Općenito o matematičkom programiranju

Problem matematičkog programiranja

Problemi opisani *matematičkim jezikom*, kod kojih je zadatak naći *ekstrem* (*minimum* ili *maksimum*) neke funkcije uz *zadana ograničenja*, nazivaju se problemima (zadacima) ***matematičkog programiranja***. Vrlo je važno razlikovati značenje riječi “*programiranje*” u ovom kontekstu od njenog značenja u *računarskom programiranju* (u suštini, matematičko programiranje kao pojam je nastalo još u doba kada računari nisu bili ni na vidiku). Ovdje “programiranje” treba shvatiti kao *planiranje*, tj. kao *postupak* i *plan* koji nam pomaže da dođemo *do određenog cilja*. U suštini, to je *procedura traženja* takvih vrijednosti *promenljivih* u zadatku koje istovremeno *zadovoljavaju postavljena ograničenja* i *daju funkciji cilja optimalnu vrijednost*. S druge strane, računarsko programiranje predstavlja *pisanje kôda*, odnosno *niza instrukcija* koje računar treba da obavi.

Matematičko programiranje istovremeno pripada *matematici*, *operacionim istraživanjima, teoriji upravljanja, sistemskoj analizi* i drugim tehničkim naukama i disciplinama, a po značaju i obimu korištenja, svakako je jedan od najznačajnijih alata ukupne nauke i tehnike.

Problemi matematičkog programiranja se, najkraće rečeno, odnose na probleme *efikasnog korištenja ograničenih resursa*. Pod resursima se mogu podrazumijevati ljudi, mašine, novac, vrijeme, robe, itd. Potreba za rješavanjem ovakvih zadataka javlja se raznim oblastima: industriji, državnoj upravi, biznisu, vojsci, poljoprivredi, policiji, zdravstvu.

Problematika *nalaženja ekstrema funcija u prisustvu dodatnih ograničenja* (tzv. ***uvjetnih ekstrema***) stara je koliko i sama matematička analiza. Mada su problemi matematičkog programiranja upravo ovog tipa, *klasični analitički matematički pristupi*, koji se koriste za nalaženje *ekstrema funkcija pod ograničenjima* (***Lagrange-ov metod***, ***metod Kuhn-Tucker-a***), nisu pogodni za rješavanje *praktičnih zadataka matematičkog programiranja*, jer su ovi zadaci u pravilu *vrlo složeni* sa *velikim brojem promjenljivih* i *velikim brojem ograničenja*, koja su najčešće data *u obliku nejednačina*. Zbog toga je razvijena *posebna teorija* i izgrađen je *niz metoda* (uglavnom numeričke prirode) pomoću kojih se mogu uspješno rješavati mnogi takvi zadaci. Ovi metodi su uglavnom razvijeni pod pretpostavkom da će biti relizirani *uz pomoć računara*, jer se u njima javlja *velika količina izračunavanja* koja je praktično neizvodljiva ručno. Bez obzira na to, u okviru matematičkog programiranja izučavaju se i klasični analitički pristupi, jer su mnogi praktični metodi za rješavanje problema matematičkog programiranja proizašli iz takvih pristupa. Pored toga, klasični pristupi daju lijep uvid u prirodu samih problema koji se rješavaju.

U svom najopćenitijem obliku, problem matematičkog programiranja možemo predstaviti ovako. Neka je data neka funkcija f čiji je domen ma kakav skup Δ (čiji elementi uopće ne moraju nužno biti brojevi), ali čije su vrijednosti *realni brojevi*, odnosno funkcija $f : \Delta \to \mathbf{R}$ (kao primjer *nenumeričkog domena*, Δ može biti na primjer *skup fotografija*, a f može biti funkcija koja svakoj fotografiji pridružuje neki kvantitativni odnosno brojčani pokazatelj, recimo prosječni nivo svjetline fotografije). Zbog općenitosti domena Δ možemo, bez umanjenja općenitosti modela, smatrati da je f funkcija *samo jednog argumenta* $\mathbf{x} \in \Delta$, s obzirom da se, na primjer, za funkciju koja zavisi od n realnih argumenata x_1, x_2, ..., x_n može smatrati da zavisi od samo jednog argumenta $\mathbf{x}$, ali koji nije broj nego *uređena n-torka realnih brojeva*, odnosno $\mathbf{x} = (x_1, x_2, ..., x_n)$. Tada se opća forma problema matematičkog programiranja može iskazati u sljedećem obliku: Odrediti $\mathbf{x}$ iz domena Δ takav da izraz

$$Z = f(\mathbf{x})$$

dostigne svoju *optimalnu* (zavisno od situacije minimalnu ili maksimalnu) *vrijednost*, eventualno uz ograničenje da $\mathbf{x}$ ne može biti ma kakav element domena Δ nego *isključivo element nekog skupa* Ω *sadržanog u domenu* Δ (tj. $\Omega \subseteq \Delta$), odnosno uz ograničenje

$$\mathbf{x} \in \Omega$$

Ovo skraćeno možemo zapisati kao

$$\arg \text{opt } Z = f(\mathbf{x})$$

p.o.

$$\mathbf{x} \in \Omega$$

pri čemu “opt”, zavisno od problema, može biti “min” ili “max”, dok “arg” i “p.o.” imaju ranije opisana značenja. Međutim, kako vrijedi $\max f(\mathbf{x}) = -\min(-f(\mathbf{x}))$, to se *svaki problem traženja maksimuma* može *prevesti u problem traženja minimuma* prostim negiranjem funkcije cilja. Stoga se, bez umanjenja općenitosti, sva teorija matematičkog programiranja može razmatrati kao da je cilj uvijek *minimizacija* (ili, alternativno, *maksimizacija*).

Primijetimo da je ograničenje oblika $\mathbf{x} \in \Omega$ *vrlo općenito* i u realnim problemima obično je dato kao presjek više jednostavnijih ograničenja, od kojih većina ima oblik jednakosti ili nejednakosti. Moguća je i situacija da bude $\Omega = \Delta$, odnosno da je dopušteno da $\mathbf{x}$ bude *ma kakav element iz domena funkcije f*. U tom slučaju se ograničenje $\mathbf{x} \in \Omega$ obično *ne navodi eksplicitno* i za takve probleme matematičkog programiranja kažemo da su ***problemi bez*** (***dopunskih***) ***ograničenja***.

Interesantno je da se u terminologiji operacionih istraživanja *svaki element* $\mathbf{x}$ iz domena Δ funkcije f naziva ***rješenjem problema matematičkog programiranja***, bez obzira na to da li on zadovoljava ograničenje $\mathbf{x} \in \Omega$ ili ne. S druge strane, elementi $\mathbf{x}$ koji se nalaze u dopustivom prostoru Ω (tj. koji zadovoljavaju ograničenje $\mathbf{x} \in \Omega$) nazivaju se ***dopustiva rješenja***, a dopustivo rješenje koje daje *optimalnu vrijednost funkciji cilja f* naziva se ***optimalno rješenje***.

Može se, dakle, reći da se matematičko programiranje bavi *izučavanjem problema gore navedenog oblika* i *metodama za nalaženje njihovih dopustivih i optimalnih rješenja* uz *različite uvjete* koji se postavljaju na *funkciju f* i na skup Ω.

Globalni i lokalni ekstremi

U teoriji matematičkog programiranja od velike važnosti je praviti razliku između *globalnih i lokalnih ekstrema*. S obzirom da se svaki problem maksimizacije može svesti na problem minimizacije, u nastavku će biti govora samo o *minimumima*, dok analogna diskusija vrijedi i za maksimume.

Ukoliko postoji takvo $\mathbf{x}^* \in \Omega$ da je

$$(\forall \mathbf{x} \in \Omega)\ f(\mathbf{x}^*) \le f(\mathbf{x})$$

onda kažemo da je $\mathbf{x}^*$ ***tačka globalnog*** (***apsolutnog***) ***minimuma*** funkcije f na skupu Ω, dok je $Z^* = f(\mathbf{x}^*)$ ***vrijednost globalnog*** (***apsolutnog***) ***minimuma*** funkcije f na skupu Ω. Ako prethodna nejednakost vrijedi i kada se relacija nestroge nejednakosti “$\le$” zamijeni relacijom stroge nejednakosti “$<$” za $\mathbf{x} \in \Omega$ i $\mathbf{x} \ne \mathbf{x}^*$, onda kažemo da je $\mathbf{x}^*$ ***tačka strogog*** (***izoliranog***) ***globalnog*** (***apsolutnog***) ***minimuma*** funkcije f na skupu Ω, dok je $Z^* = f(\mathbf{x}^*)$ ***vrijednost strogog*** (***izoliranog***) ***globalnog*** (***apsolutnog***) ***minimuma*** funkcije f na skupu Ω.

Ukoliko, međutim, prethodna nejednakost vrijedi samo u *nekoj okolini* tačke $\mathbf{x}^*$, odnosno ukoliko *postoji okolina* $\mathrm{N}(\mathbf{x}^*)$ tačke $\mathbf{x}^*$ takva da vrijedi

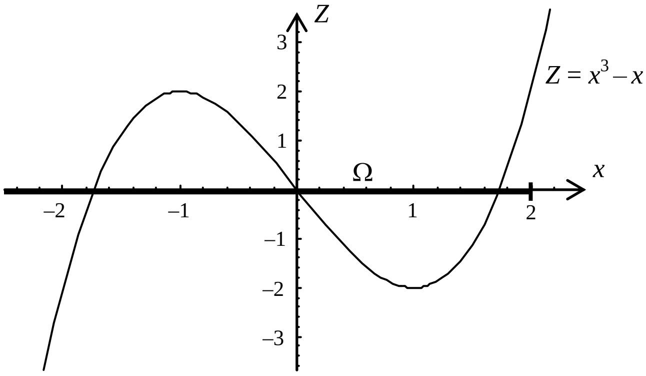
$$(\forall \mathbf{x} \in \Omega \cap \mathrm{N}(\mathbf{x}^*))\ f(\mathbf{x}^*) \le f(\mathbf{x})$$

onda kažemo da je $\mathbf{x}^*$ ***tačka lokalnog*** (***relativnog***) ***minimuma*** funkcije f, dok je $Z^* = f(\mathbf{x}^*)$ ***vrijednost lokalnog*** (***relativnog***) ***minimuma*** funkcije f. Pri tome se pod okolinom tačke $\mathbf{x}^*$ podrazumijevaju elementi domena Δ koji su u nekom smislu “blizu” elementa $\mathbf{x}^*$ (za tu svrhu, u skupu Δ pojam “blizine” mora biti na neki način definiran). Obično se kao okolina uzima tzv. ***ε-okolina*** $\mathrm{N}_\varepsilon(\mathbf{x}^*)$ koja predstavlja skup elemenata koje su na *udaljenosti* manjoj ili jednakoj od ε od elementa $\mathbf{x}^*$, odnosno $\mathrm{N}_\varepsilon(\mathbf{x}^*) = \{\mathbf{x} \,|\, \mathbf{x} \in \Omega \wedge d(\mathbf{x}, \mathbf{x}^*) \le \varepsilon\}$, gdje je $d(\mathbf{x}, \mathbf{x}^*)$ udaljenost između elemenata $\mathbf{x}$ i $\mathbf{x}^*$ (za tu svrhu, u skupu Δ mora biti definiran pojam “udaljenosti”, odnosno Δ mora biti *metrički prostor*). Ako prethodna nejednakost vrijedi i kada se relacija nestroge nejednakosti “$\le$” zamijeni relacijom stroge nejednakosti “$<$” za $\mathbf{x} \in \Omega \cap \mathrm{N}(\mathbf{x}^*)$ i $\mathbf{x} \ne \mathbf{x}^*$, onda kažemo da je $\mathbf{x}^*$ ***tačka strogog*** (***izoliranog***) ***lokalnog*** (***relativnog***) ***minimuma*** funkcije f na skupu Ω, dok je $Z^* = f(\mathbf{x}^*)$ ***vrijednost strogog*** (***izoliranog***) ***lokalnog*** (***relativnog***) ***minimuma*** funkcije f na skupu Ω.

