

DINAMIČKO PROGRAMIRANJE

Ukoliko je funkcija cilja - kriterijum, nelinearna funkcija svojih argumenata ili ukoliko su ograničenja (oblast K) određena sistemom nelinearnih jednačina ili nejednačina tada se radi o nelinearnom matematičkom programiranju.

Dinamičko programiranje se svrstava u oblast nelinearnog matematičkog programiranja. *1

Za razliku od linearnog programiranja, ne postoji standardna matematička formulacija problema dinamičkog programiranja. Može se reći da je dinamičko programiranje opšti pristup rešavanju određene vrste problema, gde se odgovarajuće jednačine, nejednačine (ograničenja) moraju ponaosob postavljati za svaki problem koji se rešava. U skladu sa tim, poznavanje opšte strukture problema dinamičkog programiranja je neophodno da bi se prepoznalo da li se i kako određeni problem može rešiti primenom dinamičkog programiranja.

Dinamičko programiranje je matematička tehnika koja omogućuje formiranje niza međusobno povezanih odluka, tj. obezbeđuje sistemsku proceduru za određivanje optimalne kombinacije povezanih odluka - strategije. *2

Drugim rečima, primenom dinamičkog programiranja moguće je upravljati odgovarajućim procesima (sistemima) koji protiču u vremenu i na čiji se tok može na određeni način uticati kroz planiranje akcija i preduzimanje potrebnih mera tj. donošenjem niza odluka sa ciljem postizanja što veće efektivnosti.

Prvi korak u formiranju matematičkog modela dinamičkog programiranja za upravljanje odgovarajućim procesom (ne samo dinamičkog programiranja) je izbor numeričkog kriterijuma (F) kojim se određuje uspešnost upravljanja datim procesom. U zavisnosti od donesenih odluka u pojedinim etapama procesa, izabrani kriterijum (F) će imati različite numeričke vrednosti. Na osnovu tih vrednosti i postavljenih ciljeva planiraju se buduće odluke tako da izabrani kriterijum (F) na kraju procesa dostigne maksimalnu ili minimalnu vrednost, što zavisi od njegovog praktičnog značenja.

Na osnovu samog izbora kriterijuma jasno je da li treba donositi odluke koje vode ka maksimalnoj ili ka minimalnoj vrednosti izabranog kriterijuma (F) na kraju procesa, npr. proizvodnja \rightarrow maksimalni dohodak, transport \rightarrow minimalni troškovi.

Osnovni preduslov za primenu dinamičkog programiranja, radi nalaženja optimalnog upravljanja odgovarajućim procesom, je da je moguće sam proces, a samim tim i upravljanje procesom, razbiti na niz etapa ili koraka gde se u svakoj

etapi donosi odgovarajuća (optimalna) odluka koja doprinosi da na kraju upravljanje celokupnim procesom bude optimalno. Neka se proces sastoji od N - etapa. U svakoj etapi n se donosi odgovarajuća odluka x_n (x_n - promenljive odlučivanja, $n = 1, 2, \dots, N$) vezana za upravljanje procesom. Na ovaj način upravljanje celokupnim procesom $\mathbf{X}(x_1, x_2, \dots, x_N)$ se sastoji iz niza odluka koje se donose po etapama. Upravljanje celokupnim procesom na ovaj način još se naziva i **strategija**. *3

Zadatak optimalnog upravljanja procesom je u stvari izbor strategije \mathbf{X}^* za koju se postiže optimalna vrednost funkcije cilja tj. kriterijuma F : $F(\mathbf{X}^*) = F_{opt}$.

Donošenjem odgovarajućih odluka x_n ($n = 1, 2, \dots, N$) u n - toj etapi ostvaruje se odgovarajući doprinos f_n funkciji cilja u pravcu njene optimizacije tj. optimizacije kriterijuma F . U praksi pri primeni dinamičkog programiranja često je funkcija cilja odnosno kriterijum F aditivan tj.: *4

$$F = f_1 + f_2 + \dots + f_N.$$

Postavlja se pitanje da li donositi onu odluku koja optimizira doprinos funkciji cilja (kriterijumu F) u datoj etapi ? U slučajevima kada sabirci aditivnog kriterijuma (F) nisu međusobno nezavisni to nije moguće.

