

UNIVERZITET U NIŠU  
PRIRODNO-MATEMATIČKI FAKULTET

Ivan P. Stanimirović

**ALGORITMI ZA SIMBOLIČKA  
MATRIČNA IZRAČUNAVANJA  
I OPTIMIZACIJU**

Doktorska disertacija

Niš, decembar 2012.

*Simboličko izračunavanje (ili kompjuterska algebra)* je važan koncept u matematici i računarskim naukama. Odnosi se na izučavanje i izradu algoritama i softvera za manipulaciju matematičkim izrazima i drugim matematičkim objektima. Simboličko izračunavanje podrazumeva ekzaktno izračunavanje izraza koji sadrže promenljive koje nemaju zadatu vrednost i time su uzimane kao simboli (otuda i naziv simboličko izračunavanje). Kada se traži rezultat izračunavanja bez numeričkih grešaka, simboličko računanje je apsolutno nezamenljiva tehnika. Zbog toga je razvijeno dosta softverskih alata koji pružaju mogućnosti simboličkog izračunavanja (sistema za kompjutersku algebru). Među njima je najmoćniji i najviše u upotrebi programski paket MATHEMATICA.

*Na samim počecima simboličkog izračunavanja, oko 1970. godine, kada su implementirani prvi složeniji algoritmi, ispostavilo se da su veoma neefikasni. Zato je obimno istraživanje bilo usmereno ka reviziji klasične algebre u smislu efikasnosti i da bi se razvili efikasni algoritmi. Tipičan primer je izračunavanje polinomijalnog najvećeg zajedničkog delioca, koji je neophodan kod simplifikacije razlomaka. Danas je kompjuterska algebra široko u upotrebi pri eksperimentima u matematici i izradi formula korišćenih u numeričkim programima. Takođe se koristi za kompletна izračunavanja, kada ne uspevaju čisto numerički metodi.*

*U ovoj disertaciji razmatrane su dve primene simboličkih transformacija: simboličko izračunavanje uopštenih inverza konstantnih, racionalnih i polinomijalnih matrica i višekriterijumska optimizacija. Disertacija je rezultat našeg višegodišnjeg bavljenja tematikom što je dovelo do publikovanja većeg broja naučnih radova u časopisima svetskog i nacionalnog značaja. Ova doktorska disertacija se prvenstveno oslanja na originalne radove [77, 78, 96, 83]. Pojedini rezultati iz ove disertacije prezentovani su na međunarodnoj konferenciji pod nazivom "14th Applied Stochastic Models and Data Analysis Conference" održanoj u Rimu juna 2011. godine.*

*Sa posebnim zadovoljstvom zahvaljujem se svom mentoru, Prof. dr Milanu Tasiću, ne samo na pomoći pri izradi ove disertacije, već i na velikoj pažnji i vremenu koje mi je posvetio uvodeći me u probleme sa simboličkom implementacijom i izračunavanjem, još od vremena kada mi je kao učeniku gimnazije predavao. Kao rezultat zajedničkog rada i njegove nesebične pomoći, publikовано је неколико naučnih radova u vrhunskim i istaknutim međunarodnim časopisima, као и u vodećim časopisima nacionalnog značaja.*

*Veliku zahvalnost dugujem Prof. dr Marku Petkoviću i Prof. dr Nebojši Stojkoviću, pre svega na dosadašnjoj uspešnoj naučnoj saradnji na polju višekriterijumske optimizacije, као и na nizu korisnih sugestija i saveta koji su drastično poboljšali kvalitet ove disertacije.*

*Zahvaljujem se profesorima i kolegama sa kojima sam naučno sarađivao: Prof. dr Miroslavu Ćiriću, Prof. dr Predragu Stanimiroviću, Doc. dr Jeleni Stefanović-Marinović, Doc. dr Miljanu Zlatanoviću i Doc. dr Aleksandru Iliću. Zahvaljujem se i grčkim kolegama sa univerziteta u Atini: Doc. dr Dimitrios Pappasu i Doc. dr Vasiliou Katsikisu, sa kojima sam sarađivao u radovima sa simboličkim izračunavanjem nekih uopštenih inverza matrica.*

*Najtoplje se zahvaljujem članovima svoje porodice, koji su razumevanjem i podrškom tokom mojih doktorskih studija takođe doprineli da ova doktorska disertacija bude kompletirana.*

*U Nišu, 24. 12. 2012.*

*Ivan Stanimirović*

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
1.1	Osnovni pojmovi . . . . .	5
1.2	Predstavljanje uopštenih inverza faktorizacijom potpunog ranga . . . . .	6
1.3	LU dekompozicija . . . . .	9
1.4	Struktura i organizacija rada . . . . .	11
<b>2</b>	<b>Simbolička matrična izračunavanja</b>	<b>13</b>
2.1	Izračunavanje $\{i, j, k\}$ inverza i uopštenih inverza racionalnih matrica . . . . .	13
2.1.1	Implementacioni detalji i rezultati testiranja . . . . .	17
2.2	$LDL^*$ dekompozicija potpunog ranga polinomijalne matrice . . . . .	19
2.2.1	Numerički primeri . . . . .	24
2.2.2	Implementacioni detalji i razultati testiranja . . . . .	26
2.3	Izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice . . . . .	29
2.3.1	Numerički primeri . . . . .	32
2.3.2	Implementacioni detalji . . . . .	34
2.4	$LDL^*$ dekompozicija potpunog ranga racionalne matrice . . . . .	35
2.5	Izračunavanje Moore-Penrose-ovog inverza racionalne matrice . . . . .	41
2.5.1	Numerički primeri i rezultati testiranja . . . . .	44
2.6	Izračunavanje $A_{T,S}^{(2)}$ inverza $LDL^*$ dekompozicijom . . . . .	46
2.6.1	Numerički primeri . . . . .	51
2.6.2	Implementacioni detalji i rezultati testiranja . . . . .	53
2.7	Reprezentacije potpunog ranga zasnovane na $QR$ dekompoziciji . . . . .	55
2.7.1	Složenost i implementacija algoritama QRATS2 i qrginv . . . . .	61
2.7.2	Od QRATS2 do qrginv i obratno . . . . .	63
2.7.3	Numerički eksperimenti za Moore-Penrose-ov inverz . . . . .	64
2.7.4	Numerički eksperimenti za Drazinov inverz . . . . .	67
2.8	Simboličko izračunavanje $A_{T,S}^{(2)}$ inverza QDR faktorizacijom . . . . .	69
2.8.1	Eksperimenti sa polinomijalnim i racionalnim matricama . . . . .	74
2.8.2	Implementacioni detalji i rezultati testiranja . . . . .	76
2.9	Uopšteni inverzi blokovskih matrica . . . . .	78
2.9.1	Blokovske reprezentacije generalisanih inverza . . . . .	81
2.10	Blokovska $LDL^*$ dekompozicija potpunog ranga . . . . .	95
2.10.1	Inverzi $n \times n$ blokovskih matrica . . . . .	100
2.10.2	Moore-Penrose-ovi inverzi $2 \times 2$ blokovskih matrica . . . . .	102
2.11	Grevilov metod pregrađivanja . . . . .	107

<b>3 Višekriterijumska optimizacija</b>	<b>112</b>
3.1 Osnovni pojmovi . . . . .	113
3.2 Simboličke transformacije u višekriterijumskoj optimizaciji . . . . .	115
3.3 Ispitivanje Pareto optimalnosti . . . . .	117
3.4 Metod težinskih koeficijenata . . . . .	119
3.4.1 Linearni metod težinskih koeficijenata . . . . .	125
3.4.2 Implementacioni detalji za dvodimenzionalni slučaj . . . . .	127
3.4.3 Implementacioni detalji za trodimenzionalni slučaj . . . . .	131
3.5 Leksikografska višekriterijumska optimizacija . . . . .	133
3.5.1 Primena geometrijskog metoda u leksikografskom metodu . . . . .	136
3.5.2 Modifikovani algoritam leksikografskog metoda . . . . .	138
3.5.3 Numerički eksperimenti . . . . .	140
3.6 Analiza drugih metoda višekriterijumske optimizacije . . . . .	142
3.6.1 Relaksirani leksikografski metod . . . . .	142
3.6.2 Metod $\varepsilon$ -ograničenja . . . . .	145
3.6.3 Metodi rastojanja . . . . .	147
<b>4 Zaključak</b>	<b>153</b>
<b>Literatura</b>	<b>156</b>

# 1. Uvod

U matematici i računarskim naukama, algoritam predstavlja korak-po-korak proceduru kojom se vrše izračunavanja. Algoritmi se primenjuju za izračunavanja, procesiranje podataka i automatsko zaključivanje. Preciznije, algoritam je efikasan metod izražen nekom konačnom listom definisanih instrukcija. Počev od inicijalnog koraka i ulaza (koji može biti i prazan), instrukcije opisuju izračunavanje, koje, nakon izvršenja, dovodi do konačnog broja uskcesivnih koraka, dajući izlazni rezultat u poslednjem koraku. Kompleksnost (složenost) nekog algoritma se može opisati kao broj primitivnih operacija ili osnovnih koraka koje je potrebno izvršiti za odredjene ulazne podatke (više o analizi algoritama može se naći u [15]).

Matrična izračunavanja nalaze primenu u većini naučnih oblasti. U svakoj grani fizike, uključujući klasičnu mehaniku, optiku, elektro-magnetizam, matrična izračunavanja se koriste za izučavanje fizičkih fenomena. U kompjuterskoj grafici, matrice se koriste za projektovanje 3-dimenzionalne slike na 2-dimenzionalni ekran. U teoriji verovatnoća i statistici, stohastičke matrice se koriste za opisivanje skupova verovatnoća. Matrice imaju veoma dugu istoriju primene u rešavanju linearnih jednačina. Veoma bitna grana numeričke analize je posvećena izradi efikasnih algoritama za matrična izračunavanja. Ovaj problem je vekovima poznat, i danas predstavlja rastuće polje istraživanja.

U linearnoj algebri, matrična dekompozicija predstavlja faktorizaciju matrice u neku kanoničku formu. Metodi matrične dekompozicije pojednostavljaju dalja izračunavanja, bilo teoretska ili praktična. Postoje algoritmi koji su pravljeni za pojedine tipove matrica, npr. retko posednute ili skoro-dijagonalne matrice. Postoji mnogo metoda za dekompoziciju matrica, gde se svaka koristi u zasebnim klasama problema. Dekompozicije vezane za rešavanje sistema linearnih jednačina su *LU dekompozicija*, *LU redukcija*, *Block LU dekompozicija*, *rang faktorizacija*, *Cholesky dekompozicija*, *QR dekompozicija*, *Singularna vrednosna dekompozicija*.

Simboličko izračunavanje je koncept koji označava upotrebu kompjuterskih alata za transformaciju matematičkih simboličkih izraza. Najčešće se primenjuje za izračunavanje eksplicitnih rezultata bez numeričkih grešaka. Zato se simboličko izračunavanje uvek primenjuje na loše uslovjenim problemima. Metodi sa simboličkim izračunavanjem imaju veliku primenu u manipulisanju komplikovanim izrazima više promenljivih. To mogu biti racionalne funkcije ili polinomi od jedne ili više promenljivih. Postoji više različitih programskih paketa za kompjutersku algebru (eng. Computer Algebra Software) koji podržavaju simboličko izračunavanje, kao što su **MATHEMATICA**, **MAPLE**, **MATLAB**.

Tradicionalni programski jezici su proceduralni jezici (**C**, **C++**, **Fortran**, **Pascal**, **Basic**). Proceduralni program je napisan kao lista instrukcija, koje se izvršavaju korak po korak

[119]. Programi u proceduralnim kompjuterskim jezicima kao što je C, se mogu koristiti u izračunavanjima, ali su ograničeni u razumevanju složenijih algoritama, jer daju malo informacije o međukoracima. Mnogo istraživača koristi mogućnost razvijanja "rapid-prototype" koda za testiranje ponašanja algoritma pre ulaganja truda u razvoj koda za algoritam u proceduralnom jeziku. Pristup sa MATHEMATICA-om ima velikih prednosti u istraživanju nad proceduralnom programiranju dostupnom u MAPLE-u i u proceduralnim jezicima [119]. MATHEMATICA dopušta nekoliko programskih paradigmi: objektno-orientisano, proceduralno, simboličko i funkcionalno programiranje [119]. Naš glavni cilj istraživanja bio je izrada algoritama podesnih za implementaciju kako u MATHEMATICA, tako i u proceduralnim programske jezicima.

Pod matricom sa simboličkim elementima podrazumevamo matricu čiji su elementi iz odgovarajućeg komutativnog prstena. Mnoge primene uključuju problem invertovanja matrice sa simboličkim elementima [1] ili slične probleme, kao npr. konstruisanje matrice minora. Faktorizacija polinomijalne matrice  $A$ , tj. matrice čiji su elementi polinomi, se često koristi za izračunavanje inverza i uopštenih inverza matrice  $A$ , [49, 67]. Kada je matrica  $A$  nesingularna, Gauss-ov algoritam je efikasan način za određivanje inverza  $A^{-1}$ . Neki polinomijalni algoritmi iz kompjuterske algebre su posmatrani u radu [112]. Programska paketi kao što su MATHEMATICA [113], MAPLE i MATLAB, koji podrazumevaju simbolička izračunavanja, sadrže ugrađene funkcije za faktorizaciju matrica. Međutim, zbog teškoća kod simplifikacije, nije praktično implementirati matrične faktorizacije na skupovima polinomijalnih ili racionalnih matrica u proceduralnim programske jezicima. Takođe, dekompozicija polinomijalne (racionalne) matrice, odnosno matrice sa elementima koji su polinomi (racionalne funkcije), je dosta drugačija i komplikovanija (videti [49, 67]). Nekoliko skorijih unapređenja na ovom polju smanjila su kompleksnosti većine problema sa polinomijalnim matricama na red veličine kompleksnosti množenja matrica, što je dosad najveće dostignuće kod kompleksnosti [24, 40, 52, 94].

Izračunavanje uopštenih inverza polinomijalne matrice  $A(x)$  je važan problem numeričke analize, koji se često javlja u inverznim sistemima [54] i rešenjima sistema diofantskih jednačina [42]. Jedan od algoritama za izračunavanje generalisanog inverza polinomijalne matrice  $A(x)$  je uveo Karampetakis u [42]. Zasnovan na ideji da se polinomijalna matrica može predstaviti kao lista konstantnih matrica, nekoliko novih algoritama za izračunavanje različitih tipova uopštenih inverza je izvedeno u radovima [99, 100]. Jedna od poteškoća kod ovih algoritama je njihova zavisnost od stepena matrice  $A(x)$ . U tom smislu smo uveli modifikacije poznatih algoritama baziranih na faktorizacijama potpunog ranga racionalnih matrica, radi poboljšanja vremena izračunavanja i stabilnosti.

Jedan od naših ciljeva bio je primena  $QDR$  i  $LDL^*$  dekompozicije potpunog ranga Hermitske polinomijalne matrice  $A$  (metod dobio u [96]) na reprezentaciju uopštenog inverza  $A_{T,S}^{(2)}$  iz rada [70]. Metod uveden u [70] je izведен na osnovu  $FG$  dekompozicije potpunog ranga matrice  $A$ . Time smo izveli proširenje ovog algoritma, primenljivu za izračunavanje široke klase  $A_{T,S}^{(2)}$  inverza, a ne samo Moore–Penrose-ovog inverza. Štaviše, razvijeno je proširenje ovog algoritma na skupu polinomijalnih matrica jedne promenljive. Umesto  $FG$  dekompozicije matrice  $A$  koristili smo  $QDR$  i  $LDL^*$  dekompozicije potpunog ranga odgovarajuće odabrane matrice  $M$ .

## 1.1 Osnovni pojmovi

Skup svih kompleksnih  $m \times n$  matrica označava se sa  $\mathbb{C}^{m \times n}$ , dok je  $\mathbb{C}_r^{m \times n} = \{X \in \mathbb{C}^{m \times n} \mid \text{rang}(X) = r\}$  njegov podskup koji se sastoji od kompleksnih matrica ranga  $k$ . Neka  $\mathbb{C}(x)$  (respektivno  $\mathbb{R}(x)$ ) označava skup racionalnih funkcija sa kompleksnim (respektivno realnim) koeficijentima. Tada se skup svih  $m \times n$  matrica sa elementima u  $\mathbb{C}(x)$  (respektivno  $\mathbb{R}(x)$ ) označava sa  $\mathbb{C}(x)^{m \times n}$  (respektivno  $\mathbb{R}(x)^{m \times n}$ ).

Sa  $I_r$  i  $I$  ćemo označavati jedniničnu matricu reda  $r$ , odnosno jediničnu matricu odgovarajućeg reda, respektivno. Sa  $\mathbf{O}$  se označava nula matrica odgovarajućeg reda veličine.

Za proizvoljnu matricu  $A \in \mathbb{C}(x)^{m \times n}$  posmatrajmo sledeće matrične jednačine po nepoznatoj  $X$ , gde  $*$  označava konjugovano transponovanu matricu:

$$(1) \quad AXA = A \quad (2) \quad XAX = X \quad (3) \quad (AX)^* = AX \quad (4) \quad (XA)^* = XA.$$

U slučaju  $m = n$  posmatraćemo i sledeće matrične jednačine

$$(5) \quad AX = XA \quad (1^k) \quad A^{k+1}X = A^k.$$

Posmatrajmo sada skup  $\mathcal{S} \subset \{1, 2, 3, 4\}$ . Tada se skup matrica koje zadovoljavaju jednačine sa indeksima iz skupa  $\mathcal{S}$  označava sa  $A\{\mathcal{S}\}$ . Proizvoljna matrica iz  $A\{\mathcal{S}\}$  se naziva  $\mathcal{S}$ -uopšteni inverz matrice  $A$ . Matrica  $X = A^\dagger$  je Moore-Penrose-ov inverz matrice  $A$  ako zadovoljava jednačine (1)–(4). Grupni inverz  $A^\#$  je jedinstveni  $\{1, 2, 5\}$  inverz matrice  $A$ , i postoji ako i samo ako je

$$\text{ind}(A) = \min_k \{k : \text{rang}(A^{k+1}) = \text{rang}(A^k)\} = 1.$$

Rang uopštenog inverza  $A^\dagger$  se često koristi u izračunavanjima, pa je potrebno posmatrati podskup  $A\{i, j, k\}_s$  skupa  $A\{i, j, k\}$ , koji se sastoji od  $\{i, j, k\}$ -inverza ranga  $s$  (videti [3]).

U ovoj disertaciji označavamo opseg (eng. range) matrice  $A \in \mathbb{C}(x)^{m \times n}$  sa  $\mathcal{R}(A)$ , a jezgro (kernel) (eng. null space) matrice  $A$  sa  $\mathcal{N}(A)$ . Takođe, sa  $\text{nrang}(A)$  ćemo označavati normalni rang matrice  $A$  (tj. rang nad skupom  $\mathbb{C}(x)$ ) a sa  $\mathbb{C}(x)_r^{m \times n}$  skup racionalnih matrica iz  $\mathbb{C}(x)^{m \times n}$  normalnog ranga  $r$ . Slično,  $\text{rang}(A)$  označava rang konstantne matrice  $A$  a  $\mathbb{C}_r^{m \times n}$  označava skup kompleksnih matrica  $\mathbb{C}^{m \times n}$  ranga  $r$ . Ovde navodimo stav iz [3, 105] kojim se definiše  $A_{T,S}^{(2)}$  uopšteni inverz.

**Stav 1.1.1.** [3, 105] *Ako je  $A \in \mathbb{C}_r^{m \times n}$ ,  $T$  podprostor prostora  $\mathbb{C}^n$  dimenzije  $t \leq r$  a  $S$  podprostor od  $\mathbb{C}^m$  dimenzije  $m-t$ , tada matrica  $A$  ima  $\{2\}$ -inverz  $X$  takav da je  $\mathcal{R}(X) = T$  i  $\mathcal{N}(X) = S$  ako i samo ako je  $AT \oplus S = \mathbb{C}^m$ , i u tom slučaju je  $X$  jedinstven i označava se sa  $A_{T,S}^{(2)}$ .*

Verovatno najviše izučavana klasa generalisanih inverza je klasa spoljašnjih inverza definisane slike i jezgra. Generalisani inverzi, kao što su Moore-Penrose-ov  $A^\dagger$ , težinski Moore-Penrose-ov  $A_{M,N}^\dagger$ , Drazinov  $A^D$  i grupni inverz  $A^\#$ , kao i Bott-Duffin-ov inverz  $A_{(L)}^{(-1)}$  i uopšteni Bott-Duffin-ov inverz  $A_{(L)}^{(\dagger)}$ , se mogu ujediniti na ovaj način i reprezentovati kao uopšteni inverzi  $A_{T,S}^{(2)}$  (pri čemu se posmatraju odgovarajuće matrice  $T$  i  $S$ ). Za datu matricu  $A \in \mathbb{C}^{m \times n}(x)$  važe sledeće reprezentacije (za više detalja videti [3, 105]):

$$A^\dagger = A_{\mathcal{R}(A^*), \mathcal{N}(A^*)}^{(2)}, \quad A_{M,N}^\dagger = A_{\mathcal{R}(A^\#), \mathcal{N}(A^\#)}^{(2)}, \quad (1.1.1)$$

gde su  $M, N$  pozitivno definitne matrice odgovarajućih dimenzija, a  $A^\sharp = N^{-1} A^* M$ .

Za datu kvadratnu matricu  $A \in \mathbb{C}^{n \times n}(x)$  Drazinov inverz i grupni inverz su dati sledećim izrazima (videti [3, 105]):

$$A^D = A_{\mathcal{R}(A^k), \mathcal{N}(A^k)}^{(2)}, \quad A^\# = A_{\mathcal{R}(A), \mathcal{N}(A)}^{(2)}, \quad (1.1.2)$$

pri čemu je  $k = \text{ind}(A)$ .

Ako je  $L$  podprostor od  $\mathbb{C}^n$  i ako je  $A$   $L$ -positivna semi-definitna matrica koja zadovoljava uslove  $AL \oplus L^\perp = \mathbb{C}^n$ ,  $S = \mathcal{R}(P_L A)$ , tada važe sledeći identiteti, [11, 105]:

$$A_{(L)}^{(-1)} = A_{L, L^\perp}^{(2)}, \quad A_{(L)}^{(\dagger)} = A_{S, S^\perp}^{(2)}. \quad (1.1.3)$$

Efektivni metod podele za izračunavanje težinskog Moore-Penrose-ovog inverza je izrađen u [61].

U ovoj disertaciji izučavani su direktni metodi za izračunavanje uopštenih inverza matrica. Primenom direktnih metoda se svaki element uopštenog inverza izračunava eksplicitno i bez iterativnih numeričkih poboljšanja. Ovo su neke od poznatih grupa metoda reprezentacije uopštenih inverza:

1. izračunavanje uopštenih inverza pomoću potpune rang faktorizacije;
2. metodi bazirani na matričnim faktorizacijama;
3. metodi koji proističu iz blokovskih dekompozicija;
4. metodi pregrađivanja (eng. *partitioning method*);
5. determinantska reprezentacija;
6. metodi zasnovani na uopštenju Frameovog rezultata;
7. metodi zasnovani na razbijanju matrica (eng. *matrix splitting*).

## 1.2 Predstavljanje uopštenih inverza faktorizacijom potpunog ranga

Opšte je poznat sledeći rezultat da svaka matrica poseduje dekompoziciju potpunog ranga.

**Lema 1.2.1.** *Svaka matrica  $A \in \mathbb{C}_r^{m \times n}$  može se predstaviti u obliku proizvoda  $A = PQ$  dve matrice potpunog ranga  $P \in \mathbb{C}_r^{m \times r}$  i  $Q \in \mathbb{C}_r^{r \times n}$ .*

Ovakva dekompozicija nije jedinstvena. Međutim, ako je jedna od matrica  $P$  ili  $Q$  zadata, tada je druga matrica jednoznačno određena. Potpuna rang faktorizacija se može upotrebiti za izračunavanje različitih klasa uopštenih inverza. Na potpunoj rang faktorizaciji su bazirane veoma korisne faktorizacije uopštenih inverza

Ovde navodimo neke rezultate o uopštenom predstavljanju generalisanih inverza, u smislu potpune rang faktorizacije i adekvatno odabrane matrice. U sledećoj lemi pokazaćemo ekvivalentnost između dva predstavljanja  $\{1,2\}$ -inverza, koja je dokazana u [64].

**Lema 1.2.2.** Neka je  $A = PQ$  potpuna rang faktorizacija od  $A \in \mathbb{C}_r^{m \times n}$ . Neka su  $U, V$  matrice koje pripadaju skupovima matrica tipa  $m \times m$  i  $n \times n$  respektivno. Takođe, neka matrice  $W_1$  i  $W_2$  pripadaju skupovima matrica tipa  $n \times r$  i  $r \times m$  respektivno. Onda skupovi

$$S_1 = \{VQ^*(P^*UAVQ^*)^{-1}P^*U : U \in \mathbb{C}^{m \times m}, V \in \mathbb{C}^{n \times n}, \text{rang}(P^*UAVQ^*) = r\}$$

$$S_2 = \{W_1(W_2AW_1)^{-1}W_2 : W_1 \in \mathbb{C}^{n \times r}, W_2 \in \mathbb{C}^{r \times n}, \text{rang}(W_2AW_1) = r\}$$

zadovoljavaju  $S_1 = S_2 = A\{1, 2\}$ .

**Dokaz.** Koristeći poznat rezultat [64], lako je zaključiti da  $X \in \{1, 2\}$  ako i samo ako se može predstaviti u obliku

$$X = VQ^*(QVQ^*)^{-1}(P^*UP)^{-1}P^*U = VQ^*(P^*UAVQ^*)^{-1}P^*U,$$

$$U \in \mathbb{C}^{m \times m}, V \in \mathbb{C}^{n \times n}, \text{rang}(P^*UAVQ^*) = r.$$

Ovo povlači  $S_1 = A\{1, 2\}$ .

Na dalje, da bismo kompletirali dokaz, pokažimo da je  $S_1 = S_2$ . Prepostavimo da je  $X \in S_1$  proizvoljno. Onda postoji matrica  $U \in \mathbb{C}^{m \times m}$  i  $V \in \mathbb{C}^{n \times n}$  takve da je  $\text{rang}(P^*UAVQ^*) = r$  i  $X$  se može predstaviti u obliku  $X = VQ^*(P^*UAVQ^*)^{-1}P^*U$ . Nakon zamene  $VQ^* = W_1 \in \mathbb{C}^{n \times r}$ ,  $P^*U = W_2 \in \mathbb{C}^{r \times m}$ , direktno dobijamo  $X \in S_2$ . Sa druge strane, prepostavimo da je  $X \in S_2$ . Onda postoji  $W_1 \in \mathbb{C}^{n \times r}$  i  $W_2 \in \mathbb{C}^{r \times m}$ , takve da je  $\text{rang}(W_2AW_1) = r$  i  $X = W_1(W_2AW_1)^{-1}W_2$ . Matrične jednačine  $W_1 = VQ^*$  i  $W_2 = P^*U$  su konzistentne. Zaista, jednačina  $W_1 = VQ^*$  je konzistentna ako i samo ako postoji  $Q^{*(1)}$  takva da je  $W_1Q^{*(1)}Q^* = W_1$ . Na primer, možemo uzeti proizvoljan levi inverz od  $Q^*$ . Konzistentnost matrične jednakosti  $W_2 = P^*U$  se dokazuje na sličan način. Dakle, za neke matrice  $U \in \mathbb{C}^{m \times m}$  i  $V \in \mathbb{C}^{n \times n}$ , proizvoljno izabrana matrica  $X \in S_2$  može se napisati u obliku  $X = VQ^*(P^*UAVQ^*)^{-1}P^*U$ ,  $\text{rang}(P^*UAVQ^*) = r$ , što povlači  $X \in S_1$ .  $\square$

U sledećoj teoremi iz [82] uvedeno je uopšteno predstavljanje klase  $\{2\}$ -inverza.

**Teorema 1.2.3.** Skup  $\{2\}$ -inverza date matrice  $A \in \mathbb{C}_r^{m \times n}$  određen je sledećom jednakošću:

$$A\{2\} = \{W_1(W_2AW_1)^{-1}W_2, W_1 \in \mathbb{C}^{n \times t}, W_2 \in \mathbb{C}^{t \times m}, \text{rang}(W_2AW_1) = t, t = 1, \dots, r\}$$

**Dokaz.** Neka je  $X$  proizvoljan  $\{2\}$ -inverz od  $A$ . Koristeći Teoremu 3.4.1 iz [64], može se predstaviti u uopštenom obliku  $X = C(DAC)^{(1,2)}D$ , gde su  $C \in \mathbb{C}^{n \times p}$  i  $D \in \mathbb{C}^{q \times m}$  proizvoljni. Takođe, poznato je da  $\text{rang}(X) = \text{rang}(DAC)$  [64] [Teorema 3.4.1]. Prepostavimo da je  $\text{rang}(DAC) = t \leq \min\{p, q\} \leq r$ . Primenom Leme 1.1.2 dobijamo uopšteno predstavljanje za  $X$ :

$$X = CJ(HDACJ)^{-1}HD, J \in \mathbb{C}^{p \times t}, H \in \mathbb{C}^{t \times q}, \text{rang}(HDACJ) = t.$$

Nakon zamene  $CJ = W_1, HD = W_2$ , lako je primetiti da je

$$X \in \{W_1(W_2AW_1)^{-1}W_2, W_1 \in \mathbb{C}^{n \times t}, W_2 \in \mathbb{C}^{t \times m}, \text{rang}(W_2AW_1) = t, t = 1, \dots, r\}.$$

Obrnuto, za proizvoljno  $X$  zadato sa

$$X \in \{W_1(W_2AW_1)^{-1}W_2, W_1 \in \mathbb{C}^{n \times t}, W_2 \in \mathbb{C}^{t \times m}, \text{rang}(W_2AW_1) = t, t = 1, \dots, r\}.$$

lako je pokazati da zadovoljava jednakost  $XAX = X$ .  $\square$

U sledećoj teoremi iz [82] uvedeno je uopšteno predstavljanje Drazinovog inverza kvadratne matrice  $A$ , u smislu potpune rang faktorizacije invarijantnog stepena  $A^l$ ,  $l \geq \text{ind}(A)$  matrice  $A$ .