Analogno se mogu sprovesti definicije za razne tipove *maksimuma* funkcije *f*, te se tako definiraju ***globalni*** (***apsolutni***) ***maksimum***, ***strogi*** (***izolirani***) ***globalni*** (***apsolutni***) ***maksimum***, ***lokalni*** (***realativni***) ***maksimum***, te ***strogi*** (***izolirani***) ***lokalni*** (***relativni***) ***maksimum***.

Interesantno je da može postojati lokalni minimum (ili maksimum) a da pri tome globalni minimum (maksimum) *ne postoji* (to se najčešće dešava kada je dozvoljena oblast Ω *neograničen skup*). Na primjer, neka je potrebno naći minimum funkcije $Z = f(x) = x^3 - 3x$ pod ograničenjem $x \le 2$ (dakle, skup Ω je

$\Omega = \{x \in \mathbf{R} \mid x \le 2\}$). Tehnikama poznatim iz matematičke analize lako se utvrđuje da je $x = 1$ *jedini lokalni minimum* (i to *strogi*) ove funkcije na skupu Ω (on je zapravo jedini lokalni minimum i na čitavom domenu **R**). Zaista, u dovoljno maloj okolini tačke $x = 1$ vrijedi $f(x) > f(1)$ za $x \ne 1$ (zapravo, pažljivijim ispitivanjem može se pokazati da ovo vrijedi sve dok je $x > -2$). S druge strane, jasno je da funkcija $Z = f(x)$ na skupu Ω *ne može imati globalni minimum*, jer $f(x)$ *neograničeno opada* za $x \to -\infty$. Ova situacija je prikazana na sljedećoj slici:



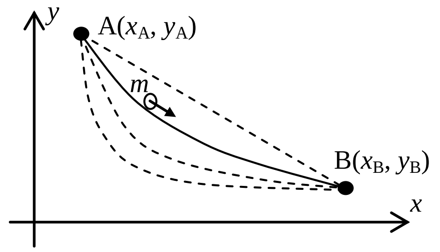
Iz same definicije slijedi da *samo globalni minimumi* (odnosno *maksmumi*, ovisno od toga šta se traži) predstavljaju *optimalna rješenja*, dok lokalni minimumi (maksimumi) *nisu nužno optimalni*. U nastavku ćemo govoriti samo o minimumima, dok analogne stvari vrijede i za maksimume. Jasno je da je svaki globalni minimum ujedno i lokalni, tako da ako globalni minimum uopće postoji, on mora biti jednak nekom od lokalnih minimuma. Stoga, ukoliko smo nekako našli *sve lokalne minimume* i ukoliko imamo garanciju da globalni minimum uopće *postoji*, tada on mora biti jednak jednom od nađenih lokalnih minimuma, i to onom u kojem je funkcija cilja *najmanja*. Nažalost, u općem slučaju je pronalaženje *svih lokalnih minimuma* veoma težak zadatak, tako da ova ideja rijetko vodi ka praktičnom načinu za nalaženje optimalnog rješenja. Naime, veliki broj metoda operacionih istraživanja uspješno pronalaze *jedan ili više lokalnih ekstrema*, ali obično *bez garancije da li ih ima još*. Stoga je u općem slučaju pronalaženje globalnih ekstrema (a samim tim i optimalnih rješenja) *veoma težak zadatak*. U nekim slučajevima čak je i ustanovljavanje *da li globalni ekstrem uopće postoji* težak zadatak. Sretna je okolnost da postoje izvjesne klase problema kod kojih se garantira da postoji *samo jedan lokalni ekstrem*, *koji je ujedno i globalni*. Takve klase problema su znatno lakše za rješavanje. O jednoj takvoj klasi biće riječi nešto kasnije.

Klasifikacija problema matematičkog programiranja

Postoji mnogo različitih klasifikacija problema matematičkog programiranja, ali na prvom mjestu ovi problemi mogu se podijeliti na ***konačnodimenzionalne*** i na ***beskonačnodimenzionalne probleme*** u ovisnosti od toga da li je domen Δ konačnodimenzionalan ili beskonačnodimenzionalan skup (intuitivno, skup je konačnodimenzionalan ukoliko mu se elementi mogu na adekvatan način kodirati sa konačno mnogo realnih brojeva, u suprotnom je beskonačnodimenzionalan). Kada se govori o problemima matematičkog programiranja, obično se prećutno podrazumijeva da se radi o konačnodimenzionalnim problemima, mada postoje i beskonačnodimenzionalni problemi (u tom slučaju, ta činjenica se *posebno naglasi*).

Beskonačnodimenzionalni problemi matematičkog programiranja ne susreću se osobito često u operacionim istraživanjima, mada se mogu susresti (npr. javljaju se u nekim modelima *dinamike borbe u ratnim operacijama*). S druge strane, ovakvi problemi se vrlo često javljaju u raznim oblastima nauke i tehnike, a naročito u *automatici* (posebno u tzv. *teoriji optimalnog upravljanja*). U ovakvim problemima elementi domena Δ su obično *funkcije* (najčešće *realne funkcije jedne ili više realnih promjenljivih*), dok je funkcija cilja $f(\mathbf{x})$ funkcija koja elemente $\mathbf{x}$ iz domena Δ (koji su i sami funkcije) preslikava u neku brojčanu vrijednost koja pokazuje koliko je funkcija $\mathbf{x}$ koja je ponuđena funkciji f kao argument “dobra” sa aspekta postavljenog cilja optimizacije. Dakle, f je u ovom slučaju funkcija koja kao svoj argument prima *druge funkcije* (iz domena Δ) i daje *brojčani rezultat* (takve funkcije se u matematičkoj analizi nazivaju ***funkcionali***). Funkcija (funkcional) f se u većini praktičnih problema izražava u formi *integrala* (i to obavezno *određenog*, da bi rezultat bio čisto brojčana vrijednost) u kojem se u okviru izraza pod integralom javlja funkcija-argument $\mathbf{x}$ i eventualno njeni izvodi. Mada se u nastavku nećemo baviti ovakvom vrstom problema, dva primjera će biti korisna da se stekne uvid o kakvoj vrsti problema se ovdje uopće radi.

- ***Primjer***: Od svih mogućih krivulja koje prolaze kroz dvije zadane tačke A(x_A, y_A) i B(x_B, y_B) u x–y koordinatnom sistemu pri čemu je $x_B > x_A$ i $y_B < y_A$, naći onu koja ima osobinu da će se tačkasto tijelo mase m koje se kliže po toj krivulji pod dejstvom gravitacije spustiti od tačke A do tačke B za najmanje moguće vrijeme.



Problem nalaženja krivulje sa traženim svojstvom, koja se u fizici naziva ***brahistohrona*** (od grčkih riječi *brachistos* i *chronos* sa značenjem *najkraći* i *vrijeme* respektivno) vrlo je star (postavljen je i riješen krajem 17. vijeka) i rješavali su ga mnogi poznati matematičari, kao što su *Newton, Leibniz, L'Hospital* i braća *Bernoulli*. Pogledajmo kako izgleda *matematski model* ovog problema. Za tu svrhu će nam trebati i neka elementarna znanja iz fizike.

Neka se tražena krivulja može opisati kao $y = \mathbf{y}(x)$, gdje je $\mathbf{y}$ nepoznata *funkcija*. Brzina tijela v može se izraziti kao *izvod pređenog puta po vremenu*, dakle $v = \mathrm{d}s/\mathrm{d}t$, Na osnovu ovoga možemo pisati $\mathrm{d}t = \mathrm{d}s/v$. S druge strane, brzinu v u proizvoljnom trenutku možemo dobiti i iz zakona o održanju energije, izjednačavanjem potencijalne energije koju je tijelo imalo na početku kretanja (tj. u tački A) i kinetičke energije u posmatranoj poziciji, pri čemu se posmatrana pozicija uzima kao referentna za računanje potencijalne energije. Drugim riječima, imamo

$$\frac{m v^2}{2} = m g\,(y_A - y)$$

odakle je

$$v = \sqrt{2 g\,(y_A - y)}$$

Dalje, poznato je da je $\mathrm{d}s^2 = \mathrm{d}x^2 + \mathrm{d}y^2$ (*Pitagorina teorema* "*u malom*"), te $\mathrm{d}y/\mathrm{d}x = \mathbf{y}'(x)$. Kada sve ovo uzmemo u obzir, dobijamo

$$\mathrm{d}t = \sqrt{\frac{1 + \mathbf{y}'(x)^2}{2 g\,(y_A - \mathbf{y}(x))}}\,\mathrm{d}x$$

Ukupno vrijeme spuštanja dobićemo integracijom ovog elementarnog vremenskog intervala po svim mogućim vrijednostima koje koordinata x može uzeti odnosno

$$t = \int_{x_1}^{x_2} \sqrt{\frac{1 + \mathbf{y}'(x)^2}{2 g\,(y_A - \mathbf{y}(x))}}\,\mathrm{d}x$$

Cilj je minimizirati ovo vrijeme, pod uvjetom da grafik funkcije $y = \mathbf{y}(x)$ prolazi kroz tačke A i B. Dakle, problem možemo prikazati u obliku

$$\arg\min t = \int_{x_1}^{x_2} \sqrt{\frac{1 + \mathbf{y}'(x)^2}{2 g\,(y_A - \mathbf{y}(x))}}\,\mathrm{d}x$$

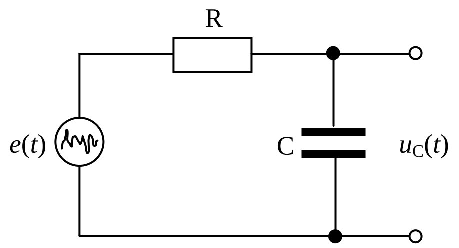
p.o.

