Operaciona istraživanja 2

- skripta za I kolokvijum 2012/13 (by Stepke) -

Poglavlje 8 - Teorija igara

8.1. Predmet teorije igara i osnovni pojmovi

Na primer, u igri šaha, rezultat igre ne zavisi samo od poteza jednog igrača već zavisi i od poteza drugog, a njihovi interesi su konfliktni jer svaka strana želi da pobedi drugu.

Ovakav slučaj neizvesnosti u odlučivanju nazivamo **igrom**, a oblast koja se bavi analizom ovakvih problema i nalaženjem optimalnih rešenja se naziva *teorijom igara*.

Igru definišu:

1) ***Igrači*** - koji predstavljaju strane u konfliktu
2) ***Dobitak*** (ili *gubitak*) - koji predstavlja rezultat igre
3) ***Skup strategija*** (poteza, alternativa) - koji predstavljaju ponašanje svakog igrača

Uzroci neizvesnosti rezultata igre su:

1) *Kombinatornost* - postoji izuzetno veliki broj varijanti odvijanja igre tako da je nemoguće predvideti njen rezultat
2) Prisustvo *slučajnih faktora* (*hazardne igre*, kao na primer: igre kockom, rulet i sl.)
3) *Odsustvo informacija* o *mogućim akcijama protivnika* (tj. o *protivničkoj strategiji*)

Prema broju igrača razlikujemo igre:

- Dva igrača (ili strane)
- Više igrača (ili strana)

Prema broju raspoloživih strategija razlikujemo:

- Konačne igre
- Beskonačne igre

Prema rezultatu igre razlikujemo:

- Igre nulte sume (dobitak jednog igrača = gubitku drugog igrača)
- Igre nenulte sume (dobitak jednog igrača ≠ gubitku drugog igrača)

Postoje dva osnovna načina predstavljanja igara:

- Normalna (strateška) forma - kada igrači istovremeno povlače svoje poteze (mogući ishodi igre se tada predstavljaju u obliku matrice)
- Ekstezivna forma - kada igrači naizmenično povlače svoje poteze (mogući ishodi igre se tada predstavljaju u obliku grafa, tj. stabla)

8.2. Matrične igre

8.2.1. Matrične igre nulte sume

Igre nulte sume dve strane sa konačnim brojem strategija se mogu pogodno razmatrati predstavljajući funkciju $C(a_i, b_j)$ kao sledeću matricu (*matrica plaćanja*):

$$C = \begin{bmatrix} c11 \ \ c12 \dots \ c1n \\ c21 \ \ c22 \dots c2n \\ . \\ . \\ . \\ cm1 \ \ cm2 \dots cmn \end{bmatrix}$$

gde elementi:

$$\mathbf{c_{ij} = C(a_i, b_j)}$$

predstavljaju dobitak **I** igrača (koji bira strategiju $\mathbf{a_i}$), odnosno gubitak **II** igrača (koji bira strategiju $\mathbf{b_j}$).

Normalna forma igre dva igrača sa nultom sumom je trojka (**A**, **B**, **C**) gde je:

1. **A** = {$\mathbf{a_i}$} , (**i** = 1, 2, …, **m**), neprazan skup strategija igrača **I**
2. **B** = {$\mathbf{b_j}$} , (**j** = 1, 2, …, **n**), neprazan skup strategija igrača **II**
3. **C** je funkcija definisana na Dekartovom proizvodu **A x B**, tako da je $\mathbf{c_{ij} = C(a_i, b_j)}$, realan broj za svako $\mathbf{a_i} \in \mathbf{A}$ i $\mathbf{b_j} \in \mathbf{B}$

Ako igrač **I** izabere strategiju $\mathbf{a_i}$, tj. bira red **i** u matrici plaćanja, tada u najgorem slučaju on ima osiguran dobitak:

$$\boldsymbol{\alpha_i = \min_j c_{ij}}$$

koji je jednak minimalnom od svih rezultata koji se dobijaju za sve moguće strategije igrača **II**.

Igrač **I** želi da maksimizira vrednost dobitka $\boldsymbol{\alpha_i}$, te bira onu strategiju $\mathbf{a_i}$ kojom maksimizira minimalni dobitak:

$$\boldsymbol{\alpha = \max_i \min_j c_{ij}}$$

Veličina **α** koja predstavlja garantovani dobitak igrača **I** se naziva *donja granica vrednosti igre*.

Slično, igrač **II** birajući svoju strategiju $\mathbf{b_j}$, tj. kolonu **j** u matrici plaćanja, ima osigurano da njegov gubitak bude maksimalno:

$$\boldsymbol{\beta_j = \max_i c_{ij}}$$

bez obzira koju strategiju bira igrač **I**.

Želeći da njegov gubitak bude minimalan, igrač **II** bira svoju strategiju $\mathbf{b_j}$ na osnovu *minmax* principa, tako da osigurava da njegov gubitak ne bude veći od:

$$\boldsymbol{\beta} = \min_{j} \max_{i} c_{ij}$$

Veličina **β** predstavlja najveći mogući gubitak igrača **II** i naziva se *gornja granica vrednosti igre*.