Primer:

Preduzeće raspolaže određenom količinom materijalnih sredstava na početku godine. Ta sredstva treba utrošiti na odgovarajući način u toku naredne tri godine.

Ulaganja se mogu vršiti na dva načina:

1. Maksimalno angažovanje postojećih kapaciteta za proizvodnju artikala koji su već "osvojeni".
2. Modernizacija postojećih kapaciteta za proizvodnju sa ciljem osvajanja proizvodnje novih tipova artikala za koje postoje pokazatelji da će biti traženi u narednom trogodišnjem periodu.

Kriterijum za donošenje odluka (upravljanje poslovanjem) na koji način utrošiti postojeća materijalna sredstva je da preduzeće u toku trogodišnjeg perioda ostvari maksimalnu dobit.

Strategija #1 poslovanja

Polazeći od uskih interesa, ulaganje svih sredstava u angažovanje kapaciteta radi proizvodnje što veće količine već "osvojenih" artikala, bi dovelo do maksimalne dobiti u prvoj godini. Postavlja se pitanje šta ako u drugoj godini bude osetno

smanjena potražnja datih artikala, a npr. u trećoj godini ti artikli uopšte ne budu traženi na tržištu ?

Strategija #2 poslovanja

Pored ulaganja u proizvodnju već "osvojenih" artikala, ulaganje izvesnog dela sredstava u modernizaciju kapaciteta, već u prvoj godini, tako da se što ranije počne sa proizvodnjom novih artikala potrebnih tržištu.

Odgovarajuće odluke x_n ($n = 1, 2, \dots, N$) u svakoj etapi se moraju donositi tako da se obezbedi optimalno nastavljanje procesa tj. da ukupna strategija bude optimalna.

Procedura rešavanja zadataka dinamičkog programiranja počinje donošenjem (optimalne) odluke - upravljanja x_N za poslednju etapu N . Donošenje optimalne odluke tj. određivanje optimalne vrednosti upravljanja za poslednju etapu (optimalnog doprinosa f_N funkciji cilja) je uvek relativno jednostavno jer je uslovljeno trenutnim i krajnjim stanjem procesa (sistema).

Za donošenje (optimalnih) odluka - upravljanja u prethodnim etapama ($N-1, N-2, \dots, 2, 1$) koristi se rekurentna formula na osnovu koje se donosi (optimalna) odluka - upravljanje za etapu n ako je poznata optimalna odluka - upravljanje za etapu $n+1$, ($n = 1, 2, \dots, N$).

Rekurentna formula u opštem slučaju je oblika: *5

$$f_n^*(s_n) = \underset{x_n}{\text{opt}} \{ c_{s_n x_n} + f_{n+1}^*(s_{n+1}) \} \quad (1)$$

gde je:

- N - broj etapa,
- n - oznaka za tekuću etapu ($n = 1, 2, \dots, N$),
- s_n - trenutno stanje za etapu n ,
- x_n - promenljiva odlučivanja za etapu n ,
- x_n^* - optimalna vrednost promenljive odlučivanja x_n za stanje s_n u etapi n ,
- $c_{s_n x_n}$ - doprinos etape n funkciji cilja, ako je početno stanje s_n i promenljiva odlučivanja x_n ,
- $f_{n+1}^*(s_{n+1})$ - optimalni doprinos etape $n+1, \dots, N$ funkciji cilja ako je proces na početku etape $n+1$ u stanju s_{n+1} ,
- $f_n(s_n, x_n)$ - doprinos etape $n, n+1, \dots, N$ funkciji cilja ako je proces (sistem) u stanju s_n , u n -toj etapi, kome odgovara promenljiva odlučivanja x_n , uzimajući u

obzir prethodno određene optimalne vrednosti promenljivih odlučivanja u narednim etapama - $f_{n+1}^*(s_{n+1})$,

- $f_n(s_n, x_n^*)$ - optimalni doprinos etapa $n, n+1, \dots, N$ funkciji cilja ako je proces (sistem) u stanju s_n , u n -toj etapi, kome odgovara optimalna promenljiva odlučivanja x_n^* , uzimajući u obzir prethodno određene optimalne vrednosti promenljivih odlučivanja u narednim etapama - $f_{n+1}^*(s_{n+1})$.