$\mathbf{y}(x_1) = y_1$
$\mathbf{y}(x_2) = y_2$

Kao što se vidi, u ovom problemu funkcija cilja ovisi od *nepoznate funkcije* $\mathbf{y}$ koju treba odrediti (i njenog izvoda), dok je skup ograničenja Ω skup *svih realnih funkcija jedne realne promjenljive čiji grafik prolazi kroz tačke* A i B.

Grana matematike koja se bavi rješavanjem ovakvih problema naziva se ***varijacioni račun***. Ovdje se ne možemo baviti načinom rješavanja ovakvih problema. Recimo samo da rješenje ovog problema nije ništa što bi neko možda na prvi pogled očekivao, odnosno tražena krivulja nije niti *prava linija*, niti *luk kružnice*, niti *luk parabole*, niti luk ma kakve druge često susretane krivulje, nego luk krivulje koja se naziva ***cikloida*** (to je krivulja koja predstavlja trag tačke na rubu točka koji se kotrlja bez klizanja po ravnoj površini).

- ***Primjer***: RC kolo na slici napaja se iz vremenski promjenljivog naponskog izvora $e(t)$ kod kojeg se oblik vremenske promjene napona može podešavati, što samim tim utiče i na oblik vremenske promjene napona $u_C(t)$ na kondenzatoru. Kakav treba biti oblik vremenske promjene napona $u_C(t)$ na kondenzatoru da bi srednja vrijednost ovog napona u razmatranom vremenskom intervalu (0, T) bila što je god moguće veća, ali da pri tome srednja snaga disipacije razvijena na otporniku R ne premaši maksimalnu dozvoljenu vrijednost P_{max}? Kakav treba biti vremenski oblik napona izvora $e(t)$ da obezbijedi takvu vremensku promjenu napona $u_C(t)$?



Srednja vrijednost $\overline{U}_C$ napona na kondenzatoru dobija se integracijom vremenske promjene tog napona na razmatranom vremenskom intervalu i dijeljenjem te vrijednosti sa dužinom intervala T, odnosno imamo

$$\overline{U}_C = \frac{1}{T}\int_0^T u_C(t)\,dt$$

Ovo je očito naša funkcija cilja koju trebamo maksimizirati, pri čemu ulogu nepoznate ima nepoznata funkcija $u_C(t)$. Što se tiče prosječne snage disipacije na otporniku, ona se može na analogan način dobiti integracijom trenutne snage i dijeljenjem sa dužinom intervala. Za trenutnu snagu $p_R(t)$ na otporniku vrijedi $p_R(t) = i_R(t)^2\,R$, gdje je $i_R(t)$ struja kroz otpornik. Kako je ova struja ista kao struja kroz kondenzator, vrijedi $i_R(t) = i_C(t) = C\,du_C(t)/dt = C\,u_C'(t)$. Dakle, srednja snaga disipacije na otporniku iznosi

$$\overline{P}_R = \frac{1}{T}\int_0^T p(t)\,dt = \frac{R\,C^2}{T}\int_0^T u_C'(t)^2\,dt$$

Kako ova vrijednost ne smije da premaši P_{max}, matematski model problema koji treba riješiti glasi:

$$\arg\max\ \overline{U}_C = \frac{1}{T}\int_0^T u_C(t)\,dt$$

p.o.

$$\frac{R\,C^2}{T}\int_0^T u_C'(t)^2\,dt \le P_{max}$$

Nakon što se riješi ovaj problem po nepoznatoj funkciji $u_C(t)$, lako možemo odrediti i traženi oblik napona $e(t)$ s obzirom da vrijedi

$$e(t) = R\,i_R(t) + u_C(t) = R\,C\,u_C'(t) + u_C(t)$$

Ni ovaj put nećemo ulaziti u načine rješavanja problema ovog tipa. Recimo samo da se za ovaj problem dobija da su optimalni oblici napona $u_C(t)$ i $e(t)$ *kvadratne funkcije*. Tačnije, optimalno rješenje ovog problema ima oblik $u_C(t) = p\,t^2 + q\,t$ i $e(t) = k\,t^2 + l\,t + m$, gdje su p, q, k, l i m neke konstante čija vrijednost ovisi od vrijednosti R, C i P_{max}.

U nastavku se više nećemo baviti beskonačnodimenzionalnom optimizacijom. Recimo samo da se rješavanje takvih problema zasniva na primjeni i rješavanju *diferencijalnih jednačina* (običnih ili parcijalnih), ili se, uz pomoć raznih aproksimacija, takvi problemi približno svode na probleme *konačnodimenzionalne optimizacije*, čije rješenje daje približno rješenje polaznog problema.

Vratimo se nazad na probleme konačnodimenzionalne optimizacije. Kako je kod takvih problema domen Δ *konačnodimenzionalan skup*, čiji se elementi (tačke) mogu adekvatno kodirati sa *konačno mnogo* (recimo n) *realnih brojeva*, možemo bez umanjivanja općenitosti uzeti da je domen Δ zapravo skup $\mathbf{R}^n$, što se uvijek može postići uz pogodno kodiranje. Broj n se naziva *dimenzija prostora*. Dakle, kod problema konačnodimenzionalne optimizacije, cilj je naći tačku $\mathbf{x} \in \mathbf{R}^n$ za koju neka zadana funkcija (funkcija cilja) f sa domenom $\mathbf{R}^n$ dobija svoju *ekstremnu vrijednost*, eventualno uz dodatno ograničenje da $\mathbf{x}$ mora pripadati *nekom podskupu skupa* $\mathbf{R}^n$, odnosno $\mathbf{x} \in \Omega$ gdje je $\Omega \subseteq \mathbf{R}^n$. Alternativno se f može posmatrati i kao funkcija n realnih promjenljivih, tj. kao $f(x_1, x_2, ..., x_n)$, gdje su x_i, $i = 1 .. n$ individualne *komponente* (*koordinate*) tačke $\mathbf{x}$. Tačka $\mathbf{x}$ se može opisati kao uređena n-torka vrijednosti x_i, $i = 1 .. n$, ali se za potrebe matematičkog programiranja češće predstavlja u matrično-vektorskoj formi kao

$$\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}$$

pri čemu se oznaka transpozicije koristi zbog činjenice da se tačke u linearnoj algebri obično predstavljaju kao vektori kolone, dok je u pisanju praktičnije koristiti vektore redove.

U nastavku ćemo, bez umanjenja općenitosti, smatrati da rješavamo *problem minimizacije*, jer smo već vidjeli se problem maksimizacije uvijek može svesti na problem minimizacije, s obzirom da vrijedi $\max f(\mathbf{x}) = -\min(-f(\mathbf{x}))$. Na ovaj način nećemo morati posebno diskutovati probleme minimizacije i maksimizacije.

Prema tome kakva je funkcija f (funkcija cilja), problemi konačnodimenzionalne optimizacije mogu se podijeliti u sljedeće klase, od lakših ka težim:

(1) Funkcija f je *linearna*;
(2) Funkcija f je *kvadratna* i *svuda konveksna*;
(3) Funkcija f *nije ni linearna ni kvadratna*, ali je *svuda konveksna i diferencijabilna*;
(4) Funkcija f *nije svuda diferencijabilna*, ali je *svuda konveksna*;
(5) Funkcija f je *kvadratna*, ali *nije svuda konveksna*;
(6) Funkcija f *nije ni linearna ni kvadratna, niti je svuda konveksna*, ali je *svuda diferencijabilna*;
(7) Funkcija f *nije ni svuda konveksna, ni svuda diferencijabilna*.

Diferencijabilnost zapravo znači da je funkcija *svuda* “*glatka*” u smislu da u svakoj tački postoji *jasno definirana tangenta* (tj. da *nema* “*šiljaka*” i sličnih tvorevina). *Konveksnost* zapravo znači da funkcija uvijek izgleda “*ispupčena naniže*” u okolini svake posmatrane tačke. Ukoliko u toj tački postoji i tangenta, to zapravo znači da se funkcija u okolini te tačke nalazi “*iznad*” *svoje tangente*. Kako je pojam konveksnosti *od velikog značaja za matematičko programiranje*, o njemu ćemo nešto kasnije detaljnije govoriti.

Prema tome kakav je skup ograničenja Ω, problemi konačnodimenzionalne optimizacije mogu se također podijeliti u sljedeće klase, od lakših ka težim:

(1) Skup Ω je *čitav prostor* $\mathbf{R}^n$ (ovo faktički znači da *ograničenja nema*);
(2) Skup Ω je *poliedarski*, odnosno definiran je skupom *linearnih jednačina i nejednačina*;
(3) Skup Ω *nije poliedarski*, ali je *konveksan* (odnosno, za svake dvije tačke koje pripadaju skupu Ω, *duž koja spaja te dvije tačke* također *leži čitava* u skupu Ω);
(4) Skup Ω *nije konveksan*, ali je *povezan* (odnosno, za svake dvije tačke koje pripadaju skupu Ω, postoji *neprekidna linija koja povezuje te dvije tačke* koja *leži čitava* u skupu Ω);
(5) Skup Ω *nije povezan*, ali *ne sadrži izolirane tačke* (tj. tačke koje se *ne mogu povezati niti sa jednom drugom tačkom iz skupa*);
(6) Skup Ω *sadrži i izolirane tačke*, ali se *ne sastoji sav isključivo od izoliranih tačaka*;
(7) Skup Ω se *sav sastoji isključivo od izoliranih tačaka*, ali te tačke formiraju *rešetkasti raspored* (npr. tačke sa cjelobrojnim koordinatama), i sve se nalaze *unutar nekog poliedarskog skupa*;
(8) Skup Ω se *sav sastoji isključivo od izoliranih tačaka, bez ikakvih dodatnih ograničenja gdje se te tačke mogu nalaziti*.

Problemi kod kojih je funkcija f u klasi (1) a skup Ω u klasi (2) spadaju u ***linearno programiranje***. Svi ostali problemi spadaju u ***nelinearno programiranje***. Inače, podjela konačnodimenzionalnog matematičkog programiranja na linearno i nelinearno programiranje je najčešće susretana klasifikacija matematičkog programiranja, iako je takva klasifikacija *izuzetno gruba* (to je isto kao kada bismo rekli da u svemiru postoje dvije vrste stvari: banane i stvari koje nisu banane).

Primijetimo da nema smisla da sa funkcijom f u klasi (1) skup Ω također bude u klasi (1), s obzirom da u odsustvu ograničenja linearna funkcija može uzimati po volji male ili velike vrijednosti. S druge strane, sa nelinearnom funkcijom cilja sasvim je moguće imati smislene probleme bez ograničenja. Na primjer, funkcija $f(x_1, x_2) = x_1^2 + x_2^2$ dostiže u tački $x_1 = 0$, $x_2 = 0$ minimalnu vrijednost 0, bez ikakvih dodatnih ograničenja.