Može se pokazati da je uvek:

$$\boldsymbol{\alpha \leq \beta}$$

jer je očigledno da za bilo koje **k** i **l** važi:

$$\min_{j} c_{kj} \leq c_{kl} \leq \max_{i} c_{il}$$

8.2.2. Proste matrične igre

Najjednostavniji slučaj matričnih igara nulte sume su *proste matrične igre* koje poseduju *sedlastu tačku*.

Naime, ukoliko neki element $\mathbf{c_{ij}}$ matrice plaćanja **C** ima osobinu da je:

1) $\mathbf{c_{ij}}$ minimalan element u redu **i** matrice **C**
2) $\mathbf{c_{ij}}$ maksimalan element u koloni **j** matrice **C**

tada kažemo da je $\mathbf{c_{ij}}$ **sedlasta tačka** matrice plaćanja **C**

Ako u matrici plaćanja $\mathbf{C} = \{\mathbf{c_{ij}}\}$, (**i** = 1, …, **m** ; **j** = 1, …, **n**) postoji par (**i***, **j***) takav da je za svako (**i** = 1, …, **m** ; **j** = 1, …, **n**):

$$\mathbf{c_{ij^*} \leq c_{i^*j^*} \leq c_{i^*j}}$$

tada je par (**i***, **j***) sedlasta tačka i određuje *par optimalnih strategija* u matrici **C**.

Drugim rečima, u sedlastoj tački važi:

$$\max_{i} \min_{j} \mathbf{c_{ij}} = \min_{j} \max_{i} \mathbf{c_{ij}} = \mathbf{c_{i^*j^*}}$$

Znači, kod proste matrične igre optimalna vrednost igre se nalazi u sedlastoj tački i za oba igrača je optimalno da biraju strategije koje definišu sedlastu tačku.

8.2.3. Mešovite matrične igre

Teorema minimaksa:

1. Postoji realni broj **v** koji se naziva *vrednost igre*
2. Postoji mešovita strategija za igrača **I** koja mu osigurava najveći očekivani minimalni dobitak jednak vrednosti igre **v** bez obzira koju mešovitu strategiju igra igrač **II**
3. Postoji mešovita strategija za igrača **II** koja mu osigurava najmanji očekivani maksimalni gubitak jednak vrednosti igre **v** bez obzira koju mešovitu strategiju igra igrač **I**
4. Bilo koja matrična igra sa matricom plaćanja **C** ima sedlastu tačku u prostoru mešovitih strategija, tj. postoje vektori verovatnoća **p** i **q** takvi da je:

$$\max_{p}\min_{q} \mathbf{p}^T\mathbf{C}\mathbf{q} = \min_{q}\max_{p} \mathbf{p}^T\mathbf{C}\mathbf{q} = \mathbf{v}$$

Ako je vrednost igre **v** pozitivna, ona predstavlja dobitak igrača **I**, odnosno gubitak igrača **II**, i obrnuto ukoliko je negativna. Za igru kažemo da je fer ako je $\mathbf{v} = \mathbf{0}$.

Mešovita startegija za igrača **I** se može izraziti vektorom verovatnoća:

$$\mathbf{p} = (\mathbf{p_1} , \mathbf{p_2} , \ldots, \mathbf{p_i} ,\ldots, \mathbf{p_m})$$

gde $\mathbf{p_i}$ predstavlja učestanost sa kojom igrač **I** igra strategiju $\mathbf{a_i}$.

Mešovita startegija za igrača **II** se može izraziti vektorom verovatnoća:

$$\mathbf{q} = (\mathbf{q_1} , \mathbf{q_2} , \ldots, \mathbf{q_j} ,\ldots, \mathbf{q_n})$$

gde $\mathbf{q_j}$ predstavlja učestanost sa kojom igrač **II** igra strategiju $\mathbf{b_j}$.

S obzirom na definiciju verovatnoća, važi:

$$\sum_{i=1}^{m} pi = 1$$

$$\sum_{j=1}^{n} qj = 1$$

Očekivani rezultat igrača **I** kada igra mešovitu strategiju predstavljenu vektorom **p** u slučaju da igrač **II** igra strategiju $\mathbf{b_j}$ je:

$$\sum_{i=1}^{m} pi \cdot cij$$

Očekivani rezultat igrača **II** kada igra mešovitu strategiju predstavljenu vektorom **q** u slučaju da igrač **I** igra strategiju $\mathbf{a_i}$ je:

$$\sum_{j=1}^{n} qj \cdot cij$$

U opštem slučaju, ako igrač **I** igra mešovitu strategiju **p** , a igrač **II** mešovitu strategiju **q** , očekivani rezultat igre je:

$$\mathbf{p^T C q} = \sum_{i=0}^{m} \sum_{j=0}^{n} pi \cdot qj \cdot cij$$

8.2.7. Rešavanje matričnih igara linearnim programiranjem

Sa stanovišta igrača **I** imamo sledeći matematički program za koji je potrebno naći vektor **p** = (p_1 , p_2 , ... , p_m) :

$$(\max)\ \min_{j} \sum_{i=1}^{m} pi \cdot cij$$

p.o.