**Teorema 1.2.4.** *Neka je  $A \in \mathbb{C}_r^{n \times n}$ ,  $k = \text{ind}(A)$  i  $l \geq k$  je proizvoljno celobrojno. Ako je  $A^l = PQ$  potpuna rang faktorizacija matrice  $A^l$ , onda Drazinov inverz od  $A$  ima sledeće predstavljanje:*

$$A^D = P(QAP)^{-1}Q.$$

**Dokaz.** Neka je  $T \in \mathbb{C}^{n \times n}$  nesingularna matrica takva da je

$$A = T \begin{bmatrix} C & \mathbb{O} \\ \mathbb{O} & N \end{bmatrix} T^{-1}$$

jezgro-nilpotent razlaganje od  $A$ , gde je  $C$  nesingularna i  $N$  nilpotentna matrica indeksa  $k$ . Lako je primetiti da je

$$A^l = T \begin{bmatrix} C^l & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} T^{-1}$$

za svako  $l \geq \text{ind}(A)$ . Ovo povlači sledeće potpuno rang razlaganje  $A^l = PQ$ :

$$P = T \begin{bmatrix} C^l \\ \mathbb{O} \end{bmatrix}, Q = [ I_r, \mathbb{O} ] T^{-1}.$$

Koristeći ovu činjenicu i poznati kanonički oblik predstavljanja za  $A^D$  [6], dobijamo

$$A^D = T \begin{bmatrix} C^{-1} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} T^{-1} = T \begin{bmatrix} C^{-1} \\ \mathbb{O} \end{bmatrix} [ I_r, \mathbb{O} ] T^{-1} = PC^{-(l+1)}Q.$$

Dokaz kompletiramo koristeći

$$C^{l+1} = [ C, \mathbb{O} ] \begin{bmatrix} C^l \\ \mathbb{O} \end{bmatrix} = [ I_r, \mathbb{O} ] T^{-1} T \begin{bmatrix} C & \mathbb{O} \\ \mathbb{O} & N \end{bmatrix} T^{-1} T \begin{bmatrix} C^l \\ \mathbb{O} \end{bmatrix} = QAP.$$

□

**Napomena 1.2.1.** Predstavljanje Drazinovog inverza uvedeno u Teoremi 1.2.4 je prirodno uopštenje sledeće karakterizacije inverza grupe, uvedeno u [14]:

Ako je  $A = PQ$  potpuna rang faktorizacija od  $A$ , onda postoji  $A^\#$  ako i samo ako je matrica  $QP$  invertibilna, i važi

$$A^\# = P(QP)^{-2}Q = P(QAP)^{-1}Q.$$

Faktorizacija potpunog ranga matrice se može odrediti na više načina. Neki metodi potpune rang faktorizacije matrica mogu se uzeti različite matrične dekompozicije, kao na primer *LU* dekompozicija, *QR* faktorizacija, *SVD Singularno Vrednosna Dekompozicija*, i druge. Potpuna rang faktorizacija se može dobiti i koristeći bilo koju od blokovskih reprezentacija matrica. Ovi metodi će biti izučavani u drugom poglavljju.

### 1.3 LU dekompozicija

U ovoj disertaciji posmatraćemo kako uopštenu LU dekompoziciju, tako i njene varijante: Cholesky, odnosno LL\* dekompoziciju i LDL\* dekompoziciju. Sledеće tvrđenje je dobro poznato iz literature.

**Stav 1.3.1.** *Svaka matrica  $A \in \mathbb{C}_r^{m \times n}$  se može transformisati u oblik  $A = LU$ , gde je  $L$  donja trougaona matrica sa jedinicama na glavnoj dijagonali, a  $U$  je gornja trougaona matrica.*

**Dokaz.** LU faktorizacija se može izvesti iz Gaussovog metoda eliminacije sa potpunim izborom glavnog elementa. Matrica  $A = A_0 \in \mathbb{C}_r^{m \times n}$  se transformiše redom u matrice  $A_1, \dots, A_r$ . Matrica  $A_k$ ,  $0 \leq k \leq r$  jeste  $m \times n$  matrica oblika

$$A_k = \begin{pmatrix} a_{ij}^{(k)} \end{pmatrix} = \begin{bmatrix} U_k & V_k \\ \mathbb{O} & W_k \end{bmatrix},$$

gde je  $U_k$  trougaona matrica dimenzija  $k \times k$ . Označimo  $(i, j)$ -element u bloku  $W_k$  sa  $w_{ij}$ . Ako je  $w_{\alpha\beta}$  najveći po modulu element u bloku  $W_k$  vrši se zamena  $\alpha$  vrste sa  $k+1$  vrstom i  $\beta$  vrste sa  $k+1$  kolonom matrice  $A_k$ . Izvršenjem analognih zameni nad vrstama jedinične  $m \times m$  matrice dobija se permutaciona matrica  $E_k = P(k+1, \alpha)$ , dok se analognom zamenom kolona jedinične  $n \times n$  matrice dobija permutaciona matrica  $F_k = P(k+1, \beta)$ . Uz pretpostavku  $a_{ij}^{(1)} = a_{ij}$ ,  $1 \leq i \leq m$ ,  $1 \leq j \leq n$ , Gaussova eliminacija matrice  $A_k$  jeste sledeća transformacija:

$$v_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - v_{ik} a_{kj}^{(k)}, \quad \begin{pmatrix} i = k+1, \dots, m \\ j = k+1, \dots, n \end{pmatrix}, \quad k = 1, \dots, r.$$

Kao rezultat ove transformacije dobija se matrica

$$A_{k+1} = \begin{bmatrix} U_{k+1} & V_{k+1} \\ \mathbb{O} & W_{k+1} \end{bmatrix}.$$

Posle  $r$  koraka dobija se matrica

$$A_r = \begin{bmatrix} U_r & V_r \\ \mathbb{O} & W_r \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1r}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2r}^{(2)} & \dots & a_{2n}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{rr}^{(r)} & \dots & a_{rn}^{(r)} \end{bmatrix}$$

Za matricu  $U$  se može uzeti prvih  $r$  vrsta matrice  $A_r$ .

Matrica  $L$  je definisana sa

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ v_{21} & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ v_{r1} & v_{r1} & \dots & v_{r,r-1} & 1 \\ \dots & \dots & \dots & \dots & \dots \\ v_{m1} & v_{m2} & \dots & v_{m,r-1} & v_{m,r} \end{bmatrix}. \quad \square$$

Na kraju transformacije dobija se:

$$EAF = LU \iff A = E^* LUF^*,$$

Gde su  $E$  i  $F$  permutacione matrice definisane kao proizvod elementarnih matrica:

$$E = E_1 \cdots E_r, \quad F = F_1 \cdots F_r.$$

**Stav 1.3.2.** *Ako  $A = LU$  predstavlja LU faktorizaciju matrice  $A$ , tada je*

$$A^\dagger = U^\dagger L^\dagger = U^*(UU^*)^{-1}(L^*L)^{-1}L^*.$$

**Primer 1.3.1.** Za matricu  $A = \begin{bmatrix} 1 & 3 & 6 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & -2 & 0 & -2 \\ -1 & 0 & -1 & -3 \end{bmatrix}$  dobija se sledeća LU faktorizacija:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{4}{9} & 1 & 0 & 0 \\ -\frac{2}{9} & 4 & 1 & 0 \\ -\frac{1}{3} & 3 & \frac{1}{4} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 3 & 6 & 9 \\ \frac{5}{9} & \frac{2}{3} & \frac{1}{3} & 0 \\ -2 & -4 & 0 & 0 \\ -\frac{11}{6} & 0 & 0 & 0 \end{bmatrix}.$$

LU faktorizacija je u programskom paketu MATHEMATICA implementirana pomoću standardne funkcije `LUDecomposition`.

Cholesky faktorizacija podrazumeva dekompoziciju simetrične pozitivno definitne matrice  $A$  oblika  $A = LL^*$ , gde je  $L$  donja trougaona matrica sa strogo pozitivnim dijagonalnim elementima. Očigledno, elementi matrice  $L$  mogu sadržati kvadratne korene, što izaziva poteškoće u daljem simboličkom izračunavanju. Iz ovog razloga se često koristi "bez-korena" Cholesky dekompozicija u obliku  $A = LDL^*$ . Proizvoljna racionalna Hermitska matrica  $A$  se može predstaviti u obliku proizvoda matrica  $A = LDL^*$ , gde je  $L$  donja trougaona, a  $D$  dijagonalna matrica. Ova forma premošćava upotrebu elemenata koji sadrže kvadratne korene. Jednostavno je pokazati da, u slučaju pozitivno definitne matrice  $A$ , dijagonalni elementi u  $D$  moraju biti nenegativni.

Neka je data polinomijalna Hermitska matrica  $A(x) \in \mathbb{C}(x)^{n \times n}$  sa elementima po promenljivoj  $x$ :

$$A(x) = A_0 + A_1x + \cdots + A_qx^q = \sum_{i=0}^q A_i x^i \tag{1.3.1}$$

pri čemu su  $A_i$ ,  $i = 0, \dots, q$  konstantne matrice dimenzija  $n \times n$ . Alternativna forma Cholesky faktorizacije simetrične polinomijalne matrice  $A(x)$  generiše dodatnu dijagonalnu matricu  $D(x)$ , u cilju izbegavanja elemenata sa kvadratnim korenima (videti [25]). Prema tome,  $LDL^*$  faktorizacija matrice  $A(x)$  je bez-korena Cholesky dekompozicija sledećeg oblika:

$$A(x) = L(x)D(x)L^*(x),$$

gde je  $L(x)$  donja trougaona, a  $D(x)$  dijagonalna matrica,

$$L(x) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21}(x) & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1}(x) & l_{n2}(x) & \cdots & 1 \end{bmatrix}, \quad D(x) = \begin{bmatrix} d_1(x) & 0 & \cdots & 0 \\ 0 & d_2(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n(x) \end{bmatrix} \quad (1.3.2)$$

pri čemu su  $l_{ij}$ ,  $d_j$ ,  $1 \leq j < i \leq n$  racionalne funkcije. Pritom,  $L^*(x)$  označava konjugovano transponovanu matricu  $L(x)$ .

Zamena  $LL^*$  dekompozicije  $LDL^*$  faktorizacijom radi izbegavanja izračunavanja korenih elemenata je dobro poznata tehnika numeričke analize, opisana na primer od strane Goluba i Van Loana u [25]. Ova tehnika je od esencijalne važnosti kod polinomijalnih i racionalnih elemenata matrica  $L$  i  $D$ . Ukoliko se radi sa npr.  $\sqrt{\sum_{i=0}^q A_i x^i}$  simbolički, gde su  $A_i$ ,  $i = 0, \dots, q$  konstantne  $n \times n$  matrice, to može biti veoma teško. Zato je naša motivacija modifikovanje algoritama sa direktnom Cholesky faktorizacijom polinomijalne matrice  $A(x)$ .

Prema tome, navedene dekompozicije potpunog ranga se mogu koristiti za simboličko izračunavanje uopštenih inverza koristeći brojne tehnike, uvedene u radovima kao što su [87, 70]. Implementacija ovih algoritama rešava problem simboličkih izračunavanja uopštenih inverza u proceduralnim programskim jezicima.

## 1.4 Struktura i organizacija rada

Doktorska disertacija podeljena je u pet glava, svaka glava je podeljena na nekoliko poglavlja a poglavlja se sastoje od odeljaka.

- U prvoj glavi, koja je podeljena u četiri poglavlja, predstavljeni su uvodni pojmovi. Prva tri poglavlja sadrže definicije i opšta svojstva pojmoveva iz glavne oblasti ove doktorske disertacije: dekompozicija potpunog ranga matrica,  $LU$  dekompozicija i izračunavanje uopštenih inverza matrica. U poslednjem poglavlju dat je opšti prikaz organizacije same doktorske disertacije.
- Rezultati sadržani u drugoj glavi odnose se na efikasno izračunavanje dekompozicija matrica i uopštene inverze matrica i njihovo efikasno računanje.
  - U prvom poglavlju ove glave dat je prikaz osnovnih definicija i pojmoveva generalisanih inverza proizvoljnih matrica sa posebnim osvrtom na izracunavanje  $\{i, j, \dots, k\}$  inverza i Moore-Penrose-ovog inverza.
  - Drugo poglavlje prezentuje rezultate za  $LDL^*$  dekompoziciju potpunog ranga polinomijalne matrice, kao i numeričke primere i implementacione detalje.
  - Treće poglavlje sadrži nove metode za izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice zasnovane na metodu iz drugog poglavlja, kao i neke numeričke primere.

- Četvrto poglavlje prezentuje rezultate dobijene o  $LDL^*$  dekompoziciji potpunog ranga racionalnih matrica.
  - Peto poglavlje sadrži nove metode za izračunavanje Moore-Penrose-ovog inverza racionalne matrice zasnovane na metodu iz četvrtog poglavlja, kao i neke numeričke primere i implementacione detalje.
  - Šesto poglavlje prezentuje nove metode za izracunavanje  $A_{T;S}^{(2)}$  inverza  $LDL^*$  dekompozicijom koristeći metode iz drugog i četvrtog poglavlja.
  - Sedmo poglavlje prezentuje nove reprezentacije potpunog ranga uopštenih inverza zasnovane na QR dekompoziciji.
  - Osmo poglavlje podrazumeva simboličko izračunavanje  $A_{T;S}^{(2)}$  inverza  $QDR$  faktorizacijom, kao i numeričke primere i implementacione detalje.
  - Deveto poglavlje daje rezultate vezane za uopštene inverze blokovskih matrica i blokovske reprezentacije generalisanih inverza.
  - Deseto poglavlje sadrži blokovsku  $LDL^*$  dekompoziciju potpunog ranga i izračunavanje inverza  $n \times n$  blokovskih matrica.
  - Jedanaesto poglavlje sadrži opis i implementaciju Greville-ovog metoda podele.
- Treća glava ovog doktorata posvećena je opštim metodama višekriterijumske optimizacije i njihovim implementacijama.
    - U prvom poglavlju ove glave definisani su osnovni pojmovi vezani za više-kriterijumsku optimizaciju i Pareto optimalnost.
    - U drugom poglavlju dat je opšti pregled prednosti simboličke implementacije u višekriterijumskoj optimizaciji.
    - Treće poglavlje sadrži metod i algoritam za ispitivanje Pareto optimalnosti problema višekriterijumske optimizacije.
    - Četvrto poglavlje je posvećeno poznatom metodu težinskih koeficijenata, gde su navedeni novi teorijski rezultati i načini implementacije.
    - U petom poglavlju prikazan je i detaljno proučen leksikografski metod više-kriterijumske optimizacije, i navedena modifikacija ovog metoda, kao i numerički primeri i rezultati testiranja.
    - Šesto poglavlje sadrži analizu i implementaciju drugih metoda višekriterijumske optimizacije: relaksirane leksikografske metode, metoda  $\epsilon$ -ograničenja i metoda rastojanja.
  - U četvrtoj glavi izveden je zaključak i izvršena sistematizacija svih predstavljenih rezultata; takođe, dato je nekoliko predloga za dalja izučavanja i na taj način su prikazane smernice za budući naučno-istraživački rad.

## 2. Simbolička matrična izračunavanja

Simboličko izračunavanje uopštenih inverza se najčešće izvodi preko dekompozicije potpunog ranga odgovarajuće matrice. U ovoj glavi su korišćene prednosti  $LDL^*$  i  $QDR$  dekompozicije u odnosu na njihove klasične varijante (videti [77, 78, 74, 96]). Pritom su korišćene modifikacije u smislu da su sve matrice iz dekompozicija potpunog ranga.

Metod iz [91], u kome je primenjena  $LU$  faktorizacija za izračunavanje pseudo-inverza matrica, proširen je u radu [77]. Očigledan je cilj napraviti algoritam za simboličko izračunavanje uopštenih inverza korišćenjem  $LDL^*$  dekompozicije racionalne matrice  $A \in \mathbf{C}(x)^{m \times n}$ . Motivacija je proistekla iz prednosti korišćenja bez-korene dekompozicije (dekompozicije koja ne sadrži korene) u algoritmima za polinomijalne i racionalne matrice. Na ovaj način se mogu izbeći poteškoće sa simboličkim izračunavanjima polinomijanih matrica i njihovih inverza, koje nastaju kroz pojavljivanje elemenata sa korenima. Takođe, pokazano je da  $LDL^*$  dekompozicija povećava efikasnost izračunavanja uopštenih inverza.

### 2.1 Izračunavanje $\{i, j, k\}$ inverza i uopštenih inverza racionalnih matrica

Za datu kompleksnu Hermitsku matricu  $A$ , sledeće rekurzivne relacije važe za elemente matrica  $D$  i  $L$ , dobijenih iz  $LDL^*$  dekompozicije matrice  $A$ :

$$d_{jj} = a_{jj} - \sum_{k=1}^{j-1} l_{jk} l_{jk}^* d_{kk}, \quad l_{ij} = \frac{1}{d_j} (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}^* d_{kk}), \text{ za } i > j. \quad (2.1.1)$$

Ideja da je ova izračunavanja dovoljno izvoditi za  $j = \overline{1, r}$ , gde je  $r = \text{rang}(A)$ , izneta je u radu [77]. Ovim se dobija faktorizacija potpunog ranga matrice  $A$ , pri čemu  $L$  ne sadrži nula-kolone, a matrica  $D$  ne sadrži nula-vrstе i nula-kolone. Dakle, za datu matricu  $A \in \mathbf{C}_r^{m \times n} = \{X \in \mathbf{C}^{m \times n} \mid \text{rang}(X) = r\}$ , njena dekompozicija potpunog ranga bez kvadratnih korena je data sa  $A = LDL^*$ , gde je  $L \in \mathbf{C}^{m \times r}$  a  $D \in \mathbf{C}^{r \times r}$  je odgovarajuća dijagonalna matrica.

Representacije  $\{1, 2, 3\}$  i  $\{1, 2, 4\}$ -inverza, uvedene u radu [3], modifikovane su sledećom lemom iz rada [91]. Takođe su ove reprezentacije, poznate za kompleksne matrice, proširene na skup racionalnih matrica jedne nepoznate.

**Lema 2.1.1.** Neka je data matrica  $A \in \mathbf{C}(x)_r^{m \times n}$  i neka su  $m_1, n_1 \geq r$  fiksirani prirodni brojevi. Tada važe sledeća tvrdjenja za skupove  $A\{1, 2, 4\}$ ,  $A\{1, 2, 3\}$  i za Moore-Penrose-ov inverz matrice  $A$ :

- (a)  $A\{1, 2, 4\} = \{(YA)^\dagger Y \mid Y \in \mathbf{C}(x)^{n_1 \times m}, YA \in \mathbf{C}(x)_r^{n_1 \times n}\}$ .
- (b)  $A\{1, 2, 3\} = \{Z(AZ)^\dagger \mid Z \in \mathbf{C}(x)^{n \times m_1}, AZ \in \mathbf{C}(x)_r^{m \times m_1}\}$ .
- (c)  $A^\dagger = (A^* A)^\dagger A^* = A^* (A A^*)^\dagger$ .

Sada je moguće izvesti sledećih nekoliko tvrdjenja za izračunavanje  $\{1, 2, 3\}$  i  $\{1, 2, 4\}$  inverza date matrice  $A \in \mathbf{C}(x)_r^{m \times n}$ .

**Teorema 2.1.2.** [77] Neka je data racionalna matrica  $A \in \mathbf{C}(x)_r^{m \times n}$  i proizvoljna  $m \times n_1$  racionalna matrica  $R$ , gde je  $n_1 \geq r$ . Neka je  $LDL^*$  bez-korena Cholesky faktorizacija potpunog ranga matrice  $G = (R^* A)^*(R^* A)$ , pri čemu su  $L \in \mathbf{C}(x)^{n \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$ . Tada važe sledeća tvrdjenja:

$$A\{1, 2, 4\} = \{L(L^* GL)^{-1} L^* (R^* A)^* R^* \mid R \in \mathbf{C}(x)^{m \times n_1}, R^* A \in \mathbf{C}(x)_r^{n_1 \times n}\}. \quad (2.1.2)$$

**Dokaz.** Posmatrajmo sledeći izraz za izračunavanje Moore-Penrose-ovog inverza proizvoda matrica  $A$  i  $B$  (tj.  $(AB)^\dagger$ ), koji je uveden u [65]:

$$(AB)^\dagger = B^* (A^* A B B^*)^\dagger A^*. \quad (2.1.3)$$

Primenom ove jednačine za slučaj  $A = R^* A$ ,  $B = I$ , Moore-Penrose-ov inverz  $(R^* A)^\dagger$  se može dobiti kao

$$(R^* A)^\dagger = ((R^* A)^* (R^* A))^\dagger (R^* A)^*. \quad (2.1.4)$$

Koristićemo  $LDL^*$  dekompoziciju matrice  $(R^* A)^* (R^* A) = LDL^*$ . Zamenom ovih vrednosti u jednačini (2.1.4), dobijamo

$$(R^* A)^\dagger = (LDL^*)^\dagger (R^* A)^*. \quad (2.1.5)$$

Dalje, zamenom  $A = L$ ,  $B = DL^*$  u jednačini (2.1.3), imamo

$$(LDL^*)^\dagger = (DL^*)^* (L^* LDL^* (DL^*)^*)^{-1} L^* = LD^* (L^* LDL^* LD^*)^{-1} L^*.$$

Tada množenjem  $(R^* A)^\dagger$  sa  $R^*$  s desne strane, saglasno sa poslednjom jednačinom i jednačinom (2.1.5) dobijamo

$$(R^* A)^\dagger R^* = LD^* (L^* LDL^* LD^*)^{-1} L^* (R^* A)^* R^* = L (L^* LDL^* L)^{-1} L^* (R^* A)^* R^*.$$

Ovde priemtimo da matrica  $L$  ne sadrži nula kolone,  $L^*$  ne sadrži nula vrste, a  $D$  je bez nula kolona i nula vrsta. Pritom je matrica  $L^* LDL^* L$  kvadratna, što povlači da je invertibilna. Sada dokaz našeg tvrdjenja sledi na osnovu Leme 2.1.1, dela pod (a).  $\square$

**Posledica 2.1.3.** [77] Neka je data racionalna matrica  $A \in \mathbf{R}(x)_r^{m \times n}$  i proizvoljna  $m \times n_1$  realna matrica  $R$ , gde je  $n_1 \geq r$ . Prepostavimo da je  $LDL^T$  bez-korena Cholesky faktorizacija potpunog ranga matrice  $G = (R^T A)^T (R^T A)$ , pri čemu je  $L \in \mathbf{R}(x)^{n \times r}$  i  $D \in \mathbf{R}(x)^{r \times r}$ . Tada su zadovoljena sledeća tvrdjenja:

$$A\{1, 2, 4\} = \{L(L^T GL)^{-1} L^T (R^T A)^T R^T \mid R \in \mathbf{R}^{m \times n_1}, R^T A \in \mathbf{R}(x)_r^{n_1 \times n}\}. \quad (2.1.6)$$

**Teorema 2.1.4.** [77] Neka je data racionalna matrica  $A \in \mathbf{C}(x)_r^{m \times n}$  i proizvoljna  $m_1 \times n$  racionalna matrica  $T$ , pri čemu je  $m_1 \geq r$ . Prepostavimo da je  $LDL^*$  bez-korena Cholesky faktorizacija potpunog ranga matrice  $G = (AT^*)(AT^*)^*$ , gde je  $L \in \mathbf{C}(x)^{m \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$ . Tada su sledeća tvrđenja validna:

$$A\{1, 2, 3\} = \left\{ T^*(AT^*)^* L (L^* GL)^{-1} L^* \mid T \in \mathbf{C}(x)^{m_1 \times n}, AT^* \in \mathbf{C}(x)_r^{m \times m_1} \right\}. \quad (2.1.7)$$

**Dokaz.** Dokaz je sličan dokazu Teoreme 2.1.2, sa razlikom u primeni tvrđenja pod (b) Lemme 2.1.1. Uzimanjem  $A = I$  i  $B = AT^*$  u jednačini (2.1.3), dobijamo

$$(AT^*)^\dagger = TA^*(AT^*(AT^*)^*)^\dagger. \quad (2.1.8)$$

Uzimajući da je  $LDL^*$  dekompozicija potpunog ranga matrice  $(AT^*)(AT^*)^*$ , imamo da je

$$(AT^*)^\dagger = TA^*(LDL^*)^\dagger. \quad (2.1.9)$$

Sada, dokaz sledi na osnovu tvrđenja (b) Leme 2.1.1, analogno kao u dokazu Teoreme 2.1.2. Primetimo da se u tom slučaju,  $m$  i  $m_1$  pojavljuju umesto  $n$  i  $n_1$ , respektivno.  $\square$

**Posledica 2.1.5.** [77] Neka je data racionalna matrica  $A \in \mathbf{R}(x)_r^{m \times n}$  i proizvoljna  $m_1 \times n$  realna matrica  $T$ , gde je  $m_1 \geq r$ . Prepostavimo da je  $LDL^T$  bez-korena Cholesky faktorizacija potpunog ranga matrice  $G = (AT^T)(AT^T)^T$ , pri čemu su  $L \in \mathbf{R}(x)^{m \times r}$  i  $D \in \mathbf{R}(x)^{r \times r}$ . Tada važe sledeća tvrđenja:

$$A\{1, 2, 3\} = \left\{ T^T (AT^T)^T L (L^T GL)^{-1} L^T \mid T \in \mathbf{R}^{m_1 \times n}, AT^T \in \mathbf{R}(x)_r^{m \times m_1} \right\}. \quad (2.1.10)$$

**Teorema 2.1.6.** [77] Neka je data racionalna matrica  $A \in \mathbf{C}(x)_r^{m \times n}$ . Ako je  $LDL^*$  bez-korena Cholesky faktorizacija matrice  $G = (A^*A)^*(A^*A)$ , gde je  $L \in \mathbf{C}(x)^{m \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$ , tada važi:

$$A^\dagger = L(L^* GL)^{-1} L^* (A^* A)^* A^*. \quad (2.1.11)$$

Ako je  $LDL^*$  bez-korena Cholesky faktorizacija matrice  $H = (AA^*)(AA^*)^*$ ,  $L \in \mathbf{C}(x)^{m \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$ , tada je zadovoljeno:

$$A^\dagger = A^*(AA^*)^* L (L^* HL)^{-1} L^*. \quad (2.1.12)$$

**Dokaz.** Dokaz sledi na osnovu Leme 2.1.1, tvrđenja pod (c).  $\square$

**Posledica 2.1.7.** [77] Neka je data racionalna matrica  $A \in \mathbf{R}(x)^{m \times n}$ . Prepostavimo da je  $LDL^T$  bez-korena Cholesky faktorizacija potpunog ranga matrice  $G = (A^T A)^T (A^T A)$ , gde je  $L \in \mathbf{R}(x)^{m \times r}$  i  $D \in \mathbf{R}(x)^{r \times r}$ . Sledeće tvrđenje je zadovoljeno:

$$A^\dagger = L(L^T GL)^{-1} L^T (A^T A)^T A^T. \quad (2.1.13)$$

Ako je  $LDL^T$  bez-korena Cholesky faktorizacija matrice  $H = (AA^T)(AA^T)^T$ , gde je  $L \in \mathbf{R}(x)^{m \times r}$  i  $D \in \mathbf{R}(x)^{r \times r}$ , tada važi:

$$A^\dagger = A^T (AA^T)^T L (L^T HL)^{-1} L^T. \quad (2.1.14)$$

Na osnovu prethodno dokazanih tvrđenja, uveden je sledeći Algoritam 2.1 nazvan LDLGInverse za izračunavanje skupova  $A\{1, 2, 4\}$  i  $A\{1, 2, 3\}$  za datu racionalnu  $m \times n$  matricu  $A$ , kao i za izračunavanje Moore-Penrose-ovog inverza matrice  $A$ .

---

**Algoritam 2.1** Izračunavanje  $\{i, j, k\}$  i Moore-Penrose-ovog inverza racionalne matrice - Algoritam LDLGInverse

---

**Ulaz:** Racionalna matrica  $A \in \mathbf{C}(x)_r^{m \times n}$ .

- 1: **if**  $\ddagger = \{1, 2, 4\}$  **then**
  - 2: Nasumično generisati matricu  $R \in \mathbf{C}^{m \times n_1}$ ,  $n_1 \geq r$
  - 3:  $G := (R^* A)^*(R^* A)$
  - 4: **else if**  $\ddagger = \{1, 2, 3\}$  **then**
  - 5: Nasumično generisati matricu  $T \in \mathbf{C}^{m_1 \times n}$ ,  $m_1 \geq r$
  - 6:  $G := (AT^*)(AT^*)^*$
  - 7: **else**
  - 8:  $G := (A^* A)^*(A^* A)$  {ovaj korak je za MP-inverz}
  - 9: **end if**
  - 10: Izračunati  $LDL^*$  faktorizaciju potpunog ranga matrice  $G$ .
  - 11: **if**  $\ddagger = \{1, 2, 4\}$  **then**
  - 12:  $M = L(L^* GL)^{-1}L^*(R^* A)^*R^*$
  - 13: **else if**  $\ddagger = \{1, 2, 3\}$  **then**
  - 14:  $M = T^*(AT^*)^*L(L^* GL)^{-1}L^*$
  - 15: **else**
  - 16:  $M = L(L^* GL)^{-1}L^*(A^* A)^*A^*$
  - 17: **end if**
  - 18: **Rezultat:** Rezultujuća matrica  $M$ .
- 

**Primer 2.1.1.** Posmatrajmo matricu  $A_6$  iz [117], gde je  $a$  neka konstanta. Da bismo izračunali  $\{1, 2, 4\}$  i  $\{1, 2, 3\}$  inverze matrice  $A_6$ , konstantna matrica  $R = T^T$  je nasumično generisana.

$$A_6 = \begin{bmatrix} a+5 & a+3 & a+2 & a+4 & a+3 & a+2 \\ a+3 & a+4 & a+2 & a+3 & a+3 & a+2 \\ a+2 & a+2 & a+2 & a+2 & a+2 & a+1 \\ a+4 & a+3 & a+2 & a+3 & a+3 & a+2 \\ a+3 & a+3 & a+2 & a+3 & a+2 & a+2 \\ a+2 & a+2 & a+1 & a+2 & a+2 & a \end{bmatrix}, \quad R = T^T = \begin{bmatrix} 1 & 5 & 3 & 1 & 2 & -1 \\ 1 & -1 & 4 & 1 & -2 & 1 \\ 3 & 1 & -3 & 1 & 2 & -1 \\ -2 & -1 & 3 & 2 & -2 & 1 \\ 1 & -1 & 3 & -1 & 2 & -1 \\ 3 & -1 & -1 & 6 & -2 & 1 \end{bmatrix}.$$

Rezultati za  $\{1, 2, 4\}$ ,  $\{1, 2, 3\}$  i MP inverze matrice  $A_6$ , dobijeni primenom algoritma LDLGInverse, su sledeći:

$$A_6^{(1,2,4)} = \begin{bmatrix} -3-a & -\frac{754+543a}{55} & \frac{479+433a}{55} & 4+a & 3+a & \frac{3(233+181a)}{55} & -\frac{589+488a}{55} \\ -3-a & -\frac{847+543a}{55} & \frac{57}{5} + \frac{433a}{55} & 3+a & 3+a & \frac{77}{5} + \frac{543a}{55} & -\frac{737+488a}{55} \\ -2-a & -2 - \frac{543a}{55} & \frac{433a}{55} & 2+a & 2+a & 1 + \frac{543a}{55} & -\frac{488a}{55} \\ 4+a & \frac{413+543a}{55} & -\frac{138+433a}{55} & -5-a & -3-a & -\frac{358+543a}{55} & \frac{8(31+61a)}{55} \\ 3+a & \frac{3(382+181a)}{55} & -\frac{871+433a}{55} & -3-a & -4-a & -\frac{1091+543a}{55} & \frac{981+488a}{55} \\ 2+a & \frac{695+543a}{55} & -\frac{530+433a}{55} & -2-a & -2-a & -\frac{695+543a}{55} & \frac{585+488a}{55} \end{bmatrix},$$

$$A_6^{(1,2,3)} = A_6^\dagger = \begin{bmatrix} -3 - a & -\frac{3(3+a)}{4} & -\frac{11+5a}{4} & 4 + a & 3 + a & \frac{5+3a}{4} & \frac{3+a}{4} \\ -3 - a & -\frac{3(2+a)}{4} & -\frac{5(2+a)}{4} & 3 + a & 3 + a & \frac{3(2+a)}{4} & \frac{2+a}{4} \\ -2 - a & -\frac{5+3a}{4} & -\frac{3+5a}{4} & 2 + a & 2 + a & \frac{1+3a}{4} & \frac{3+a}{4} \\ 4 + a & \frac{3(3+a)}{4} & \frac{11+5a}{4} & -5 - a & -3 - a & -\frac{5+3a}{4} & -\frac{3+a}{4} \\ 3 + a & \frac{3(3+a)}{4} & \frac{11+5a}{4} & -3 - a & -4 - a & -\frac{5+3a}{4} & -\frac{3+a}{4} \\ 2 + a & \frac{5+3a}{4} & \frac{7+5a}{4} & -2 - a & -2 - a & -\frac{5+3a}{4} & -\frac{3+a}{4} \end{bmatrix}.$$

Primetimo da su dobijeni  $\{1, 2, 3\}$ -inverz i Moore-Penrose-ov inverz matrice  $A_6$  jednaki u prethodnom primeru.