Nekoliko specijalnih tipova nelinearnog programiranja imaju i posebna imena (treba obratiti pažnju da se ove kategorije ponekad preklapaju, odnosno postoje problemi koji istovremeno spadaju u više kategorija):

- ***Nelinearno programiranje bez ograničenja***: Ω iz klase (1)
- ***Nelinearno programiranje sa linearnim ograničenjima***: Ω iz klase (2)
- ***Kvadratno programiranje***: f iz klase (2) ili (5), Ω iz klase (1) ili (2)
- ***Konveksno programiranje***: f iz klase (3) ili (4), Ω iz klase (1), (2) ili (3)
- ***Ne-glatko programiranje***: f iz klase (4) ili (7)
- ***Cjelobrojno linearno programiranje***: f iz klase (1), Ω iz klase (7)
- ***Diskretno programiranje***: Ω iz klase (7) ili (8)

Teškoća problema koji se rješava rapidno raste sa porastom klase kojoj pripadaju f odnosno Ω. Dok je danas sasvim izvodljivo rješavati probleme linearnog programiranja sa preko 100000 promjenljivih odnosno ograničenja, za probleme u kojima je f iz klase (7) a Ω iz klase (8), već i uspješno rješavanje problema sa 15-20 promjenljivih predstavlja ogroman uspjeh. Inače, treba još reći da sve navedene klase problema imaju svoje podklase, a široki spektar algoritama za rješavanje svakog od ovih segmenata matematičkog programiranja još uvijek se širi, odnosno novi algoritmi za rješavanje ovih problema se i danas otkrivaju.

Činjenica je da je rješavanje problema *bez ograničenja* tipično mnogo jednostavnije nego rješavanje problema sa ograničenjima. Međutim, mada se problemi bez ograničenja ne susreću često u praksi, postupci za rješavanje problema bez ograničenja su veoma značajni zbog činjenice da postoje pristupi kojima se problemi sa ograničenjima svode na *ekvivalentne probleme bez ograničenja*.

Posebnu klasu (nelinearnog) matematičkog programiranja čini ***dinamičko programiranje***, koje rješava probleme optimalnog *sekvencijalnog* odlučivanja, odnosno odlučivanja koje se, na izvjestan način, obavlja *po etapama*. Mada se problemi tog tipa često susreću i sami po sebi, takvi se problemi često dobijaju i kao rezultat aproksimacije beskonačnodimenzionalnh problema pomoću problema konačne dimenzije.

Na kraju treba ukazati i na neke *generalizacije* prethodno opisanog općeg modela matematičkog programiranja. Sve dosada smo prećutno pretpostavljali da su parametri koji se eventualno pojavljuju u funkciji cilja i u ograničenjima *precizno definirane* (*determinističke*) *vrijednosti*. Zbog toga se često govori i o ***determinističkim problemima***, odnosno o ***determinističkom programiranju***. Međutim, u praksi se često dešava da parametri *nisu tačno poznati*, nego da predstavljaju manje ili više *slučajne* (*stohastičke*) *vrijednosti* koje se podvrgavaju *određenim statističkim zakonitostima* (*raspodjelama*). U tom slučaju govorimo o ***stohastičkim problemima***, odnosno o ***stohastičkom programiranju***. Jasno je da u takvim slučajevima ne možemo očekivati da dobijemo *tačna rješenja*, već samo neke *statističke pokazatelje* o rješenjima, kao što su npr. *očekivane prosječne vrijednosti* pojedinih promjenljivih, njihove *statističke raspodjele*, itd.

Druga moguća generalizacija problema matematičkog programiranja je sljedeća. U svim dosada razmotrenim varijantama matematičkog programiranja, uvijek smo imali *samo jednu funkciju cilja* (čiji je rezultat uvijek *jedan realan broj*). Zbog toga se govori o ***jednokriterijskim problemima***, odnosno o ***jednokriterijskom programiranju***. Moguće je zamisliti situacije u kojima se umjesto jedne funkcije cilja javlja *više funkcija cilja* (*više kriterija*), ili što je analogno tome, *jedna funkcija cilja* čiji rezultat nije samo jedan broj, već *skupina od više brojeva*. U takvim slučajevima govorimo o ***višekriterijskim problemima***, odnosno o ***višekriterijskom programiranju***. Treba napomenuti da je višekriterijska optimizacija posebno problematična zbog činjenice da u takvim problemima često *uopće nije jasno šta je tačno cilj optimizacije*. Na primjer, ukoliko postoje dva nezavisna kriterija, veoma je nejasno šta recimo uopće znači *istovremeno maksimizirati oba kriterija*, s obzirom da gotovo uvijek poboljšanje jednog kriterija ide nauštrb drugog kriterija i obrnuto.

Standardna forma problema matematičkog programiranja

Već smo rekli da se opći oblik zadataka (konačnodimenzionalnog, determinističkog, jednokriterijskog) matematičkog programiranja može prikazati u sljedećem obliku:

arg opt $Z = f(\mathbf{x})$

p.o.

$\mathbf{x} \in \Omega$

Pri tome je $\mathbf{x} \in (x_1, x_2, ..., x_n)^{\mathrm{T}} \in \mathbf{R}^n$ vektor kolona, $\Omega \subseteq \mathbf{R}^n$ je neki podskup n-dimenzionalnog Euklidskog prostora $\mathbf{R}^n$, dok je $f(\mathbf{x})$ opća (realna) funkcija vektora (tačke) $\mathbf{x}$. Međutim, u praksi, skup Ω je gotovo uvijek zadan kao skup tačaka $\mathbf{x}$ koje zadovoljavaju sistem jednačina i/ili nejednačina oblika

$g_j(\mathbf{x}) \le 0,\ \ j = 1\,..\,m$
$h_k(\mathbf{x}) = 0,\ \ k = 1\,..\,p$

gdje su $g_j(\mathbf{x})$, $j = 1\,..\,m$ i $h_k(\mathbf{x})$, $k = 1\,..\,p$ realne funkcije vektora $\mathbf{x}$. Primijetimo da bez umanjenja općenitosti možemo razmatrati samo nejednakosti tipa "$\le$", jer se nejednakosti tipa "$\ge$" lako svode na ovakve množenjem sa -1. Na ovaj način se zadatak matematičkog programiranja može prikazati u sljedećem obliku:

arg opt $Z = f(\mathbf{x})$

p.o.

$g_j(\mathbf{x}) \le 0,\ \ j = 1\,..\,m$
$h_k(\mathbf{x}) = 0,\ \ k = 1\,..\,p$

Treba primijetiti da se u ovom obliku mogu zapisati čak i mnoga ograničenja za koja se na prvi pogled ne vidi da se mogu zapisati u ovom obliku. Tako se, na primjer, ograničenje poput $x_i \in \mathbf{Z}$ može zapisati u obliku $\sin \pi x_i = 0$. Dalje, ograničenja poput $x_i \in \{0, 1\}$, koja se često javljaju u praktičnim problemima, mogu se zapisati u obliku $x_i^{\,2} = x_i$ odnosno $x_i\,(x_i - 1) = 0$, itd.

Uvođenjem vektorskih funkcija $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), ..., g_m(\mathbf{x}))^{\mathrm{T}}$ i $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_p(\mathbf{x}))^{\mathrm{T}}$, te konvencije da je vektor manji ili jednak od nule ukoliko su mu sve komponente manje ili jednake od nule, prethodni problem se može kompaktnije zapisati kao

arg opt $Z = f(\mathbf{x})$

p.o.

$\mathbf{g}(\mathbf{x}) \le 0$
$\mathbf{h}(\mathbf{x}) = 0$

Interesantno je da je *svako ograničenje tipa jednakosti* moguće *prevesti* u *dva ograničenja tipa nejednakosti*. Zaista, ograničenje tipa $h_k(\mathbf{x}) = 0$ možemo zamijeniti sa dva ograničenja $h_k(\mathbf{x}) \le 0$ i $h_k(\mathbf{x}) \ge 0$, odnosno $h_k(\mathbf{x}) \le 0$ i $-h_k(\mathbf{x}) \le 0$. Stoga je, bez umanjenja općenitosti, moguće razmatrati *samo ograničenja tipa nejednakosti*. Kako se pored toga svaki problem traženja maksimuma može prevesti u problem traženja minimuma, to probleme matematičkog programiranja možemo posmatrati u obliku

arg min $Z = f(\mathbf{x})$

p.o.

$g_j(\mathbf{x}) \le 0,\ \ j = 1\,..\,m$

ili, kompaktnije zapisano,

arg min $Z = f(\mathbf{x})$

p.o.

$\mathbf{g}\,(\mathbf{x}) \le 0$

Konačno, kako u najvećem broju slučajeva, vrijednosti promjenljivih predstavljaju *količine raznih vrsta resursa* (ljudi, novca, mašina, itd.) koji po svojoj prirodi *ne mogu biti negativni*, većinu tipičnih problema matematičkog programiranja možemo napisati u obliku

$$\arg\min Z = f(\mathbf{x})$$

p.o.

$$g_j(\mathbf{x}) \le 0,\ j = 1\,..\,m$$
$$x_i \ge 0,\ i = 1\,..\,n$$

ili, kompaktnije zapisano,

$$\arg\min Z = f(\mathbf{x})$$

p.o.

$$\mathbf{g}(\mathbf{x}) \le 0$$
$$\mathbf{x} \ge 0$$

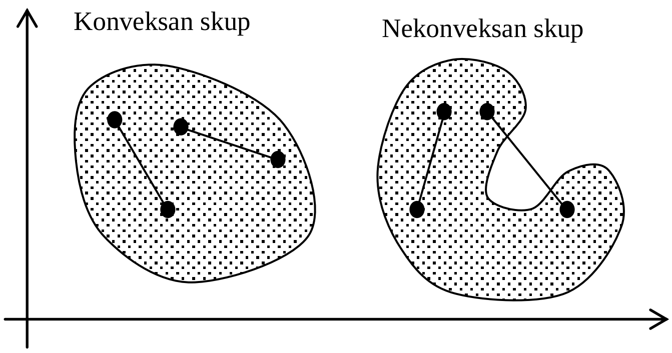
Ovako zapisan problem naziva se ***standardna forma problema matematičkog programiranja*** (alternativno se nekada umjesto minimizacije u standardnoj formi koristi maksimizacija).