$$\sum_{i=1}^{m} pi = 1$$

$$p_i \geq 0\ ,\ i = 1, 2, ..., m$$

Ovaj matematički program ima linearno ograničenje, ali funkcija cilja nije linearna jer je definisana uz pomoću operatora minimuma. Međutim, ovaj matematički program se može učiniti linearnim uvođenjem nove promenljive **v** koja predstavlja vrednost igre. Pokazano je da za vrednost igre važi sledeća nejednakost:

$$v \leq \min_{j} \sum_{i=1}^{m} pi \cdot cij$$

te se gornji nelinearni program pretvara u sledeći linearni:

$$\max v$$

p.o.

$$v \leq \sum_{i=1}^{m} pi \cdot ci1$$
$$v \leq \sum_{i=1}^{m} pi \cdot ci2$$
$$.$$
$$.$$
$$v \leq \sum_{i=1}^{m} pi \cdot cin$$

$$\sum_{i=1}^{m} pi = 1$$

$$p_i \geq 0\ ,\ i = 1, 2, ..., m$$

Ovo je sada linearni program koji se može rešavati nekom od poznatih metoda.

Zadatak određivanja optimalne strategije za igrača I se može prevesti u uobičajeni oblik linearnog programa uvođenjem smene:

$$x_i = \frac{pi}{v}$$

uz pretpostavku da je vrednost igre **v** > 0 .

S obzirom da je:

$$\sum_{i=1}^{m} pi = 1$$

sledi

$$\sum_{i=1}^{m} xi = \frac{1}{v}$$

pa je maksimizacija vrednosti igre **v** identična sa minimizacijom sume promenljivih $\mathbf{x_i}$, te imamo za rešavanje sledeći linearni program:

min $\mathbf{x_1} + \mathbf{x_2} + \ldots + \mathbf{x_m}$

p.o.

$$\sum_{i=1}^{m} ci1 \cdot xi \geq 1$$

$$\sum_{i=1}^{m} ci2 \cdot xi \geq 1$$

.

.

$$\sum_{i=1}^{m} cin \cdot xi \geq 1$$

$\mathbf{x_i} \geq 0$, $\mathbf{i} = 1, 2, \ldots, \mathbf{m}$

Rešavanjem ovog linearnog problema lako se dobijaju vrednosti igre **v** originalne matrice plaćanja, kao i vektor mešovitih strategija **p** .

Analognim rezonovanjem, linearni program za igrača **II** postaje:

$$\max \mathbf{y_1} + \mathbf{y_2} + \dots + \mathbf{y_n}$$

p.o.

$$\sum_{j=1}^{n} c1j \cdot yj \leq 1$$

$$\sum_{j=1}^{n} c2j \cdot yj \leq 1$$

.

.

$$\sum_{j=1}^{n} cmj \cdot yj \leq 1$$

$\mathbf{y_j} \geq 0$, $\mathbf{j} = 1, 2, \dots, \mathbf{n}$

8.3.1. Definicija bimatrične igre

Kod ove vrste igara, suma nije konstantna, odnosno nije jednaka 0 .
Drugim rečima, dobitak jednog igrača nije jednak gubitku drugog igrača.

Nju definišu dva konačna skupa strategija: **x** ϵ **X** za igrača **I** i **y** ϵ **Y** za igrača **II**, i dve funkcije sa realnim vrednostima $\mathbf{y_1}$ (**x** , **y**) i $\mathbf{y_2}$ (**x** , **y**) koje predstavljaju rezultate igre igrača **I** i igrača **II**.

Ovakva igra se modelira bimatricom plaćanja **C** čiji su elementi uređeni parovi ($\mathbf{y_1}$ (**x** , **y**) , $\mathbf{y_2}$ (**x** , **y**)) :

$$\mathbf{C} = \begin{bmatrix} (\mathrm{u1}\ (\mathrm{x1}\ ,\mathrm{y1}), \mathrm{u2}\ (\mathrm{x1}\ ,\mathrm{y1})) & (\mathrm{u1}\ (\mathrm{x1}\ ,\mathrm{y2})\ ,\mathrm{u2}\ (\mathrm{x1}\ ,\mathrm{y2}))\ \ldots\ (\mathrm{u1}\ (\mathrm{x1}\ ,\mathrm{yn})\ ,\mathrm{u2}\ (\mathrm{x1}\ ,\mathrm{yn})) \\ (\mathrm{u1}\ (\mathrm{x2}\ ,\mathrm{y1}), \mathrm{u2}\ (\mathrm{x2}\ ,\mathrm{y1})) & (\mathrm{u1}\ (\mathrm{x2}\ ,\mathrm{y2})\ ,\mathrm{u2}\ (\mathrm{x2}\ ,\mathrm{y2}))\ \ldots\ (\mathrm{u1}\ (\mathrm{x2}\ ,\mathrm{yn})\ ,\mathrm{u2}\ (\mathrm{x2}\ ,\mathrm{yn})) \\ & . \\ & . \\ & . \\ (\mathrm{u1}\ (\mathrm{xm}\ ,\mathrm{y1}), \mathrm{u2}\ (\mathrm{xm}\ ,\mathrm{y1})) & (\mathrm{u1}\ (\mathrm{xm}\ ,\mathrm{y2})\ ,\mathrm{u2}\ (\mathrm{xm}\ ,\mathrm{y2}))\ \ldots\ (\mathrm{u1}\ (\mathrm{xm}\ ,\mathrm{yn})\ ,\mathrm{u2}\ (\mathrm{xm}\ ,\mathrm{yn})) \end{bmatrix}$$

