letalskih kart so samo nekateri primeri, ki pokažejo, da simetrični problem PTP ni vedno pravi model. Kljub temu, se bomo v nadaljevanju omejili na simetrične PTP.

- **Metrični problem trgovskega potnika**

  V metričnem PTP cenovna funkcija zadošča trikotniški neenakosti, tj. za poljubno trojico vozlišč (A, B, C) velja $d(A, C) \leq d(A, B) + d(B, C)$, kjer je z d dana razdalja na metričnem prostoru. Uvedba metrike pomeni, da je direktna povezava od A do B vedno najkrajša. Tako kot običajen PTP je tudi metrični PTP NP- težek problem.

  V posebnem za d vzamemo evklidsko metriko. Evklidski PTP je PTP, pri katerem so točke grafa točke v ravnini, cene pa so kar razdalje v običajni evklidski metriki. Evklidski PTP je poseben primer metričnega PTP, saj za poljubno metriko velja trikotniška neenakost. Izkaže se, da je tudi evklidski NP-težak problem.

4 Načini reševanja

Reševanja PTP se lahko lotimo s pomočjo eksaktnih algoritmov, ki nam najdejo točno (optimalno) rešitev, vendar pa so lahko neprimerni za velike probleme. V to skupino spadajo algoritmi, ki delujejo po načelu razveji in omeji. To pomeni, da za iskanje rešitve problema, ki ga skušamo rešiti s pomočjo tovrstnih algoritmov tvorimo drevo rešitev, ki jih nato sistematično pregledamo in poiščemo najboljšo rešitev. Lahko pa uporabimo hitrejše, hevristične algoritme, ki nam lahko najdejo zelo dober približek poti v sprejemljivem času, ne moremo pa zagotoviti, da je rezultat res optimalen. Najbolj direktna možna rešitev bi bila sledeča: preizkusimo vse možne permutacije (načine povezovanja oglišč v sklenjeno celoto) in izberemo najboljšo. Postopek ima eksponentno časovno zahtevnost $O(n!)$. Izkaže se za neučinkovitega že pri problemih ko $n = 10$.

V nadaljevanju bomo izmed hevrističnih algoritmov predstavili

- **Metoda najbližjega soseda** Hevristika najbližjega soseda ali t.i. požrešni algoritem deluje tako, da pri izdelavi poti vedno izbere najbližje mesto, ki ga še nismo obiskali kot točko premika iz trenutnega mesta. Poti, ki jih najdemo na ta način, so v povprečju za 25 % daljše od optimalnih, je pa precej hitra. Ima to omejitev, da je najkrajša pot odvisna od ogljišča v katerem začnemo.

- **Metoda obratnih podpoti**

Pri eksaktnih algoritmih se bomo osredotočili na **B&B algoritem**.