Konveksno programiranje

Za neki skup $C \subseteq \mathbf{R}^n$ kažemo da je ***konveksan*** ako za bilo koje dvije tačke $\mathbf{x}_1$ i $\mathbf{x}_2$, koje se nalaze u skupu C, vrijedi da se u skupu C nalazi i cijela duž koja spaja te dvije tačke. Kako se svaka tačka duži koja spaja tačke $\mathbf{x}_1$ i $\mathbf{x}_2$ može predstaviti u obliku $\lambda\,\mathbf{x}_1 + (1-\lambda)\ \mathbf{x}_2$ gdje je λ neki realni broj iz segmenta $[0, 1]$, možemo reći da je skup $C \subseteq \mathbf{R}^n$ konveksan ukoliko za svako $\mathbf{x}_1, \mathbf{x}_2 \in C$ i svako $\lambda \in [0, 1]$ vrijedi

$$\lambda\,\mathbf{x}_1 + (1-\lambda)\ \mathbf{x}_2 \in C$$

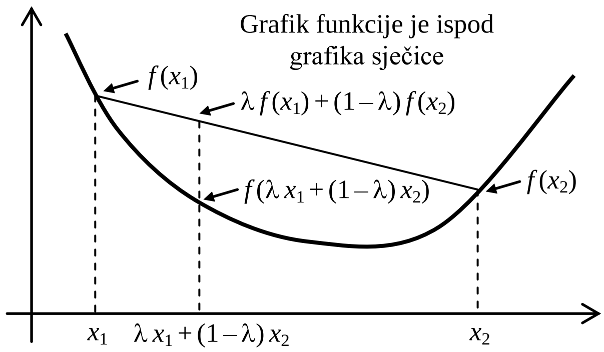
Ova definicija zapravo vrijedi ne samo za podskupove skupa $\mathbf{R}^n$, nego za podskupove proizvoljnog domena Δ takvog da su u njemu definirane *operacija sabiranja* i *operacija množenja sa realnim brojem*. Takvi skupovi Δ nazivaju se (***realni***) ***vektorski prostori***. Sljedeća slika ilustrira razliku između konveksnih i nekonveksnih skupova na primjeru dva podskupa skupa $\mathbf{R}^2$.



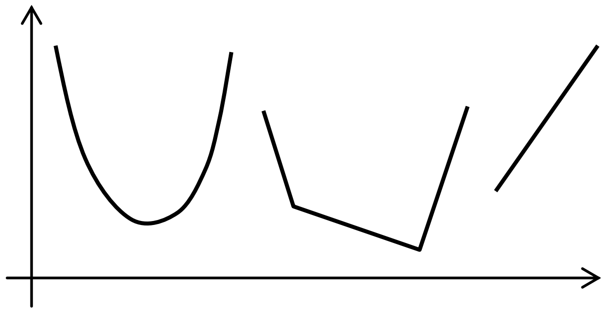
Za funkciju $f: \mathbf{R}^n \to \mathbf{R}$ kažemo da je *konveksna* na konveksnom skupu C ukoliko za svako $\mathbf{x}_1, \mathbf{x}_2 \in C$ i svako $\lambda \in [0, 1]$ vrijedi

$$f(\lambda\,\mathbf{x}_1 + (1-\lambda)\ \mathbf{x}_2) \le \lambda\,f(\mathbf{x}_1) + (1-\lambda)\,f(\mathbf{x}_2)$$

Pojednostavljeno rečeno, to znači da se grafik funkcije uvijek nalazi *ispod* tačaka ma koje duži (sječice) koja spaja ma koje dvije tačke koje leže na grafiku funkcije, kao što je to ilustrirano na sljedećoj slici na primjeru jedne funkcije jedne realne promjenljive, tj. funkcije $f: \mathbf{R} \to \mathbf{R}$.



Definicija konveksnosti se očigledno može primijeniti i na funkcije sa ma kakvim domenom Δ koji je vektorski prostor. Sljedeća slika prikazuje neke oblike konveksnih funkcija jedne realne promjenljive.



Za funkciju $f : \mathbf{R}^n \to \mathbf{R}$ (ili, općenitije, $f : \Delta \to \mathbf{R}$ gdje je Δ neki vektorski prostor) kažemo da je ***strogo konveksna*** na konveksnom skupu C ukoliko za svako $\mathbf{x}_1, \mathbf{x}_2 \in$ C, $\mathbf{x}_1 \neq \mathbf{x}_2$ i svako $\lambda \in (0, 1)$ vrijedi

$$f(\lambda\, \mathbf{x}_1 + (1-\lambda)\; \mathbf{x}_2) < \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2)$$

Slučajevi $\mathbf{x}_1 = \mathbf{x}_2$, $\lambda = 0$ i $\lambda = 1$ su isključeni jer su tada u prethodnoj nejednakosti lijeva i desna strana uvijek jednake, te tada nejednakost ne može vrijediti ni za kakvu funkciju *f*. Iz predhodnih definicija se lako može zaključiti da je *linearna funkcija koveksna* ali da *nije strogo konveksna*.

Ukoliko u prethodnim definicijama obrnemo smisao nejednakosti, tj. zamijenimo "$\le$" i "$<$" sa "$\ge$" i "$>$", dobijamo definiciju ***konkavne*** odnosno ***strogo konkavne*** funkcije. Lako se vidi da ako je funkcija *f* konveksna odnosno strogo konveksny, tada je funkcija $-f$ konkavna odnosno strogo konavna. Vrijedi i obrnuto.

Ako je funkcija *f* diferencijabilna, konveksnost funkcije se može definirati i na sljedeći način. Funkcija *f*, diferencijabilna na skupu C, je konveksna ako i samo ako za svako $\mathbf{x}_1, \mathbf{x}_2 \in$ C vrijedi

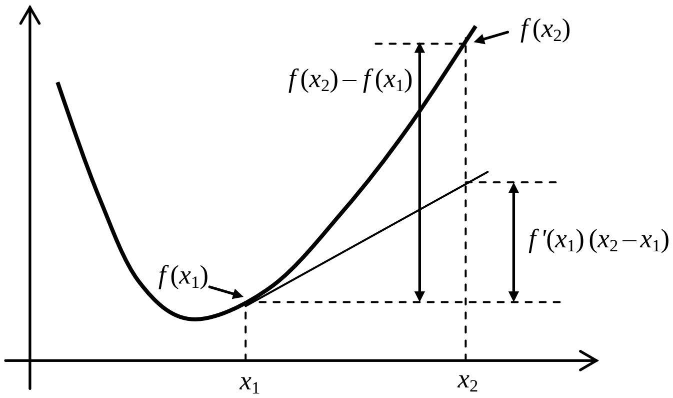
$$\nabla f(\mathbf{x}_1)^{\mathrm{T}}\, (\mathbf{x}_2 - \mathbf{x}_1) \le f(\mathbf{x}_2) - f(\mathbf{x}_1)$$

Ovdje je ∇f ***gradijent*** funkcije *f*, odnosno vektor koji se sastoji od parcijalnih izvoda funkcije *f* po svim komponentama x_i, $i = 1\,..\,n$ promjenljive $\mathbf{x}$:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)^{\mathrm{T}}$$

Slično se može definirati *stroga konveksnost*. Naime ako u gornjoj relaciji znak "$\le$" zamjenimo znakom "$<$" dobićemo *uvjet za strogu konveksnost funkcije f.*

Na sljedećoj slici je data geometrijska interpretacija konveksnosti za diferencijabilnu funkciju jedne promjenljive. Za taj specijalan slučaj imamo prosto da je $\nabla f(\mathbf{x}_1)^{\mathrm{T}} = f'(\mathbf{x}_1)$.



Ustanovljavanje konveksnosti (konkavnosti) neke funkcije f na osnovu prethodno prikazanih relacija *nije nimalo jednostavno*. Ukoliko je funkcija *f dva puta diferencijabilna*, tada se konveksnost (konkavnost) može ustanoviti na osnovu matrice $\nabla^2 f$ sastavljene od *drugih parcijalnih izvoda* funkcije f koja se naziva ***Hesseova matrica*** ili ***Hessian*** funkcije f i ponekad obilježava sa **H**:

$$\nabla^2 f = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1{}^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2{}^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n{}^2} \end{pmatrix}$$

Za tu svrhu, od velike važnosti je pojam ***definitnosti*** (***određenosti***) ***matrice*** (ovaj pojam se najčešće definira samo za simetrične matrice, ali za naše potrebe to će biti dovoljno). Neka je data neka (simetrična) matrica

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

Svakoj takvoj matrici možemo pridružiti jednu kvadratnu funkciju

$$Q(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j}\, x_i\, x_j$$

koja se, u matrično-vektorskoj notaciji, može kompaktno zapisati i u sljedećem obliku (što je lako provjeriti):

$$Q(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\, \mathbf{x}$$

Sada, za matricu **A** kažemo da je:

- ***pozitivno definitna*** ukoliko za sve tačke **x** iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) > 0$;
- ***pozitivno semidefinitna*** ukoliko za sve tačke **x** iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) \geq 0$;
- ***negativno definitna*** ukoliko za sve tačke **x** iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) < 0$;
- ***pozitivno semidefinitna*** ukoliko za sve tačke **x** iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) \leq 0$;
- ***indefinitna*** (ili ***nedefinitna***) ukoliko za neke tačke **x** iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) > 0$, a za neke druge tačke iz $\mathbb{R}^n$ vrijedi $Q(\mathbf{x}) < 0$.

Očigledno je provjeriti definitnost matrice po definiciji veoma teško. Srećom, postoje kriteriji koji se lako provjeravaju, a pomoću kojih je moguće testirati definitnost matrice. Jedan od najlakših za primjenu za matrice manjih dimenzija je ***Sylvesterov kriterij***, koji traži da se izračunaju *vodeći glavni dijagonalni minori* matrice **A**

$$\Delta_1 = a_{1,1} \qquad \Delta_2 = \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \qquad \Delta_3 = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{vmatrix} \qquad \ldots \qquad \Delta_n = \begin{vmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n} \end{vmatrix}$$

Prema Sylvesterovom kriteriju, matrica **A** je:

- *pozitivno definitna* ukoliko vrijedi $\Delta_1 > 0$, $\Delta_2 > 0$, $\Delta_3 > 0$, ..., $\Delta_n > 0$;
- *negativno definitna* ukoliko vrijedi $\Delta_1 < 0$, $\Delta_2 > 0$, $\Delta_3 < 0$, ..., $(-1)^n \Delta_n > 0$ (tj. ukoliko sekvenca minora Δ_1, Δ_2, Δ_3, ..., Δ_n alternativno mijenja znak, pri čemu je prvi negativan);
- *indefinitna* ukoliko sekvenca minora Δ_1, Δ_2, Δ_3, ..., Δ_n mijenja znak bez ikakvog određenog reda.

U slučajevima kada među vodećim glavnim dijagonalnim minorima ima nekih koji su jednaki nuli, Sylvesterov kriterij ne daje odgovor o prirodi matrice **A**. Matrica tada može biti semidefinitna (pozitivno ili negativno) ili indefinitna, te su potrebna dalja ispitivanja. Pozitivnu (negativnu) semidefinitnost je moguće ustanoviti (kod velikih matrica to iziskuje prilično mnogo računanja) ispitivanjem znakova *svih minora simetričnih u odnosu na glavnu dijagonalu*, a ne samo *vodećih glavnih dijagonalnih minora* (matrica je pozitivno semidefinitna ukoliko su svi takvi minori nenegativni).

Sada možemo dati kriterij za ispitivanje konveksnosti (konkavnosti) funkcije f na osnovu definitnosti matrice $\nabla^2 f$. Ukoliko je $\nabla^2 f(\mathbf{x})$ *pozitivno* (*negativno*) *semidefinitna* matrica za svako $\mathbf{x} \in$ C, tada je funkcija f *konveksna* (*konkavna*) na skupu C. Ukoliko je $\nabla^2 f(\mathbf{x})$ *pozitivno* (*negativno*) *definitna* matrica za svako $\mathbf{x} \in$ C, tada je f *strogo konveksna* (*strogo konkavna*) na skupu C.