Rekurentna formula (1) se na osnovu prethodnog može napisati kao:

$$f_n^*(s_n) = \underset{x_n}{\text{opt}} \{ f_n(s_n, x_n) \} \text{ ili } f_n^*(s_n) = f_n(s_n, x_n^*) \quad (2)$$

gde se $f_n(s_n, x_n)$ izražava u zavisnosti od $s_n, x_n, f_{n+1}^*(s_{n+1})$, i verovatno neke vrednosti (mere) doprinosa $c_{s_n x_n}$ promenljive odlučivanja x_n funkciji cilja kao:

$$f_n(s_n, x_n) = c_{s_n x_n} + f_{n+1}^*(s_{n+1}). \quad (3)$$

Kako je $f_n^*(s_n)$ definisano u zavisnosti od $f_{n+1}^*(s_{n+1})$ to je izraz za $f_n^*(s_n)$ rekurentan. Rekurentna formula (2) vrši rekurziju kretajući se unazad etapu po etapu. Kada se tekući broj etape smanji za 1, nova funkcija $f_n^*(s_n)$ se određuje koristeći $f_{n+1}^*(s_{n+1})$ koja je određena u prethodnoj iteraciji.

Za poslednju etapu (N), optimalan doprinos funkciji cilja određuje se preko izraza (3) kada se stavi da je $f_{N+1}^*(s_{N+1}) = 0$, odnosno:

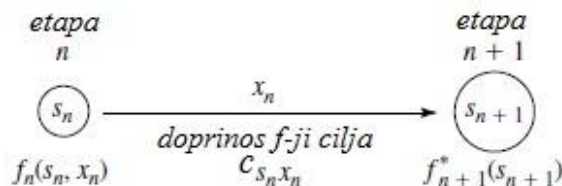
$$f_N^*(s_N) = \underset{x_N=0,1,\dots,s_N}{\text{opt}} c_{s_N x_N}. \quad (3a)$$

Koristeći rekurentnu formulu (2), procedura rešavanja se pomera unatrag etapu po etapu, određujući (optimalne) odluke za datu etapu, sve dok se ne odredi optimalna odluka za početnu etapu. Na ovaj način se dolazi do niza odluka koje se donose po etapama odnosno dolazi se do upravljanja celokupnim procesom tj. strategije. *6

Determinističko dinamičko programiranje

Kod determinističkog dinamičkog programiranja stanje procesa (sistema) u narednoj etapi ($n+1$) je u potpunosti određeno stanjem procesa (sistema) i vrednošću promenljive odlučivanja u tekućoj etapi (n). Determinističko dinamičko programiranje može se šematski prikazati kao na slici VI-1. *7

Pretpostavka je da se proces (sistem) nalazi u etapi n u stanju s_n . Donošenjem odgovarajuće odluke x_n , proces (sistem) prelazi u neko stanje s_{n+1} u etapi $n+1$. Optimalni doprinos funkciji cilja, kriterijumu F , etapa $n+1, n+2, \dots, N$ je prethodno izračunat i iznosi $f_{n+1}^*(s_{n+1})$. Odluka x_n takođe daje neki doprinos funkciji cilja - $c_{s_n x_n}$. Kombinacijom ovih dveju vrednosti na odgovarajući način dobija se $f_n(s_n, x_n)$, doprinos etapa $n, n+1, \dots, N$ funkciji cilja.



Slika VI-1. Determinističko dinamičko programiranje. *8

Optimizacijom $f_n(s_n, x_n)$ u zavisnosti od x_n dobija se $f_n^*(s_n) = f_n(s_n, x_n^*)$. Pošto se odredi optimalna odluka x_n^* i optimalni doprinos $f_n^*(s_n)$ funkciji cilja za svako moguće stanje s_n u etapi n procedura rešavanja se pomera za jednu etapu u nazad.

