PRVI KOLOKVIJ

Model je aproksimacija sustava. Nemoguće je napraviti model koji savršeno odgovara sustavu. Cilj optimizacije je postići optimalno rješenje koje zapravo predstavlja najbolju aproksimaciju sustava. Ta aprokasimacija more biti dovoljno dobra, pronađena na dovoljno brz način i da je praktično primjenjiva.
Optimiranje se sastoji od 5 koraka:
1. Modeliranje zadatka
2. Zadavanje funkcije cilja
3. Uvođenje ograničenja
4. Izbor metode rješavanja (sve metode sastoje se od postupka treženja osnovnog mogućeg riješenja i od postupka traženja poboljšanog rješenja)
5. Izbor računalne podrške

Ako imamo neki model tada je potrebno definirati metodu rješavanja koja je općenita odnosno na višoj razini (ne odnosi se isključivo na konkretan problem). Također, program isto tako mora biti definiran na višoj razini. Gledajući zajedno ove dvije komponente moraju biti definirane na razini sustava - sustavski pristup.
U slučaju da imamo još jedan model tada je ideja taj model preoblikovati da se mogže koristiti već postojeća metoda i program, a ne da se razvija sve ponovno za novi model.

U slučaju da imamo nelinearne modele ideja ih je linearizrati (snažna aproksimacija)
METODE:
- razvijamo ih da riješe osnovni zadatak
- pokušavamo ih i ubrzati

PROSTOR RJEŠENJA oderđuje rezultate, koji se najčešće nalaze na rubovima prostora.
Zadaci optimiziranja:
Cilj (definirati cilj) -> prihvatiti informacije (model) -> provjera ograničenja (nije nužna) -> metoda (koja dovodi do optimalnih rezultata) -> program -> optimalni rezultati -> postoptimalna analiza -> (zaključci, primjena)

OSNOVNO MOGUĆE RJEŠENJE - "pozicionirati se negdje u podnožju pravog brežuljka i tražiti poboljšana rješenja do optimuma" - Predstavlja početno rješenje koje može poslužiti, dakle zadovoljava sve uvjete, ali nije optimalno. Ovo rješenje predstavlja početak od kojeg se dalje ide prema optimalnom rješenju.

Simplex

Simplex traži poboljšano riješenje pomoći dva kriterija:
KRITERIJ OPTIMALNOSTI: gledanje funkcije cilja kako bi vidjeli u kojem smijeu idemo da bi postigli max => negativni koeficijenti vode prema max, a pozitivni prema min. (odnosi se na stupce)
KRITERIJ IZVEDIVOSTI: definira način na koji treba birati retke kako bi se zadržali unirat prostora rješenja (odnosi se na retke)

Postupak rješavanja simplex-a:
1. odaberemo stupac po kriteriju optimalnosti (za min najveći, za max najmanji)
2. ne gledamo sve sta je <=0
3. odaberemo neki redak po kriteriju izvedivosti

Vrijedi:
- min(x0) = max(g0)

Dualni simplex:
Ako imamo ograničenja tip >= ne možemo dobro definirati točku (postavlja se pitanje koje je osnovno moguće rješenje). Problem je sta nema gornje granice - beskonačni prostor.
Takav sustav potrebno je premodelirati u sustav s ograničenjima tipa <=.

Kod simplexa:

- zadaci su u cijelosti matematički modelirani (model, funkcija cilja, ograničenja)
- rješavanje se provodi pomoću matematike (dig. rač. to lako riješe pomoću matrica)
- postoptimalna naliza se vrši pomoću matematike.

OSJETLJVOST NA PROMJENE PARAMETARA (postoptimalna analiza):
- osjetljivost na promjene koeficijenata funkcije cilja kod neishodišnih varijabli: Sve dok se koeficijenti u polaznoj fuknkciji cilja uz neishodišnu varijablu mjenjaju unutar izračunatih granica, raspored optimalnih vrijabli ostaje isti, kao i vrijednosti neishodišnih varijabli, samo se dobije nova optimalna vrijednost funkcije cilja. (optimalan plan ostaje potpuno isti mjenja se samo iznos funkcije cilja)
- osjetljivost na promjene koeficijenata funkcije cilja kod ishodišnih varijabli: Sve dok se koeficijent u fukviji cilja mijenja u toj granici optimalno rješenje uopće se neće promjeniti. (neće se promjeniti baš ništa)
- osjetljivost na promjene konstanti desne srane ograničenja: Sve dok se definirani element(i) desne strane ograničenja mjenja do te granice raspored optimalnih vrijabli ostaje isti, ali neishodišne varijable i fukcija cilja mijenjaju vrijednosti. (optimalan plan ostaje isti, ali se mjenjaju svi optimalni iznosi).

Postoptimalna analiza:
- rezultati moraju biti u prostoru rješenja (ne može se zeti samo obično zaokruživanje)
- najveći problem traženja optimalnih rješenja je ne preskočiti sva potencijalna rješenja

Zahtjev za cjelobrojnm rezultatima nije praktično pisati unaprije. To se definira postoptimalno. Koristi se 0-1 način.