Skup problema matematičkog programiranja, kod kojeg je potrebno *minimizirati konveksnu funkciju* (ili *maksimizirati konkavnu funkciju*) funkciju na *konveksnom skupu*, naziva se problemima ***konveksnog programiranja***. Ovaj tip problema matematičkog programiranja je zanimljiv jer spada u porodicu tzv. *jednoekstremalnih zadataka*, kod kojih je *lokalni minmum* (*maksimum*), ukoliko takav postoji, sigurno i *globalni maksimum* (*minimum*). To znači da je kod problema konveksnog programiranja dovoljno za neku tačku dokazati da predstavlja tačku lokalnog minimuma (maksimuma), pa da time bude dokazano da je ta tačka ujedno i tačka globalnog minimuma (maksimuma), odnosno da predstavlja *optimalno rješenje*. Na ovaj način je izbjegnut težak zadatak nalaženja globalnog ekstrema.

Opći oblik problema konveksnog programiranja je:

$$\arg\min f(\mathbf{x})$$
$$\text{p.o.}$$
$$\mathbf{x} \in \Omega$$

gdje je $f : \mathbf{R}^n \rightarrow \mathbf{R}$ *konveksna funkcija* (ili *konkavna* ako se umjesto minimizacije traži maksimizacija) a $\Omega \subseteq \mathbf{R}^n$ je *konveksan skup*. Predstavi li se skup Ω pomoću nejednakosti, ovo se svodi na

$$\arg\min f(\mathbf{x})$$
$$\text{p.o.}$$
$$g_j(\mathbf{x}) \le 0, \;\; j = 1 \,..\, m$$

gdje su ne samo funkcija cilja f nego i funkcije ograničenja g_j, $j = 1 \,..\, m$ konveksne funkcije.

Problemi konveksnog programiranja posjeduju sljedeće važne osobine:

- *Svaki lokalni ekstrem* na skupu Ω je *ujedno i globalni ekstrem*, pa prema tome on predstavlja traženo *optimalno rješenje*;
- Ukoliko je f *strogo konveksna funkcija*, onda problem može imati *najviše jedan lokalni ekstrem*, pa samim tim i *najviše jedno optimalno rješenje*;
- Ukoliko je skup Ω *neprazan* i *ograničen*, onda problem ima *barem jedan lokalni ekstrem*, pa samim tim i *barem jedno optimalno rješenje*.

Grafička interpretacija problema matematičkog programiranja

Dvodimenzionalni problemi matematičkog programiranja mogu se lijepo grafički interpretirati. Kroz takvu interpretaciju mogu se lijepo prikazati osnovne osobenosti zadataka traženja ekstrema uz ograničenja, a u slučaju *jednostavnijih problema*, moguće je grafičkim putem i *naći rješenje*, koristeći neka *elementarna geometrijska razmatranja*. Ovo će biti ilustrirano kroz nekoliko posve jednostavnih primjera. Grafička interpretacija je načelno moguća i za *trodimenzionalne probleme* (uz pomoć *prostornih crteža*) samo je takva interpretacija najčešće *iznimno nepregledna*, dok za probleme čija je dimenzionalnost veća od tri nikakva grafička interpretacija nije moguća.

➢ ***Primjer*** : Grafički predstaviti i riješiti problem matematičkog programiranja

$\arg\min Z = (x_1 - 8)^2 + (x_2 - 6)^2$

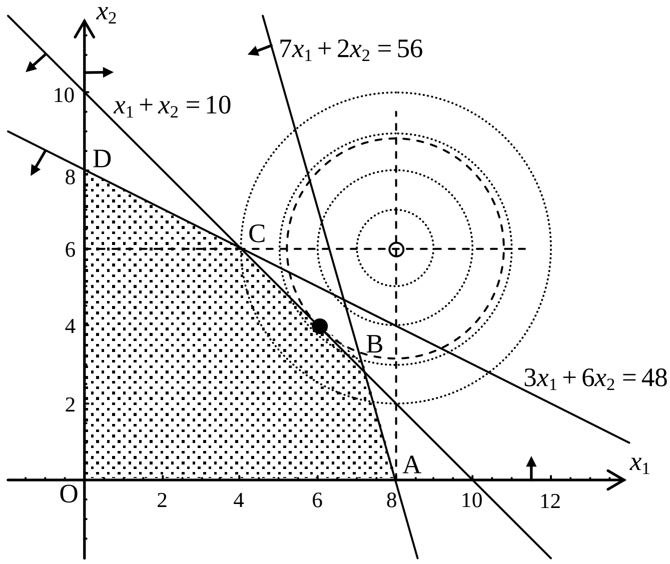
p.o.

$3x_1 + 6x_2 \le 48$
$x_1 + x_2 \le 10$
$7x_1 + 2x_2 \le 56$
$x_1, x_2 \ge 0$

U ovom primjeru, *dopustivi prostor* odnosno *skup mogućih rješenja* određen je pomoću *pet ograničenja* zadatih u vidu *nejedačina*. Svaka nejednačina dijeli x_1–x_2 ravan na *dva poluprostora* od kojih je jedan *dopustiv* a drugi *nedopustiv* u odnosu na razmatranu nejednačinu. Granice dopustivih poluprostora su:

- *koordinatne ose*; zbog nenegativnosti promjenljivih dopustivi prostor se nalazi u *prvom kvadrantu*;
- *pravac* $3x_1 + 6x_2 = 48$; dopustivi poluprostor je *ispod ovog pravca*, s obzirom da tačka (0,0), koja *zadovoljava* nejednačinu $3x_1 + 6x_2 \le 48$, leži također ispod ovog pravca;
- *pravac* $x_1 + x_2 = 10$; dopustivi poluprostor je *ispod ovog pravca*;
- *pravac* $7x_1 + 2x_2 = 56$; dopustivi poluprostor je *ispod ovog pravca*.

Presjek ovih dopustivih poluprostora daje *dopustivi prostor zadatka*. U ovom slučaju, taj prostor je lik (poligon) OABCD prikazan osjenčeno na sljedećoj slici. Tjemena ovog poligona imaju redom koordinate O(0, 0), A(8, 0), B(7.2, 2.8), C(4, 6) i D(0, 8).



Pravci oblika $p\,x_1 + q\,x_2 = r$ najlakše se crtaju koristeći činjenicu da takvi pravci moraju prolaziti kroz tačke $(0, r/q)$ i $(r/p, 0)$. Tako, na primjer, pravac $3x_1 + 6x_2 = 48$ prolazi kroz tačke (0, 8) i (16, 0). Što se tiče koordinata tjemena dozvoljene oblasti OABCD, one se mogu naći rješavanjem sistema jednačina koje obrazuju pravci koji se sijeku u razmatranom tjemenu. Recimo, koordinata tjemena B dobija se rješavanjem sistema jednačina $x_1 + x_2 = 10$ i $7x_1 + 2x_2 = 56$ (rješenje ovog sistema je $x_1 = 7.2$ i $x_2 = 2.8$, a upravo su to koordinate tačke B).

Funkcija cilja u ovom primjeru predstavlja funkciju dvije promjenljive čiji je grafik *rotacioni paraboloid*. Da bismo izbjegli crtanje funkcija koje zavise od dvije promjenljive, a koje bi se moglo obaviti jedino *u trodimenzionalnom prostoru*, ovu funkciju ćemo predstaviti ***nivo linijama***, odnosno linijama koje predstavljaju geometrijska mjesta tačaka u kojima funkcija ima *konstantnu vrijednost*. Nivo linije se dobiju kada se razmatrana funkcija (u našem slučaju funkcija cilja) izjednači sa *konstantom*. U ovom primjeru, za nivo linije dobijamo jednačinu kružnice oblika

$$(x_1-8)^2+(x_2-6)^2=c$$

gdje je c proizvoljna konstanta koja u ovom konkretnom slučaju mora biti veća ili jednaka nuli (u suprotnom, gornja relacija ne bi imala realna rješenja za x_1 i x_2). Pustimo li da konstanta c uzima različite konkretne vrijednosti, dobićemo *niz međusobno koncentričnih kružnica*, čiji poluprečnici iznose $\sqrt{c}$. Na prethodnoj slici su tačkastim linijama prikazane ove kružnice za $c = 1$, $c = 4$, $c = 9$ i $c = 16$, što daje kružnice sa poluprečnicima 1, 2, 3 i 4 respektivno. Za $c = 0$ ove kružnice se svode samo na jednu tačku sa koordinatama $x_1 = 8$ i $x_2 = 6$, koja predstavlja zajednički centar svih ovih kružnica. Ta tačka je ujedno i *minimum funkcije cilja*, odnosno *bezuvjetni minimum*, jer je iz oblika funkcije cilja jasno da funkcija cilja ne može imati vrijednosti manje od 0. Međutim, ta tačka *nije optimalno rješenje* postavljenog problema, jer se uopće *ne radi o dozvoljenom rješenju*. Zaista, sa slike se vidi da ova tačka ne leži u dozvoljenoj oblasti, odnosno ne zadovoljava postavljena ograničenja. Ovo nije teško provjeriti i analitički, stavljajući vrijednosti koordinata u nejednačine koje opisuju ograničenja (na primjer, $8+6=14$ nije manje ili jednako od 10, tako da ograničenje $x_1 + x_2 \le 10$ nije zadovoljeno u ovoj tački).

Sa porastom c, poluprečnik kružnica koje predstavljaju nivo linije se povećava, odnosno *vrijednost funkcije cilja je tim veća* što je *veća kružnica koja predstavlja nivo liniju*, odnosno što je ona *udaljenija od tačke bezuvjetnog minimuma*. Pošto je naš cilj da nađemo *dopustivu tačku sa minimalnom vrijednosti funkcije cilja*, to će ona ležati na kružnici koja ima *što je god moguće manji poluprečnik*, a na kojoj se nalazi *makar jedna dopustiva tačka*. To je očigledno kružnica koja *tangira dopustivi prostor*, a koja je na prethodnoj slici prikazana *crtkanom linijom*.

Naše traženo optimalno rješenje je, dakle, upravo *tačka u kojoj ta kružnica tangira dopustivi prostor*. Sa slike vidimo da se ta tačka nalazi na presjeku pravca $x_1 + x_2 = 10$ koji predstavlja granicu jednog od ograničenja i pravca koji prolazi kroz centar kružnice a okomit je na ovaj pravac. Nađimo kako glasi jednačina takvog pravca. Jednačina pravca $x_1+x_2=10$ može se napisati u obliku $x_2=-x_1+10$, iz čega vidimo da koeficijent smjera ovog pravca iznosi $k = -1$. Koeficijent smjera normale na ovaj pravac iznosi $k'=-1/k$, tako da jednačina normale koja prolazi kroz centar kružnice (8,6) glasi $x_2-6=k'(x_1-8)$ odnosno $x_2-6=x_1-8$ ili, drugačije zapisano, $x_2=x_1+2$. Presječnu tačku ova dva pravca dobijamo rješavanjem sistema jednačina

$$\begin{aligned}x_2&=-x_1+10\\ x_2&=x_1+2\end{aligned}$$

Rješenje ovog sistema je $x_1 = 6$ i $x_2 = 4$, tako da je tačka preskeka $\mathbf{x}^* = (6,4)$, a vrijednost funkcije cilja u ovoj tački iznosi $Z^* = 8$. Očigledno je da je tačka $\mathbf{x}^*$ tačka *strogog lokalnog minimuma*, jer se vidi da se minimum nalazi samo u jednoj tački. Ujedno, s obzirom da su dopustivi prostor i funkcija cilja *konveksni*, ta tačka je ujedno i tačka *strogog globalnog minimuma*, odnosno *optimalno rješenje* postavljenog problema.