Jedan od načina kategorizacije problema determinističkog dinamičkog programiranja je po formi funkcije cilja u smislu da li kriterijum F treba minimizirati ili maksimizirati. Drugi mogući način kategorizacije problema determinističkog dinamičkog programiranja je prema prirodi skupa stanja procesa (sistema) za odgovarajuću etapu. Skup stanja procesa može biti diskretan (konačan) odnosno kontinualan (beskonačan). *9

RASPODELA OGRANIČENE KOLIČINE RESURSA NA AKTIVNOSTI

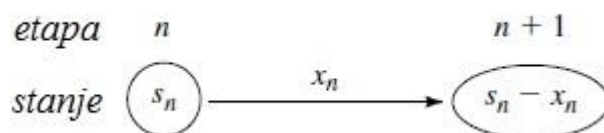
Jedan od najčešćih problema za čije se rešavanje koristi determinističko dinamičko programiranje je zadatak raspodele ograničene količine resursa na aktivnosti. Zadatak se sastoji u tome da se određena količina *jedne* vrste resursa raspodeli na određeni broj aktivnosti (N). Cilj je da se odredi kako raspodeliti resurse između postojećih aktivnosti na najefikasniji način. *10

Pošto ova vrsta zadatka uvek podrazumeva raspodelu jedne vrste resursa na određeni broj aktivnosti, problem raspodele ograničene količine resursa na aktivnosti ima uvek, kao deo determinističkog dinamičkog programiranja, istu formulaciju - postavku:

- etapa $n \rightarrow$ aktivnost n ($n = 1, 2, \dots, N$),
 $x_n \rightarrow$ količina resursa koja se raspodeljuje na aktivnost n (promenljiva odlučivanja),
stanje $s_n \rightarrow$ količina resursa trenutno dostupna (preostala) za raspodelu na ostale aktivnosti (n, \dots, N).

Razlog zašto se stanje s_n definiše na ovaj način je taj što je trenutno dostupna količina resursa za raspodelu u stvari informacija o trenutnom stanju procesa (tekuća etapa n) koja je potrebna za donošenje odluka o raspodeli resursa na preostale aktivnosti.

Kada se proces (sistem) nalazi u n -toj etapi u stanju s_n , izborom vrednosti promenljive odlučivanja x_n proces prelazi u etapu $n+1$ u stanje $s_{n+1} = s_n - x_n$, kao što je to prikazano na slici VI-2.



Slika VI-2. Promena stanja pri raspodeli ograničene količine resursa.

Ako se sa $c_n(x_n)$ označi doprinos funkciji cilja u slučaju da se količina resursa x_n dodeli n -toj aktivnosti, tada je cilj da se x_1, \dots, x_N izaberu tako da maksimiziraju:

$$\sum_{n=1}^N c_n(x_n),$$

pri ograničenju:

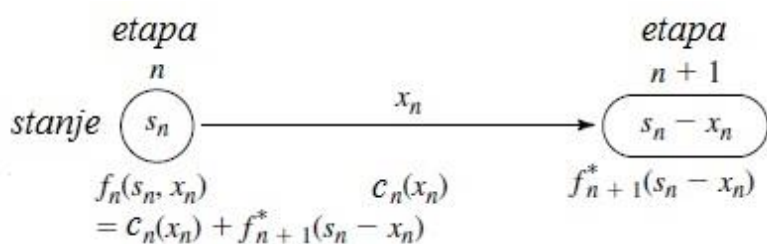
$$\sum_{n=1}^N x_n = S,$$

gde je: x_n - nenegativna celobrojna veličina,

S - ukupna raspoloživa količina resursa.