Ukoliko označimo sa:

$$\mathbf{a_{ij}} = \mathbf{y_1}\ (\mathbf{x_i}\ ,\ \mathbf{y_j})$$
$$\mathbf{b_{ij}} = \mathbf{y_2}\ (\mathbf{x_i}\ ,\ \mathbf{y_j})$$

tada se bimatrica plaćanja **C** može dekomponovati na dve obične matrice:

$$\mathrm{A} = \begin{bmatrix} a11\ \ a12\ \ldots\ a1n \\ a21\ \ a22\ \ldots a2n \\ . \\ . \\ . \\ am1\ \ am2\ \ldots amn \end{bmatrix}$$

$$\mathrm{B} = \begin{bmatrix} b11\ \ b12\ \ldots\ b1n \\ b21\ \ b22\ \ldots b2n \\ . \\ . \\ . \\ bm1\ \ bm2\ \ldots bmn \end{bmatrix}$$

Rešavanje igara nenulte sume ili bimatričnih igara je složenije od igara nulte sume, a i princip minimaksa se ovde ne može primeniti.

Jedna od najpoznatijih bimatričnih igara je takozvana „dilema zatvorenika“ (pročitati iz knjige na stranicama 342. i 343.)

Poglavlje 9 - Mrežno planiranje

9.1. Uvod

Međunarodna komisija za standarde preporučuje sledeću definiciju *projekta*:

Projekat je jedinstveni proces, koji se sastoji od skupa koordinisanih i kontrolisanih aktivnosti, sa određenim datumom početka i završetka, koje se preduzimaju da bi se isporučio proizvod u skladu sa postavljenim zahtevima, pri čemu postoje ograničenja na vreme, troškove i resurse.

Ciljevi (*zahtevi*) *projekta* se jasno iskazuju na početku i treba da opišu ono što se želi ili mora postići i treba da budu izraženi u merljivim pojmovima. Oni mogu biti:

- *Minimalno vreme*
- *Minimalni troškovi*
- *Minimalni utrošci resursa*
- Postići *zahtevani kvalitet*

Upravljanje projektom je proces kojim se sprovodi:

- *Planiranje projekta*
- *Praćenje realizacije* (uz određena podešavanja plana)
- *Analiziranje*, *procenjivanje* i *izveštavanje o rezultatima*

Rukovodilac projekta je osoba sa odgovornošću za upravljanje projektom i postizanje postavljenih ciljeva.

Projekat se uvek tiče većeg broja pojedinaca ili grupa koji su zainteresovani za rezultate, efekte i ostvarenja projekata. Oni se nazivaju *učesnici na projektu* ili *interesne grupe* (*stakeholders*) i obuhvataju:

- *Korisnike*
- *Vlasnike*
- *Partnere*
- moguće *snabdevače* i *podugovarače*
- *Interne učesnike*
- *Društvo*

Za planiranje realizacije projekta razvijen je skup metoda koje se nazivaju ***tehnike mrežnog planiranja*** (***TMP***).

*** Bitne skraćenice koje treba znati:

1. **TMP** - *tehnike mrežnog planiranja*
2. **CPM** (Critical Path Metod) - *metod kritičnog puta*
3. **PERT** (Programme Evaluation and Review Technique) - *tehnika za ocenu* i *pregled* (*reviziju*) *programa*
4. **PDM** (Precedance diagramming Method) - *metoda za analizu vremena*
5. **MDČ** - *mrežni dijagram sa aktivnostima na čvorovima*
6. **MDG** - *mrežni dijagram sa aktivnostima na granama*

9.5. Analiza vremena

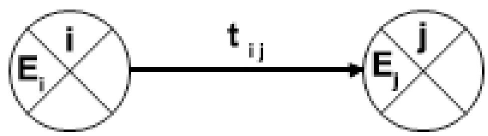
Analiza vremena obuhvata procenu i utvrđivanje vremena potrebnog za izvršenje pojedinih aktivnosti, kao i određivanje vremenskih parametara na osnovu kojih se može kontrolisati vremensko odvijanje projekta i uticati na održavanje rokova.