### 2.1.1 Implementacioni detalji i rezultati testiranja

Na osnovu skraćene iterativne procedure (2.1.1), moguće je izvesti sledeću implementaciju  $LDL^*$  dekompozicije proizvoljne matrice u MATHEMATICA. Sledeća funkcija kao jedini argument podrazumeva matricu zadatu u obliku liste.

```
LDLDecomposition[A_List] :=
Module[{i, j, k, n = MatrixRank[A], m = Length[A], L, D},
L = Table[0, {m}, {n}];
D = Table[0, {n}, {n}];
For[j = 1, j ≤ n, j++,
L[[j, j]] = 1; D[[j, j]] = Simplify[A[[j, j]] - Sum[L[[j, k]]^2 D[[k, k]], {k, 1, j - 1}]];
For[i = j + 1, i ≤ m, i++,
L[[i, j]] = Simplify[1/D[[j, j]] (A[[i, j]] - Sum[L[[i, k]] L[[j, k]] D[[k, k]], {k, 1, j - 1}])];
];
Return[{L, D}];]
```

Na osnovu Algoritma 2.1 imamo sledeći modul kojim se određuje Moore-Penrose-ov inverz proizvoljne matrice  $A$ .

```
LDLGInverse[A_List] :=
Module[{L, D, G, G1, M, Y},
G1 = Transpose[A].A;
G = G1.G1;
{L, D} = LDLDecomposition[G];
M = Simplify[Inverse[Transpose[L].L.D.Transpose[L].L]];
Y = L.M.Transpose[L].G1.Transpose[A];
Return[Simplify[Y]]];
```

**Primer 2.1.2.** Uporedimo različite algoritme za izračunavanje uopštenih inverza sa algoritmom  $LDLGInverse$ . U sledećoj tabeli data su vremena izvršavanja na različitim test matricama dobijena primenom implementiranih algoritama u programskom paketu MATHEMATICA. Matrice za testiranje su uzete iz [117], pri čemu je posmatran parcijalni slučaj  $a = 1$ .

Test matrica	$A_{10}$	$A_{50}$	$A_{100}$	$S_{10}$	$S_{50}$	$S_{100}$	$F_{10}$	$F_{50}$	$F_{100}$
PseudoInverse [113]	1.344	-	-	1.031	-	-	1.094	-	-
Partitioning [87]	0.015	0.484	2.750	0.031	1.187	9.204	0.016	0.485	2.812
Lev.-Faddeev [41]	0.001	2.766	43.781	0.001	2.516	44.375	0.001	2.672	42.844
Courrieu [16]	0.015	0.766	5.844	0.015	0.375	2.297	0.015	0.703	5.782
ModCholesky [91]	0.015	2.218	16.954	0.015	0.687	5.781	0.015	2.328	17.594
LDLGInverse	0.015	1.875	11.672	0.014	0.487	4.553	0.014	1.966	12.643

Tabela 2.1.1. Vremena izvršenja (u sekundama) dobijena na osnovu nekoliko algoritama i LDLGInverse algoritma.

Prva vrsta Tabele 2.1.1 sadrži nazive test matrica iz [117]. Primetimo da su ispitivane tri grupe test matrica. Poslednja vrsta tabele sadrži vremena dobijena primenom LDLGInverse algoritma. Znak '-' označava "veliko" vreme izračunavanja.

**Primer 2.1.3.** Posmatrajmo sada nasumično generisane retko posednute matrice iz skupa  $A \in \mathbf{R}_r^{m \times n}$  različitih dimenzija i gustina, pri čemu je najveća sopstvena vrednost matrice  $A^*A$  data sa  $k_r$ , a najmanja ne-nula sopstvena vrednost matrice  $A^*A$  je jednaka 1 (videti [17]). Neki od dobijenih rezultata su prikazani u Tabeli 2.1.2. Tabela 2.1.3 prikazuje razlike između vremena izvršavanja algoritama za određene matrice iz kolekcije Matrix-Market (videti [53]).

m	128			256			512		
$k_r$	16	256	4096	16	256	4096	16	256	4096
Potpuni rang	4.259	4.539	4.681	30.011	30.248	30.312	227.387	233.908	234.298
Rang-deficientna	3.044	3.754	3.862	25.101	25.241	25.537	185.907	190.369	190.462

Tabela 2.1.2. Prosečna vremena izračunavanja (u sekundama) za nasumično generisane retko posednute matrice.

Matrica	gr_30_30	illc1850	watt_1	well1033	well1850
Veličina	$900 \times 900$	$1850 \times 712$	$1856 \times 1856$	$1033 \times 320$	$1850 \times 712$
Gustina	0.0096	0.0066	0.0033	0.0143	0.0066
Vremena	233.3	122.1	40.4	11.9	127.5

Tabela 2.1.3. Prosečna vremena izračunavanja (u sekundama) za neke test matrice iz kolekcije Matrix-Market.

Primećeno je da su neki vrlo brzi metodi numerički nestabilni ukoliko je matrica (ne singularna ali) loše uslovljena (za pojašnjenje videti [17, 103]). Uvedeni metod za izračunavanje uopštenih inverza u [77] je vrlo efikasan, međutim nije i najefikasniji. Očigledno su rang-deficientne matrice procesuirane brže nego matrice potpunog ranga istih dimenzija, s obzirom da su matrice  $L$  i  $D$  manjih dimenzija, u slučaju rang-deficientnih matrica. Međutim, vremena izračunavanja rapidno rastu povećanjem dimenzija i gustina matrica. U opštem slučaju, izračunavanje inverza racionalne matrice je  $O(n^3)$  problem, a takođe i veoma osetljiv na loše uslovljene matrice.

Dobro je poznato da kondicioni broj za inverziju matrica u odnosu na matričnu normu  $\|\cdot\|$  kvadratne matrice  $A$ , definisan kao  $\kappa(A) = \|A\| \|A^{-1}\|$ , predstavlja meru stabilnosti ili osetljivosti inverzne matrice od numeričkih operacija [25]. Matrice sa kondicionim brojem bliskim 1 se nazivaju dobro-uslovljene, a one sa većim kondicionim brojem od jedan se nazivaju loše-uslovljene. Primetimo da svih pet matrica uzetih iz "MatrixMarket" testiranih u Tabeli 2.1.3 imaju kondicione brojeve veće od jedan, pri čemu MATHEMATICA

prikazuje informaciju ”*results for inverse of badly conditioned matrix may contain significant numerical errors*”. Algoritam LDLGInverse je napravljen primarno za simbolička izračunavanja sa racionalnim i polinomijalnim matricama gde je poželjno izbeći kvadratne korene.

## 2.2 *LDL<sup>\*</sup>* dekompozicija potpunog ranga polinomijalne matrice

Kao što je rečeno, zamena  $LL^*$  faktorizacije sa  $LDL^*$ , tako da se izbegnu izračunavanja elemenata sa kvadratnim korenima, je od velike važnosti pri radu sa polinomijalnim i racionalnim matricama. Prema tome, motivacija je modifikovati metod Cholesky dekompozicije polinomijalne matrice  $A(x)$ , uvođenjem dodatne dijagonalne matrice  $D$  koja osigurava ne-pojavljivanje elemenata koji sadrže kvadratne korene. Očigledno,  $LDL^*$  dekompozicija je mnogo prikladnija za rad sa polinomijalnim matricama, a kasnije se može iskoristiti za nalaženje uopštenih inverza faktorisane matrice.

Podsetimo se da za slučaj kompleksne Hermitske matrice  $A$ , moguće je odrediti njenu  $LDL^*$  dekompoziciju koristeći recurzivne relacije (2.1.1). U radovima [77] i [96] predložili smo da je izračunavanja (2.1.1) potrebno izvršavati samo za  $j = \overline{1, r}$ , gde je  $r = \text{rang}(A)$ . U tom slučaju, iterativna procedura (2.1.1) generiše faktorizaciju potpunog ranga matrice  $A$ , pri čemu će matrica  $L$  biti bez nula kolona, a matrica  $D$  bez nula vrsta i nula kolona (s obzirom da su matrice  $L$  i  $D$  tada takođe ranga  $r$ ).

Posmatrajmo polinomijalnu Hermitsku matricu  $A \in \mathbf{C}(x)_r^{n \times n}$  ranga  $r$  sa elementima:

$$a_{ij}(x) = \sum_{k=0}^{a_q} a_{k,i,j} x^k, \quad 1 \leq i, j \leq n, \quad (2.2.1)$$

pri čemu je maksimalni eksponent matrice  $A(x)$  označen sa  $a_q$ . Tada je bez-korena Cholesky dekompozicija potpunog ranga matrice  $A$  data sa  $A = LDL^*$ , gde je  $L \in \mathbf{C}(x)^{n \times r}$ ,  $l_{ij} = 0$  za  $i < j$ , a  $D \in \mathbf{C}(x)^{r \times r}$  je odgovarajuća dijagonalna racionalna matrica. Ne-nula elementi racionalnih matrica  $L(x)$  i  $D(x)$  su oblika:

$$\begin{aligned} d_{jj}(x) &= \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j,j} x^k}{\sum_{k=0}^{\bar{\bar{d}}_q} \bar{\bar{d}}_{k,j,j} x^k}, \quad l_{jj}(x) = 1, \quad 1 \leq j \leq r, \\ l_{ij}(x) &= \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k,i,j} x^k}{\sum_{k=0}^{\bar{\bar{l}}_q} \bar{\bar{l}}_{k,i,j} x^k}, \quad 1 \leq j \leq n, \quad 1 \leq i \leq r, \quad j < i. \end{aligned} \quad (2.2.2)$$

Greville-ov metod pregrađivanja [27] i Leverrier–Faddeev algoritam su najčešće korišćeni u simboličkim implementacijama generalisanih inverza. Nekoliko ekstenzija algoritma particije na skupovima racionalnih i polinomijalnih matrica je uvedeno. Prva generalizacija je

ekstenzija Greville-ovog algoritma na skup polinomijalnih i/ili racionalnih matrica jedne nepoznate, uvedena u [87]. Dalje primene i modifikacije algoritama za izračunavanje težinskih Moore-Penrose-ovih inverza se mogu naći u radovima [85, 98]. Za više informacija o izračunavanju Drazinovog inverza, Moore-Penrose-ovog inverza i težinskog Moore-Penrose-ovog inverza, videti [19, 21, 20, 86, 90].

Motivacija je usavršiti efikasni metod za simboličko izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice. Korišćenjem  $LDL^*$  faktorizacije umesto klasične Cholesky dekompozicije, izbegnuto je dobijanje elemenata sa kvadratnim korenima, što je od esencijalne važnosti u simboličkom polinomijalnom izračunavanju. Za simboličko izračunavanje Moore-Penrose-ovog inverza primenjena je Teorema 2.1.6, uvedena u radu [77], na polinomijalnim matricama koje se javljaju tokom izračunavanja.

Već navedena iterativna procedura (2.1.1) se može modifikovati tako da proizvodi kao rezultat dekompoziciju potpunog ranga polinomijalne matrice  $A(x)$ . Prema tome, važe sledeće relacije za racionalne elemente matrica  $D(x)$  and  $L(x)$  za svako  $j = \overline{1, r}$ :

$$f_{ij}(x) = \sum_{k=1}^{j-1} l_{ik}(x) l_{jk}^*(x) d_{kk}(x), \text{ za } i = \overline{j, n}, \quad (2.2.3)$$

$$d_{jj}(x) = a_{jj}(x) - f_{jj}(x), \quad (2.2.4)$$

$$l_{ij}(x) = \frac{1}{d_{jj}(x)}(a_{ij}(x) - f_{ij}(x)), \text{ za } i = \overline{j+1, n}. \quad (2.2.5)$$

Ove rekurentne relacije su iskorišćene u sledećoj teoremi za izračunavanje koeficijenata racionalnih matrica  $L$  i  $D$ . U nastavku disertacije, promenljive sa jednom crtom će označavati koeficijente u brojiocu, dok će promenljive sa dve crte označavati koeficijente u imeniocu razlomka.

**Teorema 2.2.1.** [96]  *$LDL^*$  dekompozicija potpunog ranga Hermitske polinomijalne matrice  $A(x) \in \mathbf{C}(x)_r^{n \times n}$  sa elementima oblika (2.2.1) jednaka je  $A(x) = L(x)D(x)L(x)^*$ , gde su  $L(x)$  i  $D(x)$  racionalne matrice oblika (2.2.2), pri čemu su koeficijenti elemenata  $d_{jj}(x)$  i  $l_{ij}(x)$  dati sledećim izrazima:*

$$\bar{d}_{k,j} = \sum_{k_1=0}^k a_{k-k_1,j,j} \bar{\bar{f}}_{k_1,j,j} - \bar{f}_{k,j,j}, \quad 0 \leq k \leq \bar{d}_q = \max(a_q + \bar{\bar{f}}_q, \bar{f}_q), \quad (2.2.6)$$

$$\bar{\bar{d}}_{k,j} = \bar{\bar{f}}_{k,j,j}, \quad 0 \leq k \leq \bar{\bar{d}}_q = \bar{\bar{f}}_q, \quad (2.2.7)$$

$$\bar{l}_{k,i,j} = \sum_{k_1=0}^k \bar{\bar{d}}_{k-k_1,j} \left( \sum_{k_2=0}^{k_1} a_{k_1-k_2,i,j} \bar{\bar{f}}_{k_2,i,j} - \bar{f}_{k_1,i,j} \right), \quad 0 \leq k \leq \bar{l}_q = \bar{\bar{d}}_q + \max(a_q + \bar{\bar{f}}_q, \bar{f}_q), \quad (2.2.8)$$

$$\bar{\bar{l}}_{k,i,j} = \sum_{k_1=0}^k \bar{d}_{k-k_1,j} \bar{\bar{f}}_{k_1,i,j}, \quad 0 \leq k \leq \bar{\bar{l}}_q = \bar{d}_q + \bar{\bar{f}}_q, \quad (2.2.9)$$

gde su koeficijenti  $\bar{f}_{k,i,j}$  jednaki

$$\bar{f}_{k,i,j} = \sum_{k_2=0}^k \sum_{k_3=0}^{j-1} \bar{p}_{k-k_2,i,j,k_3} q_{k_2,i,j,k_3}, \quad 0 \leq k \leq \bar{f}_q = 2\bar{l}_q + \bar{d}_q + \bar{\bar{f}}_q - 2\bar{\bar{l}}_q - \bar{\bar{d}}_q, \quad (2.2.10)$$

$a \bar{\bar{f}}_{k,i,j}$ ,  $0 \leq k \leq \bar{\bar{f}}_q$ , su koeficijenti sledećeg polinoma:

$$PolinomijalniNZS \left( \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,1} x^k, \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,2} x^k, \dots, \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,j-1} x^k \right), \quad (2.2.11)$$

pri čemu je

$$\begin{aligned} \bar{p}_{t_1,i,j,k} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{l}_{t_3,i,k} \bar{l}_{t_1-t_2-t_3,j,k}^* \bar{d}_{t_2,k}, \quad 0 \leq t_1 \leq 2\bar{l}_q + \bar{d}_q, \\ \bar{\bar{p}}_{t_1,i,j,k} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{l}}_{t_3,i,k} \bar{\bar{l}}_{t_1-t_2-t_3,j,k}^* \bar{\bar{d}}_{t_2,k}, \quad 0 \leq t_1 \leq 2\bar{l}_q + \bar{d}_q, \end{aligned}$$

$$dok su vrednosti q_{k,i,j,t} koeficijenti polinoma q_{i,j,t}(x) = \frac{\sum_{k=0}^{\bar{\bar{f}}_q} \bar{\bar{f}}_{k,i,j} x^k}{\sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,t} x^k}.$$

**Dokaz.** Kako su elementi matrica  $L(x)$  i  $D(x)$  racionalne funkcije, jednačina (2.2.3) dobija sledeći oblik:

$$\begin{aligned} f_{ij}(x) &= \\ &= \sum_{k=1}^{j-1} \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} x^t \sum_{t=0}^{\bar{l}_q} \bar{l}_{t,j,k}^* x^t \sum_{t=0}^{\bar{d}_q} \bar{d}_{t,k} x^t}{\sum_{t=0}^{\bar{l}_q} \bar{\bar{l}}_{t,i,k} x^t \sum_{t=0}^{\bar{l}_q} \bar{\bar{l}}_{t,j,k}^* x^t \sum_{t=0}^{\bar{d}_q} \bar{\bar{d}}_{t,k} x^t} = \sum_{k=1}^{j-1} \frac{\sum_{t_1=0}^{j-1} \sum_{t_2=0}^{t_1-t_1} \bar{l}_{t_3,i,k} \bar{l}_{t_1-t_2-t_3,j,k}^* \bar{d}_{t_2,k} x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \sum_{t_2=0}^{t_1-t_1} \bar{\bar{l}}_{t_3,i,k} \bar{\bar{l}}_{t_1-t_2-t_3,j,k}^* \bar{\bar{d}}_{t_2,k} x^{t_1}} \\ &= \sum_{k=1}^{j-1} \frac{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{p}_{t_1,i,j,k} x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{t_1,i,j,k} x^{t_1}} = \frac{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{p}_{t_1,i,j,1} x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{t_1,i,j,1} x^{t_1}} + \frac{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{p}_{t_1,i,j,2} x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{t_1,i,j,2} x^{t_1}} + \dots + \frac{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{p}_{t_1,i,j,j-1} x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{t_1,i,j,j-1} x^{t_1}}. \end{aligned}$$

S obzirom da je najmanji zajednički sadržalac (NZS) polinoma u imeniciima označen kao:

$$PolinomijalniNZS \left( \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,1} x^k, \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,2} x^k, \dots, \sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,j-1} x^k \right) = \sum_{k=0}^{\bar{\bar{f}}_q} \bar{\bar{f}}_{k,i,j} x^k,$$

i važe jednakosti

$$q_{i,j,t}(x) = \frac{\sum_{k=0}^{\bar{\bar{f}}_q} \bar{\bar{f}}_{k,i,j} x^k}{\sum_{k=0}^{2\bar{l}_q+\bar{d}_q} \bar{\bar{p}}_{k,i,j,t} x^k} = \sum_{k=0}^{\bar{\bar{f}}_q - 2\bar{l}_q - \bar{d}_q} q_{k,i,j,t} x^k, \quad 1 \leq t < j < i,$$

polinomi  $f_{ij}(x)$  se mogu izraziti kao

$$\begin{aligned} f_{ij}(x) &= \frac{\sum_{k=1}^{j-1} \sum_{k_1=0}^{\bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{p}_{k_1-k_2, i, j, k} q_{k_2, i, j, k} \right) x^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k} \\ &= \frac{\sum_{k_1=0}^{\bar{f}_q} \sum_{k_2=0}^{k_1} \sum_{k_3=0}^{j-1} \bar{p}_{k_1-k_2, i, j, k_3} q_{k_2, i, j, k_3} x^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k} = \frac{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k}. \end{aligned}$$

Primetimo da iz jednačine (2.2.4) sledi sledeća jednačina:

$$\begin{aligned} d_{jj}(x) &= \sum_{k=0}^{a_q} a_{k, j, j} x^k - \frac{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k} = \frac{\sum_{k=0}^{a_q} a_{k, j, j} x^k \sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k - \sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k} \\ &= \frac{\sum_{k_1=0}^{\max(a_q + \bar{f}_q, \bar{f}_q)} \left( \sum_{k_2=0}^{k_1} a_{k_1-k_2, j, j} \bar{f}_{k_2, j, j} - \bar{f}_{k_1, j, j} \right) x^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k} = \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k}. \end{aligned}$$

Konačno, na osnovu jednačine (2.2.5), naredna jednakost je validna:

$$\begin{aligned} l_{ij}(x) &= \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k} \left( \sum_{k=0}^{a_q} a_{k, i, j} x^k - \frac{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, i, j} x^k} \right) \\ &= \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k, j} x^k} \frac{\sum_{k_1=0}^{\max(a_q + \bar{f}_q, \bar{f}_q)} \left( \sum_{k_2=0}^{k_1} a_{k_1-k_2, i, j} \bar{f}_{k_2, i, j} - \bar{f}_{k_1, i, j} \right) x^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k, j, j} x^k} \\ &= \frac{\sum_{k=0}^{\bar{d}_q + \max(a_q + \bar{f}_q, \bar{f}_q)} \left( \sum_{k_1=0}^k \bar{d}_{k-k_1, j} \left( \sum_{k_2=0}^{k_1} a_{k_1-k_2, i, j} \bar{f}_{k_2, i, j} - \bar{f}_{k_1, i, j} \right) \right) x^k}{\sum_{k=0}^{\bar{d}_q + \bar{f}_q} \left( \sum_{k_1=0}^k \bar{d}_{k-k_1, j} \bar{f}_{k_1, i, j} \right) x^k} = \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k, i, j} x^k}{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k, i, j} x^k}. \end{aligned}$$

□

Teorema 2.2.1 pruža praktičan metod za kalkulaciju koeficijenata koji se pojavljuju u elementima  $l_{ij}(x)$  i  $d_{jj}(x)$  na osnovu prethodno izračunatih  $l_{ik}(x)$ ,  $l_{jk}(x)$  i  $d_{kk}(x)$ , za  $k < j$ . Sada se algoritam za izračunavanje  $LDL^*$  dekompozicije potpunog ranga date Hermitske polinomijalne matrice  $A(x) = \{a_{ij}(x)\}_{i,j=1}^n$  može formulisati.

---

**Algoritam 2.2** Izračunavanje  $LDL^*$  dekompozicije potpunog ranga zadate Hermitske polinomijalne matrice

---

**Ulaz:** Hermitska polinomijalna matrica  $A(x)$  ranga  $r$ .

- 1: Inicijalizacija:  $d_1(x) := a_{11}(x)$ ,  $l_{11}(x) := \frac{a_{11}(x)}{d_1(x)}$  za  $i = \overline{2, n}$ . Postaviti dijagonalne elemente matrice  $L(x)$  na vrednost 1. Za  $j = \overline{2, r}$  ponavljati korake 2, 3, 4.
- 2: Izračunati koeficijente polinoma  $f_{ij}(x)$  kroz sledeće korake. Za  $i = \overline{j, n}$  ponoviti Korak 2.1, Korak 2.2, Korak 2.3 i Korak 2.4.
  - 2.1: Za svako  $k = \overline{1, j-1}$  izračunati

$$\begin{aligned}\bar{p}_{t_1,i,j,k} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{l}_{t_3,i,k} \bar{l}_{t_1-t_2-t_3,j,k}^* \bar{d}_{t_2,k}, \quad 0 \leq t_1 \leq 2\bar{l}_q + \bar{d}_q, \\ \bar{\bar{p}}_{t_1,i,j,k} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{l}}_{t_3,i,k} \bar{\bar{l}}_{t_1-t_2-t_3,j,k}^* \bar{\bar{d}}_{t_2,k}, \quad 0 \leq t_1 \leq 2\bar{\bar{l}}_q + \bar{\bar{d}}_q.\end{aligned}$$

- 2.2: Odrediti polinomijalni najmanji zajdenički sadržalac generisanih polinoma koji određuju imenojice:

$$\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} x^k = PolinomijalniNZS \left( \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,1} x^k, \dots, \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,j-1} x^k \right).$$

- 2.3: Za svako  $t = \overline{1, j-1}$  i svako  $i > j$  podeliti polinom  $\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} x^k$  sledećim polinomom:  $\sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,t} x^k$ , i količnik označiti kao

$$q_{i,j,t}(x) = \sum_{k=0}^{\bar{f}_q - 2\bar{l}_q - \bar{d}_q} q_{k,i,j,t} x^k.$$

- 2.4: Uvesti smenu  $\bar{f}_q = 2\bar{l}_q + \bar{d}_q + \bar{\bar{f}}_q - 2\bar{\bar{l}}_q - \bar{\bar{d}}_q$ , i izračunati

$$\bar{f}_{k,i,j} = \sum_{k_2=0}^k \sum_{k_3=0}^{j-1} \bar{p}_{k-k_2,i,j,k_3} q_{k_2,i,j,k_3}, \quad 0 \leq k \leq \bar{f}_q.$$

- 3: Uvesti smene  $\bar{d}_q = \max(a_q + \bar{\bar{f}}_q, \bar{f}_q)$  i  $\bar{\bar{d}}_q = \bar{\bar{f}}_q$ . Izvršiti sledeće evaluacije:

$$\begin{aligned}\bar{d}_{k,j} &= \sum_{i=0}^k a_{k-i,j,j} \bar{\bar{f}}_{i,j,j} - \bar{f}_{k,j,j}, \quad 0 \leq k \leq \bar{d}_q \\ \bar{\bar{d}}_{k,j} &= \bar{\bar{f}}_{k,j,j}, \quad 0 \leq k \leq \bar{\bar{d}}_q.\end{aligned}$$

4: Uvesti smene  $\bar{l}_q = \bar{\bar{d}}_q + \max(a_q + \bar{\bar{f}}_q, \bar{f}_q)$  i  $\bar{\bar{l}}_q = \bar{d}_q + \bar{\bar{f}}_q$ . Za svako  $i = \overline{j+1, n}$  izvršiti sledeće evaluacije:

$$\begin{aligned}\bar{l}_{k,i,j} &= \sum_{k_1=0}^k \bar{\bar{d}}_{k-k_1,j} \left( \sum_{k_2=0}^{k_1} a_{k_1-k_2,i,j} \bar{\bar{f}}_{k_2,i,j} - \bar{f}_{k_1,i,j} \right), \quad 0 \leq k \leq \bar{l}_q, \\ \bar{\bar{l}}_{k,i,j} &= \sum_{k_1=0}^k \bar{d}_{k-k_1,j} \bar{\bar{f}}_{k_1,i,j}, \quad 0 \leq k \leq \bar{\bar{l}}_q.\end{aligned}$$

5: **Rezultat:**

$$d_j(x) = \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} x^k}{\sum_{k=0}^{\bar{\bar{d}}_q} \bar{\bar{d}}_{k,j} x^k}, \quad l_{i,j}(x) = \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k,i,j} x^k}{\sum_{k=0}^{\bar{\bar{l}}_q} \bar{\bar{l}}_{k,i,j} x^k}, \quad j = \overline{1, r}, \quad i = \overline{j+1, n}.$$

Ovde navedeni algoritam primenjuje samo operacije množenja, deljenja i sumiranja matričnih elemenata. Pritom, ovaj metod ne podrazumeva korišćenje kvadratnih korena, što kao posledicu podrazumeva kreiranje nove dijagonalne matrice. Očigledno, ovaj algoritam je vrlo pogodan za implementaciju u proceduralnim programskim jezicima.

### 2.2.1 Numerički primeri

**Primer 2.2.1.** Posmatrajmo simetričnu polinomijalnu matricu ranga 2 generisanu u radu [117]:

$$S_3 = \begin{bmatrix} 1+x & x & 1+x \\ x & -1+x & x \\ 1+x & x & 1+x \end{bmatrix}.$$

Za  $j = 1$  je  $d_{11} = 1 + x$ , i prema tome je

$$l_{21}(x) = \frac{x}{1+x}, \quad l_{31}(x) = \frac{1+x}{1+x} = 1.$$

Za slučaj  $j = 2$  imamo da je

$$f_{22}(x) = \frac{x^2}{1+x}, \quad f_{32}(x) = x.$$

Na osnovu ovih rezultata je zadovoljeno:

$$d_{22}(x) = -\frac{1}{1+x}, \quad l_{32}(x) = \frac{1}{d_{22}(x)}(x - f_{32}(x)) = 0,$$

pa se prema tome dobijaju sledeće racionalne matrice:

$$L(x) = \begin{bmatrix} 1 & 0 \\ \frac{x}{1+x} & 1 \\ 1 & 0 \end{bmatrix}, \quad D(x) = \begin{bmatrix} 1+x & 0 \\ 0 & -\frac{1}{1+x} \end{bmatrix}.$$

**Primer 2.2.2.** Posmatrajmo simetričnu polinomijalnu matricu

$$A(x) = \begin{bmatrix} 1+x^2 & x & 2+x & -3+x \\ x & -1+x & 3+x & x \\ 2+x & 3+x & -1+x & 1+x \\ -3+x & x & 1+x & x \end{bmatrix}$$

Na osnovu Koraka 1 Algoritam 2.2 stavićemo  $d_1(x) = 1 + x^2$  i

$$l_{21}(x) = \frac{x}{1+x^2}, \quad l_{31}(x) = \frac{2+x}{1+x^2}, \quad l_{41}(x) = \frac{-3+x}{1+x^2}.$$

Za  $j = 2$  imamo da je

$$f_{22}(x) = \frac{x^2}{1+x^2}, \quad f_{32}(x) = \frac{x(2+x)}{1+x^2}, \quad f_{42}(x) = \frac{(-3+x)x}{1+x^2}.$$

Prema tome, jednostavno se dobija

$$d_{22}(x) = (-1 + x - 2x^2 + x^3)/(1 + x^2)$$

a ne-nula elementi matrice  $L$  su

$$l_{32}(x) = \frac{3 - x + 2x^2 + x^3}{-1 + x - 2x^2 + x^3}, \quad l_{42}(x) = \frac{x(4 - x + x^2)}{-1 + x - 2x^2 + x^3}.$$

Za  $j = 3$  se dobija

$$f_{33}(x) = \frac{5 - 6x + 3x^2 + 5x^3 + x^4}{-1 + x - 2x^2 + x^3}, \quad f_{43}(x) = \frac{6 + 7x - 3x^2 + 2x^3 + x^4}{-1 + x - 2x^2 + x^3}$$

pa prema tome sledi

$$d_{33}(x) = \frac{-4 + 4x - 8x^3}{-1 + x - 2x^2 + x^3}, \quad l_{43}(x) = \frac{7 + 7x - 2x^2 + 3x^3}{4 - 4x + 8x^3}.$$

Konačno, za  $j = 4$  se dobija  $f_{44}(x) = (85 + 87x - 2x^2 - x^3 + 8x^4)/(4 - 4x + 8x^3)$ . Ovaj rezultat vodi do

$$d_{44}(x) = \frac{-85 - 83x - 2x^2 + x^3}{4 - 4x + 8x^3}.$$

Prema tome, dobijene su sledeće matrice iz faktorizacije:

$$L(x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{x}{1+x^2} & 1 & 0 & 0 \\ \frac{2+x}{1+x^2} & \frac{3-x+2x^2+x^3}{-1+x-2x^2+x^3} & 1 & 0 \\ \frac{-3+x}{1+x^2} & \frac{x(4-x+x^2)}{-1+x-2x^2+x^3} & \frac{7+7x-2x^2+3x^3}{4-4x+8x^3} & 1 \end{bmatrix},$$

$$D(x) = \begin{bmatrix} 1+x^2 & 0 & 0 & 0 \\ 0 & \frac{-1+x-2x^2+x^3}{1+x^2} & 0 & 0 \\ 0 & 0 & \frac{-4+4x-8x^3}{-1+x-2x^2+x^3} & 0 \\ 0 & 0 & 0 & \frac{-85-83x-2x^2+x^3}{4-4x+8x^3} \end{bmatrix}.$$

### 2.2.2 Implementacioni detalji i razultati testiranja

S obzirom da je *LDL<sup>\*</sup>* dekompozicija iste složenosti kao i Cholesky faktorizacija, efikasnost našeg algoritma neće biti drugačija od efikasnosti algoritma *LL<sup>\*</sup>* dekompozicije. Algoritam 2.2 podrazumeva jedno više množenje matrice i dijagonalne matrice. Prema tome, algoritamska složenost će biti reda veličine  $O(n^3m^2)$ , pri čemu je  $O(n^3)$  složenost *LDL<sup>\*</sup>* dekompozicije, a  $m$  je maksimalni eksponent svih polinoma dobijenih u algoritmu.