U skladu sa oznakama u prethodnom poglavlju (izraz 3), doprinos funkciji cilja $f_n(s_n, x_n)$ aktivnosti $n, n+1, \dots, N$ ako je trenutno stanje resursa s_n , na početku aktivnosti n , i ako se aktivnosti n dodeljuje količina resursa x_n , iznosi: (slika VI-3)

$$f_n(s_n, x_n) = c_n(x_n) + \max_{k=n+1}^N c_k(x_k) = c_n(x_n) + f_{n+1}^*(s_{n+1}). \quad (4)$$



Slika VI-3. Doprinos funkciji cilja $f_n(s_n, x_n)$. *11

U prethodnom izrazu maksimum se odnosi na promenljive odlučivanja tj. količine resursa x_{n+1}, \dots, x_N koje moraju da zadovolje uslov:

$$\sum_{k=n+1}^N x_k = s_{n+1}.$$

Optimalan doprinos funkciji cilja $f_n^*(s_n)$ aktivnosti $n, n+1, \dots, N$ ako je trenutno stanje resursa s_n , na početku aktivnosti n , i ako se aktivnosti n dodeljuje količina resursa x_n^* za koju je doprinos funkciji cilja maksimalan (količina resursa koja se dodeljuje aktivnosti n može biti $0, 1, \dots, s_n$), iznosi:

$$f_n^*(s_n) = f_n(s_n, x_n^*) = \max_{x_n=0,1,\dots,s_n} f_n(s_n, x_n).$$

Optimalan doprinos funkciji cilja aktivnosti $n+1, \dots, N$ se može napisati kao:

$$f_{n+1}^*(s_{n+1}) = f_{n+1}^*(s_n - x_n) \leftarrow s_{n+1} = s_n - x_n,$$

što daje:

$$f_n(s_n, x_n) = c_n(x_n) + f_{n+1}^*(s_n - x_n).$$

Zamenom prethodnog izraza u (4) dobija se rekurentna formula za izračunavanje optimalnog doprinosa aktivnosti n funkciji cilja ako je na početku aktivnosti n raspoloživa količina resursa bila s_n : *12

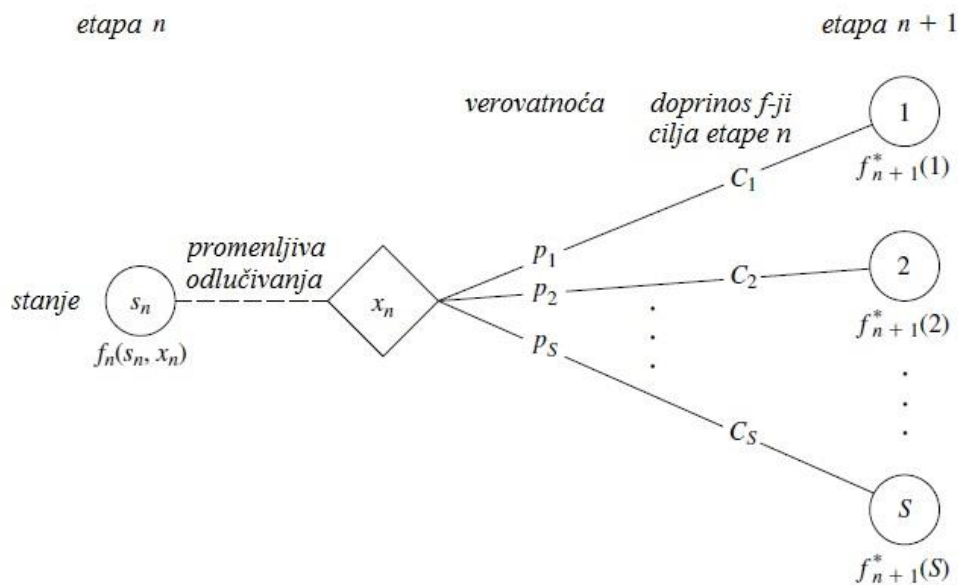
$$f_n^*(s_n) = \max_{x_n=0,1,\dots,s_n} [c_n(x_n) + f_{n+1}^*(s_n - x_n)] \quad n=1,2, \dots, N-1. \quad (5)$$