U proceduri određivanje trajanja projekta, u prostor koji označava čvor grafa upisuju se podaci koji se koriste u daljoj analizi. Koji će se podaci upisivati i gde zavisi od konvencije, a dva najčešća načina su:

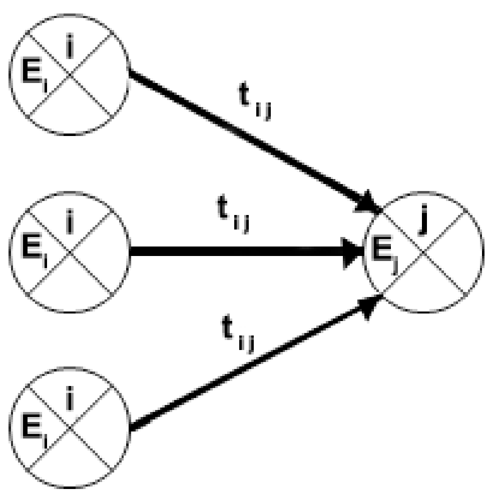
gde je:

- **i** - broj čvora
- E_i - najraniji trenutak događaja
- L_i - najkasniji trenutak događaja
- S_i - vremenska rezerva

U prvoj fazi primene metode kritičnog puta određuju se *najraniji trenuci događaja*:



$$E_j = E_i + t_{ij}$$



$$E_j = \max_{i} \{ E_i + t_{ij} \}$$

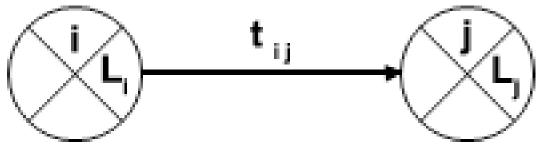
U opštem slučaju je:

$$E_j = \max_{i \in \Gamma^{-1}_j} \{ E_i + t_{ij} \} \quad , j \in N \backslash 1 , E_1 = 0$$

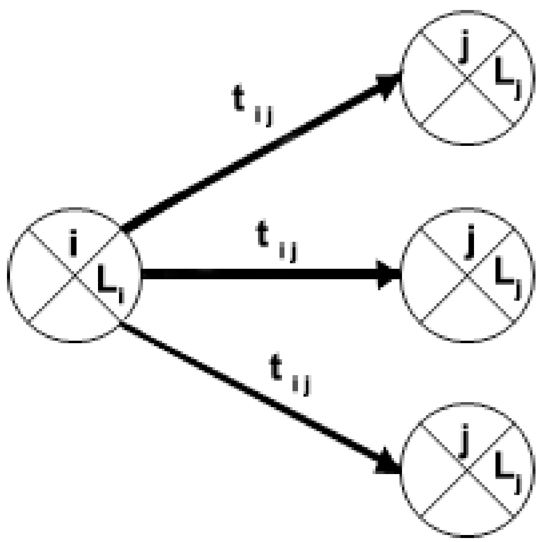
Za čvor **i** = **n** , koji označava završetak projekta, dobija se E_n tako da je trajanje projekta $T_n = E_n$.

U drugoj fazi metode kritičnog puta traže se odgovori na pitanje da li sme da zakasni završetak svake pojedine aktivnosti, i ako sme, koliko je dozvoljeno a da to ne izazove kašnjenje projekta.
Aktivnosti čije kašnjenje nije dozvoljeno nazivaju se **kritične aktivnosti**, a put koje one sačinjavaju **kritičan put**.

U ovoj fazi primene metode kritičnog puta određuju se *najkasniji trenuci događaja*:



$$\mathbf{L_i = L_j - t_{ij}}$$



$$\mathbf{L_i = \min_{j} \{ L_j - t_{ij} \}}$$

U opštem slučaju je:

$$\mathbf{L_i = \min_{j \in \Gamma_i} \{ L_j - t_{ij} \}} \quad , \mathbf{i} \in \mathbf{N \backslash n} \text{ , } \mathbf{L_n = E_n}$$

Za čvor **i** = **n** , koji označava završetak projekta, po konvenciji se stavlja da je $\mathbf{L_i} = \mathbf{E_i} = \mathbf{T\pi}$.

Pošto se odrede najranija i najkasnija vremena za svaki čvor ($\mathbf{E_i}$ i $\mathbf{L_i}$), prelazi se na računanje vremeskih rezervi za događaje $\mathbf{S_i}$:

$$\mathbf{S_i = L_i - E_i}$$

Za kritični put važi da su:

- vremena *najranijeg trenutka događaja* i *najkasnijeg trenutka događaja* jednaki $\mathbf{E_i} = \mathbf{L_i}$
- na kritičnom putu nema vremenskih rezervi događaja $\mathbf{S_i} = 0$

9.5.2. Analiza vremenskih rezervi

Za svaku aktivnost potrebno je odrediti trenutke najranijeg i najkasnijeg početka aktivnosti (**i, j**) (oznake $\mathbf{E_s(i, j)}$ i $\mathbf{L_s(i, j)}$), kao i najranijeg i najkasnijeg završetka aktivnosti (**i, j**) (oznake $\mathbf{E_f(i, j)}$ i $\mathbf{L_f(i, j)}$), tako da trajanje projekta ostane nepromenjeno.