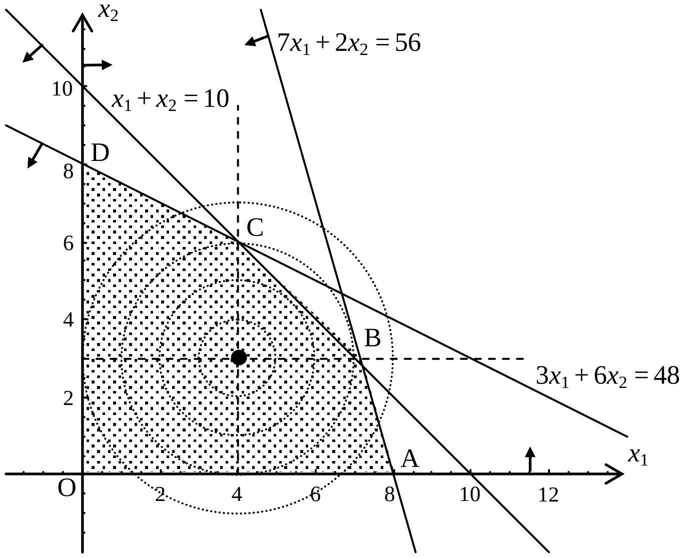
- ***Primjer*** : Grafički predstaviti i riješiti problem matematičkog programiranja

$$\arg\min Z=(x_1-4)^2+(x_2-3)^2$$

p.o.

$$\begin{aligned}3x_1+6x_2&\le 48\\ x_1+x_2&\le 10\\ 7x_1+2x_2&\le 56\\ x_1,x_2&\ge 0\end{aligned}$$

U ovom primjeru imamo *ista ograničenja* kao u prethodnom primjeru, tako da je dopustivi prostor odnosno skup mogućih rješenja *isti kao u prethodnom primjeru*, što je prikazano i na sljedećoj slici.



Grafik funkcije cilja je ponovo rotacioni paraboloid, a njene nivo linije su koncentrične kružnice

$$(x_1 - 8)^2 + (x_2 - 6)^2 = c$$

gdje je c ponovo proizvoljna konstanta koja mora biti veća ili jednaka nuli. Minimalna vrijednost funkcije cilja se dobija za vrijednost kostante $c = 0$. Ova vrijednost daje tačku sa koordinatama $x_1 = 4$ i $x_2 = 3$, koja je zajednički centar svih koncentričnih kružnica koje predstavljaju nivo linije. Ta tačka je ujedno i *bezuvjetni minimum funkcije cilja*. Vrijednost funkcije cilja u toj tački je dakle jednaka nuli. Pored toga, sa slike se vidi da ova tačka također i *zadovaljava ograničenja*, što se može provjeriti i analitički, stavljajući vrijednosti njenih koordinata u individualne nejednačine koje opisuju ograničenja ($3 \cdot 4 + 6 \cdot 3 = 40 \le 48$, $4 + 3 = 7 \le 10$, $7 \cdot 4 + 2 \cdot 3 = 34 \le 56$, $4 \ge 0$, $3 \ge 0$).

Iz izloženog slijedi da je tačka $\mathbf{x}^* = (4, 3)$ *barem lokalni minimum*, i to *strogi lokalni minimum* jer se minumum nalazi samo u jednoj tački. S druge strane, kako su funkcija cilja i dozvoljena oblast *konveksni*, to je ova tačka ujedno i tačka *strogog globalnog minimuma*. Dakle, optimalno rješenje ovog problema je $\mathbf{x}^* = (4, 3)$, a optimalna vrijednost funkcije cilja je $Z^* = 0$. Primijetimo da se, za razliku od prethodnog primjera u kojem se optimalno rješenje nalazilo *na rubu dozvoljene oblasti*, u ovom primjeru se optimalno rješenje nalazi *u unutrašnjoj tački dozvoljene oblasti*.

➢ ***Primjer*** : Grafički predstaviti i riješiti problem matematičkog programiranja

$$\arg\max Z = x_1 + 4x_2$$

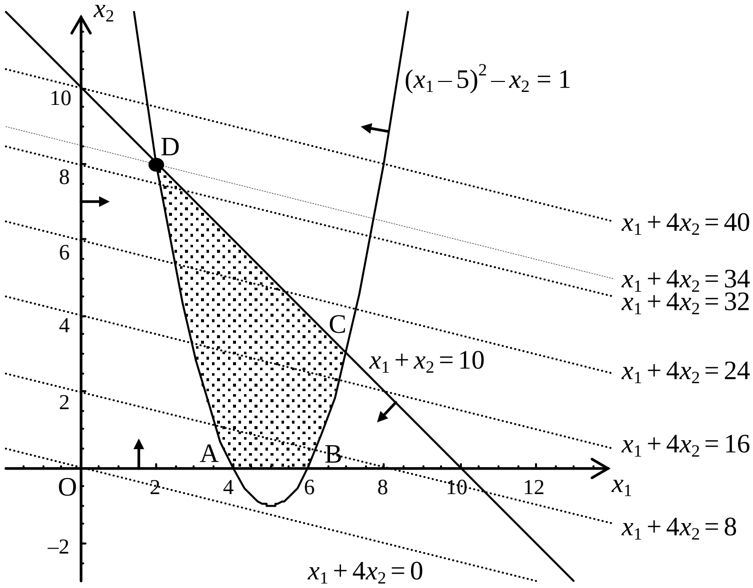
p.o.

$$(x_1 - 5)^2 - x_2 \le 1$$
$$x_1 + x_2 \le 10$$
$$x_1, x_2 \ge 0$$

U ovom primjeru, *dopustivi prostor* formiraju *četiri ograničenja* zadana u vidu nejednačina. Kao i obično, svaka od tih nejednačina dijeli x_1–x_2 ravan na dva poluprostora od kojih je jedan dopustiv a drugi nedopustiv u odnosu na razmatranu nejednačinu. Granice dopustivih poluprostora su:

- *koordinatne ose*; zbog nenegativnosti promjenljivih dopustivi prostor se nalazi u *prvom kvadrantu*;
- *pravac* $x_1 + x_2 = 10$; dopustivi poluprostor je *ispod ovog pravca*, s obzirom da tačka (0,0), koja *zadovoljava* nejednačinu $x_1 + x_2 \le 10$, leži također ispod ovog pravca;
- *kriva linija* (tačnije, *parabola*) $(x_1 - 5)^2 - x_2 = 1$ odnosno, drugačije zapisano, $x_2 = (x_1 - 5)^2 - 1$; dopustivi prostor je *unutar* (odnosno *iznad*) ove *parabole*, s obzirom da tačka $(0, 0)$ koja *ne zadovoljava* nejednačinu $(x_1 - 5)^2 - x_2 \le 1$ leži *izvan* (odnosno *ispod*) ove parabole.

Presjek ovih dopustivih poluprostora daje *dopustivi prostor* zadatka, što je prikazano osjenčeno na sljedećoj slici.



Kao što vidimo, dopustivi prostor je u ovom primjeru neka vrsta *krivolinijskog poligona* ABCD čije su dvije stranice AB i CD *ravne* (*duži*), a dvije stranice BC i AD *zakrivljene* (*lukovi parabole*). Tačke A, B, C i D imaju redom koordinate A(4, 0), B(6, 0), C(7, 3) i D(2,8), što se ponovo može odrediti rješavanjem sistema jednačina koje obrazuju granice dopustivih poluprostora koje se sijeku u razmatranoj tački. Na primjer, koordinate tačaka C i D nalaze se na sjecištu pravca $x_1 + x_2 = 10$ i parabole $(x_1 - 5)^2 - x_2 = 1$. Rješavanjem sistema koji obrazuju ove dvije jednačine dobijaju se dva rješenja $x_1 = 7$ i $x_2 = 3$, odnosno $x_1 = 2$ i $x_2 = 8$, a to su upravo koordinate tačaka C i D.

Funkcija cilja u ovom primjeru predstavlja linearnu funkciju dvije promjenljive, a poznato je da je grafik takvih funkcija (u trodimenzionalnom prostoru) *ravan*. Ponovo, da bismo izbjegli komplikovano crtanje funkcija dvije promjenljive u trodimenzionalnom prostoru, ovu funkciju ćemo predstaviti nivo linijama, koje dobijamo kada funkciju izjednačimo sa konstantom. U ovom slučaju dobićemo jednačinu pravca

$$x_1 + 4\,x_2 = c$$

gdje je c proizvoljno izabrana konstanta. Pustimo li da ta konstanta uzima različite konkretne vrijednosti, dobićemo *niz međusobno paralelnih pravaca* (da su ti pravci paralelni, može se zaključiti iz činjenice da koeficijent smjera za proizvoljan pravac $p\,x_1 + q\,x_2 = r$ iznosi $k = -p/q$, odnosno ovisi samo od p i q a ne i od r). Ovi pravci, naravno, prolaze kroz tačke $(0, c/4)$ i $(c, 0)$. Neki od tih pravaca će *mimoilaziti* dopustivi prostor, neki će ga *sjeći*, a neki će ga *tangirati*. Pošto je funkcija cilja *linearna*, povećavanje konstante c će davati pravce koji će se pomjerati *u istom smjeru*, koji je *okomit na same pravce*. Na osnovu činjenice da pravci prolaze kroz tačke $(0, c/4)$ i $(c, 0)$ koje se *udaljavaju* od koordinatnog početka sa porastom c, to se i sami pravci udaljavaju od koordinatnog početka sa porastom c. Pošto je zadatak pronaći *dopustivu tačku* u kojoj funkcija cilja ima *najveću moguću vrijednost*, povećavaćemo konstantu c *koliko god je to moguće*, a da pri tome pravac koji predstavlja nivo liniju funkcije cilja ima *zajedničkih tačaka sa dopustivim prostorom*. Drugim riječima, povećavanjem c pomjeraćemo taj pravac *preko dopustivog prostora*, sve dok ne dobijemo pravac čije bi dalje pomjeranje dovelo do *napuštanja dopustivog prostora*. U našem primjeru, to je pravac koji tangira dopustivi prostor u tački D. Uvrštavanjem koordinata tačke D(2, 8) u jednačinu pravca $x_1 + 4\,x_2 = c$ dobijamo da taj pravac ima jednačinu $x_1 + 4\,x_2 = 34$. To znači da je D(2, 8) tačka u kojoj funkcija cilja ima (barem lokalni) *maksimum* i da vrijednost funkcije cilja u toj tački iznosi 34.