Očito ponekad koeficijenti polinoma mogu veoma rasti u među-koracima, i pored toga što se izvršava simplifikacija. One possible solutions to this problem is to consider implementacija large number operations. Another possibility is to handle this with matrix elements being quotients two polinomijalnas, thereat doing simplification after each step. polinomijalna matrices containing a relativno small number nonzero entries are often considered in practical computations. In this case, previous algoritam is not very effective because many redundant operations.

Procedura **PolynomialLDLDecomposition[A\_List]** za testiranje i proveru koraka algoritma u paketu MATHEMATICA, navedena je u [78], i ovde je data u celosti.

```
PolynomialLDLDecomposition [A_List] :=
Module[{i, j, k, n = Length[A], r = MatrixRank[A], L, D,
f, p, q, Num, Den, DNum, LNum, LDen, UpperLim = 0, x, r1},
L = f = Table[0, {n}, {r}];
D = Table[0, {r}, {r}];
p = q = Table[0, {3 n}, {3 n}];
D[[1, 1]] = A[[1, 1]];
For[j = 1, j <= r, j++, L[[j, j]] = 1];
For[i = 2, i <= n, i++, L[[i, 1]] = Simplify[{{A[[i, 1]]} / {A[[1, 1]]}}];
For[j = 2, j <= r, j++, 
For[i = j, i <= n, i++, 
For[k = 1, k <= j - 1, k++, 
Num[k] = 0; Den[k] = 0;
For[r1 = 0, r1 <= 2 * Exponent[L[[i, k]], x] + Exponent[D[[k, k]], x], r1++,
p[[r1 + 1, k]] = 0; q[[r1 + 1, k]] = 0;
p[[r1 + 1, k]] = Sum[Sum[Coefficient[Numerator[L[[i, k]]], x, r3] *
Coefficient[Numerator[L[[j, k]]], x, r1 - r2 - r3] *
Coefficient[Numerator[D[[k, k]]], x, r2],
{r2 = 0, r3 = 0}], {r1 = 0, r2 = r1 - r3}]; 
q[[r1 + 1, k]] = Sum[Sum[Coefficient[Denominator[L[[i, k]]], x, r3] *
Coefficient[Denominator[L[[j, k]]], x, r1 - r2 - r3] *
Coefficient[Denominator[D[[k, k]]], x, r2],
{r2 = 0, r3 = 0}], {r1 = 0, r2 = r1 - r3}]; 
Num[k] += p[[r1 + 1, k]] * x^r1;
Den[k] += q[[r1 + 1, k]] * x^r1;
];
];
];
f[[i, j]] = Simplify[Sum[Num[k] / Den[k], {k, 1, j - 1}]];
Print["f[", i, j, "]", f[[i, j]]];
];
];
```

```

DNum = 0;
UpperLim = Max[Max[Exponent[A, x]] + Max[Exponent[Denominator[f[[j, j]]], x]], 
    Max[Exponent[Numerator[f[[j, j]]], x]]];
For[k = 0, k ≤ UpperLim, k++,
    DNum += ((Sum[Coefficient[A[[j, j]], x, k - i] * 
        Coefficient[Denominator[f[[j, j]]], x, i]) - 
    Coefficient[Numerator[f[[j, j]]], x, k]) * x^k];
D[[j, j]] = Simplify[DNum / Denominator[f[[j, j]]]];
For[i = j + 1, i ≤ n, i++,
    LNum = 0;
    For[k = 0, k ≤ Max[Exponent[Denominator[f[[i, j]]], x]] + UpperLim, k++,
        LNum += Sum[Coefficient[Denominator[D[[j, j]]], x, k - k1] * 
            ((Sum[Coefficient[A[[i, j]], x, k1 - k2] * Coefficient[Denominator[f[[i, j]]], 
                x, k2]) - Coefficient[Numerator[f[[i, j]]], x, k1]) * x^k];
    LDen = 0;
    For[k = 0, k ≤ Max[Exponent[Denominator[f[[i, j]]], x]] + UpperLim, k++,
        LDen += Sum[Coefficient[Numerator[D[[j, j]]], x, k - k1] * 
            Coefficient[Denominator[f[[i, j]]], x, k1]) * x^k];
    L[[i, j]] = Simplify[LNum / LDen];
];
];
Return[{Simplify[L], Simplify[D]}];

```

Dizajniran je i paket `PolynomialMatrix` u Java za simboličke operacije na maticama sa polinomijalnim elementima iz skupa  $Z[x]$ . Ovaj paket sadrži četiri glavne klase: `Polynomial`, `RationalFunction`, `Matrix` i `Decompositions`. Polinomijalna implementacija je implementirana na dva načina: koristeći nizove koeficijenata i heš tabele za nenula koeficijente kod retkih polinoma. Čak i ako su ulazni i izlazni polinomi u algoritmima vrlo kratki, koeficijenti u među-koracima mogu biti ogromni. Zato su korišćena dva tipa za koeficijente: `long` and `BigInteger` (paket za celobrojne operacije u JAVA). Prema tome, imamo četiri dobijene klase, pri čemu uvek koristimo najpogodniju. Četiri osnovne operacije su podržane *Karatsuba* algoritmom za množenje i polinomijalno deljenje za određivanje ostataka.

Korišćen je brzi modularni  $nzd$  algoritam za određivanje najvećeg zajedničkog delioca dva polinoma (primenjujući Kinesku teoremu o ostacima i Euklidov algoritam). Polinom  $P(x)$  je prost ako su svi njegovi koeficijenti uzajamno prosti. Koeficijenti i parcijalni rezultati rastu vrlo brzo u Euklidovom algoritmu kod polinomijalne sekvence ostataka. Zato je moguće izračunati prosti deo ostatka u svakom koraku. Međutim, izračunavanje prostog dela zahteva izračunavanje najvećih zajedničkih delilaca polinoma čiji koeficijenti

mogu biti veliki. Konačno, koristili smo modularni pristup iz rada [112], a koristeći Landay–Mignotte granicu, svaki koeficijent nzd-a celobrojnih polinoma  $a(x) = \sum_{i=0}^m a_i x^i$  i  $b(x) = \sum_{i=0}^n b_i x^i$ , gde je  $a_m \neq 0$  i  $b_n \neq 0$ , je ograničen po absolutnoj vrednosti sa

$$2^{\min(m,n)} \cdot \text{nzd}(a_m, b_n) \cdot \min\left(\frac{1}{a_m} \sqrt{\sum_{i=0}^m a_i^2}, \frac{1}{b_n} \sqrt{\sum_{i=0}^n b_i^2}\right).$$

Sledeća lema je fundamentalna za implementaciju u JAVA.

**Lema 2.2.2.** *Neka su  $a, b \in Z[x]$  dva zadata polinoma, i neka je  $p$  prost broj koji ne deli vodeće koeficijente polinoma  $a(x)$  i  $b(x)$ . Neka su  $a_{(p)}$  i  $b_{(p)}$  slike od  $a$  i  $b$  po modulu  $p$ , respektivno. Neka je  $c = \text{nzd}(a, b)$  nad  $Z$ . Tada je  $\text{st}(\text{nzd}(a_{(p)}, b_{(p)})) \geq \text{st}(\text{nzd}(a, b))$ . Štaviše, ako  $p$  ne deli rezultante  $a/c$  i  $b/c$ , tada  $\text{nzd}(a_{(p)}, b_{(p)}) = c \bmod p$ .*

Naravno, nzd polinoma nad  $Z_p$  je određen sa tačnošću samo do proizvoda sa konstantom. Tada koristimo Kinesku teoremu o ostacima za rekonstrukciju koeficijenata najvećeg zajedničkog delioca.

Racionalne funkcije su sačuvane kao uređeni parovi brojilaca i imenilaca, koji su prosti polinomi. Nakon svake operacije, izvršava se simplifikacija razlomka deljenjem sa nzd-om brojioca i imenioca. Kako radimo sa matricama dimenzije reda maksimalno 100, koristili smo regularno matrično množenje složenosti  $O(n^3)$ , i Gauss-Jordan-ovu eliminaciju za izračunavanje inverza, ranga i determinante matrice.

**Primer 2.2.3.** Ovde prezentujemo uporedne rezultate kod izračunavanja posmatrajući vremena izvršavanja različitih implementacija algoritma 2.2 u MATHEMATICA i JAVA. Svi testovi su izvršeni na sistemu Intel Core 2 Duo T5800 2.0 GHz sa 2 GB RAM memorijom, na Windows operativnom sistemu, koristeći JRE 1.6.0\_24 i MATHEMATICA 7.0. Svako vreme je dobijeno kao srednja vrednost od 20 nezavisnih izvršavanja.

Naš skup test matrica se sastojao od  $n \times n$  matrica sa slučajnim ne-nula koeficijentima iz intervala  $[-10, 10]$  i stepana  $d$ . Ovde je MATHEMATICA implementacija algoritma 2.2 bila superiorna u odnosu na ostale implementacije. Glavni razlog su česte simplifikacije koje su izvršavane u MATHEMATICA. Implementacija osnovne procedure (2.2.3)-(2.2.5) je označena kao MATH. osnovni algoritam i očigledno je dosta sporija od implementacije algoritma 2.2, koja je označena kao MATH. algoritam 2.2.

<i>n</i>	<i>d</i>	MATH. osnovni algoritam	MATH. algoritam 2.2	Java algoritam 2.2
5	10	0.27	0.04	0.03
5	25	0.80	0.09	0.17
5	50	1.86	0.18	0.67
5	100	5.86	0.44	2.95
5	200	19.91	1.12	12.42
10	2	0.21	0.09	0.24
10	5	0.48	0.15	0.89
10	10	1.14	0.23	4.35
10	25	3.36	0.54	31.42
10	50	8.81	1.24	118.46
10	100	29.72	2.81	-

Tabela 2.2.1. Vremena izračunavanja u sekundama dobijena različitim implementacijama Algoritma 2.2 na nasumičnim test matricama.

## 2.3 Izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice

Na osnovu prethodne Teoreme 2.2.1 moguće je izvesti sledeću posledicu Teoreme 2.1.6.

**Teorema 2.3.1.** [96] Neka je  $A \in \mathbf{C}(x)_r^{m \times n}$  polinomijalna matrica sa elementima oblika (2.2.1). Posmatrajmo  $LDL^*$  faktorizaciju potpunog ranga matrice  $(A^*A)^*(A^*A)$ , gde su  $L \in \mathbf{C}(x)^{n \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$  matrice sa elementima oblika (2.2.2). Označimo elemente inverzne matrice  $N = (L^*LDL^*)^{-1} \in \mathbf{C}(x)^{r \times r}$  sa

$$n_{ij}(x) = \frac{\sum_{t=0}^{\bar{n}_q} \bar{n}_{t,k,l} x^t}{\sum_{t=0}^{\bar{n}_q} \bar{n}_{t,k,l} x^t}.$$

Tada se proizvoljni element Moore-Penrose-ovog inverza matrice  $A$  može izračunati kao  $A_{ij}^\dagger(x) = \frac{\bar{\Gamma}_{ij}(x)}{\bar{\Gamma}_i(x)}$ , gde je

$$\begin{aligned} \bar{\Gamma}_{ij}(x) &= \sum_{t=0}^{\bar{\Gamma}_q - \bar{b}_q + \bar{b}_q} \left( \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t_1=0}^t \bar{\beta}_{t_1, i, j, k, l, \mu, \kappa, \lambda} \gamma_{t-t_1, i, k, l, \mu} \right) x^t, \\ \bar{\bar{\Gamma}}_i(x) &= \text{Polinomijalni NZS} \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\bar{\beta}}_{t, i, k, l, \mu} x^t \mid \mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}} \right\} \end{aligned} \quad (2.3.1)$$

pri čemu je  $\bar{\Gamma}_q$  maksimalni eksponent u polinomima  $\bar{\bar{\Gamma}}_i(x)$ ,  $1 \leq i \leq m$ , vrednosti  $\gamma_{t, i, k, l, \mu}$ ,  $0 \leq t \leq \bar{\Gamma}_q - \bar{b}_q$ , su koeficijenti polinoma  $\Gamma_{i, k, l, \mu}(x) = \frac{\bar{\Gamma}_i(x)}{\sum_{t=0}^{\bar{b}_q} \bar{\bar{\beta}}_{t, i, k, l, \mu} x^t}$ , za svako  $\mu = \overline{1, n}$ ,  $k = \overline{1, \min\{i, r\}}$ ,  $l = \overline{1, \min\{\mu, r\}}$

pri čemu su korišćene sledeće notacije za  $\kappa = \overline{1, m}$ ,  $\lambda = \overline{1, n}$ :

$$\begin{aligned} \bar{\beta}_{t, i, j, k, l, \mu, \kappa, \lambda} &= \sum_{t_1=0}^t \bar{p}_{t_1, i, k, l, \mu} \alpha_{t-t_1, j, \mu, \kappa, \lambda}, & 0 \leq t \leq \bar{b}_q = 2\bar{l}_q + \bar{n}_q + 3\bar{a}_q, \\ \bar{\bar{\beta}}_{t, i, k, l, \mu} &= \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{l}_{t_1, i, k} \bar{n}_{t-t_1-t_2, k, l} \bar{l}_{t_2, \mu, l}^*, & 0 \leq t \leq \bar{b}_q = 2\bar{l}_q + \bar{n}_q, \\ \bar{p}_{t, i, k, l, \mu} &= \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{l}_{t_1, i, k} \bar{n}_{t-t_1-t_2, k, l} \bar{l}_{t_2, \mu, l}^*, & 0 \leq t \leq 2\bar{l}_q + \bar{n}_q \\ \alpha_{t, j, \mu, \kappa, \lambda} &= \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} a_{t_1, \kappa, \lambda} a_{t-t_1-t_2, \mu, \kappa, \lambda}^* a_{t_2, j, \lambda}^*, & 0 \leq t \leq 3\bar{a}_q. \end{aligned} \quad (2.3.2)$$

**Dokaz.** Primetimo da su sledeće jednakosti zadovoljene:

$$\begin{aligned} (LNL^*)_{ij} &= \sum_{l=1}^r \left( \sum_{k=1}^r l_{ik} n_{kl} \right) l_{jl}^* = \sum_{l=1}^r \sum_{k=1}^r l_{ik} n_{kl} l_{jl}^*, \\ ((A^* A)^* A^*)_{ij} &= \sum_{l=1}^n \left( \sum_{k=1}^m a_{kl} a_{ki}^* \right) a_{jl}^* = \sum_{l=1}^n \sum_{k=1}^m a_{kl} a_{ki}^* a_{jl}^*. \end{aligned}$$

Na osnovu prvog tvrdjenja Teoreme 2.1.6, proizvoljni  $(i, j)$ -ti element Moore-Penrose-ovog inverza matrice  $A$  se može izračunati kao:

$$\begin{aligned} A_{ij}^\dagger &= \sum_{\mu=1}^n \left( \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} l_{ik} n_{kl} l_{\mu l}^* \right) \left( \sum_{\lambda=1}^n \sum_{\kappa=1}^m a_{\kappa\lambda} a_{\kappa\mu}^* a_{j\lambda}^* \right) \\ &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m l_{ik} n_{kl} l_{\mu l}^* a_{\kappa\lambda} a_{\kappa\mu}^* a_{j\lambda}^*. \end{aligned}$$

Dalje, razvojem polinoma u prethodnom izrazu,  $(i, j)$ -ti element matrice  $A^\dagger$  se može izračunati kao:

$$\begin{aligned} A_{ij}^\dagger(x) &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} x^t \sum_{t=0}^{\bar{n}_q} \bar{n}_{t,k,l} x^t \sum_{t=0}^{\bar{l}_q} \bar{l}_{t,\mu,l}^* x^t}{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} x^t \sum_{t=0}^{\bar{n}_q} \bar{n}_{t,k,l} x^t \sum_{t=0}^{\bar{l}_q} \bar{l}_{t,\mu,l}^* x^t} \\ &\quad \cdot \sum_{t=0}^{a_q} a_{t,\kappa,\lambda} x^t \sum_{t=0}^{a_q} a_{t,\kappa,\mu}^* x^t \sum_{t=0}^{a_q} a_{t,j,\lambda}^* x^t \\ &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{2\bar{l}_q + \bar{n}_q} \bar{p}_{t,i,k,l,\mu} x^t}{\sum_{t=0}^{2\bar{l}_q + \bar{n}_q} \bar{\beta}_{t,i,k,l,\mu} x^t} \sum_{t=0}^{3a_q} \alpha_{t,j,\mu,\kappa,\lambda} x^t \\ &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t}{\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,k,l,\mu} x^t}. \end{aligned}$$

Prema tome, imamo da je  $A_{ij}^\dagger(x) = \frac{\bar{\Gamma}_{ij}(x)}{\bar{\Gamma}_i(x)}$ , pri čemu važi

$$\begin{aligned} \bar{\Gamma}_i(x) &= Polinomijalni NZS \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,k,l,\mu} x^t \mid \mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}} \right\} \\ &= \sum_{t=0}^{\bar{\Gamma}_q} \bar{\gamma}_{t,i} x^t, \end{aligned}$$

### 2.3. IZRAČUNAVANJE MOORE-PENROSE-OVOG INVERZA POLINOMIJALNE MATRICE 31

$$\bar{\Gamma}_{ij}(x) = \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu,r\}} \sum_{k=1}^{\min\{i,r\}} \left( \Gamma_{i,k,l,\mu}(x) \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \right),$$

gde je svaki polinom  $\Gamma_{i,l,k,\mu}(x)$  jednak  $\bar{\Gamma}_i(x)/\left(\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,k,l,\mu} x^t\right) = \sum_{t=0}^{\bar{b}_q} \gamma_{t,i,k,l,\mu} x^t$ . Prema tome

$$\bar{\Gamma}_{ij}(x) = \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu,r\}} \sum_{k=1}^{\min\{i,r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t=0}^{\bar{b}_q} \left( \sum_{t_1=0}^t \bar{\beta}_{t_1,i,j,k,l,\mu,\kappa,\lambda} \gamma_{t-t_1,i,k,l,\mu} \right) x^t,$$

što se poklapa sa formom (2.3.26), čime je dokaz kompletan.  $\square$

Očigledno je moguće izvesti sličnu teoremu posmatrajući drugo tvrđenje Teoreme 2.1.6. Dobijene rezultate je moguće sumirati sledećim algoritmom.

---

**Algoritam 2.3** Simboličko izračunavanje Moore-Penrose-ovog inverza korišćenjem  $LDL^*$  dekompozicije potpunog ranga

---

**Ulaz:** Polinomijalna matrica  $A(x) \in \mathbf{C}(x)_r^{m \times n}$  sa elementima oblika  $a_{ij}(x) = \sum_{t=0}^{a_q} a_{t,i,j} x^t$ .

- 1: Generisati  $LDL^*$  dekompoziciju potpunog ranga matrice  $(A^* A)^*(A^* A)$ , gde su  $L \in \mathbf{C}(x)^{n \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$  matrice sa elementima oblika (2.2.2), primenom metoda datog jednačinama (2.3.2) iz Teoreme 2.2.1.
- 2: Transformisati racionalnu matricu  $M = L^* LDL^* L$  u oblik  $M = \frac{1}{p(x)} M_1$ , pri čemu je  $p(x)$  polinom a  $M_1$  polinomijalna matrica.
- 3: Izračunati inverz matrice  $M_1$  primenom Algoritma 3.2 iz rada [98]. Generisati inverznu matricu  $N = M^{-1} = p(x) M_1^{-1}$ , a onda je redukovati na formu:  $n_{ij}(x) = \left( \sum_{k=0}^{\bar{n}_q} \bar{n}_{k,i,j} x^k \right) / \left( \sum_{k=0}^{\bar{n}_q} \bar{n}_{k,i,j} x^k \right)$ .
- 4: Za svako  $i = \overline{1, m}$ ,  $\mu = \overline{1, n}$ ,  $k = \overline{1, \min\{i, r\}}$ ,  $l = \overline{1, \min\{\mu, r\}}$  izračunati sledeće koeficijente:

$$\bar{p}_{t,i,k,l,\mu} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{l}_{t_1,i,k} \bar{n}_{t-t_1-t_2,k,l} \bar{l}_{t_2,\mu,l}^*, \quad 0 \leq t \leq 2\bar{l}_q + \bar{n}_q \quad (2.3.3)$$

$$\bar{\bar{\beta}}_{t,i,k,l,\mu} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{\bar{l}}_{t_1,i,k} \bar{\bar{n}}_{t-t_1-t_2,k,l} \bar{\bar{l}}_{t_2,\mu,l}^*, \quad 0 \leq t \leq 2\bar{\bar{l}}_q + \bar{\bar{n}}_q. \quad (2.3.4)$$

- 5: Za svako  $j = \overline{1, n}$ ,  $\mu = \overline{1, n}$ ,  $\kappa = \overline{1, m}$ ,  $\lambda = \overline{1, n}$  izvršiti sledeće evaluacije:

$$\alpha_{t,j,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} a_{t_1,\kappa,\lambda} a_{t-t_1-t_2,\mu,\kappa,\lambda}^* a_{t_2,j,\lambda}^*, \quad 0 \leq t \leq 3a_q. \quad (2.3.5)$$

### 2.3. IZRAČUNAVANJE MOORE-PENROSE-OVOG INVERZA POLINOMIJALNE MATRICE 32

6: Uvesti označke  $\bar{b}_q = 2\bar{l}_q + \bar{n}_q + 3\bar{a}_q$ ,  $\bar{\bar{b}}_q = 2\bar{\bar{l}}_q + \bar{\bar{n}}_q$  i za svako  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ ,  $\mu = \overline{1, n}$ ,  $k = \overline{1, \min\{i, r\}}$ ,  $l = \overline{1, \min\{\mu, r\}}$ ,  $\kappa = \overline{1, m}$ ,  $\lambda = \overline{1, n}$  izračunati

$$\bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \bar{p}_{t_1,i,k,l,\mu} \alpha_{t-t_1,j,\mu,\kappa,\lambda}, \quad 0 \leq t \leq \bar{b}_q. \quad (2.3.6)$$

7: Za  $i = \overline{1, m}$  evaluirati polinome u imenocima elementa  $A_{i,j}^\dagger$  na sledeći način:

$$\bar{\bar{\Gamma}}_i(x) = \text{Polinomijalni NZS} \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,k,l,\mu} x^t \mid \mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}} \right\} \quad (2.3.7)$$

i označiti ih sa  $\bar{\bar{\Gamma}}_i(x) = \sum_{t=0}^{\bar{b}_q} \bar{\bar{\gamma}}_{t,i} x^t$ .

8: Za svako  $i = \overline{1, m}$ ,  $\mu = \overline{1, n}$ ,  $k = \overline{1, \min\{i, r\}}$ ,  $l = \overline{1, \min\{\mu, r\}}$  odrediti sledeći polinom:  $\bar{\bar{\Gamma}}_i(x) / \left( \sum_{t=0}^{\bar{b}_q} \bar{\bar{\beta}}_{t,i,k,l,\mu} x^t \right)$ , and denote it as  $\Gamma_{i,l,k,\mu}(x) = \sum_{t=0}^{\bar{\bar{\Gamma}}_i - \bar{b}_q} \gamma_{t,i,k,l,\mu} x^t$ .

9: Za  $i = \overline{1, m}$ ,  $j = \overline{1, n}$  izračunati polinome brojilaca:

$$\bar{\Gamma}_{ij}(x) = \sum_{t=0}^{\bar{\Gamma}_i - \bar{b}_q + \bar{b}_q} \left( \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^m \sum_{\lambda=1}^{\min\{i, r\}} \sum_{\kappa=1}^n \sum_{\lambda=1}^m \sum_{t_1=0}^t \bar{\beta}_{t_1,i,j,k,l,\mu,\kappa,\lambda} \gamma_{t-t_1,j,k,l,\mu} \right) x^t. \quad (2.3.8)$$

10: Za  $i = \overline{1, m}$ ,  $j = \overline{1, n}$  postaviti  $(i, j)$ -ti element Moore-Penrose-ovog inverza  $A^\dagger$  na  $\bar{\Gamma}_{ij}(x) / \bar{\bar{\Gamma}}_i(x)$ .

---

$LDL^*$  dekompozicija je jednake složenosti kao i Cholesky dekompozicija. Primetimo da  $LDL^*$  dekompozicija daje jednu dodatnu dijagonalnu matricu, ali vraća rezultat bez kvadratnih korena, pogodniji za dalja simbolička izračunavanja. Takođe, totalni broj ne-nula elemenata je jednak kao i za Cholesky dekompoziciju.

Primetimo još da se  $(i, j)$ -ti element matrice  $(A^*A)^*(A^*A)$  može izračunati kao

$$\sum_{l=1}^n \sum_{k=1}^m \sum_{\kappa=1}^m a_{kl}^* a_{ki} a_{\kappa l}^* a_{\kappa j}.$$

Tada su ovi polinomi ulazne vrednosti algoritma 2.3, korišćenog za određivanje  $LDL^*$  dekompozicije u Koraku 1. Slična ideja je korišćena i u Koraku 3 gore navedenog algoritma, pri određivanju ulaznih vrednosti Algoritma 3.2 iz [98].

#### 2.3.1 Numerički primeri

**Primer 2.3.1.** Neka je data polinomijalna matrica  $S_3$  iz [117]. Da bismo izračunali elemente matrice  $A^\dagger$ ,  $LDL^*$  dekompozicija matrice  $(S_3^* S_3)^* S_3^* S_3$  se određuje direktnim

### 2.3. IZRAČUNAVANJE MOORE-PENROSE-OVOG INVERZA POLINOMIJALNE MATRICE 33

množenjem elemenata matrica  $L \in \mathbf{C}(x)^{3 \times 2}$  i  $D \in \mathbf{C}(x)^{2 \times 2}$  na osnovu Algoritma 2.2. Prema tome,

$$L = \begin{bmatrix} 1 & 0 \\ \frac{5x+21x^2+27x^3+27x^4}{8+32x+57x^2+54x^3+27x^4} & 1 \\ 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 8 + 32x + 57x^2 + 54x^3 + 27x^4 & 0 \\ 0 & \frac{8}{8+32x+57x^2+54x^3+27x^4} \end{bmatrix}.$$

Nakon transformisanja matrice  $L^* LDL^* L$  na način opisan u algoritmu, primenom Algoritma 3.2 iz [98] imamo:

$$N = (L^* LDL^* L)^{-1} = \left[ \begin{array}{c} \frac{1}{32}(1 - 4x + 12x^2 + 27x^4) \\ \frac{-85x - 657x^2 - 2349x^3 - 5265x^4 - 7695x^5 - 8019x^6 - 5103x^7 - 2187x^8}{256 + 1024x + 1824x^2 + 1728x^3 + 864x^4} \\ \frac{-85x - 657x^2 - 2349x^3 - 5265x^4 - 7695x^5 - 8019x^6 - 5103x^7 - 2187x^8}{256 + 1024x + 1824x^2 + 1728x^3 + 864x^4} \\ \frac{2048 + 24576x + 142905x^2 + 532782x^3 + 1420335x^4 + 2858328x^5 + 4466826x^6 + 5484996x^7 + 5288166x^8 + 3936600x^9 + 2184813x^{10} + 826686x^{11} + 177147x^{12}}{2048 + 16384x + 61952x^2 + 144384x^3 + 228384x^4 + 252288x^5 + 191808x^6 + 93312x^7 + 23328x^8} \end{array} \right]$$

Evaluacija koeficijenata iz Koraka 4-6 se jednostavno izvodi, kao i evaluacija polinomijalnog najmanjeg zajedničkog sadržaoca potrebnog u Koraku 7. Primetimo da je simplifikacija od ključnog značaja u Koraku 8, gde se izračunavaju koeficijenti  $\gamma_{t,i,k,l,\mu}$ ,  $i = \overline{1,3}$ ,  $k = \overline{1,2}$ ,  $l = \overline{1,3}$ ,  $\mu = \overline{1,3}$ . Konačno, generalisani inverz

$$S_3^\dagger = \begin{bmatrix} \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \\ \frac{x}{2} & -1-x & \frac{x}{2} \\ \frac{1-x}{4} & \frac{x}{2} & \frac{1-x}{4} \end{bmatrix}$$

je dobijen nakon simplifikacije svakog elementa koji je u formi razlomka  $\bar{\Gamma}_{ij}(x)/\bar{\Gamma}_i(x)$ ,  $i = \overline{1,3}$ ,  $j = \overline{1,3}$ , računanjem najvećeg zajedničkog delioca svakog para brojilaca i imenilaca.