1. *Najraniji početak* aktivnosti (**i, j**) (oznaka $\mathbf{E_s(i, j)}$) je:

a) za početnu aktivnost: $\mathbf{E_s}(1, \mathbf{j}) = 0$

b) za aktivnost (**i, j**): $\mathbf{E_s(i, j) = E_i}$

Drugim rečima, $\mathbf{E_s(i, j)}$ je maksimum najranijih završetaka aktivnosti koje prethode aktivnosti (**i, j**):

$$\mathbf{E_s(i, j)} = \max_{\mathbf{l} \in \Gamma^{-1}_{\mathbf{i}}} \mathbf{E_f(l, i)}$$

2. *Najkasniji početak* aktivnosti (**i, j**) (oznaka $\mathbf{L_s(i, j)}$) je:

a) za završnu aktivnost: $\mathbf{L_s(i, j) = L_f(i, j) - t_{ij}}$

b) za ostale aktivnosti: $\mathbf{L_s(i, j) = L_f(i, j) - t_{ij}}$

3. *Najraniji završetak* aktivnosti (**i, j**) (oznaka $\mathbf{E_f(i, j)}$) je:

a) za početnu aktivnost: $\mathbf{E_f}(1, \mathbf{j}) = \mathbf{t_{1j}}$

b) za ostale aktivnosti: $\mathbf{E_f(i, j) = E_s(i, j) + t_{ij}}$

4. *Najkasniji završetak* aktivnosti (**i, j**) (oznaka $\mathbf{L_f(i, j)}$) je:

a) za završnu aktivnost: $\mathbf{L_f(i, j) = E_f(i, j) = T\pi}$

b) za ostale aktivnosti: $\mathbf{L_f(i, j) = L_j}$

Na osnovu ovoga, računaju se tri različite vremenske rezerve za aktivnosti:

1. *Ukupna vremenska rezerva* $\mathbf{F_t}$ aktivnosti (**i, j**) govori koliko se najviše može promeniti trajanje aktivnosti, a da se trajanje projekta ne promeni (pod uslovom da se trajanja drugih aktivnosti ne menjaju). Računa se na sledeći način:

$$\mathbf{F_t(i, j) = L_f(i, j) - E_f(i, j)}$$
$$\mathbf{F_t(i, j) = L_j - E_i - t_{ij}}$$

2. *Slobodna vremenska rezerva* $\mathbf{F_f}$ aktivnosti (**i, j**) pokazuje koliko je maksimalno moguće produžiti aktivnost (**i, j**), a da to ne utiče na početak sledeće aktivnosti. Računa se na sledeći način:

a) za završnu aktivnost:

$$\mathbf{F_f(i, j)} = 0$$

b) za ostale aktivnosti:

$$\mathbf{F_f(i, j) = E_j - E_i - t_{ij}}$$

3. *Nezavisna vremenska rezerva* $\mathbf{F_i}$ aktivnosti (**i, j**) ne zavisi od početka i završetka drugih aktivnosti i po konvenciji je nenegativna, pa se računa po obrascu:

$$\mathbf{F_i(i, j)} = \max \{ 0, \mathbf{E_j - L_i - t_{ij}} \}$$

Svaka aktivnost koja ima ovu rezervu može se produžiti za taj iznos, a da to ne utiče na trajanje projekta.

9.5.3.1 Procena trajanja aktivnosti

U mrežnom planiranju je pri primeni metode **PERT** (metoda *stohastičkog* karaktera) ustaljeno da se za trajanje aktivnosti daju sledeće tri procene:

a - optimistička procena trajanja aktivnosti (vreme koje bi se postiglo ako bi pri izvršavanju aktivnosti sve išlo na najbolji mogući način)

m - procena najverovatnijeg trajanja aktivnosti (vreme koje ima najveću verovatnoću događanja, i ne treba ga mešati sa prosečnim trajanjem ili matematičkim očekivanjem)

b - pesimistička procena trajanja aktivnosti (vreme koje bi se postiglo ako bi ostvarile sve nepovoljne okolnosti)

Tri navedene procene trajanja aktivnosti se koriste za opisivanje funkcije raspodele.

Na osnovu centralne granične teoreme može se dokazati sledeće:

1. Trajanje projekta (**Tπ**) je slučajna promenljiva približno raspodeljena po zakonu Normalne raspodele.

2. Matematičko očekivanje trajanja projekta **Tπ** jednako je zbiru matematičkih očekivanja trajanja aktivnosti na kritičnom putu **P**:

$$\mathbf{T\pi} = \sum_{(i,j)\,\epsilon\, P} \mathbf{t_{ij}}$$

3. Varijansa trajanja projekta $\boldsymbol{\delta\pi^2}$ je jednaka zbiru varijansi trajanja aktivnosti na kritičnom putu P:

$$\boldsymbol{\delta\pi^2} = \sum_{(i,j)\,\epsilon\, P} \boldsymbol{\delta_{ij}^2}$$

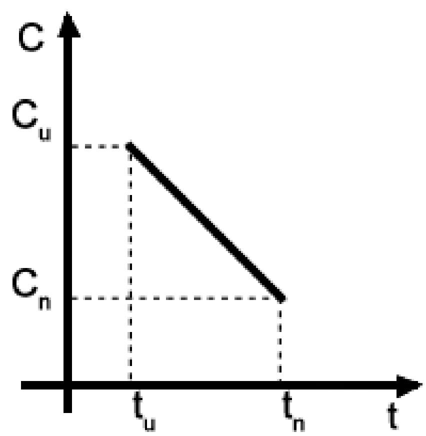
9.6. Analiza vremena i troškova

9.6.1. Normalno i usiljeno trajanje aktivnosti

Normalno trajanje aktivnosti t_n se ostvaruje normalnom upotrebom resursa i izaziva normalne troškove C_n