S obzirom da se traži *maksimizacija*, da je dopustivi prostor *konveksan* skup a da je funkcija cilja *konkavna* (linearna funkcija je *istovremeno* i konveksna i konkavna, ali *nije strogo* ni jedno ni drugo), to nam garantira da je nađeni maksimum ujedno i optimalno rješenje. Dakle, optimalno rješenje iznosi $x_1 = 2$ i $x_2 = 8$ odnosno $\mathbf{x}^* = (2, 8)$, dok optimalna vrijednost funkcije cilja iznosi $Z^* = 34$.

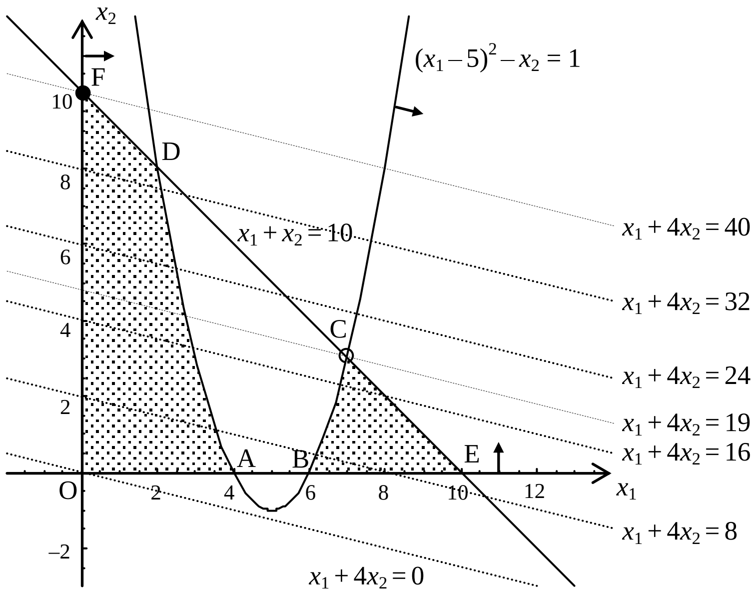
➢ ***Primjer***: Grafički predstaviti i riješiti problem matematičkog programiranja

$$\arg\max Z = x_1 + 4x_2$$

p.o.

$$(x_1 - 5)^2 - x_2 \geq 1$$
$$x_1 + x_2 \leq 10$$
$$x_1, x_2 \geq 0$$

Ovaj problem je gotovo isti kao prethodni, samo što je promijenjen smjer nejednakosti u prvom ograničenju. Stoga su granice dopustivih poluprostora *iste kao u prethodnom primjeru*, uz jedinu razliku što se dopustivi poluprostor čija je granica parabola $(x_1 - 5)^2 - x_2 = 1$ sada nalazi *izvan* (odnosno *ispod*) ove parabole, s obzirom da tačka $(0, 0)$ koja *zadovoljava* nejednačinu $(x_1 - 5)^2 - x_2 \geq 1$ također leži *izvan* (odnosno *ispod*) ove parabole. Posljedica ovoga je da će se u ovom primjeru dozvoljeni prostor sastojati od *dva nepovezana lika*, od kojih je jedan *krivolinijski četverougao* OADF (tri stranice su mu *duži* OA, OF i FD, dok mu je četvrta stranica luk parabole AB), a drugi *krivolinijski trougao* BCE (dvije stranice su mu duži BE i CE, dok mu je treća stranica luk parabole BC). Slijedi da dopustivi prostor *nije povezan skup*, pa samim tim *nije ni konveksan* (inače, može se primijetiti da čak ni dvije povezane cjeline na koje se dopustivi prostor raspada *nisu ni same po sebi konveksne*). Ova situacija je prikazana na sljedećoj slici.



Koordinate karakterističnih tačaka O, A, B, C, D, E i F mogu se naći na isti način kao u prethodnim primjerima i one glase O$(0, 0)$, A$(4, 0)$, B$(6, 0)$, C$(7, 3)$, D$(2, 8)$, E$(10, 0)$ i F$(0, 10)$. Ako sada na isti način kao u prethodnom primjeru ponovimo postupak pomjeranja nivo linija funkcije cilja povećavajući vrijednost konstante c, vidjećemo da će posljednja dopustiva tačka biti tačka F$(0, 10)$. Jednačina nivo linije (pravca) koji prolazi kroz tačku F glasi $x_1 + 4x_2 = 40$, što je lako utvrditi. To znači da je tačka F$(0, 10)$ tačka u kojoj funkcija cilja ima (barem lokalni) *maksimum* i da je vrijednost funkcije cilja u tom maksimumu jednaka 40. Iz načina kako smo dobili rješenje lako se vidi da je ovaj lokalni maksimum ujedno i globalni. Slijedi da je tačka $\mathbf{x}^* = (0, 10)$ *optimalno rješenje*, dok je optimalna vrijednost funkcije cilja $Z^* = 40$.

Treba obratiti pažnju na sljedeće. Kako dopustivi prostor nije konveksan, *svaki lokalni maksimum ne mora biti ujedno i globalni*. Zbog grafičkog načina rješavanja (gdje cijelo vrijeme imamo uvid u cjelokupni dopustivi prostor) jasno je da je gore pronađeno rješenje zaista i globalni maksimum. Međutim, ovo *nije jedini lokalni maksimum* za ovaj problem. Zaista, nije teško uvidjeti da ova funkcija cilja na zadanom dopustivom prostoru *ima još jedan strogi lokalni maksimum* koji se nalazi u tački C$(7, 3)$. Zaista, vidimo da se funkcija cilja ne može dalje povećavati *ukoliko ostajemo samo u bliskoj okolini ove tačke*. U ovoj tački funkcija cilja ima vrijednost $Z = 19$, tako da ovo nije i globalni maksimum. Međutim, ukoliko ovaj problem ne bismo rješavali grafički nego *nekim numeričkim postupkom*, sasvim je moguće da bi taj postupak pronašao *upravo ovaj lokalni maksimum*, a da ne bismo imali ikakvu indikaciju da *on nije optimalno rješenje*, nego da se optimalno rješenje *nalazi u nekom drugom lokalnom maksimumu*. Ovo ilustrira razlog zašto su problemi matematičkog programiranja *mnogo teži* ukoliko se *ne radi o problemima konveksnog programiranja* nego u slučaju problema konveksnog programiranja.

➢ ***Primjer*** : Grafički predstaviti i riješiti problem matematičkog programiranja

$$\arg\max Z = (x_1 - 0.4)^2 + (x_2 - 0.7)^2$$

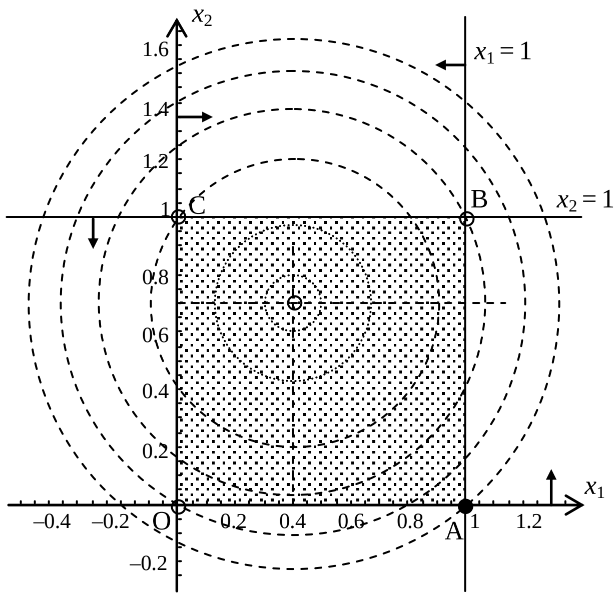
p.o.

$$0 \le x_1 \le 1$$
$$0 \le x_2 \le 1$$

Ovo je još jedan vrlo karakterističan primjer koji spada u probleme *nekonveksnog programiranja.* Naime, mada se lako vidi da je dopustivi prostor kvadrat čija su tjemenima O(0, 0), A(1, 0), B(1, 1) i C(0, 1), što je konveksan skup, funkcija cilja je također konveksna, a za problem maksimizacije ona bi trebala biti konkavna da problem bude problem konveksnog programiranja. Stoga se može očekivati da problem može imati više lokalnih ekstrema koji nisu globalni. Nivo linije u ovom primjeru su koncentrične kružnice

$$(x_1 - 0.4)^2 + (x_2 - 0.7)^2 = c$$

gdje je c proizvoljna nenegativna konstanta. Sa sljedeće slike, lako se može uočiti da sva četiri tjemena O, A, B i C predstavljaju *lokalne maksimume,* dok se globalni maksimum nalazi u tački A. Dakle, optimalno rješenje problema je u tački $\mathbf{x}^* = (1, 0)$ koje daje optimalnu vrijednost funkcije cilja $Z^* = 0.85$.



Prethodni primjer se može lako generalizirati. Naime, neka je dat problem sa n promjenljivih sljedećeg oblika, pri čemu su a_i, $i = 1 .. n$ neke konstante za koje vrijedi $0 < a_i < 1$, $i = 1 .. n$:

$$\arg\max Z = \sum_{i=1}^{n} (x_i - a_i)^2$$

p.o.

$$0 \le x_i \le 1, \ \ i = 1 .. n$$

Rezonovanjem analognim kao u prethodnom primjeru nije teško zaključiti da dopustivi prostor predstavlja n-dimenzionalnu kocku sa tjemenima u tačkama oblika $(x_1^*, x_2^*, ..., x_n^*)$ gdje su sve vrijednosti x_i^*, $i = 1 .. n$ nule i jedinice, pri čemu je svaka takva tačka ujedno i lokalni maksimum. Međutim, takvih tačaka očigledno ima 2^n. Ovaj primjer pokazuje da čak i problemi kvadratnog programiranja (u koje ovaj problem očigledno spada) ali koji nisu konveksni problemi mogu imati eksponencijalno mnogo lokalnih ekstrema (posmatrano kao funkcija dimenzionalnosti problema n), što je neprihvatljivo mnogo za iole veće vrijednosti n. U ovom konkretnom primjeru, nije teško uvidjeti kako se može utvrditi koja je od tih tačaka globalni maksimum (probajte sami otkriti kako), ali da je slučajno dozvoljena oblast bila iole nepravilnijeg oblika, to ne bi bilo moguće. Odavde slijedi da nalaženje globalnog ekstrema za probleme nekonveksnog programiranja može biti izuzetno težak zadatak čak i u slučaku kada je funkcija cilja sasvim jednostavna, kao što je npr. kvadratna funkcija. Drugim riječima, već su i problemi kvadratnog programiranja u odsustvu konveksnosti iznimno teški za rješavanje, zbog neprihvatljivo velikog broja potencijalnih rješenja.