**Primer 2.3.2.** Posmatrajmo sledeću  $4 \times 3$  polinomijalnu matricu  $A_3$  generisanu u [117]:

$$A_3 = \begin{bmatrix} 3+x & 2+x & 1+x \\ 2+x & 1+x & x \\ 1+x & x & -1+x \\ x & -1+x & -2+x \end{bmatrix}.$$

S obzirom da je rang matrice  $A_3$  jednak 2, naša  $LDL^*$  dekompozicija potpunog ranga matrice  $(A_3^* A_3)^* A_3^* A_3$  daje matrice  $L \in \mathbf{C}(x)^{3 \times 2}$  i  $D \in \mathbf{C}(x)^{2 \times 2}$  sa sledećim elementima:

$$L = \begin{bmatrix} 1 & 0 \\ \frac{21+38x+37x^2+18x^3+6x^4}{33+60x+52x^2+24x^3+6x^4} & 1 \\ \frac{9+16x+22x^2+12x^3+6x^4}{33+60x+52x^2+24x^3+6x^4} & 2 \end{bmatrix}, D = \begin{bmatrix} 264+480x+416x^2+192x^3+48x^4 & 0 \\ 0 & \frac{300}{33+60x+52x^2+24x^3+6x^4} \end{bmatrix}.$$

Primenom Algoritma 2.3 na matrice  $A_3$ ,  $L$  i  $D$ , dobijamo Moore-Penrose-ov inverz matrice  $A_3$ :

$$A_3^\dagger = \begin{bmatrix} -\frac{3}{20}(-1+x) & \frac{1}{60}(8-3x) & \frac{1}{60}(7+3x) & \frac{1}{20}(2+3x) \\ \frac{1}{10} & \frac{1}{30} & -\frac{1}{30} & -\frac{1}{10} \\ \frac{1}{20}(1+3x) & \frac{1}{60}(-4+3x) & \frac{1}{60}(-11-3x) & -\frac{3}{20}(2+x) \end{bmatrix}.$$

### 2.3.2 Implementacioni detalji

Na osnovu formulacije Algoritma 2.3 moguće je implementirati sledeću proceduru za izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice. Jedinstveni argument sledeće funkcije je polinomijalna matrica  $A$  zadata u obliku liste. Primetimo da je korišćena funkcija `LDLDecomposition` za izračunavanje  $LDL^*$  faktorizacije matričnog proizvoda  $(A^*A)^2$ .

```

LDLGINverse[A_List] := Module[{t, i, j, k, l, mu, m = Length[A], n = Length[A[[1]]]},
  r = MatrixRank[A], L, D, N, GInv, f, p, beta1, beta2, alpha, kap, lam, r1},
{L, D} = LDLDecomposition[Conjugate[Transpose[A]].A.Conjugate[Transpose[A]].A];
L = ExpandDenominator[ExpandNumerator[L]];
D = ExpandDenominator[ExpandNumerator[Together[D]]];
N = Simplify[Inverse[Conjugate[Transpose[L]].L.D.Conjugate[Transpose[L]].L]];
N = ExpandDenominator[ExpandNumerator[N]];
p = Table[0,
 {2 * Max[Exponent[L, x]] + Max[Exponent[N, x]] + 1}, {m + 1}, {r + 1}, {r + 1}, {n + 1}];
alpha = Table[0, {3 * Max[Exponent[A, x]] + 35}, {n + 1}, {n + 1}, {m + 1}, {n + 1}];
beta1 = Table[0, {m}, {n}, {r}, {r}, {n}, {m}, {n}];
beta2 = Table[0, {m}, {r}, {r}, {n}];
GInv = Table[0, {m}, {n}];
For[i = 1, i ≤ m, i++,
  For[k = 1, k ≤ Min[i, r], k++,
    For[l = 1, l ≤ Min[mu, r], l++,
      For[mu = 1, mu ≤ n, mu++,
        beta2[[i, k, l, mu]] = 0;
        For[t = 0, t ≤ Max[Exponent[L[[i, k]], x]] +
          Max[Exponent[N[[k, l]], x]] + Max[Exponent[L[[mu, l]], x]], t++,
          p[[t + 1, i, k, l, mu]] = Sum[Sum[(Coefficient[Numerator[L[[i, k]]],
            x, t1] * Coefficient[Numerator[N[[k, l]]], x, t - t1 - t2] *
            Conjugate[Coefficient[Numerator[L[[mu, l]]], x, t2]]),
          {t1 = 0, t2 = 0}], {t1 = 0, t2 = 0}], t - t1 - t2] * Coefficient[Denominator[L[[i, k]]],
          x, t1] * Coefficient[Denominator[N[[k, l]]], x, t - t1 - t2] *
          Conjugate[Coefficient[Denominator[L[[mu, l]]], x, t2]]];
        beta2[[i, k, l, mu]] += x^t * Sum[Sum[(Coefficient[Denominator[L[[i, k]]],
          x, t1] * Coefficient[Denominator[N[[k, l]]], x, t - t1 - t2] *
          Conjugate[Coefficient[Denominator[L[[mu, l]]], x, t2]]),
          {t1 = 0, t2 = 0}], {t1 = 0, t2 = 0}], t - t1 - t2];
      ];
    ];
  ];
For[j = 1, j ≤ n, j++,
  For[mu = 1, mu ≤ n, mu++,
    For[kap = 1, kap ≤ m, kap++,
      For[lam = 1, lam ≤ n, lam++,
        For[t = 0, t ≤ 3 * Max[Exponent[A, x]], t++,
          alpha[[t + 1, j, mu, kap, lam]] =
          Sum[Sum[(Coefficient[A[[kap, lam]], x, t1] * Conjugate[Coefficient[A[[kap,
            mu]], x, t - t1 - t2]] * Conjugate[Coefficient[A[[j, lam]], x, t2]]),
          {t1 = 0, t2 = 0}], {t1 = 0, t2 = 0}], t - t1 - t2];
      ];
    ];
  ];

```

```

For[i = 1, i ≤ m, i++,
  For[j = 1, j ≤ n, j++,
    For[k = 1, k ≤ Min[i, r], k++,
      For[l = 1, l ≤ Min[mu, r], l++,
        For[mu = 1, mu ≤ n, mu++,
          For[kap = 1, kap ≤ m, kap++,
            For[lam = 1, lam ≤ n, lam++,
              For[t = 0, t ≤ 2 * Max[Exponent[Numerator[L], x]] + Max[Exponent[
                Numerator[N], x]] + 3 * Max[Exponent[Numerator[A], x]], t++,
                beta1[[i, j, k, l, mu, kap, lam]] += x^t * Sum(p[[t1 + 1, i,
                  k, l, mu]] * alpha[[t - t1 + 1, j, mu, kap, lam]]),
                t1 = 0;
              ];
            ];
          ];
        ];
      ];
    ];
  ];
Return[Simplify[GInv]];
];

```

## 2.4 $LDL^*$ dekompozicija potpunog ranga racionalne matrice

Neka je data racionalna Hermitska matrica  $A(s) \in \mathbb{C}[s]_r^{n \times n}$ , sa elementima koji predstavljaju količnik dva polinoma po nepoznatoj  $s$ :

$$a_{ij}(s) = \frac{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,i,j} s^k}{\sum_{k=0}^{\bar{\bar{a}}_q} \bar{\bar{a}}_{k,i,j} s^k}, \quad 1 \leq i, j \leq n. \quad (2.4.1)$$

Alternativna forma Cholesky faktorizacije, korišćena u radu [74], podrazumeva  $LDL^*$

dekompoziciju simetrične racionalne matrice  $A(s)$  oblika

$$A(s) = L(s)D(s)L^*(s),$$

gde su  $L(s)$  i  $D(s)$  donja trougaona i dijagonalna matrica, respektivno, i imaju oblik dat sa (1.3.2).

Koristeći  $LDL^*$  faktorizaciju, izbegava se izračunavanje elemenata sa kvadratnim ko- renima. Na primer, raditi sa izrazom  $\sqrt{\sum_{i=0}^q A_i s^i}$  simbolički, gde su  $A_i$ ,  $i = 0, \dots, q$  konstantne  $n \times n$  matrice, je veoma teško. Zbog navedenog je u [74] dat algoritam za simboličko izračunavanje uopštenih inverza racionalnih matrica korišćenjem tehnike opisane u [87]. Ovaj metod pojednostavljuje problem simboličkog izračunavanja uopštenih inverza racionalnih matrica u proceduralnim programskim jezicima.

Kao što je ranije uočeno, rekurentne relacije (2.2.3)–(2.2.5) je potrebno izvršavati samo za  $j = 1, \text{rang}(A)$ . Ovaj metod generiše reprezentaciju potpunog ranga matrice  $A$ , gde je  $L$  bez nula kolona, a matrica  $D$  ne sadrži nula vrste i nula kolone. Prema tome, za datu racionalnu matricu  $A \in \mathbf{C}(s)_r^{m \times n} = \{X \in \mathbf{C}^{m \times n} \mid \text{rang}(X) = r\}$ , posmatrajmo dekompoziciju  $A = LDL^*$  potpunog ranga, pri čemu  $L \in \mathbf{C}(s)^{m \times r}$ ,  $l_{ij} = 0$  za  $i < j$ , a  $D \in \mathbf{C}(s)^{r \times r}$  je dijagonalna racionalna matrica.

Kako su matrice  $L(s)$ ,  $D(s)$  ranga  $r$ , one su oblika

$$L(s) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{2,1}(s) & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{r,1}(s) & l_{r,2}(s) & \cdots & 1 \\ l_{r+1,1}(s) & l_{r+1,2}(s) & \cdots & l_{r+1,r}(s) \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1}(s) & l_{n,2}(s) & \cdots & l_{n,r}(s) \end{bmatrix}, D(s) = \begin{bmatrix} d_1(s) & 0 & \cdots & 0 \\ 0 & d_2(s) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_r(s) \end{bmatrix}, \quad (2.4.2)$$

pri čemu su ne-nula elementi racionalnih matrica  $L(s)$  i  $D(s)$  oblika (2.2.2).

Upravo ćemo relacije (2.2.3)–(2.2.5) primenjivati za direktno izračunavanje koeficijenata matričnih elemenata u  $L(s)$  i  $D(s)$ . Kao i ranije, promenljive sa jednom crtom će označavati koeficijente polinoma u brojiocu, a promenljive sa dve crte koeficijente polinoma u imeniocu.

Jednačina (2.2.3) se u polinomijalnom obliku može zapisati kao:

$$\begin{aligned} f_{i,j}(s) &= \sum_{k=1}^{j-1} l_{ik}(s) l_{jk}^*(s) d_k(s) = \sum_{k=1}^{j-1} \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} s^t}{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} s^t} \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,j,k}^* s^t}{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,j,k}^* s^t} \frac{\sum_{t=0}^{\bar{d}_q} \bar{d}_{t,k} s^t}{\sum_{t=0}^{\bar{d}_q} \bar{d}_{t,k} s^t} \\ &= \sum_{k=1}^{j-1} \frac{\sum_{r_1=0}^{2\bar{l}_q+\bar{d}_q} \left( \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{l}_{r_3,i,k} \bar{l}_{r_1-r_2-r_3,j,k}^* \bar{d}_{r_2,k} \right) s^{r_1}}{\sum_{r_1=0}^{2\bar{l}_q+\bar{d}_q} \left( \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{l}_{r_3,i,k} \bar{l}_{r_1-r_2-r_3,j,k}^* \bar{d}_{r_2,k} \right) s^{r_1}}. \end{aligned}$$

Uvedimo sledeće oznake:

$$\begin{aligned}\bar{p}_{r_1,i,j,k} &= \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{l}_{r_3,i,k} \bar{l}_{r_1-r_2-r_3,j,k}^* \bar{d}_{r_2,k}, \quad 0 \leq r_1 \leq 2\bar{l}_q + \bar{d}_q, \\ \bar{\bar{p}}_{r_1,i,j,k} &= \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{\bar{l}}_{r_3,i,k} \bar{l}_{r_1-r_2-r_3,j,k}^* \bar{\bar{d}}_{r_2,k}, \quad 0 \leq r_1 \leq 2\bar{\bar{l}}_q + \bar{\bar{d}}_q.\end{aligned}$$

Tada je zadovoljeno

$$\begin{aligned}f_{i,j}(s) &= \sum_{k=1}^{j-1} \frac{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{r_1,i,j,k} s^{r_1}}{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{r_1,i,j,k} s^{r_1}} \\ &= \frac{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{r_1,i,j,1} s^{r_1}}{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{r_1,i,j,1} s^{r_1}} + \frac{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{r_1,i,j,2} s^{r_1}}{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{r_1,i,j,2} s^{r_1}} + \dots + \frac{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{r_1,i,j,j-1} s^{r_1}}{\sum_{r_1=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{r_1,i,j,j-1} s^{r_1}}.\end{aligned}$$

Sada odredimo najmanji zajednički sadržalac (NZS) polinoma u imeniciima i označimo ga na sledeći način:

$$Polinomijalni NZS \left( \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,1} s^k, \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,2} s^k, \dots, \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,j-1} s^k \right) = \sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k,i,j} s^k.$$

Takođe uvedimo oznake

$$q_{i,j,t}(s) = \frac{\sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k,i,j} s^k}{\sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{\bar{p}}_{k,i,j,t} s^k} = \sum_{k=0}^{\bar{f}_q - 2\bar{l}_q - \bar{d}_q} q_{k,i,j,t} s^k, \quad 1 \leq t \leq j-1, \quad i > j.$$

Neka je sada  $\bar{f}_q = 2\bar{l}_q + \bar{d}_q + \bar{\bar{f}}_q - 2\bar{\bar{l}}_q - \bar{\bar{d}}_q$ . Nastavljujući sa izračunavanjima elemenata  $f_{i,j}(s)$  imamo

$$\begin{aligned}f_{i,j}(s) &= \frac{\sum_{k=1}^{j-1} \sum_{k_1=0}^{\bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{p}_{k_1-k_2,i,j,k} q_{k_2,i,j,k} \right) s^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k,i,j} s^k} \\ &= \frac{\sum_{k_1=0}^{\bar{f}_q} \sum_{k_2=0}^{k_1} \sum_{k_3=0}^{j-1} \bar{p}_{k_1-k_2,i,j,k_3} q_{k_2,i,j,k_3} s^{k_1}}{\sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k,i,j} s^k} = \frac{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} s^k}{\sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k,i,j} s^k},\end{aligned}$$

gde je  $\bar{f}_{k,i,j} = \sum_{k_2=0}^k \sum_{k_3=0}^{j-1} \bar{p}_{k-k_2,i,j,k_3} q_{k_2,i,j,k_3}$ ,  $0 \leq k \leq \bar{f}_q$ .

Primetimo sada da za svako  $j = \overline{1, r}$ , na osnovu jednakosti (2.2.4), sledi sledećih nekoliko evaluacija

$$\begin{aligned} d_j(s) &= \frac{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k - \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k}{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k - \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k} = \frac{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k - \sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k}{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k} \\ &= \frac{\sum_{k_1=0}^{\bar{a}_q + \bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{a}_{k_1-k_2,j,j} \bar{f}_{k_2,j,j} \right) s^{k_1} - \sum_{k_1=0}^{\bar{a}_q + \bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{a}_{k_1-k_2,j,j} \bar{f}_{k_2,j,j} \right) s^{k_1}}{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,j,j} s^k \sum_{k=0}^{\bar{f}_q} \bar{f}_{k,j,j} s^k} \\ &= \frac{\sum_{k_1=0}^{\max(\bar{a}_q + \bar{f}_q, \bar{a}_q + \bar{f}_q)} \left( \sum_{k_2=0}^{k_1} (\bar{a}_{k_1-k_2,j,j} \bar{f}_{k_2,j,j} - \bar{a}_{k_1-k_2,j,j} \bar{f}_{k_2,j,j}) \right) s^{k_1}}{\sum_{k_1=0}^{\bar{a}_q + \bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{a}_{k_1-k_2,j,j} \bar{f}_{k_2,j,j} \right) s^{k_1}} = \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k}, \end{aligned}$$

pri čemu je

$$\begin{aligned} \bar{d}_q &= \max(\bar{a}_q + \bar{f}_q, \bar{a}_q + \bar{f}_q), \quad \bar{d}_q = \bar{a}_q + \bar{f}_q, \\ \bar{d}_{k,j} &= \sum_{i=0}^k (\bar{a}_{k-i,j,j} \bar{f}_{i,j,j} - \bar{a}_{k-i,j,j} \bar{f}_{i,j,j}), \quad 0 \leq k \leq \bar{d}_q, \\ \bar{d}_{k,j} &= \sum_{i=0}^k \bar{a}_{k-i,j,j} \bar{f}_{i,j,j}, \quad 0 \leq k \leq \bar{d}_q. \end{aligned}$$

Konačno, na osnovu jednakosti (2.2.5), zadovoljeno je

$$\begin{aligned} l_{i,j}(s) &= \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k} \left( \frac{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,i,j} s^k}{\sum_{k=0}^{\bar{a}_q} \bar{a}_{k,i,j} s^k} - \frac{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} s^k}{\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} s^k} \right) \\ &= \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k}{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k} \frac{\sum_{k_1=0}^{\max(\bar{a}_q + \bar{f}_q, \bar{a}_q + \bar{f}_q)} \left( \sum_{k_2=0}^{k_1} (\bar{a}_{k_1-k_2,i,j} \bar{f}_{k_2,i,j} - \bar{a}_{k_1-k_2,i,j} \bar{f}_{k_2,i,j}) \right) s^{k_1}}{\sum_{k_1=0}^{\bar{a}_q + \bar{f}_q} \left( \sum_{k_2=0}^{k_1} \bar{a}_{k_1-k_2,i,j} \bar{f}_{k_2,i,j} \right) s^{k_1}} \end{aligned}$$

$$\begin{aligned}
&= \frac{\bar{d}_q + \max(\bar{a}_q + \bar{\bar{f}}_q, \bar{\bar{a}}_q + \bar{f}_q)}{\sum_{k=0}^{\bar{l}_q} \bar{\bar{d}}_{k_1, j} \left( \sum_{k_2=0}^{k-k_1} (\bar{a}_{k-k_1-k_2, i, j} \bar{\bar{f}}_{k_2, i, j} - \bar{\bar{a}}_{k-k_1-k_2, i, j} \bar{f}_{k_2, i, j}) \right)} s^k \\
&= \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k, i, j} s^k}{\sum_{k=0}^{\bar{\bar{l}}_q} \bar{\bar{l}}_{k, i, j} s^k},
\end{aligned}$$

gde je

$$\begin{aligned}
\bar{l}_q &= \bar{\bar{d}}_q + \max(\bar{a}_q + \bar{\bar{f}}_q, \bar{\bar{a}}_q + \bar{f}_q), \quad \bar{\bar{l}}_q = \bar{d}_q + \bar{\bar{a}}_q + \bar{\bar{f}}_q \\
\bar{l}_{k, i, j} &= \sum_{k_1=0}^k \bar{\bar{d}}_{k_1, j} \left( \sum_{k_2=0}^{k-k_1} (\bar{a}_{k-k_1-k_2, i, j} \bar{\bar{f}}_{k_2, i, j} - \bar{\bar{a}}_{k-k_1-k_2, i, j} \bar{f}_{k_2, i, j}) \right), \quad 0 \leq k \leq \bar{l}_q, \\
\bar{\bar{l}}_{k, i, j} &= \sum_{k_1=0}^k \bar{d}_{k_1, j} \left( \sum_{k_2=0}^{k-k_1} \bar{\bar{a}}_{k-k_1-k_2, i, j} \bar{\bar{f}}_{k_2, i, j} \right), \quad 0 \leq k \leq \bar{\bar{l}}_q.
\end{aligned}$$

Sledeći algoritam za izračunavanje  $LDL^*$  dekompozicije date Hermitske polinomijalne matrice  $A(s) = \{a_{ij}(s)\}_{i,j=1}^n$  je uveden u radu [74].

---

**Algoritam 2.4**  $LDL^*$  faktorizacija potpunog ranga racionalne Hermitske matrice

---

**Ulaz:** Racionalna Hermitska matrica  $A(s) \in \mathbf{C}[s]_r^{n \times n}$ .

- 1: Inicijalizacija:  $d_1(s) := a_{11}(s)$  i  $l_{11}(s) := \frac{a_{11}(s)}{d_1(s)}$  za  $i = \overline{2, n}$ . Postaviti  $L_{ii}(s) := 1$ , za svako  $i = \overline{1, n}$ . Za  $j = \overline{2, r}$  izvršiti Korake 2, 3, 4.
- 2: Izračunati koeficijente polinoma  $f_{ij}(s)$  kroz sledeća četiri koraka, tj. za  $i = \overline{j, n}$  izvršiti Korak 2.1, Korak 2.2, Korak 2.3 i Korak 2.4.
  - 2.1: Za svako  $k = \overline{1, j-1}$  izvršiti sledeća izračunavanja:

$$\begin{aligned}
\bar{p}_{r_1, i, j, k} &= \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{l}_{r_3, i, k} \bar{l}_{r_1-r_2-r_3, j, k}^* \bar{d}_{r_2, k}, \quad 0 \leq r_1 \leq 2\bar{l}_q + \bar{d}_q, \\
\bar{\bar{p}}_{r_1, i, j, k} &= \sum_{r_2=0}^{r_1} \sum_{r_3=0}^{r_1-r_2} \bar{\bar{l}}_{r_3, i, k} \bar{\bar{l}}_{r_1-r_2-r_3, j, k}^* \bar{\bar{d}}_{r_2, k}, \quad 0 \leq r_1 \leq 2\bar{\bar{l}}_q + \bar{\bar{d}}_q.
\end{aligned}$$

2.2: Odrediti najmanji zajednički sadržalac polinoma imenilaca:

$$\sum_{k=0}^{\bar{f}_q} \bar{\bar{f}}_{k, i, j} s^k = PolinomijalniNZS \left( \sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{k, i, j, 1} s^k, \dots, \sum_{k=0}^{2\bar{\bar{l}}_q + \bar{\bar{d}}_q} \bar{\bar{p}}_{k, i, j, j-1} s^k \right).$$

2.3: Za svako  $t = \overline{1, j-1}$  i  $i = \overline{j+1, n}$  podeliti dobijeni polinom  $\sum_{k=0}^{\bar{f}_q} \bar{f}_{k,i,j} s^k$  sledećim polinomom  $\sum_{k=0}^{2\bar{l}_q + \bar{d}_q} \bar{p}_{k,i,j,t} s^k$ , i označiti količnik kao

$$q_{i,j,t}(s) = \sum_{k=0}^{\bar{f}_q - 2\bar{l}_q - \bar{d}_q} q_{k,i,j,t} s^k.$$

2.4: Uvesti smenu  $\bar{f}_q = 2\bar{l}_q + \bar{d}_q + \bar{\bar{f}}_q - 2\bar{l}_q - \bar{d}_q$ , i izračunati

$$\bar{f}_{k,i,j} = \sum_{k_2=0}^k \sum_{k_3=0}^{j-1} \bar{p}_{k-k_2, i, j, k_3} q_{k_2, i, j, k_3}, \quad 0 \leq k \leq \bar{f}_q.$$

3: Uvesti sledeće smene:  $\bar{d}_q = \max(\bar{a}_q + \bar{\bar{f}}_q, \bar{\bar{a}}_q + \bar{f}_q)$  i  $\bar{\bar{d}}_q = \bar{\bar{a}}_q + \bar{\bar{f}}_q$ , i izračunati

$$\begin{aligned} \bar{d}_{k,j} &= \sum_{i=0}^k (\bar{a}_{k-i, j, j} \bar{\bar{f}}_{i, j, j} - \bar{\bar{a}}_{k-i, j, j} \bar{f}_{i, j, j}), \quad 0 \leq k \leq \bar{d}_q, \\ \bar{\bar{d}}_{k,j} &= \sum_{i=0}^k \bar{\bar{a}}_{k-i, j, j} \bar{\bar{f}}_{i, j, j}, \quad 0 \leq k \leq \bar{\bar{d}}_q. \end{aligned}$$

4: Uvesti smene  $\bar{l}_q = \bar{\bar{d}}_q + \max(\bar{a}_q + \bar{\bar{f}}_q, \bar{\bar{a}}_q + \bar{f}_q)$  i  $\bar{\bar{l}}_q = \bar{d}_q + \bar{\bar{a}}_q + \bar{\bar{f}}_q$ , a za svako  $i = \overline{j+1, n}$  izračunati sledeće:

$$\begin{aligned} \bar{l}_{k,i,j} &= \sum_{k_1=0}^k \bar{\bar{d}}_{k_1,j} \left( \sum_{k_2=0}^{k-k_1} (\bar{a}_{k-k_1-k_2, i, j} \bar{\bar{f}}_{k_2, i, j} - \bar{\bar{a}}_{k-k_1-k_2, i, j} \bar{f}_{k_2, i, j}) \right), \quad 0 \leq k \leq \bar{l}_q, \\ \bar{\bar{l}}_{k,i,j} &= \sum_{k_1=0}^k \bar{d}_{k_1,j} \left( \sum_{k_2=0}^{k-k_1} \bar{\bar{a}}_{k-k_1-k_2, i, j} \bar{\bar{f}}_{k_2, i, j} \right), \quad 0 \leq k \leq \bar{\bar{l}}_q. \end{aligned}$$

5: **Rezultat:**

$$d_j(s) = \frac{\sum_{k=0}^{\bar{d}_q} \bar{d}_{k,j} s^k}{\sum_{k=0}^{\bar{\bar{d}}_q} \bar{\bar{d}}_{k,j} s^k}, \quad l_{i,j}(s) = \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k,i,j} s^k}{\sum_{k=0}^{\bar{\bar{l}}_q} \bar{\bar{l}}_{k,i,j} s^k}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq r.$$

Implementacija Algoritma 2.4 je veoma slična implementaciji algoritma 2.2. Vremena izračunavanja kod primene Algoritma 2.4 se ne razliku od njegove polinomijalne verzije na skupu polinomijalnih test matrica. S obzirom da je složenost algoritma  $LDL^*$  dekompozicije u najgorem slučaju  $O(n^3)$ , složenost Algoritma 2.4 će biti aproksimativno  $O(n^3 m^2)$ , gde je  $m$  maksimalni eksponent polinoma dobijenih algoritmom.

## 2.5 Izračunavanje Moore-Penrose-ovog inverza racionalne matrice

Naša motivacija je iskoristiti dekompoziciju bez korena za dobijanje algoritma za izračunavanje Moore-Penrose-ovog inverza racionalne matrice. Takav metod, prezentovan u radu [74], je veoma primenljiv u proceduralnim programskim jezicima, kao i u simboličkim programskim paketima, kao što je MATHEMATICA. Sledeća posledica Teoreme 2.1.6 za slučaj racionalnih matrica je uvedena u radu [74].

**Teorema 2.5.1.** [74] *Posmatrajmo racionalnu matricu  $A \in \mathbf{C}(x)_r^{m \times n}$  i  $LDL^*$  dekompoziciju potpunog ranga matrice  $(A^*A)^*(A^*A)$ , pri čemu je proizvoljan element inverzne matrice  $N = (L^*LDL^*L)^{-1} \in \mathbf{C}(x)^{r \times r}$  označen kao*

$$n_{ij}(x) = \left( \sum_{t=0}^{\bar{n}_q} \bar{n}_{t,i,j} x^t \right) / \left( \sum_{t=0}^{\bar{\bar{n}}_q} \bar{\bar{n}}_{t,i,j} x^t \right). \quad (2.5.1)$$

Neka su uvedene sledeće oznake za  $\mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}}, \kappa = \overline{1, m}, \lambda = \overline{1, n}$ :

$$\bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \bar{p}_{t_1,i,k,l,\mu} \bar{\alpha}_{t-t_1,j,\mu,\kappa,\lambda}, \quad 0 \leq t \leq \bar{b}_q = 2\bar{l}_q + \bar{n}_q + 3\bar{a}_q, \quad (2.5.2)$$

$$\bar{\bar{\beta}}_{t,i,j,k,l,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \bar{\bar{p}}_{t_1,i,k,l,\mu} \bar{\bar{\alpha}}_{t-t_1,j,\mu,\kappa,\lambda}, \quad 0 \leq t \leq \bar{\bar{b}}_q = 2\bar{\bar{l}}_q + \bar{\bar{n}}_q + 3\bar{\bar{a}}_q, \quad (2.5.3)$$

$$\bar{p}_{t,i,k,l,\mu} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{l}_{t_1,i,k} \bar{n}_{t-t_1-t_2,k,l} \bar{l}_{t_2,\mu,l}^*, \quad 0 \leq t \leq 2\bar{l}_q + \bar{n}_q \quad (2.5.4)$$

$$\bar{\bar{p}}_{t,i,k,l,\mu} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{\bar{l}}_{t_1,i,k} \bar{\bar{n}}_{t-t_1-t_2,k,l} \bar{\bar{l}}_{t_2,\mu,l}^*, \quad 0 \leq t \leq 2\bar{\bar{l}}_q + \bar{\bar{n}}_q, \quad (2.5.5)$$

$$\bar{\alpha}_{t,j,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{a}_{t_1,\kappa,\lambda} \bar{a}_{t-t_1-t_2,\kappa,\mu}^* \bar{a}_{t_2,j,\lambda}^*, \quad 0 \leq t \leq 3\bar{a}_q, \quad (2.5.6)$$

$$\alpha_{t,j,\mu,\kappa,\lambda} = \sum_{t_1=0}^t \sum_{t_2=0}^{t-t_1} \bar{\bar{a}}_{t_1,\kappa,\lambda} \bar{\bar{a}}_{t-t_1-t_2,\kappa,\mu}^* \bar{\bar{a}}_{t_2,j,\lambda}^*, \quad 0 \leq t \leq 3\bar{\bar{a}}_q. \quad (2.5.7)$$

Tada je proizvoljni  $(i, j)$ -ti element Moore-Penrose-ovog inverza matrice  $A$  dat sa

$$A_{ij}^\dagger(x) = \frac{\Phi_{ij}(x)}{\Psi_{ij}(x)},$$

pri čemu je

$$\Phi_{ij}(x) = \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t=0}^{\Psi_q - \bar{b}_q + \bar{b}_q} \left( \sum_{t_1=0}^t \bar{\beta}_{t_1,i,j,k,l,\mu,\kappa,\lambda} \gamma_{t-t_1,i,j,k,l,\mu,\kappa,\lambda} \right) x^t, \quad (2.5.8)$$

$$\Psi_{ij}(x) = \text{PolinomijalniNZS} \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \mid k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}}, \mu = \overline{1, n}, \kappa = \overline{1, m}, \lambda = \overline{1, n} \right\}, \quad (2.5.9)$$

dok su  $\gamma_{t,i,j,k,l,\mu,\kappa,\lambda}$ ,  $0 \leq t \leq \Psi_q - \bar{b}_q$ , koeficijenti polinoma

$$\Gamma_{i,j,k,l,\mu,\kappa,\lambda}(x) = \frac{\Psi_{ij}(x)}{\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t},$$

a  $\Psi_q$  predstavlja maksimalni eksponent polinoma  $\Psi_{ij}(x)$ .

**Dokaz.** Pretpostavimo, bez umanjenja opštosti, da su elementi racionalne matrice  $A$  dati u formi (2.4.1). Kako  $LDL^*$  predstavlja dekompoziciju potpunog ranga matrice  $(A^*A)^*(A^*A) \in \mathbf{C}(x)_r^{n \times n}$ , sledi da su  $L \in \mathbf{C}(x)^{n \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$  matrice sa elementima datim u obliku (2.2.2).