DRUGI KOLOKVIJ:

Transportni problem:
Cilj je minimizirati ukupni cijenu transporta neovisno o tome tko će što dobiti. Imperativ je ispreznitи sve polazne i napuniti sve dolazne - pretpostavka je da su transportna sredstva neograničenog kapaciteta.
Transportni problem - zadatak je u cijelosti matematički definirana ali je matematički nerješiv => koristi se heuristika (pretraživanje)
npr:
X1 + X2 = 6
X1 + X2 = 5

Kad X-eve pribacimo u matricu tada ta matrica ima determinantu 0.

Lokacijski problem:
Kada ne postoji put između polazišta i dolazišta. Tada je potrebno definirati minimalnu cijenu transporta + cijena gradnje puta.

Pretraživanje se organizira na općenit način (zapis ide prema matricama) da bude pregledno danas i da bude prihvatljivo digitalnom računalu.

Dvije metode:
- SJEVEROZAPADNA METODA (kreće se iz gornjeg lijevog kuta)
- METODA MINIMALNIH TROŠKOVA (kreće se s najmanjim cijenom)
Ako želimo zabraniti mogućnost transporta tada stavimo veliku cijenu (npr. 999)
Potrebno je da ukupna suma svih jedinica iz polaznog skladišta bude jednaka ukupnoj sumi svih jedinica dolaznog skladišta (tzv. zatvoreni problem).
Ako problem nije zatvoren matematika savjetuje uvođenje fiktivnih skladišta
Ove dvije metode daju osnovno moguće rješenje. Cilj je dalje trežiti rješeje do optimuma.

Jedna od metoda je i stepping stone" - relativni troškovi.
Koriste se prazna polja u tablici (ona koja nisu korištena). Dodaje se u ono polje koje nije korištno, potom se, da bi se sačuvala ravnoteža, oduzima iz polja u tom retku. Nako toga se dodaje u polje u tom stupcu, i potom se opet oduzima iz polja koje ima koodrinate

netom spometutog stupca i retka.
Ako je broj polaznih kombinacija manji od m+n-1 tada imamo degeneraciju i nemožemo problem riješiti medorom relativnih troškova.

Sale i studenti:
Cilj je postaviti grupe studenata u sale (svaka grupa i sala imaju kapacitet 1). Cijene se računaju kao broj sjedalica - broj studenata. Negativne cijene mjenjamo s (999, zabranjujemo ih). Ako nema jednak brj sala i studenata uvodimo fiktivne (sale ili grupu studenata). To predstavlja da određena grupa studenata neće biti u nijednoj sali ili će jedna sala biti prazna.

Trgovački putnik:
- skladišta nemjau kapacitete
- sve točke obilazi samo jedno sredstvo
- cilj je obići sve točke najkraćim putm, ali se pri tom nesmijemo nigdje pojavit dva puta osim u polaznoj točki
Da bismo riješili ovaj zadatak mi sami odajemo ograničenja.
Dakle najprije ograničavano iz polazišta prema svima =1, od svuh polazišta prema jednom odredištu =1 i na kraju pomoću x12+x21<=1 … definiramo da nema povratka u tu točku (da se kroz svaku točku proazi samo jednom)
Ako se vraćamo u početnu točku tada je moguće više rezultata, ako se na vraćamo tada je samo jedan.
Pretraživanje se vrši pomoću tablice istine ili stabla odlučivanja na način da se parametri koju su optimalni postavljaju u 1, a ostali u 0. Kod tablice istine ograničenja su pasivna (čekaju), kod stabla odlučvanja su aktivna.

Dinamičko programiranje:
Cilj je na optimalan način pronaći put između dije točke. Pri tome rješenje ovisi o etapama. Optimalna odluka po etapi ne mora biti i dio ukupne optimalne odluke.

Teorija igara:
Zadaci optimiranja suočavaju se sa sukobljenim stranama. Optimalna odluka za jednog igrača donosi se na slijedeći način:
- nađu se minimumi retka
- odabrere se njihov max
Strateška odluka - najmanji mogući gubitak (negativni brojevi predstavlju gubitke)

Teorija grafova:
Događaji - ne troše ni vrijeme ni sredstva, označavaju početak i završetak aktivnosti
Aktovnosti - troše sve, omeđene s dva događaja!!!
Fiktivna aktivnost - ništa ne troši, predstavlja nužno ili korisno čekanje
Aktivnosti na grafu ne smiju se križati, niti smiju postojati dvije aktivnosti na istom grafu s istim događajima u istom smjeru (dig rač - uvodi se novi događaj povezan fiktivnom vezom).
Uz događaje vežemo vrijeme.
Uz aktovnosti vežemo vrijeme (može biti i različito od t2-t1 - slobodno vrijeme) i sredstva (novac i materijal)
Tri načina:
- točno vrijme
- gornja i donja granica
- pesimistično, vjerovatno i potimistično
Kritična aktivnost - nema slobodnog vremena za eventualne probleme
OPTIMALAN REZULTAT = ukupno (minimalno) vrijeme za dovršetak zadatka + ukupna (minimalna) sredstva + kritične aktivnosti

Nelinearno programiranje:
- postoji beskonačno mnogo nelinearnih zadataka
- digitalno računalo ga teško riješava
- gradijentna metoda - opasnost - možemo, ovisno o izboru delta X, preskočiti extrem