Usiljeno trajanje aktivnosti t_u odgovara *najkraćem mogućem trajanju* i ostvaruje se uz maksimalno korišćenje resursa i odgovarajuće usiljene troškove C_u

Imajući na umu ove predpostavke, za svaku aktivnost na projektu računa se nagib linearne funkcije:



$$\Delta C = \frac{Cu - Cn}{tn - tu} = \left|\frac{Cn - Cu}{tn - tu}\right|$$

Vrednost **Δ C** predstavlja priraštaj troškova na aktivnosti kada se trajanje aktivnosti smanji za jedinicu.
Naziva se još i jedinični priraštaj troškova, jedinični trošak ili marginalni (granični) trošak.

Veće **Δ C** odgovara većoj strmini krive i znači da su za jedinicu skraćenja trajanja aktivnosti potrebna veća sredstva.
Manje **Δ C** znači da je skraćenje trajanja aktivnosti moguće postići uz manje dodatne troškove.

9.6.2. Minimizacija troškova pri zahtevanom trajanju projekta

Za problem minimizacije troškova projekta kada je njegovo trajanje zadato, može se formulisati sledeći model linearnog programiranja:

$$\min \mathbf{C}(\mathbf{t}) = \sum_{(i-j)} (\mathbf{C}_n)_{ij} + \Delta\mathbf{C}_{ij}\,((\mathbf{t}_n)_{ij} - \mathbf{t}_{ij})$$

p.o.

$$\mathbf{E}_j - \mathbf{E}_i - \mathbf{t}_{ij} \geq 0 \quad \forall(i, j)$$

$$(\mathbf{t}_u)_{ij} \leq \mathbf{t}_{ij} \leq (\mathbf{t}_n)_{ij} \quad \forall(i, j)$$

$$\mathbf{E}_1 = 0$$

$$\mathbf{E}_n = \mathbf{T}_0 \quad \text{gde } \mathbf{T}_0 \in (m, M)$$

gde je **m** trajanje projekta kada sve aktivnosti imaju usiljeno trajanje, a **M** trajanje projekta kada sve aktivnosti imaju normalno trajanje.

Zadatak je da se odrede vrednosti trajanja svih aktivnosti $\mathbf{t}_{ij}$ i vrednosti za trenutke odigravanja svih događaja na mrežnom dijagramu ($\mathbf{E}_i$) tako da se minimiziraju ukupni troškovi projekta $\mathbf{C}(\mathbf{t})$.

9.6.3. Minimizacija trajanja projekta pri dozvoljenim troškovima

Za problem minimizacije trajanja projekta kada su zadati troškovi njegove realizacije, može se formulisati sledeći model linearnog programiranja:

$$\min \mathbf{T} = \mathbf{E}_n$$

p.o.

$$\sum_{(i-j)} (\mathbf{C}_n)_{ij} + \Delta\mathbf{C}_{ij}\,((\mathbf{t}_n)_{ij} - \mathbf{t}_{ij}) \leq \mathbf{C}_0$$

$$\mathbf{E}_j - \mathbf{E}_i - \mathbf{t}_{ij} \geq 0 \quad \forall(i, j)$$

$$(\mathbf{t}_u)_{ij} \leq \mathbf{t}_{ij} \leq (\mathbf{t}_n)_{ij} \quad \forall(i, j)$$

$$\mathbf{E}_1 = 0$$

Ovde treba odrediti minimalno trajanje projekta **T** , gde su ukupni troškovi projekta ($\mathbf{C}_0$) ograničeni.

Poglavlje 13 - Heuristčke metode

13.2. Pojam heuristike

Heuristika je tehnika koja pokušava da nađe neka “dobra” rešenja problema (dopustiva rešenja koja su bliska njegovom optimumu) u okviru razumnog vremena, pri čemu se ne garantuje da će nađena rešenja biti optimalna, niti se može odrediti njihova bliskost optimalnom rešenju.

Računska složenost nekog algoritma se obično meri ukupnim brojem elementarnih koraka koje treba u okviru tog algoritma ostvariti da bi se rešio posmatrani problem. Ako za svaki konkretan primer ovog problema proizvoljne dimenzije **n** važi da je:

$$\textit{Ukupan broj elementarnih koraka algoritma} \leq \mathbf{C} \cdot \mathbf{f}(\mathbf{n})$$

gde je **C** pozitivna konstanta, a **f** neka realna funkcija, tada algoritam ima računsku složenost **O** (**f**(**n**)).
Ako je **f**(**n**) polinom po **n** , tada se algoritam naziva ***polinomijalnim*** (smatraju se računski efikasnim), a u suprotnom on je ***eksponencijalan*** (ne mogu u razumnom vremenu da reše probleme velikih dimenzija).

Da bi heuristički algoritam bio efikasan, treba da bude **polinomijalan**.