Na osnovu prvog tvrđenja Teoreme 2.1.6, proizvoljni  $(i, j)$ -ti element matrice  $A^\dagger$  se može izračunati kao:

$$\begin{aligned} A_{ij}^\dagger &= \sum_{\mu=1}^n (LN L^*)_{i\mu} ((A^*A)^* A^*)_{\mu j} \\ &= \sum_{\mu=1}^n \left( \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} l_{ik} n_{kl} l_{\mu l}^* \right) \left( \sum_{\lambda=1}^n \sum_{\kappa=1}^m a_{\kappa\lambda} a_{\kappa\mu}^* a_{j\lambda}^* \right) \\ &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m l_{ik} n_{kl} l_{\mu l}^* a_{\kappa\lambda} a_{\kappa\mu}^* a_{j\lambda}^*. \end{aligned}$$

Uzimanjem racionalne forme svih elemenata matrica,  $(i, j)$ -ti element Moore-Penrose-ovog inverza od  $A$  se može odrediti.

$$\begin{aligned} A_{ij}^\dagger(x) &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,i,k} x^t}{\sum_{t=0}^{\bar{l}_q} \bar{\bar{l}}_{t,i,k} x^t} \cdot \frac{\sum_{t=0}^{\bar{n}_q} \bar{n}_{t,k,l} x^t}{\sum_{t=0}^{\bar{n}_q} \bar{\bar{n}}_{t,k,l} x^t} \cdot \frac{\sum_{t=0}^{\bar{l}_q} \bar{l}_{t,\mu,l}^* x^t}{\sum_{t=0}^{\bar{l}_q} \bar{\bar{l}}_{t,\mu,l}^* x^t} \\ &\quad \cdot \frac{\sum_{t=0}^{\bar{a}_q} \bar{a}_{t,\kappa,\lambda} x^t}{\sum_{t=0}^{\bar{a}_q} \bar{\bar{a}}_{t,\kappa,\lambda} x^t} \cdot \frac{\sum_{t=0}^{\bar{a}_q} \bar{a}_{t,\kappa,\mu}^* x^t}{\sum_{t=0}^{\bar{a}_q} \bar{\bar{a}}_{t,\kappa,\mu}^* x^t} \cdot \frac{\sum_{t=0}^{\bar{a}_q} \bar{a}_{t,j,\lambda}^* x^t}{\sum_{t=0}^{\bar{a}_q} \bar{\bar{a}}_{t,j,\lambda}^* x^t} \\ &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu, r\}} \sum_{k=1}^{\min\{i, r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{2\bar{l}_q + \bar{n}_q} \bar{p}_{t,i,k,l,\mu} x^t \sum_{t=0}^{3\bar{a}_q} \bar{\alpha}_{t,j,\mu,\kappa,\lambda} x^t}{\sum_{t=0}^{2\bar{l}_q + \bar{n}_q} \bar{\bar{p}}_{t,i,k,l,\mu} x^t \sum_{t=0}^{3\bar{a}_q} \bar{\bar{\alpha}}_{t,j,\mu,\kappa,\lambda} x^t} \end{aligned}$$

$$A_{ij}^\dagger(x) = \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu,r\}} \sum_{k=1}^{\min\{i,r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \frac{\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t}{\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t}.$$

Pritom,  $A_{ij}^\dagger(x) = \frac{\Phi_{ij}(x)}{\Psi_{ij}(x)}$  je validno, gde je

$$\begin{aligned} \Psi_{ij}(x) &= NZS \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \mid \mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}}, \kappa = \overline{1, m}, \lambda = \overline{1, n} \right\}, \\ \Phi_{ij}(x) &= \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu,r\}} \sum_{k=1}^{\min\{i,r\}} \left( \Gamma_{i,j,k,l,\mu,\kappa,\lambda}(x) \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \right), \end{aligned}$$

a polinom  $\Gamma_{i,j,k,l,\mu,\kappa,\lambda}(x)$  je dat u obliku

$$\Psi_{ij}(x) / \left( \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \right) = \sum_{t=0}^{\Psi_q - \bar{b}_q} \gamma_{t,i,j,k,l,\mu,\kappa,\lambda} x^t.$$

Konačno, sledeća forma

$$\Phi_{ij}(x) = \sum_{\mu=1}^n \sum_{l=1}^{\min\{\mu,r\}} \sum_{k=1}^{\min\{i,r\}} \sum_{\lambda=1}^n \sum_{\kappa=1}^m \sum_{t=0}^{\Psi_q - \bar{b}_q + \bar{b}_q} \left( \sum_{t_1=0}^t \bar{\beta}_{t_1,i,j,k,l,\mu,\kappa,\lambda} \gamma_{t-t_1,i,j,k,l,\mu,\kappa,\lambda} \right) x^t,$$

je jednaka formi (2.5.8).  $\square$

**Napomena 2.5.1.** Primetno je da se slična teorema može postaviti posmatranjem drugog tvrđenja Teoreme 2.1.6.

---

**Algoritam 2.5** Izračunavanje MP-inverza primenom  $LDL^*$  faktorizacije potpunog ranga

**Ulaz:** Racionalna matrica  $A(x) \in \mathbf{C}(x)_r^{m \times n}$  (sa elementima u obliku (2.4.1)).

- 1: Primeniti Algoritam 2.4 za izračunavanje  $LDL^*$  dekompozicije potpunog ranga matrice  $(A^* A)^*(A^* A)$ , gde su  $L \in \mathbf{C}(x)^{n \times r}$  i  $D \in \mathbf{C}(x)^{r \times r}$  matrice sa elementima u obliku (2.2.2).
- 2: Predstaviti racionalnu matricu  $R = L^* LDL^* L$  u obliku:  $R(x) = \frac{1}{p(x)} P(x)$ , pri čemu je  $p(x)$  neki polinom (jednak NZS-u polinoma imenilaca u  $R$ ) a  $P(x)$  je polinomijalna matrica.
- 3: Izračunati inverz polinomijalne matrice  $P$  koristeći Algoritam 3.2 iz rada [98]. Generisati inverznu matricu  $N = R^{-1} = p(x)P^{-1}$  i transformisati je u oblik (2.5.1).
- 4: Za svako  $i = \overline{1, m}, j = \overline{1, n}, \mu = \overline{1, n}, k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}}, \kappa = \overline{1, m}, \lambda = \overline{1, n}$  izvršiti evaluacije (2.5.2)–(2.5.7).
- 5: Za  $i = \overline{1, m}, j = \overline{1, n}$  izračunati polinome imenilaca elemenata  $A_{i,j}^\dagger$  na sledeći način

$$\begin{aligned} \Psi_{ij}(x) &= NZS \left\{ \sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,j,k,l,\mu,\kappa,\lambda} x^t \mid \right. \\ &\quad \left. k = \overline{1, \min\{i, r\}}, l = \overline{1, \min\{\mu, r\}}, \mu = \overline{1, n}, \kappa = \overline{1, m}, \lambda = \overline{1, n} \right\}, \end{aligned}$$

- i označiti ih sa  $\Psi_i(x) = \sum_{t=0}^{\Psi_q} \Psi_{t,i} x^t$ .
- 6: Za svako  $i = \overline{1,m}$ ,  $\mu = \overline{1,n}$ ,  $k = \overline{1,\min\{i,r\}}$ ,  $l = \overline{1,\min\{\mu,r\}}$  odrediti sledeće polinome:  $\Psi_i(x)/\left(\sum_{t=0}^{\bar{b}_q} \bar{\beta}_{t,i,k,l,\mu} x^t\right)$ , i označiti ih sa  $\Gamma_{i,l,k,\mu}(x) = \sum_{t=0}^{\Psi_q - \bar{b}_q} \gamma_{t,i,k,l,\mu} x^t$ .
- 7: Za  $i = \overline{1,m}$ ,  $j = \overline{1,n}$  izračunati polinome brojilaca:

$$\Phi_{ij}(x) = \sum_{t=0}^{\Psi_q - \bar{b}_q + \bar{b}_q} \left( \sum_{\mu=1}^n \sum_{l=1}^m \sum_{k=1}^{\min\{\mu,r\}} \sum_{\lambda=1}^{\min\{i,r\}} \sum_{\kappa=1}^m \sum_{t_1=0}^t \bar{\beta}_{t_1,i,j,k,l,\mu,\kappa,\lambda} \gamma_{t-t_1,i,k,l,\mu} \right) x^t. \quad (2.5.10)$$

- 8: Za  $i = \overline{1,m}$ ,  $j = \overline{1,n}$  postaviti  $(i,j)$ -ti element uopštenog inverza  $A^\dagger$  na vrednost  $\Phi_{ij}(x)/\Psi_i(x)$ .
- 

Još jednom navedimo da se  $(i,j)$ -ti element proizvoda matrica  $(A^*A)^*(A^*A)$  može izračunati kao

$$\sum_{l=1}^n \sum_{k=1}^m \sum_{k'=1}^m a_{kl}^* a_{ki} a_{k'l}^* a_{k'j}.$$

Dakle, ovi polinomi će predstavljati ulaz Algoritma 2.4, korišćenog u Koraku 1 našeg algoritma. Takođe, jednostavno je odrediti i ulaz za Algoritam 3.2 iz [98] u Koraku 3 algoritma 2.5.

### 2.5.1 Numerički primeri i rezultati testiranja

Ovde ćemo ispitati Algoritam 2.4 i Algoritam 2.5 i uporediti neke različite implementacije, kroz narednih nekoliko primera. Na kraju ćemo uporediti vremena izvršenja različitih algoritama i implementacija na skupu nasumičnih test matrica.

**Primer 2.5.1.** Uočimo kvadratnu simetričnu racionalnu matricu

$$A(x) = \begin{bmatrix} \frac{1+x^2}{-1+x} & \frac{x}{\frac{3+x^2}{-2x+x^2}} & 2+x \\ \frac{x}{3+x^2} & \frac{-1+x}{-2x+x^2} & 3+x \\ 2+x & 3+x & -1+x \end{bmatrix}.$$

Primenom Algoritma 2.4 za  $j = 2$  dobijamo

$$f_{22}(x) = \frac{(-1+x)x^2}{(1+x^2)(3+x^2)^2}, \quad f_{32} = \frac{(-1+x)x(2+x)}{3+4x^2+x^4}.$$

Za  $j = 3$  se dobijaju sledeći "intermediate" rezultati:

$$f_{33}(x) = \frac{36 - 198x - 102x^2 - 192x^3 - 23x^4 - 54x^5 + 5x^6 + 8x^7 + 3x^8 + 4x^9 + x^{10}}{(-1+x)(9+15x^2+2x^3+6x^4+x^6)}.$$

Nakon tih velikih rezultata, konačan rezultat je nešto manje veličine nakon izvršene simplifikacije:

$$L(x) = \begin{bmatrix} 1 & 0 & 0 \\ \frac{(-1+x)x}{3+4x^2+x^4} & 1 & 0 \\ \frac{(-1+x)(2+x)}{1+x^2} & \frac{(-2+x)x(3+x^2)(9+5x+11x^2+3x^3+3x^4+x^5)}{-9+9x-15x^2+13x^3-4x^4+6x^5-x^6+x^7} & 1 \end{bmatrix},$$

$$D(x) = \begin{bmatrix} \frac{1+x^2}{-1+x} & 0 & 0 \\ 0 & \frac{(-1+x)(9+15x^2+2x^3+6x^4+x^6)}{(-2+x)x(1+x^2)(3+x^2)^2} & 0 \\ 0 & 0 & \frac{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}}{9-9x+15x^2-13x^3+4x^4-6x^5+x^6-x^7} \end{bmatrix}.$$

Sledeći Moore-Penrose-ov inverz matrice  $A$  je dobijen primenom Algoritma 2.5 na ulaznu matricu  $A$ :

$$A^\dagger = \begin{bmatrix} \frac{(3+x^2)^2(1+15x-12x^2-8x^3+3x^4+x^5)}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{x(-108+66x+6x^2+16x^3+29x^4-8x^5+2x^6-2x^7-x^8)}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{18-27x-6x^2+12x^3-4x^4+7x^5}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{x(-108+66x+6x^2+16x^3+29x^4-8x^5+2x^6-2x^7-x^8)}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{x(3+x^2)^2(6-x-9x^2+4x^3)}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{x(-54-3x-69x^2+14x^3-26x^4+8x^5+x^7+x^8)}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \\ \frac{9-9x+15x^2-13x^3+4x^4-6x^5+x^6-x^7}{27-180x-126x^2-164x^3-40x^4-44x^5-2x^6+10x^7+2x^8+4x^9+x^{10}} \end{bmatrix}.$$

**Primer 2.5.2.** Posmatrajmo matricu  $A(x)$  iz Primera 2.2.2 kao racionalnu matricu. Da bismo izračunali elemente Moore-Penrose-ovog inverza matrice  $A$ ,  $LDL^*$  dekompoziciju matrice  $(A^*A)^2$  treba odrediti algoritmom 2.4, koji je zahtevan u Koraku 1 Algoritam 2.5. Prema tome, primenom Algoritam 2.5 na datu matricu  $A$ , dobijen je sledeći Moore-Penrose-ov inverz matrice  $A$ :

$$A^\dagger = \begin{bmatrix} \frac{-1+x}{-85-83x-2x^2+x^3} & \frac{9+17x}{85+83x+2x^2-x^3} & \frac{3+10x}{85+83x+2x^2-x^3} & \frac{26(1+x)}{-85-83x-2x^2+x^3} \\ \frac{9+17x}{85+83x+2x^2-x^3} & \frac{4+36x-2x^2+3x^3}{-85-83x-2x^2+x^3} & \frac{27-2x+2x^2-2x^3}{85+83x+2x^2-x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} \\ \frac{3+10x}{85+83x+2x^2-x^3} & \frac{27-2x+2x^2-2x^3}{-85-83x-2x^2+x^3} & \frac{9-16x+x^2-x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} \\ \frac{26(1+x)}{-85-83x-2x^2+x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} & \frac{4-4x+8x^3}{-85-83x-2x^2+x^3} \end{bmatrix}.$$

**Primer 2.5.3.** Uporedimo sada neka vremena izračunavanja za različite implementacije algoritma 2.4 u MATHEMATICA i JAVA.

Za potrebe testiranja, u radu [74], generisali smo skup  $n \times n$  racionalnih matrica gusotine 1, sa slučajno odabranim nenula koeficijentima iz intervala  $[-10, 10]$ , pri čemu su svi polinomi u brojicima i imeniocima stepena  $d$ . Tabela 2.5.1 ilustruje razlike među vremenima dobijenim različitim implementacijama algoritma 2.4 i osnovne  $LDL^*$  dekompozicije racionalnih matrica. Testiranja su pokazala da je MATHEMATICA implementacija algoritma 2.4 dosta efikasnija u odnosu na druge implementacije. Glavni razlog su česte simplifikacije izvršene u MATHEMATICA. Implementacija osnovnog algoritma (2.2.3)-(2.2.5) je ponovo označna kao MATH. osnovni algoritam i očigledno je manje efikasna nego implementacija algoritma 2.4 u JAVA (Java algoritam 2.4) za manje dimenzije. Za slučaj većih dimenzija ( $n = 10$ ) implementacija u Java je dosta sporija nego implementacija u MATHEMATICA.

$n$	5					10				
$d$	10	25	50	100	200	2	5	10	25	50
MATH.osnovni algoritam	0.27	0.80	1.86	5.86	19.91	0.21	0.48	1.14	3.36	8.81
MATH. algoritam 2.4	0.04	0.09	0.18	0.44	1.12	0.09	0.15	0.23	0.54	1.24
Java algoritam 2.4	0.03	0.17	0.67	2.95	12.42	0.24	0.89	4.35	31.42	118.46

Tabela 2.5.1. Vremena izvršavanja u sekundama dobijena različitim implementacijama algoritma 2.4.

Test matrica	$S_{10}$	$S_{50}$	$S_{100}$	$S_{150}$	$F_{10}$	$F_{50}$	$F_{100}$	$F_{150}$
Lev.-Faddeev [41]	0.11	2.51	44.47	-	0.14	2.77	42.84	-
Partitioning [87]	0.04	1.18	8.20	74.55	0.01	0.45	2.81	15.4
LDLGInverse [77]	0.02	0.88	4.55	22.41	0.01	1.87	12.74	-
PseudoInverse [113]	0.14	2.77	41.48	-	0.08	2.44	44.54	-
<b>Algoritam 2.5</b>	0.01	0.48	2.88	17.11	0.02	0.80	7.24	40.01

Tabela 2.5.2. Prosečna vremena izračunavanja Moore-Penrose-ovog inverza, u sekundama, dobijena različitim algoritmima.

**Primer 2.5.4.** Odredimo dalje efikasnost Algoritma 2.5. Uporedili smo nekoliko različitih algoritama za izračunavanje Moore-Penrose-ovog inverza. U Tabeli 2.5.2 su data prosečna vremena izračunavanja, dobijena primenom različitih algoritama na dva skupa test matrica iz [117] (posmatrajući parcijalni slučaj  $x = 1$ ).

Prva vrsta u Tabeli 2.5.2 sadrži nazine test matrica generisanih u [117]. Vremena izračunavanja u poslednjoj vrsti su dobijena primenom Algoritma 2.5. Očito, naš algoritam je brz, ali nije i najefikasniji algoritam za svaki skup test matrica. Algoritam sa pregrađivanjem iz rada [87] je bolja opcija kod drugog skupa test matrica. Takođe, jasno je da je Algoritam 2.5 imao bolje performanse u odnosu na LDLGInverse iz [77], iz kojeg je izведен.

## 2.6 Izračunavanje $A_{T,S}^{(2)}$ inverza LDL\* dekompozicijom

Mnoge reprezentacije potpunog ranga različitih tipova generalisanih inverza su napravljene, bilo za zadati rang ili zadati opseg ili jezgro. Radi kompletnosti, navešćemo sledeću reprezentaciju potpunog ranga spoljnih inverza definisanog jezgra i slike (eng. range, null space), koju su uveli Sheng i Chen [70].

**Lema 2.6.1.** [70] Neka je data matrica  $A \in \mathbb{C}_r^{m \times n}$  i podprostori  $T \subseteq \mathbb{C}^n$  dimenzije  $s \leq r$  i  $S \subseteq \mathbb{C}^m$  dimenzije  $m - s$ . Prepostavimo da matrica  $M \in \mathbb{C}^{n \times m}$  zadovoljava  $\mathcal{R}(M) = T, \mathcal{N}(M) = S$  i da ima dekompoziciju potpunog ranga  $M = FG$ . Ako matrica  $A$  ima  $\{2\}$ -inverz  $A_{T,S}^{(2)}$ , tada važe sledeća tvrdjenja:

- (1)  $GAF$  je invertibilna matrica;
- (2)  $A_{T,S}^{(2)} = F(GAF)^{-1}G = A_{\mathcal{R}(F), \mathcal{N}(G)}^{(2)}$ .

U radu [78] data je teorema kojom je određena reprezentacija potpunog ranga spoljnih inverza, u istom smislu kao u Lemi 2.6.1. Sledеće tvrđenje važi za racionalne matrice i zasnovano je na  $LDL^*$  dekompozicije potpunog ranga proizvoljne Hermitske matrice  $M$ , što se može izvesti na sličan način opisan u Algoritmu 2.4.

**Teorema 2.6.2.** [78] Neka je data kvadratna racionalna matrica  $A \in \mathbb{C}(x)_r^{n \times n}$  normalnog ranga  $r$ . Ako je  $LDL^*$  dekompozicija potpunog ranga proizvoljne Hermitske matrice  $M \in \mathbb{C}(x)_s^{n \times n}$  ranga  $s \leq r$ , gde je  $L \in \mathbb{C}(x)_s^{n \times s}$ , a  $D \in \mathbb{C}(x)_s^{s \times s}$  je dijagonalna matrica, definišimo sledeći skup

$$\mathcal{D}_{A,M} = \{x \in \mathbb{C} \mid nrang(M) = rang(L^*(x)A(x)L(x)) = s\}. \quad (2.6.1)$$

Ako je zadovoljen uslov

$$nrang(M) = nrang(RAQ) = s, \quad (2.6.2)$$

sledeće tvrđenje je validno na skupu  $\mathcal{D}_{A,M}$ :

$$A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} = L(L^*AL)^{-1}L^* = A_{\mathcal{R}(M),\mathcal{N}(M)}^{(2)}. \quad (2.6.3)$$

**Dokaz.** S obzirom da izraz

$$M = LDL^* = (LD)(L^*), \quad (2.6.4)$$

predstavlja dekompoziciju potpunog ranga matrice  $M$  na skupu  $\mathcal{D}_{A,M}$ , tada prvi identitet u (2.6.3) važi na osnovu Leme 2.6.1 i jednačine

$$LD(L^*ALD)^{-1}L^* = L(L^*AL)^{-1}L^* = A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)}.$$

Druga jednačina  $A_{\mathcal{R}(M),\mathcal{N}(M)}^{(2)} = A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)}$  je očigledno zadovoljena na skupu  $\mathcal{D}_{A,M}$ , na osnovu jednačine (2.6.4).  $\square$

Sada navodimo algoritam za izračunavanje  $A_{T,S}^{(2)}$  inverza date matrice  $A$ , uveden u [78]

---

**Algoritam 2.6** Izračunavanje  $A_{T,S}^{(2)}$  inverza matrice  $A$  na osnovu  $LDL^*$  dekompozicije potpunog ranga matrice  $M$ . (**Algoritam LDLATS**)

---

**Ulaz:** Matrica  $A \in \mathbb{C}(x)_r^{n \times n}$  normalnog ranga  $r$ .

- 1: Odabratи proizvoljnu polinomijalnu Hermitsku matricu  $M \in \mathbb{C}(x)_s^{n \times n}$  normalnog ranga  $s \leq r$ .
  - 2: Generisati  $LDL^*$  dekompoziciju potpunog ranga polinomijalne matrice  $M$  primenom Algoritma 2.4.
  - 3: **Rezultat:** Generalisani inverz matrice  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)}$  kao proizvod matrica dat u jednačini (2.6.3).
- 

Sledeće tvrđenje iz [78] pruža praktični kriterijum za konstruisanje Hermitske matrice  $M \in \mathbb{C}(x)_s^{n \times n}$ , radi izračunavanja nekoliko tipova uopštenih inverza matrice  $A$ .

**Posledica 2.6.3.** [78] Za datu matricu  $A \in \mathbb{C}(x)_r^{n \times n}$  normalnog ranga  $r$  i proizvoljnu Hermitsku matricu  $M \in \mathbb{C}(x)_s^{n \times n}$ , gde je  $s \leq r$ , sledeća tvrdjenja su validna na skupu  $\mathcal{D}_{A,M}$ : za  $M = A^*$  identitet  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} = A^\dagger$  je zadovoljen; za  $M = A^\sharp$  identitet  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} = A_{M,N}^\dagger$  je zadovoljen; za  $M = A$  važi identitet  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} = A^\#$ ; za  $M = A^k$ , pri čemu je  $k \geq \text{ind}(A)$ , važi identitet  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} = A^\dagger$ .

**Dokaz.** Sledi na osnovu Teoreme 2.6.2 i identiteta (1.1.1) i (1.1.2).  $\square$

Na osnovu ove posledice, širok spektar generalisanih inverza matrice  $A \in \mathbb{C}(x)_r^{n \times n}$  je moguće odrediti, odabirom odgovarajuće Hermitske matrice  $M$  i izračunavanjem njene dekompozicije potpunog ranga.

U radu [78] uvedena je sledeća posledica Teoreme 2.6.2 za slučaj polinomijalnih matrica  $A$  i  $M$ .

**Teorema 2.6.4.** [78] Neka je  $A \in \mathbb{C}(x)_r^{n \times n}$  polinomijalna matrica normalnog ranga  $r$ . Neka je  $LDL^*$  dekompozicija potpunog ranga proizvoljne Hermitske polinomijalne matrice  $M \in \mathbb{C}(x)_s^{n \times n}$  normalnog ranga  $s \leq r$ , gde su  $L \in \mathbb{C}(x)_s^{n \times s}$  i  $D \in \mathbb{C}(x)_s^{s \times s}$  matrice oblika (2.4.2) sa elementima formi (2.2.2). Označimo sa  $\mathcal{D}_{A,M}$  skup kao u (2.6.1) i proizvoljni  $(i,j)$ -ti element inverzne matrice  $N = (L^*AL)^{-1}$  na sledeći način:

$$n_{i,j}(x) = \sum_{k=0}^{\bar{n}_q} \bar{n}_{k,i,j} x^k / \sum_{k=0}^{\bar{n}_q} \bar{\bar{n}}_{k,i,j} x^k, \quad (2.6.5)$$

Ako je uslov (2.6.2) zadovoljen, tada se proizvoljni  $(i,j)$ -ti element uopštenog inverza  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)}$  može izračunati kao

$$\left( A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} \right)_{ij}(x) = \frac{\bar{\Sigma}_{i,j}(x)}{\bar{\bar{\Sigma}}_{i,j}(x)},$$

za  $x \in \mathcal{D}_{A,M}$ , gde su  $\bar{\Sigma}_{i,j}(x)$  i  $\bar{\bar{\Sigma}}_{i,j}(x)$  polinomi oblika

$$\bar{\Sigma}_{i,j}(x) = \sum_{t=0}^{\bar{\Sigma}_q - \bar{\bar{\gamma}}_q + \bar{\gamma}_q} \left( \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{\min\{i,s\}} \sum_{t_1=0}^t \bar{\gamma}_{t_1,i,j,k,l} \sigma_{t-t_1,i,j,k,l} \right) x^t, \quad (2.6.6)$$

$$\bar{\bar{\Sigma}}_{i,j}(x) = \text{NZS} \left\{ \sum_{t=0}^{\bar{\bar{\gamma}}_q} \bar{\bar{\gamma}}_{t,i,j,k,l} x^t \mid k = \overline{1, \min\{j, s\}}, l = \overline{1, \min\{i, s\}} \right\} \quad (2.6.7)$$

$$= \sum_{t=0}^{\bar{\bar{\gamma}}_q} \bar{\bar{\sigma}}_{t,i,j} x^t, \quad (2.6.8)$$

gde su za  $k = \overline{1, \min\{j, s\}}$ ,  $l = \overline{1, \min\{i, s\}}$ , vrednosti  $\sigma_{t,i,j,k,l}$ ,  $0 \leq t \leq \bar{\Sigma}_q - \bar{\bar{\gamma}}_q$  koeficijenti polinoma

$$\Sigma_{i,j,k,l}(x) = \frac{\bar{\Sigma}_{i,j}(x)}{\sum_{t=0}^{\bar{\bar{\gamma}}_q} \bar{\bar{\gamma}}_{t,i,j,k,l} x^t},$$

pri čemu su korišćene oznake:

$$\bar{\gamma}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{l}_{t_2,i,l} \bar{n}_{t_1-t_2-t_3,l,k} \bar{l}_{t_3,j,k}^*, \quad 0 \leq t \leq \bar{\gamma}_q = 2\bar{l}_q + \bar{n}_q, \quad (2.6.9)$$

$$\bar{\bar{\gamma}}_{t,i,j,k,l} = \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{l}}_{t_2,i,l} \bar{\bar{n}}_{t_1-t_2-t_3,l,k} \bar{\bar{l}}_{t_3,j,k}^*, \quad 0 \leq t \leq \bar{\bar{\gamma}}_q = 2\bar{\bar{l}}_q + \bar{\bar{n}}_q. \quad (2.6.10)$$

**Dokaz.** Kako su elementi inverzne matrice  $N = (L^*AL)^{-1} = \{n_{i,j}(x)\}_{i,j=0}^s$  određeni izrazom (2.6.5), tada

$$\begin{aligned} (LN)_{ij}(x) &= \sum_{l=1}^{\min\{i,s\}} l_{i,l}(x) n_{l,j}(x) = \sum_{l=1}^{\min\{i,s\}} \frac{\sum_{k=0}^{\bar{l}_q} \bar{l}_{k,i,l} x^k \sum_{k=0}^{\bar{n}_q} \bar{n}_{k,l,j} x^k}{\sum_{k=0}^{\bar{l}_q} \bar{\bar{l}}_{k,i,l} x^k \sum_{k=0}^{\bar{n}_q} \bar{\bar{n}}_{k,l,j} x^k} \\ &= \sum_{l=1}^{\min\{i,s\}} \frac{\sum_{k=0}^{\bar{l}_q+\bar{n}_q} \left( \sum_{k_1=0}^k \bar{l}_{k_1,i,l} \bar{n}_{k-k_1,l,j} \right) x^k}{\sum_{k=0}^{\bar{l}_q+\bar{n}_q} \left( \sum_{k_1=0}^k \bar{\bar{l}}_{k_1,i,l} \bar{\bar{n}}_{k-k_1,l,j} \right) x^k}. \end{aligned}$$

Prema tome, važe sledeće jednakosti:

$$\begin{aligned} (L(L^*AL)^{-1}L^*)_{ij}(x) &= \sum_{k=1}^{\min\{j,s\}} (LN)_{ik}(x) \cdot (L^*)_{kj}(x) \\ &= \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{\min\{i,s\}} \frac{\sum_{t_1=0}^{\bar{l}_q+\bar{n}_q} \left( \sum_{t_2=0}^{t_1} \bar{l}_{t_2,i,l} \bar{n}_{t_1-t_2,l,k} \right) x^{t_1} \sum_{t_2=0}^{\bar{l}_q} \bar{l}_{t_2,j,k}^* x^{t_2}}{\sum_{t_1=0}^{\bar{l}_q+\bar{n}_q} \left( \sum_{t_2=0}^{t_1} \bar{\bar{l}}_{t_2,i,l} \bar{\bar{n}}_{t_1-t_2,l,k} \right) x^{t_1} \sum_{t_2=0}^{\bar{l}_q} \bar{\bar{l}}_{t_2,j,k}^* x^{t_2}} \\ &= \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{\min\{i,s\}} \frac{\sum_{t_1=0}^{2\bar{l}_q+\bar{n}_q} \left( \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{l}_{t_2,i,l} \bar{n}_{t_1-t_2-t_3,l,k} \bar{l}_{t_3,j,k}^* \right) x^{t_1}}{\sum_{t_1=0}^{2\bar{l}_q+\bar{n}_q} \left( \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{l}}_{t_2,i,l} \bar{\bar{n}}_{t_1-t_2-t_3,l,k} \bar{\bar{l}}_{t_3,j,k}^* \right) x^{t_1}}. \end{aligned}$$

Na osnovu Teoreme 2.6.2, jednačina (2.6.3) je zadovoljena za  $x \in \mathbb{C}_s(M)$ , a proizvoljni  $(i,j)$ -ti element inverza  $A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)}$  ima oblik

$$\left( A_{\mathcal{R}(L),\mathcal{N}(L^*)}^{(2)} \right)_{ij} = \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{\min\{i,s\}} \frac{\sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t}{\sum_{t=0}^{\bar{\bar{\gamma}}_q} \bar{\bar{\gamma}}_{t,i,j,k,l} x^t} = \frac{\bar{\Sigma}_{i,j}(x)}{\bar{\bar{\Sigma}}_{i,j}(x)},$$

gde je polinome brojioca i imenioca moguće odrediti na sledeće načine:

$$\begin{aligned}\bar{\Sigma}_{i,j}(x) &= \text{PolinomijalniNZS} \left\{ \sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t \mid k = \overline{1, \min\{j, s\}}, l = \overline{1, \min\{i, s\}} \right\} = \sum_{t=0}^{\bar{\Sigma}_q} \bar{\sigma}_{t,i} x^t, \\ \bar{\Sigma}_{i,j}(x) &= \sum_{k=1}^j \sum_{l=1}^s \left( \Sigma_{i,j,k,l}(x) \sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t \right),\end{aligned}$$

pri čemu je svaki polinom  $\Sigma_{i,j,k,l}(x)$  određen kao

$$\Sigma_{i,j,k,l}(x) = \bar{\Sigma}_{i,j}(x) / \sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t = \sum_{t=0}^{\bar{\Sigma}_q - \bar{\gamma}_q} \sigma_{t,i,j,k,l} x^t.$$

Prema tome, svaki polinom brojioca  $\bar{\Sigma}_{i,j}(x)$  je oblika (2.6.6), čime je dokaz kompletiran.  $\square$

Dalje navodimo algoritam iz [78] za izračunavanje generalisanih inverza polinomijalne matrice, na osnovu prethodne teoreme. Zasnovan je na  $LDL^*$  dekompoziciji potpunog ranga odgovarajuće polinomijalne matrice, koristeći Algoritam 2.2, i izračunavanje inverza polinomijalne matrice primenom Algoritma 3.2 iz rada [98]. Da bi se primenio Algoritam 3.2 iz [98] na racionalnu matricu  $L^*AL$ , nju bi najpre trebalo predstaviti u obliku količnika polinomijalne matrice i polinoma.

---

**Algoritam 2.7** Izračunavanje  $A_{T,S}^{(2)}$  inverza polinomijalne matrice primenom  $LDL^*$  faktorizacije potpunog ranga proizvoljne Hermitske polinomijalne matrice  $M$ . (**Algoritam LDLATS2**)

---

**Ulaz:** Polinomijalna matrica  $A(x) \in \mathbb{C}(x)_r^{n \times n}$  normalnog ranga  $r$ .

- 1: Choose arbitrary fixed Hermitska  $n \times n$  polinomijalna matrix  $M$  normal rang  $s \leq r$ .
- 2: Izračunati  $LDL^*$  dekompoziciju potpunog ranga matrice  $M$  koristeći algoritam 2.2, pri čemu su elementi racionalne matrice  $L$  dati u obliku (2.2.2).
- 3: Izračunati najmanji zajednički sadržalac polinoma imenilaca u racionalnoj matrici  $L^*AL$ , tako da je sledeća jednačina validna:  $L^*AL = \frac{1}{p(x)}P(x)$ , gde je  $p(x)$  odgovarajući polinom a  $P(x)$  je polinomijalna matrica.
- 4: Izračunati inverznu matricu  $P^{-1}(x)$  koristeći Algoritam 3.2 iz rada [98]. Odrediti inverznu matricu  $N = (L^*AL)^{-1}$  kao proizvod  $p(x) \cdot P^{-1}(x)$ , pri čemu su elementi matrice  $N$  oblika (2.6.5).
- 5: Uvesti smene  $\bar{\gamma}_q = 2\bar{l}_q + \bar{n}_q$ ,  $\bar{\bar{\gamma}}_q = 2\bar{\bar{l}}_q + \bar{\bar{n}}_q$ , i za  $i, j = \overline{1, n}$  izvršiti Korak 5.1 – Korak 5.5.
  - 5.1: Za  $k = \overline{1, \min\{j, s\}}$ ,  $l = \overline{1, \min\{i, s\}}$  izvršiti sledeća izračunavanja:

$$\begin{aligned}\bar{\gamma}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{l}_{t_2,i,l} \bar{n}_{t_1-t_2-t_3,l,k} \bar{l}_{t_3,j,k}^*, \quad 0 \leq t \leq \bar{\gamma}_q, \\ \bar{\bar{\gamma}}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{l}}_{t_2,i,l} \bar{\bar{n}}_{t_1-t_2-t_3,l,k} \bar{\bar{l}}_{t_3,j,k}^*, \quad 0 \leq t \leq \bar{\bar{\gamma}}_q.\end{aligned}$$

5.2: Izračunati polinom imenioca  $(i, j)$ -tog elementa inverzne matrice  $A_{\mathcal{R}(L), \mathcal{N}(L^*)}^{(2)}$  kao

$$\text{PolinomijalniNZS} \left\{ \sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t \mid k = \overline{1, \min\{j, s\}}, l = \overline{1, \min\{i, s\}} \right\},$$

i označiti ga sa  $\bar{\Sigma}_{i,j}(x) = \sum_{t=0}^{\bar{\Sigma}_q} \bar{\sigma}_{t,i,j} x^t$ .

