

## **DINAMIK DASTURLASH YORDAMIDA OPTIMAL BOSHQARUV MASALALARINI YECHISH**

**Mamatova Z.X., O'ktamjonova N.A.**

*Farg'ona davlat universiteti dotsenti, [mamatova.zilolakhon@gmail.com](mailto:mamatova.zilolakhon@gmail.com).*

*Farg'ona davlat universiteti talabasi, [karimberdiyevanilufar625@gmail.com](mailto:karimberdiyevanilufar625@gmail.com).*

*Annotatsiya. Ushbu maqolada dinamik dasturlash usuli yordamida optimal boshqaruv masalalarini yechish masalalari ko'rib chiqilgan. Dinamik dasturlash murakkab jarayonlarni vaqt bo'yicha bosqichlarga ajratish orqali optimal qarorlarni aniqlash imkonini beruvchi samarali matematik usullardan biri hisoblanadi. Maqolada ushbu usulning nazariy asoslari, matematik modeli va amaliy qo'llanilishi yoritilgan. Amaliy misol sifatida CAM avtomoyka faoliyati modeli tuzilib, xizmat ko'rsatish jarayonini optimal boshqarish masalasi yechilgan. Olingan natijalar dinamik dasturlash usuli xizmat ko'rsatish tizimlarida resurslardan samarali foydalanish va foydani maksimal darajaga yetkazishda muhim ahamiyatga ega ekanligini ko'rsatadi.*

*Kalit so'zlar. Dinamik dasturlash, optimal boshqaruv, matematik modellashtirish, optimallashtirish, boshqaruv tizimi, qaror qabul qilish, ko'p bosqichli masala, vaqt bo'yicha boshqaruv, resurslarni taqsimlash, iqtisodiy model, foydani maksimal qilish, xizmat ko'rsatish tizimi, avtomoyka modeli, navbat nazariyasi, amaliy masala, matematik model, boshqaruv strategiyasi, optimal qaror, jarayonni rejalashtirish, samaradorlik, tizim holati, cheklovlar, foyda funksiyasi, boshqaruv jarayoni, hisoblash usuli*

Hozirgi zamon fan va texnika taraqqiyoti sharoitida murakkab jarayonlarni samarali boshqarish muammosi dolzarb masalalardan biri hisoblanadi. Iqtisodiyot, sanoat, energetika, transport, logistika hamda texnologik jarayonlarda qarorlarni optimal qabul qilish zarurati tobora ortib bormoqda. Bunday sharoitda boshqaruv jarayonlarini matematik modellashtirish va ularning eng maqbul yechimlarini aniqlash muhim ahamiyat kasb etadi. Optimal boshqaruv masalalari odatda vaqt bo'yicha o'zgaruvchi tizimlar bilan bog'liq bo'lib, ular ma'lum cheklovlar ostida eng yaxshi natijani ta'minlovchi boshqaruv strategiyasini topishni talab etadi. Ushbu masalalarni an'anaviy usullar yordamida yechish ko'pincha murakkab yoki samarasiz bo'lib qoladi. Shu sababli zamonaviy hisoblash usullari, jumladan, dinamik dasturlash usuli keng qo'llanila boshlagan.

Optimal boshqaruv — bu tizimni boshqarish jarayonida maqsad funksiyasini maksimal yoki minimal qilishga erishadigan qarorlar to‘plamidir. Matematika nuqtai nazaridan, optimal boshqaruv 5 elementdan iborat. Obyekt — boshqariladigan tizim  $S$  masalan, cam moyka, ishlab chiqarish liniyasi, transport tizimi. Holat o‘zgaruvchisi  $x(t)$  — tizimning vaqt bo‘yicha o‘zgaruvchi parametrlar majmui, masalan, navbatdagi mashinalar soni. Boshqaruv o‘zgaruvchisi  $u(t)$  — qaror yoki harakatlar majmui, tizimga ta’sir qiluvchi omillar, masalan, kunlik yuvish hajmi. Cheklovlar — resurslar, vaqt, imkoniyatlar, xavfsizlik talablari. Maqsad funksiyasi  $J$  — boshqaruv natijasida erishilmoqchi bo‘lgan qiymat, masalan, foyda, xarajat, samaradorlik. Optimal boshqaruv tizimni boshqarish jarayonida holat va boshqaruv parametrlarini shunday tanlashki, maqsad funksiyasi maksimal yoki minimal bo‘lsin va barcha cheklovlar bajarilsin.

Optimal boshqaruvning matematik modeli optimal boshqaruv masalasi diferensial yoki diskret tenglamalar orqali ifodalanadi.