Za poslednju aktivnost (N), optimalan doprinos funkciji cilja, se odrađuje na sledeći način:

$$f_N^*(s_N) = \max_{x_N=0,1,\dots,s_N} c_N(x_N). \quad (5a)$$

Na verovatnoći zasnovano dinamičko programiranje

Na verovatnoći zasnovano dinamičko programiranje (**Probabilistic dynamic programming**) se razlikuje od determinističkog dinamičkog programiranja u tome što stanje procesa (sistema) u sledećoj etapi nije u potpunosti određeno stanjem procesa i promenljivom odlučivanja u tekućoj etapi. U ovom slučaju postoji raspodela verovatnoća na osnovu koje se definiše u koje će stanje proces preći u sledećoj etapi. Raspodela verovatnoća je u potpunosti definisana stanjem i promenljivom odlučivanja u tekućoj etapi procesa. Osnovna struktura na verovatnoći zasnovanog dinamičkog programiranja prokazana je na slici VI-4. *13



Slika VI-4. Na verovatnoći zasnovano dinamičko programiranje. *14

Na slici VI-4 sa S je označen broj mogućih stanja u etapi $n+1$, dok se sama stanja obeležavaju brojevima tj. $1, 2, \dots, S$. Ako promenljiva odlučivanja ima vrednost x_n tada proces iz stanja s_n u etapi n prelazi u stanje i u etapi $n+1$ sa verovatnoćom p_i ($i = 1, 2, \dots, S$). Ako proces pređe u stanje i , u etapi $n+1$, tada je C_i doprinos funkciji cilja etape n .

Kada se struktura prikazana na slici VI-4 proširi tako da obuhvati sva stanja i sve promenljive odlučivanja u svim etapama procesa dobija se tzv. **drvo odlučivanja**.

Iz razloga što zavisi od raspodele verovatnoća, relacija između $f_n(s_n, x_n)$ i $f_{n+1}^*(s_{n+1})$ je uvek komplikovanija nego što je to slučaj kod determinističkog dinamičkog programiranja. Tačan oblik ove relacije zavisi od oblika funkcije cilja tj. kriterijuma F .

Npr. neka je cilj da se minimizira očekivani doprinos pojedinačnih etapa. U tom slučaju $f_n(s_n, x_n)$ predstavlja očekivani doprinos etapa $n, n+1, \dots, N$ funkciji cilja ako se proces nalazi u stanju s_n , u n -toj etapi, kome odgovara promenljiva odlučivanja x_n , tj.:

$$f_n(s_n, x_n) = \sum_{i=1}^S p_i \cdot [C_i + f_{n+1}^*(i)],$$

gde je:

$$f_{n+1}^*(i) = \min_{x_{n+1}} f_{n+1}(i, x_{n+1})$$

minimalni (optimalni) očekivani doprinos etapa $n+1, \dots, N$ funkciji cilja u zavisnosti od izvodljivih vrednosti za x_{n+1} .

Pitanja:

1. U koju oblast matematičkog programiranja spada dinamičko programiranje.
2. Šta predstavlja dinamičko programiranje.
3. Osnovni preduslov za primenu dinamičkog programiranja.
4. Oblik funkcije cilja kod dinamičkog programiranja.
5. Oblik rekurentne formule u opštem slučaju dinamičkog programiranja.
6. Procedura rešavanja problema dinamičkog programiranja.
7. Osnovna karakteristika determinističkog dinamičkog programiranja.
8. Šematski prikaz determinističkog dinamičkog programiranja.
9. Kategorizacija problema determinističkog dinamičkog programiranja.
10. U čemu se sastoji zadatak raspodele ograničene količine resursa na aktivnosti.
11. Šematski prikaz zadatka raspodele ograničene količine resursa na aktivnosti.
12. Oblik rekurentne formule u slučaju zadatka raspodele ograničene količine resursa na aktivnosti.
13. Osnovna karakteristika na verovatnoći zasnovanog dinamičkog programiranja.
14. Šematski prikaz na verovatnoći zasnovanog dinamičkog programiranja.