13.4. Klasifikacija heuristika

Uobičajena klasifikacija heurističkih metoda je na:

1. ***Konstruktivne metode*** - generišu samo jedno dopustivo rešenje problema koje treba da bude blisko optimumu. Pri tome se mogu koristiti dva principa:
   - *Princip proždrljivosti* („greedy“) - u svakoj iteraciji se od trenutno mogućih izbora bira onaj koji je najbolji po nekom lokalnom kriterijumu
   - *Princip gledanja unapred* - u svakoj iteraciji među trenutno mogućim izborima prepoznaje one koji bi mogli da dovedu do lošeg krajnjeg rešenja, pa se zato takvi izbori izbegavaju

2. ***Metode lokalnog pretraživanja*** (*sekvencijalne metode*) - iterativno generišu čitav niz dopustivih rešenja problema tražeći da ona budu sve bolja i bolja. Pri tome se u svakoj iteraciji pretražuje okolina trenutnog rešenja i u njoj prema nekom lokalnom kriterijumu bira sledeće rešenje u nizu. Početno dopustivo rešenje niza se može generisati slučajno ili formirati primenom neke konstruktivne metode.

3. ***Evolutivne metode*** - u svakoj iteraciji se generiše više dopustivih rešenja problema koja čine tzv. „populaciju“, pri čemu se teži da svaka sledeća populacija bude bolja od prethodne.

4. ***Metode dekompozicije*** - razbijaju problem na više potproblema manjih dimenzija, koji se kasnije odvojeno rešavaju, pri čemu se vodi računa o njihovoj međusobnoj korelaciji.

5. ***Induktivne metode*** - rešavaju veće i složenije probleme primenom metoda razvijenih za manje i jednostavnije probleme istog tipa.

Novija klasifikacija heuristika je na:

a) ***Specijalne heuristike*** - dizajniraju se za posebne vrste optimizacionih problema i mogu rešavati samo probleme za koje su dizajnirane.

b) ***Opšte heuristike*** - su heuristike opšteg karaktera koje se mogu primeniti na bilo koji problem kombinatorne optimizacije, bez obzira na specifičnost njihove strukture.

13.5. Opšte heuristike

Opšte heuristike (*metaheuristike*) su namenjene rešavanju problema kombinatorne optimizacije koji se mogu prikazati kao:

$$\min_{x \in X} \mathbf{f}(\mathbf{x})$$

gde dopustivi skup **X** (prostor dopustivih rešenja) ima *konačno* mnogo elemenata.

Princip lokalnog pretraživanja polazi od proizvoljne tačke $\mathbf{x_1} \in \mathbf{X}$ kao od početnog rešenja, pa se u svakoj iteraciji **n** pretražuje okolina $\mathbf{N}(\mathbf{x_n})$ trenutnog rešenja $\mathbf{x_n}$ i u njoj nalazi, prema definisanom pravilu izbora, sused koji predstavlja sledeće rešenje $\mathbf{x_{n+1}}$.
Ako se takav sused ne nađe, pretraživanje staje i za aproksimaciju optimalnog rešenja se uzima rešenje iz generisanih rešenja koje daje najmanju vrednost funkcije cilja **f**(**x**).

Četiri najpopularnije opšte heuristike su (prve 3 su zasnovane na *Principu lokalnog pretraživanja*):

1) ***Simultano kaljenje*** - ova heuristika na slučajan način generiše nekog suseda iz okoline trenutne tačke, prihvatajući ga kao sledeću tačku pretraživanja ne samo u slučaju poboljšanja, već i u slučaju pogoršanja funkcije cilja, ali sa verovatnoćom koja se kontrolisano menja tokom iteracija kako bi se izbegle zamke lokalnih minimuma.

2) ***Tabu pretraživanje*** - se bazira na tzv. *adaptivnoj memoriji* koja služi za pamćenje podataka o prethodnim fazama procesa pretraživanja, a koji utiču na izbor sledećih tačaka u ovom procesu. Zapravo, u svakoj iteraciji **n** se čuva neka istorija **H** prethodnog pretraživanja, tj. zapis koji pamti izabrane karakteristike nekih od prethodno generisanih tačaka.

3) ***Metoda promenljivih okolina*** (**MPO**) - se bazira na principu da se u svakoj iteraciji može, u cilju nalaženja sledeće tačke, vršiti sistematsko prestruktuiranje okoline trenutne tačke.

4) ***Genetski algoritmi*** - svakom rešenju iz prostora dopustivih rešenja **X** se na tačno definisan način dodeljuje jedan niz konačne dužine (nad nekom konačnom azbukom simbola) koji se naziva kôd ovog rešenja. Skup kodova svih dopustiih rešenja iz **X** čini prostor kodiranih rešenja $\underline{\mathbf{X}}$. Nakon toga se realizuje takav proces pretraživanja koji generiše tačke u prostoru kodiranih rešenja $\underline{\mathbf{X}}$. Tumačeći neko rešenje iz **X** kao jedinku, a njegov kôd kao hromozom te jedinke, genetski algoritam u svakoj iteraciji generiše skup više tačaka iz $\underline{\mathbf{X}}$ koje predstavljaju *populaciju*.