Tizim dinamikasi quyidagicha,

Uzluksiz tizim:

$$x'(t) = f(x(t), u(t), t), \quad x_0 = x_{-0}$$

Diskret tizim:

$$x_{k+1} = f(x_k, u_k, k)$$

$x_0$  berilgan

bu yerda,  $x$  — holat vektori,  $u$  — boshqaruv vektori,  $f$  — tizim dinamikasini tavsiflovchi funksiya

Tizimning kelajak holati faqat joriy holat va boshqaruvga bog‘liq, ya’ni tizim xatti-harakatlari dinamik qonunlar bilan belgilanadi.

Dinamik dasturlash — bu murakkab optimallashtirish va hisoblash masalalarini yechish uchun ishlatiladigan samarali algoritmik usuldir. U katta va murakkab masalani kichik, o‘zaro bog‘liq bo‘lgan qismlarga bo‘lish va ularni ketma-ket yechish orqali umumiy natijaga erishishni ta’minlaydi. Dinamik dasturlashning mohiyati shundan iboratki, ko‘plab masalalar bir xil kichik submasalalardan tashkil topadi. Shu

sababli, kichik masalalarning yechimlarini alohida hisoblab, ularni xotirada saqlash orqali, keyinchalik ularni qayta ishlatish mumkin. Bu usul hisoblash samaradorligini sezilarli darajada oshiradi. Dinamik dasturlash shuningdek, masalaning optimal yechimi uning kichik qismlari (submasalalari) optimal yechimlaridan tashkil topishi sharti bilan ishlaydi. Bu xususiyatni matematik til bilan aytganda, masala optimal substrukturga ega bo'ladi.

Dinamik dasturlash usuli ko'plab sohalarda qo'llaniladi. Masalan, Fibonacci sonlarini hisoblash, yo'l topish va marshrutlarni optimallashtirish, resurslarni taqsimlash, sarmoya va ishlab chiqarish rejalashtirish kabi masalalarda keng qo'llaniladi. Bu usulning samaradorligi shundan kelib chiqadiki, har bir kichik masala bir marta hisoblanadi va keyinchalik uning natijasi qayta ishlatiladi, bu esa hisoblash vaqtini sezilarli darajada kamaytiradi. Tarixiy nuqtai nazardan, dinamik dasturlash 1950-yillarda rivojlangan. Shu davrda Amerika matematik va iqtisodchi Richard Bellman murakkab optimallashtirish masalalarini yechish usulini ishlab chiqqan. Bellman bu usulni ishlab chiqishda masalalarni kichik qismalarga bo'lish, har bir qismning yechimini saqlash va keyinchalik umumiy yechimni hosil qilish g'oyasidan foydalangan. U "dynamic programming" atamasini yaratgan. Shu bilan birga, u murakkab iqtisodiy va texnik masalalarni algoritmik tarzda yechish imkonini beradigan matematik asoslarni ishlab chiqqan. Hozirgi kunda dinamik dasturlash kompyuter fanining eng muhim yo'nalishlaridan biri hisoblanadi. U nafaqat matematik masalalarda, balki informatsion texnologiyalar, logistika, sun'iy intellekt va boshqa ko'plab sohalarda qo'llaniladi.

Dinamik dasturlash yordamida katta hajmdagi ma'lumotlar bilan ishlash va murakkab hisoblash masalalarini qisqa vaqt ichida yechish mumkin. Dinamik dasturlash — bu murakkab masalalarni tizimli va samarali tarzda yechishga imkon beruvchi usul bo'lib, uning mohiyati kichik submasalalarni hisoblash va natijalarni qayta ishlatishga asoslangan. Tarixiy rivojlanishi esa asosan 20-asrning o'rtalarida Richard Bellman tomonidan shakllantirilgan bo'lib, bugungi kunda u ko'plab ilmiy va amaliy sohalarda keng qo'llanilmoqda.

Masala: CAM avtomoyka kun davomida xizmat ko‘rsatadi. Har bir soatda moykaga turli miqdorda avtomobillar keladi. Moykaning imkoniyati cheklangan bo‘lib, bir soatda maksimal 10 ta avtomobil yuvish mumkin. Agar barcha avtomobillar o‘z vaqtida yuvilmasa, navbat paydo bo‘ladi va mijozlar soni kamayadi. Har bir soatda nechta avtomobil yuvishni belgilashi kerak. Maqsadimiz kun oxirida maksimal foyda olish.