5.3: Za  $k = \overline{1, \min\{j, s\}}$ ,  $l = \overline{1, \min\{i, s\}}$  izračunati polinom  $\bar{\Sigma}_{i,j}(x) / \sum_{t=0}^{\bar{\gamma}_q} \bar{\gamma}_{t,i,j,k,l} x^t$  i označiti ga sa  $\Sigma_{i,j,k,l}(x) = \sum_{t=0}^{\bar{\Sigma}_q - \bar{\gamma}_q} \sigma_{t,i,j,k,l} x^t$ .

5.4: Izračunati polinom u brojiocu  $(i, j)$ -tog elementa matrice  $A_{\mathcal{R}(L), \mathcal{N}(L^*)}^{(2)}$  kao

$$\bar{\Sigma}_{i,j}(x) = \sum_{t=0}^{\bar{\Sigma}_q - \bar{\gamma}_q + \bar{\gamma}_q} \left( \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^{\min\{i,s\}} \sum_{t_1=0}^t \bar{\gamma}_{t_1, i, j, k, l} \sigma_{t-t_1, i, j, k, l} \right) x^t,$$

5.5: **Izlaz:**  $(i, j)$ -ti element inverzne matrice  $A_{\mathcal{R}(L), \mathcal{N}(L^*)}^{(2)}$  dat u obliku  $\bar{\Sigma}_{i,j}(x) / \bar{\Sigma}_{i,j}(x)$ .

### 2.6.1 Numerički primeri

U sledećih nekoliko primera ćemo opisati naš algoritam i testirati nekoliko implementacija radi upoređivanja vremena izvršavanja na skupovima nasumičnih matrica i test matrica iz [117].

**Primer 2.6.1.** Posmatrajmo polinomijalnu matricu  $A(x)$  datu u Primeru 2.2.2. Kako je  $A$  realna simetrična matrica, njen Moore-Penrose-ov inverz se može izračunati primenom Algoritma LDLATS2 za matricu  $M = A^* = A$ . Nakon nekoliko izračunavanja potrebnih u Koraku 5 navedenog algoritma, dobijen je sledeći rezultat:

$$A^\dagger = A^\# = \begin{bmatrix} \frac{-1+x}{-85-83x-2x^2+x^3} & \frac{9+17x}{85+83x+2x^2-x^3} & \frac{3+10x}{85+83x+2x^2-x^3} & \frac{26(1+x)}{-85-83x-2x^2+x^3} \\ \frac{9+17x}{4+36x-2x^2+3x^3} & \frac{27-2x+2x^2-2x^3}{-85-83x-2x^2+x^3} & \frac{27-2x+2x^2-2x^3}{85+83x+2x^2-x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} \\ \frac{85+83x+2x^2-x^3}{3+10x} & \frac{85+83x+2x^2-x^3}{27-2x+2x^2-2x^3} & \frac{9-16x+x^2-x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} \\ \frac{85+83x+2x^2-x^3}{26(1+x)} & \frac{85+83x+2x^2-x^3}{21+19x-2x^2+5x^3} & \frac{85+83x+2x^2-x^3}{7+7x-2x^2+3x^3} & \frac{85+83x+2x^2-x^3}{4-4x+8x^3} \\ \frac{26(1+x)}{-85-83x-2x^2+x^3} & \frac{21+19x-2x^2+5x^3}{85+83x+2x^2-x^3} & \frac{7+7x-2x^2+3x^3}{85+83x+2x^2-x^3} & \frac{4-4x+8x^3}{-85-83x-2x^2+x^3} \end{bmatrix}.$$

**Primer 2.6.2.** Posmatrajmo simetričnu matricu  $F_6$  generisanu u [117] po promenljivoj  $x$ , tj.

$$F_6 = \begin{bmatrix} 6+x & 5+x & 4+x & 3+x & 2+x & 1+x \\ 5+x & 5+x & 4+x & 3+x & 2+x & 1+x \\ 4+x & 4+x & 4+x & 3+x & 2+x & 1+x \\ 3+x & 3+x & 3+x & 3+x & 2+x & 1+x \\ 2+x & 2+x & 2+x & 2+x & 1+x & x \\ 1+x & 1+x & 1+x & 1+x & x & -1+x \end{bmatrix}.$$

Sada, da bi odredili Moore-Penrose-ov inverz matrice  $F_6$ , posmatrajmo polinomijalnu matricu  $M = F_6^* = F_6$ . Kako je rang matrice  $F_6$  jednak 5, matrice  $L$  i  $D$  dobijene  $LDL^*$  faktorizacijom matrice  $F_6$  imaju sledeći oblik:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{5+x}{6+x} & 1 & 0 & 0 & 0 \\ \frac{4+x}{6+x} & \frac{5+x}{3+x} & 1 & 0 & 0 \\ \frac{6+x}{3+x} & \frac{3+x}{2+x} & \frac{3+x}{2+x} & 1 & 0 \\ \frac{6+x}{2+x} & \frac{5+x}{2+x} & \frac{4+x}{1+x} & \frac{2+x}{1+x} & 1 \\ \frac{6+x}{1+x} & \frac{5+x}{1+x} & \frac{4+x}{1+x} & \frac{3+x}{1+x} & 1 \\ \frac{6+x}{6+x} & \frac{5+x}{5+x} & \frac{4+x}{4+x} & \frac{3+x}{3+x} & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 6+x & 0 & 0 & 0 & 0 \\ 0 & \frac{5+x}{6+x} & 0 & 0 & 0 \\ 0 & 0 & \frac{4+x}{5+x} & 0 & 0 \\ 0 & 0 & 0 & \frac{3+x}{4+x} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{3+x} \end{bmatrix}.$$

Prema tome, inverzna matrica  $(L^*F_6L)^{-1}$  je  $5 \times 5$  racionalna matrica oblika

$$\begin{bmatrix} 1 & \frac{-11-2x}{6+x} & \frac{4+x}{5+x} & 0 & 0 \\ -11-2x & \frac{157+56x+5x^2}{36+12x+x^2} & \frac{-98-40x-4x^2}{30+11x+x^2} & \frac{3+x}{4+x} & 0 \\ \frac{6+x}{4+x} & \frac{-98-40x-4x^2}{30+11x+x^2} & \frac{122+54x+6x^2}{25+10x+x^2} & \frac{-352-183x-23x^2}{120+54x+6x^2} & \frac{4+3x}{18+6x} \\ \frac{5+x}{0} & \frac{3+x}{4+x} & \frac{-352-183x-23x^2}{120+54x+6x^2} & \frac{1816+932x+88x^2-9x^3}{576+288x+36x^2} & \frac{-104-62x+24x^2+9x^3}{432+252x+36x^2} \\ 0 & 0 & \frac{4+3x}{18+6x} & \frac{-104-62x+24x^2+9x^3}{432+252x+36x^2} & \frac{-44-60x-36x^2-9x^3}{324+216x+36x^2} \end{bmatrix},$$

i najzad, generisan je sledeći Moore-Penrose-ov inverz prateći poslednji korak algoritma LDLATS2:

$$F_6^\dagger = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -\frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \\ 0 & 0 & -\frac{5}{6} & \frac{1}{36}(28-9x) & \frac{4}{9} & \frac{1}{36}(4+9x) \\ 0 & 0 & -\frac{1}{3} & \frac{4}{9} & \frac{1}{9} & -\frac{2}{9} \\ 0 & 0 & \frac{1}{6} & \frac{1}{36}(4+9x) & -\frac{2}{9} & \frac{1}{36}(-20-9x) \end{bmatrix}.$$

Da bismo izračunali Drazinov inverz matrice  $F_6$ , indeks  $k \geq \text{ind}(A)$  treba odrediti. Prema tome, za  $k = 2$  treba odrediti faktorizaciju matrice  $M = F_6^2$ , a dobijeni rezultat je jednak Moore-Penrose-ovom inverzu, tj.  $F_6^D = F_6^\dagger = F_6^\#$ .

**Primer 2.6.3.** Uočimo sada  $6 \times 6$  matricu  $A_2$  ranga 2 generisanu u [117], datu u polinomijalnom obliku

$$A_2 = \begin{bmatrix} x+10 & x+9 & x+8 & x+7 & x+6 & x+5 \\ x+9 & x+8 & x+7 & x+6 & x+5 & x+4 \\ x+8 & x+7 & x+6 & x+5 & x+4 & x+3 \\ x+7 & x+6 & x+5 & x+4 & x+3 & x+2 \\ x+6 & x+5 & x+4 & x+3 & x+2 & x+1 \\ x+5 & x+4 & x+3 & x+2 & x+1 & x \end{bmatrix}.$$

Primenom Algoritma 2.2, za  $j = 2$  dobijaju se sledeći među-rezultati:

$$f_{2,2}(x) = \frac{(9+x)^2}{10+x}, \quad f_{3,2}(x) = \frac{(8+x)(9+x)}{10+x},$$

$$f_{4,2}(x) = \frac{(7+x)(9+x)}{10+x}, \quad f_{5,2}(x) = \frac{(6+x)(9+x)}{10+x}, \quad f_{6,2}(x) = \frac{(5+x)(9+x)}{10+x}.$$

Prema tome, imamo sledeće racionalne matrice uključene u  $LDL^*$  dekompoziciju polazne matrice

$$L = \begin{bmatrix} 1 & \frac{9+x}{10+x} & \frac{8+x}{10+x} & \frac{7+x}{10+x} & \frac{6+x}{10+x} & \frac{5+x}{10+x} \end{bmatrix}^T, D = \begin{bmatrix} 10+x & 0 \\ 0 & -\frac{1}{10+x} \end{bmatrix}.$$

Primenjujući Algoritam LDLATS2 i rezultate Posledice 2.6.3, dobijeni su sledeći međusobno jednaki uopšteni inverzi:

$$A_2^\dagger = A_2^\# = A^D = \begin{bmatrix} \frac{1}{147}(-8 - 3x) & \frac{1}{735}(-17 - 9x) & \frac{2-x}{245} & \frac{1}{735}(29 + 3x) & \frac{1}{735}(52 + 9x) & \frac{5+x}{49} \\ \frac{1}{735}(-17 - 9x) & \frac{-10-9x}{25-9x} & \frac{25-9x}{3675} & \frac{80+9x}{3675} & \frac{9(5+x)}{1225} & \frac{1}{735}(38 + 9x) \\ \frac{2-x}{245} & \frac{25-9x}{3675} & \frac{3675}{3675} & \frac{3675}{1225} & \frac{1225}{3675} & \frac{1}{735}(1 + 3x) \\ \frac{1}{735}(29 + 3x) & \frac{80+9x}{3675} & \frac{3675}{3675} & \frac{-50-3x}{3675} & \frac{-115-9x}{3675} & \frac{1}{245}(-12 - x) \\ \frac{1}{735}(52 + 9x) & \frac{9(5+x)}{1225} & \frac{10+9x}{3675} & \frac{-115-9x}{3675} & \frac{-80-9x}{1225} & \frac{1}{735}(-73 - 9x) \\ \frac{5+x}{49} & \frac{1}{735}(38 + 9x) & \frac{1}{735}(1 + 3x) & \frac{1}{245}(-12 - x) & \frac{1}{735}(-73 - 9x) & \frac{1}{147}(-22 - 3x) \end{bmatrix}.$$

## 2.6.2 Implementacioni detalji i rezultati testiranja

```
LDLATS2[A_List, W_List] :=
Module[{N1, L, D, i, j, k, l, n, m, r1, r2, r3, p, q, f, Num, Den, s = MatrixRank[W]},
{n, m} = Dimensions[W];
{L, D} = PolynomialLDLDecomposition[W];
N1 =
ExpandNumerator[ExpandDenominator[Together[Simplify[Inverse[Transpose[L].A.L]]]]];
f = Num = Den = Table[0, {n}, {m}];
For[i = 1, i ≤ n, i++,
For[j = 1, j ≤ m, j++,
For[k = 1, k ≤ Min[j, s], k++,
For[l = 1, l ≤ Min[i, s], l++,
Num[[k, l]] = 0; Den[[k, l]] = 0;
For[r1 = 0, r1 ≤
Max[Exponent[Numerator[L[[i, l]]], x], Exponent[Denominator[L[[i, l]]], x]] +
Max[Exponent[Numerator[N1[[l, k]]], x], Exponent[Denominator[N1[[l, k]]], x]] +
Max[Exponent[Numerator[L[[j, k]]], x], Exponent[Denominator[L[[j, k]]], x]],
r1++,
p = Sum[Sum[Coefficient[Numerator[L[[i, l]]], x, r2] *
Coefficient[Numerator[N1[[l, k]]], x, r1 - r2 - r3] *
Conjugate[Coefficient[Numerator[L[[j, k]]], x, r3]], {r2, 0, r1 - r3}, {r3, 0, r2}], {r1, 0, s};
q = Sum[Sum[Coefficient[Denominator[L[[i, l]]], x, r2] *
Coefficient[Denominator[N1[[l, k]]], x, r1 - r2 - r3] *
Conjugate[Coefficient[Denominator[L[[j, k]]], x, r3]], {r2, 0, r1 - r3}, {r3, 0, r2}], {r1, 0, s};
Num[[k, l]] += p * x^r1;
Den[[k, l]] += q * x^r1;
If[Den[[k, l]] == 0, Num[[k, l]] = 0; Den[[k, l]] = 1];
];];
];
f[[i, j]] = Together[ Simplify[ Sum[ Sum[ Simplify[ Num[[k, l]] / Den[[k, l]] ]], {k, 1, Min[j, s]}, {l, 1, Min[i, s]} ]]];
];
];
Return[f];
];
```

Gore navodena implementacija je implementacija Algoritma 2.10, nazvanog LDLATS2, u programskom paketu MATHEMATICA. Funkcija LDLATS2 podrazumeva dva parametra: zadanu polinomijalnu matricu  $A$  i proizvoljnu fiksiranu matricu  $W$ . U zavisnosti od odabira matrice  $W$ , dobija se odgovarajući generalisani inverz matrice  $A$ .

Implementacija Algoritma 2.6 (LDLATS) je krajnje jednostavna, i data je sledećim kodom, pod pretpostavkom da se matrična jednačina (2.6.3) rešava primenom MATHEMATICA funkcije Inverse.

```
ATS2Inverse[A_List, W_List] :=
Module[{N1, L, D},
{L, D} = PolynomialLDLDecomposition[W];
N1 = Inverse[Transpose[L].A.L] // Simplify;
Return[Simplify[L.N1.Transpose[L]]];
];
```

**Primer 2.6.4.** Ispitajmo sada efikasnost LDLATS algoritma, upoređujući grupu različitih algoritama za izračunavanje Moore-Penrose-ovog inverza matrice. Sledeće tabela daje srednja vremena izračunavanja dobijenih primenom ovih algoritama na tri skupa Hermit-skih test matrica iz [117], posmatrajući parcijalni slučaj  $x = 1$ .

Test matrica	$A_{10}$	$A_{50}$	$A_{100}$	$A_{150}$	$S_{10}$	$S_{50}$	$S_{100}$	$S_{150}$	$F_{10}$	$F_{50}$	$F_{100}$	$F_{150}$
Partitioning [87]	0.01	0.48	2.75	14.55	0.03	1.18	9.20	73.55	0.01	0.45	2.81	15.3
Lev.-Faddeev [41]	0.04	2.76	43.78	-	0.11	2.51	44.37	-	0.13	2.67	42.84	-
PseudoInverse [113]	0.03	1.86	25.51	-	0.13	2.77	41.39	-	0.09	2.33	33.54	-
Courrieu [16]	0.02	0.76	5.84	39.86	0.01	0.37	2.29	14.66	0.01	0.70	5.78	35.43
LDLGInverse [77]	0.02	1.87	11.67	-	0.02	0.98	4.55	22.31	0.01	1.96	12.64	-
<b>LDLATS</b>	0.02	0.85	6.05	41.58	0.01	0.48	2.89	16.11	0.02	0.90	6.23	40.01

Tabela 2.6.1. Prosečna vremena izračunavanja (u sekundama) dobijena različitim algoritmima i LDLATS algoritmom.

Prva vrsta Tabele 2.6.1 sadrži nazive test matrica, pri čemu su posmatrane tri grupe matrica iz [117]. Vremena dobijena LDLATS algoritmom su navedena u poslednjoj vrsti. Očigledno, ovo nije najefikasniji algoritam za svaki skup numeričkih matrica, ali je u rangu sa algoritmom koji je uveo Courrieu u [16]. Dugotrajna vremena izračunavanja su navedena crticama.

**Primer 2.6.5.** U sledećim komparativnim eksperimentima, nekoliko nasumičnih Hermit-skih polinomijalnih matrica  $A \in \mathbb{C}(x)_r^{n \times n}$  različitih veličina i gustina, je posmatrano. Prema tome, najveća sopstvena vrednost test matrice  $A$  je označena kao  $k_r$ , pri čemu je najmanja ne-nula sopstvena vrednost matrice  $A$  jednaka 1 (slična testiranja su izvršena u radu [17]). Neka od dobijenih vremena su ilustrovana u Tabeli 2.6.2, dajući uvid u efikasnost LDLATS i LDLATS2 algoritama. Očito je LDLATS2 algoritam neznatno efikasniji, s obzirom da je specijalizovan za polinomijalne matrice.

$n$	128			256			512		
$k_r$	16	256	4096	16	256	4096	16	256	4096
LDLATS potp. rang	4.25	4.53	4.68	30.01	30.24	30.31	227.38	231.90	234.29
LDLATS rang def.	3.04	3.75	3.86	25.1	25.24	25.53	185.90	190.36	190.46
LDLATS2 potp. rang	4.09	4.22	4.31	28.11	29.48	30.12	217.87	221.08	224.98
LDLATS2 rang def.	2.44	3.04	3.62	24.01	24.71	25.37	180.07	182.39	184.62

Tabela 2.6.2. Prosečna vremena izračunavanja (u sekundama) za nasumično generisane polinomijalne matrice.

## 2.7 Reprezentacije potpunog ranga zasnovane na $QR$ dekompoziciji

Navedimo najpre osnovne pojmove u vezi  $QR$  faktorizacije matrica.

**Teorema 2.7.1.** *Ako je  $A \in \mathbb{R}_r^{m \times n}$ , tada postoji ortogonalna matrica  $Q \in \mathbb{R}^{m \times m}$  i gornja trougaona  $R \in \mathbb{R}^{n \times n}$ , takve da je  $A = QR$ .*

**Teorema 2.7.2.** *Ako  $A = QR$  predstavlja  $QR$  faktorizaciju matrice  $A \in \mathbb{C}_r^{m \times n}$ , tada je*

$$A^\dagger = R^\dagger Q^* = R^*(RR^*)^{-1}Q^*.$$

Reprezentacije Moore–Penrose-ovog inverza  $A^\dagger$  zasnovane na  $QR$  dekompoziciji matrice  $A$  su poznate u literaturi (videti na primer [44, 105]). Numeričko testiranje ovih reprezentacija je prezentovano u [44]. Numerički metod za izračunavanje najmanjeg kvadrata rešenja  $A^\dagger b$  minimalne norme za linearne jednačine  $Ax = b$ , zasnovane na  $QR$  faktorizaciji matrice  $A^*$ , je istraženo u [33]. Ovaj metod se može upotrebiti za izračunavanje Moore–Penrose-ovog inverza  $A^\dagger$  uzimajući sukcesivne vrednosti  $b = e_i$ , gde je  $e_i$   $i$ -ti jedinični vektor, za svako  $i = 1, \dots, n$ .

Spoljni uopšteni inverzi definisanog jezgra i slike su veoma važni u teoriji matrica.  $\{2\}$ -inverzi imaju primenu u iterativnim metodima za rešavanje nelinearnih jednačina [3, 57] kao i u statistici [28, 35]. Specijalno, spoljni inverzi igraju važnu ulogu kod stabilnih aproksimacija loše uslovljenih problema, kao i u linearnim i nelinearnim problemima sa rang-deficijentnim uopštenim inverzima [56, 115]. Sa druge strane, dobro je poznato da se Moore–Penrose-ov inverz i težinski Moore–Penrose-ov inverz  $A^\dagger, A_{M,N}^\dagger$ , Drazinov i grupni inverz  $A^D, A^\#$ , kao i Bott-Duffin-ov inverz  $A_{(L)}^{(-1)}$  i uopšteni Bott-Duffin-ov inverz  $A_{(L)}^{(\dagger)}$  mogu predstaviti na jedinstven način, kao uopšteni inverzi  $A_{T,S}^{(2)}$  za odgovarajući odabir matrica  $T$  i  $S$ .

Postoji veliki broj "full-rank" reprezentacija spoljnih inverza definisanog ranga kao i za spoljne inverze zadatog jezgra i slike. I ovde ćemo koristiti reprezentaciju iz [70] datu Lemom 2.6.1 za izvođenje kasnijih rezultata. Reprezentacija  $A_{T,S}^{(2)}$  inverza data u Lemi 2.6.1 je veoma opšt rezultat. Ova reprezentacija je korišćena u istom radu za određivanje determinantskih reprezentacija  $A_{T,S}^{(2)}$  inverza. Determinanska reprezentacija nije efikasna numerička procedura za izračunavanje uopštenih inverza. Štaviše, autori rada [70] ne razmatraju različite mogućnosti za odgovarajuće faktorizacije potpunog ranga.

Ovde navodimo dva cilja koja smo naveli u radu [80]. Najpre je izrađen efikasan metod za izračunavanje spoljnih inverza primenom opšte reprezentacije potpunog ranga. Dodatno, opisani su i analogni algoritmi za izračunavanje  $\{2, 4\}$  i  $\{2, 3\}$ -inverza, kao dva posebna podskupa skupa spoljnih inverza.

Na osnovu poznatih reprezentacija iz radova [3, 64, 70, 82, 90] u radu [80] naveden je sledeći stav sa reprezentacijama (1.1.1)–(1.1.3). Ove reprezentacije karakterišu klase  $\{2\}$ ,  $\{2, 4\}$  i  $\{2, 3\}$  uopštenih inverza zadatog ranga.

**Stav 2.7.3.** [80] Neka je  $A \in \mathbb{C}_r^{m \times n}$  zadata matrica i neka je  $0 < s \leq r$ . Tada važe sledeće opšte reprezentacije za neke klase uopštenih inverza:

- (a)  $A\{2\}_s = \{A_{\mathcal{R}(F), \mathcal{N}(G)}^{(2)} = F(GAF)^{-1}G \mid F \in \mathbb{C}^{n \times s}, G \in \mathbb{C}^{s \times m}, \text{rang}(GAF)=s\};$
- (b)  $A\{2, 4\}_s = \left\{ A_{\mathcal{R}((GA)^*), \mathcal{N}(G)}^{(2,4)} = (GA)^* (GA(GA)^*)^{-1} G \mid G \in \mathbb{C}_s^{s \times m} \right\}$   
 $= \{(GA)^\dagger G \mid GA \in \mathbb{C}_s^{s \times n}\};$
- (c)  $A\{2, 3\}_s = \left\{ A_{\mathcal{R}(F), \mathcal{N}((AF)^*)}^{(2,3)} = F ((AF)^* AF)^{-1} (AF)^* \mid F \in \mathbb{C}_s^{n \times s} \right\}$   
 $= \{F(AF)^\dagger \mid AF \in \mathbb{C}_s^{m \times s}\};$
- (d)  $A\{1, 2\} = A\{2\}_r.$

U narednom tvrđenju izvedena je reprezentacija spoljnih inverza zadatog ranga, jezgra i slike. Ova reprezentacija je zasnovana na  $QR$  dekompoziciji definisanoj u Teoremi 3.3.11 iz rada [108]. Analogna  $QR$  dekompozicija za kompleksne matrice je korišćena u [26].

**Lema 2.7.4.** [80] Neka je  $A \in \mathbb{C}_r^{m \times n}$  zadata matrica. Posmatrajmo proizvoljnu matricu  $W \in \mathbb{C}_s^{n \times m}$ , gde je  $s \leq r$ . Neka je  $QR$  faktorizacija matrice  $W$  data u obliku

$$WP = QR, \quad (2.7.1)$$

gde je  $P$  neka  $m \times m$  permutaciona matrica,  $Q \in \mathbb{C}^{n \times n}$ ,  $Q^*Q = I_n$  a  $R \in \mathbb{C}_s^{n \times m}$  je gornja trapezoidna matrica. Prepostavimo da je matrica  $P$  odabrana tako da se  $Q$  i  $R$  mogu predstaviti kao

$$Q = [ Q_1 \ Q_2 ], \quad R = \begin{bmatrix} R_{11} & R_{12} \\ O & O \end{bmatrix} = \begin{bmatrix} R_1 \\ O \end{bmatrix}, \quad (2.7.2)$$

gde  $Q_1$  sadrži prvih  $s$  kolona matrice  $Q$  a matrica  $R_{11} \in \mathbb{C}^{s \times s}$  je nesingularna.

Ako matrica  $A$  ima  $\{2\}$ -inverz  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}$ , tada:

- (a)  $R_1 P^* A Q_1$  je invertibilna matrica;
- (b)  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = Q_1 (R_1 P^* A Q_1)^{-1} R_1 P^*$ ;
- (c)  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2)}$ ;
- (d)  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = Q_1 (Q_1^* W A Q_1)^{-1} Q_1^* W$ ;
- (e)  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} \in A\{2\}_s$ .

**Dokaz.** (a) Na osnovu učinjenih prepostavki je

$$W = QRP^*. \quad (2.7.3)$$

Očigledno je netrivijalni deo  $QR$  dekompozicije (2.7.1), definisan kao

$$W = Q_1(R_1 P^*), \quad (2.7.4)$$

zapravo faktorizacija potpunog ranga matrice  $W$  (videti takođe [105]). S obzirom da  $A$  ima  $\{2\}$ -inverz  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}$ , tada na osnovu dela (1) Leme 2.6.1 zaključujemo da je  $R_1 P^* A Q_1$  invertibilna.

(b), (c) Na osnovu dela (2) Leme 2.6.1, nije teško dobiti reprezentaciju

$$A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = Q_1(R_1 P^* A Q_1)^{-1} R_1 P^*$$

spoljašnjeg inverza  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}$  zadatog slike  $\mathcal{R}(W) = \mathcal{R}(Q_1)$  i jezgra  $\mathcal{N}(W) = \mathcal{N}(R_1 P^*)$ .

(d) Štaviše, na osnovu (2.7.4) imamo

$$R_1 P^* = Q_1^* W$$

i

$$Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = Q_1(Q_1^* W A Q_1)^{-1} Q_1^* W.$$

(e) Ovaj deo tvrđenja sledi iz Stava 2.7.3.  $\square$

Koristeći rezultate iz Stava 2.7.3 i uzimajući u obzir jednačine (1.1.1)–(1.1.3), dobijamo sledeće "full–rank" reprezentacije za različite spoljne uopštene inverze:

**Posledica 2.7.5.** [80] Za zadatu matricu  $A \in \mathbb{C}_r^{m \times n}$  i odabranu matricu  $W \in \mathbb{C}_s^{n \times m}$ ,  $s \leq r$ , sa reprezentacijom potpunog ranga (2.7.4) nastalu iz njene QR dekompozicije (2.7.1), sledeća tvrđenja su validna:

$$A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2)} = \begin{cases} A^\dagger, & W = A^*; \\ A_{M,N}^\dagger, & W = A^\ddagger; \\ A^\#, & W = A; \\ A^D, & W = A^k, k \geq \text{ind}(A); \\ A_{(L)}^{(-1)}, & \mathcal{R}(W) = L, \mathcal{N}(W) = L^\perp; \\ A_{(L)}^{(\dagger)}, & \mathcal{R}(W) = S, \mathcal{N}(W) = S^\perp \end{cases} \quad (2.7.5)$$

Faktorizacije potpunog ranga  $W = FG$  koje zadovoljavaju  $F^* F = I$  se nazivaju ortogonalne [63]. Prema tome, "full–rank" reprezentacija (2.7.4) je ortogonalna.

U sledećem tvrđenju je pokazano da ortogonalna "full–rank" reprezentacija zasnovana na QR dekompoziciji matrice  $W$  daje isti  $A_{T,S}^{(2)}$  inverz kao i reprezentacija iz Leme 2.6.1.

**Posledica 2.7.6.** [80] Neka su zadovoljene pretpostavke Leme 2.7.4 i neka je  $W = FG$  proizvoljna faktorizacija potpunog ranga matrice  $W$ . Tada je zadovoljeno

$$Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = F(GAF)^{-1} G \quad (2.7.6)$$

**Dokaz.** Na osnovu jedinstvenosti uopštenog inverza zadatog jezgra i slike, primenjujući tvrđenje (b) Leme 2.7.4 i Lemu 2.6.1, tvrđenje (2), zaključujemo

$$\begin{aligned} Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* &= A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} \\ &= A_{\mathcal{R}(F), \mathcal{N}(G)}^{(2)} \\ &= F(GAF)^{-1} G, \end{aligned}$$

čime je dokaz kompletan.  $\square$

U sledećoj lemi iz [80] pokazano je da je reprezentacija spoljnog inverza  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}$ , definisana u Lemi 2.6.1, invarijantna u odnosu na izbor faktorizacije potpunog ranga matrice  $W$ . U tom smislu je dobijeno uopštenje Posledice 2.7.6.

**Lema 2.7.7.** [80] *Neka su zadovoljene pretpostavke Leme 2.7.4. Ako su  $W = F_1G_1$  i  $W = F_2G_2$  dve različite faktorizacije potpunog ranga matrice  $W$ , tada važi*

$$\begin{aligned} F_1(G_1AF_1)^{-1}G_1 &= F_2(G_2AF_2)^{-1}G_2 \\ &= A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}. \end{aligned}$$

**Dokaz.** Na osnovu Teoreme 2 iz [63], postoji invertibilna  $s \times s$  matrica  $D$  takva da je

$$F_2 = F_1D, \quad G_2 = D^{-1}G_1.$$

Ova relacija povlači sledeće:

$$F_2(G_2AF_2)^{-1}G_2 = F_1D(D^{-1}G_1AF_1D)^{-1}D^{-1}G_1.$$

Kako su matrice  $D$  i  $G_1AF_1$  invertibilne, važi zakon obrnutog poretku za  $(D^{-1}G_1AF_1D)^{-1}$ , tako da je

$$F_2(G_2AF_2)^{-1}G_2 = F_1DD^{-1}(G_1AF_1)^{-1}DD^{-1}G_1 = F_1(G_1AF_1)^{-1}G_1.$$

Iz ovoga sledi dokaz uvedenog tvrđenja.  $\square$

Konačno možemo naći korelaciju između opštih reprezentacija datih u radu [70] i reprezentacije uvedene u Lemi 2.7.4.