Bizda mavjud malumotlar ish vaqti: 4 soat, har bir yuvilgan avtomobildan tushadigan foyda: 40 000 so‘m, har bir kutilgan (navbatda qolgan) avtomobil uchun jarima: 10 000 so‘m, bir soatda yuvish mumkin bo‘lgan maksimal avtomobillar soni: 10 ta

Vaqt(soat)	Kelgan mashinalar
1	8
2	14
3	6
4	12

Masalani tahlil qilishni boshlaymiz,

$$T = 4$$

Bu yerda  $T = 4$  — moykaning ishlash vaqti 4 soat ekanligini bildiradi. Dinamik dasturlashda vaqt bosqichlarga bo‘linadi. Har bir soat alohida bosqich hisoblanadi. Masala 4 bosqichdan iborat, chunki moyka 4 soat ishlaydi.

Holat o‘zgaruvchisini kiritish, ya’ni har soatda kiruvchi mashinlar soni

$$a_1 = 8$$

$$a_2 = 14$$

$$a_3 = 6$$

$$a_4 = 12$$

Bir soatda maksimal 10 ta mashina yuviladi va mavjud mashinadan ko‘pini yuvib bo‘lmaydi. Qarorlar real sharoit bilan cheklanadi.

$$x_{t+1} = x_t + a_t - u_t$$

Keyingi soatdagi navbat avvalgi navbatga kelgan mashinalar qo‘shilib, yuvilgan mashinalar ayrilishi orqali hosil bo‘ladi. Bu tenglama tizimning qanday o‘zgarishini ko‘rsatadi.

Daromad funksiyasi quyidagicha

$$F_t = 40000 \cdot u_t - 10000 \cdot x_{t+1}$$

Yuvilgan mashinalardan daromad olinadi, navbatda qolgan mashinalar uchun jarima belgilanadi. Maqsad faqat yuvish emas, balki navbatni kamaytirish hamdir. Endi 4 soat davomida olinadigan umumiy foyda maksimal qilinadi.

$$\max \sum_{t=1}^4 F_t$$

Dinamik dasturlash funksiyasi . Har bir bosqichda hozirgi foyda va kelajakdagi optimal foyda birgalikda hisobga olinadimiz

$$V_t(x_t) = \max [F_t + V_{t+1}(x_{t+1})]$$

4-soatdan keyin jarayon tugaydi va qo‘shimcha foyda ham yo‘q.

$$V_5(x_5) = 0$$

Endi formulalarga qarab hisoblab chiqamiz.

Dinamik dasturlash orqadan oldinga qarab ishlaydi va shu shart bo‘yicha hisoblaymiz.

$$4 \rightarrow 3 \rightarrow 2 \rightarrow 14$$

Biz oxirgi bosqichdan boshlaymiz, chunki kelajak ma‘lum bo‘lishi kerak.

$$4\text{-soatda } V_4(x_4) = \max [40000 \cdot u_4 - 10000 \cdot x_4 + 12 - u_4]$$

$$3\text{-soatda } V_3(x_3) = \max [40000 \cdot u_3 - 10000 \cdot x_4 + V_4(x_4)]$$

$$2\text{-soatda } V_2(x_2) = \max [40000 \cdot u_2 - 10000 \cdot x_3 + V_3(x_3)]$$

$$1\text{-soatda } V_1(0) = \max [40000 \cdot u_1 - 10000 \cdot 8 - u_1 + V_2(8 - u_1)]$$

Va natijalar quyidagicha bo‘ladi.

$$u_1 = 8$$

$$u_2 = 10$$

$$u_3 = 10$$

$$u_4 = 10$$

Bundan kelib chiqib formulaga qo‘yib, yakuning max foydani topamiz.

$$V_1(0) = 1460000 \text{ so‘m}$$

Bu kuniga 4 soat ishlaydigan CAM moykaning maksimal kunlik foydasi.

## FOYDALANILGAN ADABIYOTLAR

1. Bellman R. Dynamic Programming. — Princeton University Press, 1957.

2. Bertsekas D. P. Dynamic Programming and Optimal Control. — Athena Scientific, 2012.
3. Pontryagin L. S., Boltyanskiy V. G., Gamkrelidze R. V., Mishchenko E. F. Optimal Processes Theory. — Moscow: Nauka, 1969.
4. Taha H. A. Operations Research: An Introduction. — Pearson Education, 2017.
5. Hillier F. S., Lieberman G. J. Introduction to Operations Research. — McGraw-Hill, 2015.
6. Kirk D. E. Optimal Control Theory: An Introduction. — Dover Publications, 2004.
7. Лукьянов А. Г. Методы оптимального управления. — Москва: Физматлит, 2006.
8. Юдин Д. Б., Гольштейн Е. Г. Задачи и методы оптимизации. — Москва: Наука, 1979.
9. Сафаров А. А., Каримов Ш. Х. Operatsion tadqiqotlar asoslari. — Toshkent: O‘zbekiston, 2018.