**Posledica 2.7.8.** [80] *Neka su zadovoljeni uslovi Leme 2.7.4. Neka je  $W = FG$  proizvoljna reprezentacija potpunog ranga matrice  $W$  koja zadovoljava uslove Leme 2.6.1. Ukoliko je (2.7.4) faktorizacija potpunog ranga matrice  $W$ , zadovoljena su sledeća tvrđenja:*

$$F = Q_1D, \quad G = D^{-1}R_1P^*, \tag{2.7.7}$$

pri čemu je matrica  $D$  jednaka  $D = Q_1^*F$ .

**Primer 2.7.1.** Posmatrajmo matricu

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 4 & 6 & 2 \\ 2 & 3 & 4 & 5 & 3 \\ 3 & 4 & 5 & 6 & 4 \\ 4 & 5 & 6 & 7 & 6 \\ 6 & 6 & 7 & 7 & 8 \end{bmatrix} \tag{2.7.8}$$

i odaberimo matrice

$$F = \begin{bmatrix} 3 & -2 \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.7.9}$$

Pretpostavimo da je matrica  $W$  definisana kao  $W = FG$ . Matrice  $Q_1, R_1$  i  $P$  koje definišu netrivialni deo QR dekompozicije (2.7.1) matrice  $WP$  su date sa

$$\{Q_1, R_1, P\} = \left\{ \begin{bmatrix} \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ -\frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \sqrt{10} & -\frac{7}{\sqrt{10}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{10}} & 0 & 0 & 0 & 0 \end{bmatrix}, I_6 \right\}.$$

Reprezentacije date u tvrdjenjima (b) i (c) Leme 2.7.4 daju sledeći {2}-inverz  $A$ :

$$A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Iako su  $W = FG$  i  $W = Q_1(R_1P^*)$  dve različite reprezentacije potpunog ranga matrice  $W$ , na osnovu Posledice 2.7.6 imamo

$$A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = Q_1(R_1P^*AQ_1)^{-1}R_1P^* = F(GAF)^{-1}G.$$

Označimo sada sa  $I(n)$  složenost algoritma za invertovanje zadate  $n \times n$  matrice (kao u [15]). Takođe sa  $\mathcal{A}(n)$  označićemo složenost dodavanja/oduzimanja dve  $n \times n$  matrice, a sa  $\mathcal{M}(m, n, k)$  složenost množenja  $m \times n$  matrice  $n \times k$  matricom. Jednostavnija notacija  $\mathcal{M}(n)$  (uzeta iz [15]) se koristi umesto  $\mathcal{M}(n, n, n)$ .

**Napomena 2.7.1.** Reprezentacija data u tvrdjenju (d) Leme 2.7.4 podrazumeva  $M(s, n, m)$  operacija za formiranje proizvoda matrica  $Q_1^*W$ . Sa druge strane, reprezentacija definisana tvrdjenjem (b) Leme 2.7.4 zahteva  $M(s, m, m)$  operacija za formiranje matričnog proizvoda  $R_1P^*$ . Prema tome, reprezentacija (d) je bolja nego reprezentacija (b) u slučaju kada je  $n < m$ . U suprotnom slučaju  $m < n$  je reprezentacija (b) pogodnija u odnosu na reprezentaciju (d).

Efikasan metod za izračunavanje  $A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}$  je definisan kao što sledi. Za slučaj  $m < n$ , uvezši u obzir složenost matričnog množenja, bolje je rešavati skup jednačina

$$R_1P^*AQ_1X = R_1P^*. \quad (2.7.10)$$

Za slučaj  $n < m$ , efikasnije je rešavati sledeći skup jednačina

$$Q_1^*WAQ_1X = Q_1^*W. \quad (2.7.11)$$

Kada se generiše matrica  $X$ , neophodno je izračunati proizvod matrica

$$A_{\mathcal{R}(Q_1), \mathcal{N}(R_1P^*)}^{(2)} = A_{\mathcal{R}(Q_1), \mathcal{N}(Q_1^*W)}^{(2)} = Q_1X. \quad (2.7.12)$$

**Posledica 2.7.9.** [80] Neka je  $A \in \mathbb{C}_r^{m \times n}$  zadata matrica,  $s \leq r$  zadan ceo broj i neka su matrice  $F, G$  odabrane kao u Stava 2.7.3.

(a) Ako je (2.7.4) faktorizacija potpunog ranga matrice  $W = (GA)^*G \in \mathbb{C}_s^{n \times m}$  tada važi sledeće:

$$A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2,4)} = Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = (GA)^\dagger G \in A\{2, 4\}_s. \quad (2.7.13)$$

(b) Ako je (2.7.4) faktorizacija potpunog ranga matrice  $W = F(AF)^* \in \mathbb{C}_s^{n \times m}$  tada važi sledeće:

$$A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2,3)} = Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = F(AF)^\dagger \in A\{2, 3\}_s. \quad (2.7.14)$$

**Dokaz.** (a) U ovom slučaju, na osnovu Posledice 2.7.6 i Stava 2.7.3 imamo

$$Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = (GA)^*(GA(GA)^*)^{-1} G = (GA)^\dagger G.$$

(b) Ovaj deo dokaza se proverava na sličan način.  $\square$

**Primer 2.7.2.** Neka je data matrica  $A$  definisana u (2.7.8) i odaberimo matrice  $F, G$  definisane u (2.7.9).

(i) Reprezentacija potpunog ranga (2.7.4) matrice  $W = (GA)^*G$  je data sa

$$\{Q_1, R_1, P\} = \left\{ \begin{bmatrix} \frac{1}{\sqrt{31}} & -2\sqrt{\frac{7}{93}} \\ \frac{2}{\sqrt{31}} & \sqrt{\frac{3}{217}} \\ \frac{3}{\sqrt{31}} & -\frac{11}{\sqrt{651}} \\ \frac{4}{\sqrt{31}} & 2\sqrt{\frac{3}{217}} \\ \frac{1}{\sqrt{31}} & \frac{17}{\sqrt{651}} \end{bmatrix}, \begin{bmatrix} \sqrt{31} & \frac{45}{\sqrt{31}} & 0 & 0 & 0 & 0 \\ 0 & \sqrt{\frac{21}{31}} & 0 & 0 & 0 & 0 \end{bmatrix}, I_6 \right\}.$$

Reprezentacija (b) iz Leme 2.7.4 daje sledeći  $\{2, 4\}$ -inverz od  $A$ :

$$\begin{aligned} A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2,4)} &= Q_1(R_1 P^* A Q_1)^{-1} R_1 P^* = \begin{bmatrix} 1 & -\frac{2}{3} & 0 & 0 & 0 & 0 \\ -\frac{1}{7} & \frac{1}{7} & 0 & 0 & 0 & 0 \\ \frac{6}{7} & -\frac{11}{21} & 0 & 0 & 0 & 0 \\ -\frac{2}{7} & \frac{2}{7} & 0 & 0 & 0 & 0 \\ -\frac{8}{7} & \frac{17}{21} & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= (GA)^\dagger G \\ &= A_{\mathcal{R}((GA)^*), \mathcal{N}(G)}^{(2,4)}. \end{aligned}$$

(ii) Odaberimo matricu  $W$  kao  $W = F(AF)^*$ . Reprezentacija potpunog ranga (2.7.4) matrice  $W$  je data sa

$$\{Q_1, R_1, P\} \left\{ \begin{bmatrix} \frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ -\frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} \sqrt{10} & -\frac{7}{\sqrt{10}} & \frac{37}{\sqrt{10}} & 32\sqrt{\frac{2}{5}} & \frac{91}{\sqrt{10}} & 81\sqrt{\frac{2}{5}} \\ 0 & \frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\sqrt{\frac{2}{5}} & -\frac{3}{\sqrt{10}} & -3\sqrt{\frac{2}{5}} \end{bmatrix}, I_6 \right\}.$$

Reprezentacija iz Leme 2.7.4, tvrđenja (b) dovodi do sledećeg  $\{2, 3\}$ -inverza matrice  $A$ :

$$\begin{aligned} A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2,3)} &= Q_1 (R_1 P^* A Q_1)^{-1} R_1 P^* = \begin{bmatrix} -\frac{59}{392} & -\frac{69}{196} & -\frac{39}{392} & -\frac{19}{392} & \frac{1}{392} & \frac{15}{49} \\ \frac{55}{392} & \frac{61}{196} & \frac{43}{392} & \frac{31}{392} & \frac{19}{392} & -\frac{9}{49} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ &= F(AF)^\dagger \\ &= A_{\mathcal{R}(F), \mathcal{N}((AF)^*)}. \end{aligned}$$

U nastavku navodimo algoritam za izračunavanje  $A_{T,S}^{(2)}$  inverza date matrice  $A$ .

---

**Algoritam 2.8** Izračunavanje  $A_{T,S}^{(2)}$  inverza matrice  $A$  koristeći QR dekompoziciju matrice  $W$ . (algoritam QRATS2)

---

**Ulaz:** Matrica  $A$  dimenzija  $m \times n$  i ranga  $r$ .

- 1: Odabratи proizvoljnu fiksiranu  $n \times m$  matricу  $W$  ranga  $s \leq r$ .
  - 2: Izračunati QR dekompoziciju matrice  $W$  oblika (2.7.1).
  - 3: Odreditи dekompoziciju potpunog ranga matrice  $W$  као у (2.7.4).
  - 4: Rešiti matričnu jednačину облика (2.7.10) за slučaj  $m < n$  или мatričnu jednačину (2.7.11) за slučaj  $n < m$ .
  - 5: Odreditи излазно решење  $A_{\mathcal{R}(Q_1), \mathcal{N}(R_1 P^*)}^{(2)} = A_{\mathcal{R}(Q_1), \mathcal{N}(Q_1^* W)}^{(2)}$  као у (2.7.12).
- 

### 2.7.1 Složenost i implementacija algoritama QRATS2 i qrginv

Algoritam QRATS2 koristi QR dekompoziciju матrice  $W$  облика (2.7.4) и онда израчунава  $A_{T,S}^{(2)}$  inverz на основу тврђења (b) у Леми 2.7.4. Prema томе, njегова сложеност израчунавања је једнака

$$E(QRATS2(W)) = \mathcal{M}(s, m, m) + \mathcal{M}(s, m, n) + \mathcal{M}(s, n, s) + I(s) + \mathcal{M}(n, s, s) + \mathcal{M}(n, s, m). \quad (2.7.15)$$

У Табели 2.7.1 су описане сложености алгоритма за израчунавање спонјног inverza  $X = Q_1(R_1 P^* A Q_1)^{-1} R_1 P^*$ .

Izraz	Složenost
$\Lambda_1 = R_1 P^*$	$\mathcal{M}(s, m, m)$
$\Lambda_2 = \Lambda_1 A$	$\mathcal{M}(s, m, m) + \mathcal{M}(s, m, n)$
$\Lambda_3 = \Lambda_2 Q_1$	$\mathcal{M}(s, m, m) + \mathcal{M}(s, m, n) + \mathcal{M}(s, n, s)$
$\Lambda_4 = (\Lambda_3)^{-1}$	$\mathcal{M}(s, m, m) + \mathcal{M}(s, m, n) + \mathcal{M}(s, n, s) + I(s)$
$\Lambda_5 = Q_1 \Lambda_4$	$\mathcal{M}(s, m, m) + \mathcal{M}(s, m, n) + \mathcal{M}(s, n, s) + I(s) + \mathcal{M}(n, s, s)$
$X = \Lambda_5 \Lambda_1$	$\mathcal{M}(s, m, m) + \mathcal{M}(s, m, n) + \mathcal{M}(s, n, s) + I(s) + \mathcal{M}(n, s, s) + \mathcal{M}(n, s, m)$

Tabela 2.7.1. Složenost izračunavanja  $X = Q_1(R_1 P^* A Q_1)^{-1} R_1 P^*$ .

Dobro je poznato да је inverzija матрица једнаке тежине као и мatričно množenje.

Tačnije, običan inverz realne nesingularne  $n \times n$  matrice se može izračunati u vremenu  $I(n) = \mathcal{O}(\mathcal{M}(n))$  [15]. Notacija  $\mathcal{O}(f(n))$  je objašnjena u [15], na primer. U našoj implementaciji koristimo uobičajene metode za množenje matrica, tako da je  $\mathcal{M}(m, n, k) = m \cdot n \cdot k$ . Ovo povlači, na osnovu (2.7.15)

$$E(QRATS2(W)) = 2s^2n + sm(m + 2n) + I(s) \approx 2s^2n + sm(m + 2n) + s^3. \quad (2.7.16)$$

Za poseban slučaj  $W = A^*$ , algoritam *QRATS2* uzima  $QR$  dekompoziciju matrice  $A^*$  oblika (2.7.4) i izračunava  $A^\dagger$ . Sa druge strane, algoritam *qrginv* počinje sa  $QR$  dekompozicijom matrice  $A$  oblika

$$A = Q_A R_A P_A^*$$

i izračunava Moore–Penrose-ov inverz

$$A^\dagger = P_A R_A^\dagger Q_A^*.$$

Njegova složenost izračunavanja je data sa

$$E(qrginv) = pinv(r, n) + \mathcal{M}(n, n, r) + \mathcal{M}(n, r, m) = rn(m + n) + pinv(r, n), \quad (2.7.17)$$

gde  $pinv(r, n)$  označava složenost izračunavanja pravougaone  $r \times n$  realne matrice.

Za poseban slučaj  $W = A^*$  algoritma *QRATS2*, njegova složenost je

$$E(QRATS2(A^*)) = r^2(m + 2n) + 2rmn + pinv(r, r). \quad (2.7.18)$$

Implementacija Algoritma *QRATS2* u Matlab okruženu je ovde navedena u celosti. Pritom se sistem matričnih jednačina (2.7.10) rešava korišćenjem Matlab "backslash" operatora. Svi rezultati su generisani za slučaj  $m < n$ . Slični rezultati se mogu dobiti i za slučaj  $m > n$  pod blagom modifikacijom navedenog Matlab koda.

```
function ATS2 =ATS2(A)

% Sinopsis:
% ATS2 = ATS2(A)
% Ulazni podaci:
%   A = inicijalna ulazna matrica.
%   U = A' ili A ili A^k itd. radi dobijanja A^+ ili A^# ili A^D itd.,
%   respektivno.
% Izlaz:
%   ATS2 = ATS2 koriscenjem QR faktorizacije matrice U.
%   Slučaj za Moore-Penrose-ov inverz A^+

U=A';
[Q,R,P] = qr(U);
r = rang(U);
Q1 = Q(:,1:r);
R1 = R(1:r,:);
G = R1*P';
X = (G*A*Q1)\(G);
ATS2 = Q1*X;
```

Indeks zadate matrice se može izračunati primenom sledeće Matlab funkcije:

```
function [k,r,A1] = index(A)

% Sinopsis:
% [k,r,A1] = index(A)
% Ulazni podaci:
%     A = inicijalna ulazna matrica.
% Izlaz:
%     k = indeks matrice A
%     r = rang matrice A^k
%     A1 = A^k

k = 0;
n = length(A);
r= n;
A0 = A;
r1 = rang(A);
A1 = zeros(n);
while r~=r1
r=r1;
A1 = A;
A = A * A0;
r1 = rang(A);
k = k+1;
end
```

### 2.7.2 Od QRATS2 do qrginv i obratno

Pretpostavimo da je  $AP = QR$  data  $QR$  dekompozicija definisana matricom  $Q$  sa ortonormalnim kolonama i  $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ , gde je  $R_1$  potpunog ranga vrsta (videti [108], Teoremu 3.3.11). Tada je  $AP = Q_1 R_1$ , nakon odsecanja vrsta i kolona i uvezši u obzir rang od  $A$ . Na osnovu [44] (komentar nakon Teoreme 4), zadovoljeno je

$$\begin{aligned} A^\dagger &= PR^\dagger Q^* = P \begin{bmatrix} R_1^\dagger & 0 \end{bmatrix} Q^* \\ &= PR_1^\dagger Q_1^*. \end{aligned} \tag{2.7.19}$$

U radu [80] definisane su dve reprezentacije Moore–Penrose-ovog inverza. Prva reprezentacija je definisana kao zaseban slučaj  $W = A^*$  algoritma QRATS2. Preciznije, ona kreće od  $QR$  dekompozicije  $A^* = Q_1(R_1 P^*)$  matrice  $A^*$  a onda se izračunava Moore–Penrose-ov inverz na osnovu dela (b) Leme 2.7.4.

Druga reprezentacija Moore–Penrose-ovog inverza je zasnovana na dekompoziciji matrice  $A^*$ , dobijene iz  $QR$  dekompozicije matrice  $A$ . Ako je  $AP = Q_1 R_1$ ,  $Q_1^* Q_1 = I$  tada  $A^* = (PR_1^*) Q_1^*$  je faktorizacija potpunog ranga  $A^*$ . Iz Leme 2.7.4 dobijamo

$$QRATS2(A^*) = PR_1^* (Q_1^* A P R_1^*)^{-1} Q_1^*. \tag{2.7.20}$$

Nije teško proveriti da reprezentacija (2.7.20) proizvodi isti rezultat kao i reprezentacija

(2.7.19). Zaista, primenjujući

$$\begin{aligned} QRATS2(A^*) &= PR_1^* (Q_1^* A P R_1^*)^{-1} Q_1^* \\ &= PR_1^* (Q_1^* Q_1 R_1 P^* P R_1^*)^{-1} Q_1^* \\ &= PR_1^* (R_1 R_1^*)^{-1} Q_1^* \end{aligned}$$

i uvezši u obzir da je matrica  $R_1$  potpunog ranga vrsta, sledi da je  $QRATS2(A^*) = PR_1^\dagger Q_1^*$ , što je upravo jednako sa (2.7.19).

### 2.7.3 Numerički eksperimenti za Moore-Penrose-ov inverz

U ovoj sekciji navodimo rezultate korišćenja jezika Matlab za izračunavanja, kao i za testiranje tačnosti dobijenih rezultata. Svi numerički zadaci su izvršeni koristeći Matlab R2009a okruženje na Intel(R) Pentium(R) Dual CPU T2310 1.46 GHz 1.47 GHz 32-bit sistemu sa 2 GB RAM memorije na Windows Vista Home Premium Operating sistemu.

U radu [80] izvedeni su numerički eksperimenti koji upoređuju algoritam  $QRATS2$  sa algoritmom  $qrginv$  iz [44] i sa iterativnim metodima prezentovanim od strane Wei i Wu u [110], za izračunavanje Moore-Penrose-ovog inverza singularne matrice  $A$ . U radu autora Wei-Wu, [110], nekoliko iterativnih metoda je navedeno, a nakon testiranja sa različitim skupovima matrica zaključeno je da je najpouzdanija jednačina (3.8) u [110] sa stanovišta tačnosti i vremena izvršenja. Prema tome, samo je ovaj metod iz [110] uključen u našim numeričkim testovima.

Testiranje algoritama  $QRATS2$ ,  $qrginv$  i Wei-Wu metoda je izvršeno odvojeno za nasumične singularne matrice i za singularne test matrice sa velikim kondicionim brojem iz *Matrix Computation Toolbox* (see [32]). Takođe smo testirali predloženi metod na retkim matricama iz Matrix Market kolekcije (videti [53]), i pritom su dobijeni vrlo efikasno tačni rezultati u svim slučajevima.

Konačno, navedeni su numerički rezultati za slučaj Drazinovog inverza tj. za slučaj  $W = A^k$ , gde je  $k$  indeks matrice  $A$ . Primetimo da znak ' $-$ ' označava duga procesorska vremena izvršavanja algoritma. Tolerancija za iterativni metod (3.8) iz [109] je postavljena na  $10^{-6}$ . Implementacija  $QRATS2$  metoda u Matlab-u, kao i Matlab funkcija za izračunavanje matričnog indeksa su navedene u dodatku. Rezultati za tačnost su određeni primenom sledeće četiri relacije koje karakterišu Moore-Penrose inverz:

$$AA^\dagger A = A, \quad A^\dagger AA^\dagger = A^\dagger, \quad (AA^\dagger)^* = AA^\dagger, \quad (A^\dagger A)^* = A^\dagger A.$$

#### Slučajno generisane singularne test matrice

Upoređivane su performanse predloženog metoda  $QRATS2$  sa algoritmom  $qrginv$  iz [44] i metodom navedenog jednačinom (3.8) u [109], na skupu nasumičnih singularnih matrica ranga  $2^n$ ,  $n = 8, 9, 10, 11$ .

U Tabeli 2.7.2, tačnost metoda je testirana na osnovu greške 2-norme. Kao i ranije, *rang* (resp. *Cond*) kolona sadrži rang (resp. kondicioni broj) svake testirane matrice. Evidentno je da predloženi metod  $QRATS2$  daje odličnu aproksimaciju u svakom izvedenom testu. Uvedeni metod izvodi brza i tačna izračunavanja, pa prema tome može biti dobra alternativa za izračunavanje Moore-Penrose-ovog inverza.

rang\kond.	Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
$2^8 \backslash 1.14e+04$	QRATS2	0.1796	1.1969 e-12	3.8694 e-12	3.1776 e-12	2.2822 e-12
	qrginv	0.1929	2.6305 e-12	6.4950 e-12	4.7533 e-12	5.7056 e-13
	Wei,Wu	5.4895	4.4502 e-12	4.6559 e-08	2.8868 e-13	1.1914 e-11
$2^9 \backslash 1.29e+04$	QRATS2	1.3490	6.7212 e-12	7.1291 e-11	5.9296 e-12	1.2068 e-11
	qrginv	1.2848	2.9506 e-12	2.0125 e-12	3.4956 e-12	4.4940 e-13
	Wei,Wu	45.2327	5.0381 e-11	1.2878 e-07	2.7502 e-13	6.9487 e-12
$2^{10} \backslash 6.31e+05$	QRATS2	10.0298	6.5701 e-11	1.0135 e-09	1.4507 e-10	9.7971 e-11
	qrginv	8.5909	1.4615 e-10	3.4266 e-10	1.4075 e-10	1.8104 e-11
	Wei,Wu	477.6512	2.6840 e-10	3.7996 e-10	7.5295 e-12	6.2981 e-09
$2^{11} \backslash 8.88e+04$	QRATS2	76.1483	5.6069 e-11	2.1023 e-10	5.7148 e-11	9.1536 e-11
	qrginv	64.5475	3.2243 e-11	9.3141 e-12	2.6953 e-11	2.3243 e-12
	Wei,Wu	-				

Tabela 2.7.2. Greška i vreme izračunavanja; Nasumične singularne matrice.

### Singularne test matrice iz Matrix Computation Toolbox-a

Kondicioni brojevi ovih matrica su reda koji je u opsegu od  $10^{15}$  do  $10^{135}$ . Radi uporednih rezultata smo takođe primenili QRATS2 algoritam, qrginv funkciju i Wei-Wu metod. Greške su navedene u Tabelama 2.7.3–2.7.11.

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0284	9.4187 e-13	2.0040 e-13	1.8662 e-13	1.3462 e-12
qrginv	0.0695	4.3737 e-13	1.1428 e-13	6.0657 e-13	1.4614 e-13
Wei,Wu	1.6319	3.9456 e-13	5.8480 e-13	8.9149 e-15	1.5847 e-14

Tabela 2.7.3. Greška i vreme izračunavanja za chow (rang=199, Cond=8.01849 e+135).

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0154	3.5703 e-14	8.3310 e-17	1.7152 e-15	3.8605 e-15
qrginv	0.0479	3.3422 e-14	6.3095 e-17	1.5658 e-15	1.2189 e-15
Wei,Wu	0.5201	5.1488 e-07	2.3601 e-09	2.4669 e-15	1.0209 e-15

Tabela 2.7.4. Greška i vreme izračunavanja za cycol (rang=50, Cond=2.05 e+48)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0172	4.1331 e-15	4.0777 e-13	1.0614 e-14	1.2640 e-13
qrginv	0.0683	3.3368 e-15	1.6225 e-13	9.5534 e-14	2.1089 e-14
Wei,Wu	1.2365	4.3255 e-13	4.3828 e-10	4.8506 e-15	5.3844 e-15

Tabela 2.7.5. Greška i vreme izračunavanja za gearmat (rang=199, Cond=3.504 e+17)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0199	3.1363 e-14	1.1835 e-09	6.5937 e-10	3.9813 e-14
qrginv	0.0535	2.1280 e-05	1.8366 e-09	0.6923	7.5374 e-15
Wei,Wu	4.0873	7.6304 e-15	2.9332 e-10	3.6749 e-10	3.2342 e-14

Tabela 2.7.6. Greška i vreme izračunavanja za `kahan` (rang=199, Cond=2.30018 e+24)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0912	8.5749 e-06	9.5852 e-08	3.1702 e-11	0.2384
qrginv	0.0461	8.3470 e-06	8.6095 e-09	0.0463	1.1078 e-11
Wei,Wu	-				

Tabela 2.7.7. Greška i vreme izračunavanja za `lotkin` (rang=19, Cond=8.97733 e+21)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.1151	1.3837 e-06	1.4334 e-06	5.9357 e-11	0.0477
qrginv	0.1219	1.3837 e-06	2.9980 e-07	0.0477	4.4909 e-11
Wei,Wu	-				

Tabela 2.7.8. Greška i vreme izračunavanja za `prolate` (rang=117, Cond=5.61627 e+17)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0108	7.8780 e-06	1.8662 e-07	2.0383 e-11	0.1005
qrginv	0.0565	7.8780 e-06	1.0846 e-08	0.1005	6.3029 e-12
Wei,Wu	-				

Tabela 2.7.9. Greška i vreme izračunavanja za `hilb` (rang=20, Cond=1.17164 e+19)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0856	1.0460 e-08	2.4776 e-19	7.6620 e-14	2.9086 e-13
qrginv	0.0455	1.1003 e-08	5.6690 e-19	4.0822 e-13	3.0107 e-14
Wei,Wu	1.0240	2.5179 e+06	2.5641 e-07	2.5206 e-17	1.9203 e-17

Tabela 2.7.10. Greška i vreme izračunavanja za `magic` (rang=3, Cond=4.92358 e+19)

Metod	Vreme	$\ AA^\dagger A - A\ _2$	$\ A^\dagger AA^\dagger - A^\dagger\ _2$	$\ AA^\dagger - (AA^\dagger)^*\ _2$	$\ A^\dagger A - (A^\dagger A)^*\ _2$
QRATS2	0.0570	1.2808 e-05	1.4616 e-06	1.1470e-10	0.5799
qrginv	0.0601	1.2730 e-05	2.5570 e-07	0.5330	5.4429 e-11
Wei,Wu	-				

Tabela 2.7.11. Greška i vreme izračunavanja za `vand` (rang=34, Cond=1.16262 e+20)

Interesantno je posmatrati komplementarnost algoritama *QRATS2* i *qrginv* u Tabellama 2.7.7–2.7.11. Preciznije, rezultati iz primene algoritma *QRATS2* predstavljaju  $\{1, 2, 3\}$  inverze dok *qrginv* algoritam daje rezultate koji predstavljaju  $\{1, 2, 4\}$  inverze. Algoritam *QRATS2* koristi *QR* dekompoziciju matrice  $A^*$  dok algoritam *qrginv* počinje *QR* dekompozicije matrice  $A$ . Ova činjenica može biti razlog za razlike u konvergenciji algoritma *QRATS2* za slučaj  $W = A^*$  i algoritam *qrginv*. Ako se koristi jednačina (2.7.20), tada nema komplementarnih rezultata u navedenim tabelama.

#### 2.7.4 Numerički eksperimenti za Drazinov inverz

U radu [80] dati su i numerički eksperimenti kod izračunavanja Drazinovog inverza  $A^D$  kvadratne matrice  $A$ , primenom navedenog algoritma za određivanje  $A_{T,S}^{(2)}$  inverza. U slučaju Drazinovog inverza, uzimamo da je  $W = A^k$ , gde je  $k$  indeks matrice  $A$ . Radi provere tačnosti rezultata, koristili smo sledeće tri relacije iz definicije Drazinovog inverza:

$$A^D A = AA^D, \quad A^D AA^D = A^D, \quad A^{k+1} A^D = A^k.$$

Najpre navodimo primer za izračunavanje Drazinovog inverza kvadratne matrice malih dimenzija, koristeći metod *QRATS2*.

**Primer 2.7.3.** Neka je data singularna matrica

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & -1 & -1 \end{bmatrix}.$$

Tada, indeks matrice  $A$  je jednak  $\text{ind}(A) = 2$  i

$$W = A^2 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

*QR* dekompozicija potpunog ranga matrice  $W$  je određena sa

$$W = Q_1 R_1 P^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Prema tome, na osnovu dela pod (b) Leme 2.7.4 i Posledice 2.7.5, imamo

$$A^D = Q_1 (R_1 P^* A Q_1)^{-1} R_1 P^* = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

U nastavku navodimo dve tabele kojim upoređujemo greške i vremena izračunavanja navedenog metoda sa metodom prezentovanim u [109]. Ponovo koristimo skup od osam test matrica dimenzija  $200 \times 200$  dobijenih funkcijom `matrix` u Matrix Computation

Matrica	Indeks	Rang	Metod	Vreme	$\ AA^D - A^DA\ _2$	$\ A^DAA^D - A^D\ _2$	$\ A^{k+1}A^D - A^k\ _2$
CYCOL	1	50	QRATS2 Wei,Wu	0.2035 -	2.2280 e-12 -	2.9286 e-12 -	6.3953 e-12 -
GEARMAT	2	199	QRATS2 Wei,Wu	0.3617 -	1.8144 e-10 -	3.9470 e-10 -	6.3876 e-13 -
KAHAN	25	199	QRATS2 Wei,Wu	1.1954 -	5.5349 e+03 -	9.7653 e-07 -	2.1696 e-10 -
LOTKIN	5	19	QRATS2 Wei,Wu	0.4343 -	3.2649 e-04 -	1.1547 e-13 -	5.2916 e-06 -
PROLATE	5	117	QRATS2 Wei,Wu	0.3693 -	9.2475 e-04 -	1.6430 e-14 -	5.4875 e-08 -
HILB	7	20	QRATS2 Wei,Wu	0.5537 -	5.2832 e-05 -	2.0200 e-14 -	1.9039 e-08 -
MAGIC	1	3	QRATS2 Wei,Wu	0.1868 0.1621	4.1704 e-13 6.2739 e-18	1.7757 e-17 7.3154 e-08	5.7818 e-07 2.3074 e+06
VAND	5	34	QRATS2 Wei,Wu	0.4404 -	2.2305 e+05 -	4.6483 -	2.3188 e-05 -

Tabela 2.7.12. Greške i vremena izračunavanja Drazinovog inverza.

Toolbox (Tabela 2.7.12). Koristili smo i niz nasumičnih singularnih matrica rangova  $2^n$ ,  $n = 8, 9, 10, 11$  (Tabela 2.7.13).

Kao što je očigledno iz Tabele 2.7.12, iterativni metod Wei-Wu ([109]) ne konvergira u većini slučajeva. Prema tome, on ne daje numerički tačan odgovor. Nasuprot tome,  $A_{T,S}^{(2)}$  algoritam daje veoma tačne rezultate u većini slučaja i veoma je efikasan.

Wei-Wu iterativni metod ne daje rezultate za neko razumljivo vreme za slučaj slučajno generisanih singularnih matrica, tako da je taj metod isključen iz poređenja iz Tabele 2.7.13.

Index	rang	Vreme	$\ AA^D - A^DA\ _2$	$\ A^DAA^D - A^D\ _2$	$\ A^{k+1}A^D - A^k\ _2$
1	$2^8$	0.8124	1.9117 e-10	2.4519 e-09	3.5369 e-10
1	$2^9$	2.4586	5.8145 e-11	1.4301 e-10	2.2431 e-10
1	$2^{10}$	13.4005	2.2166 e-09	1.6988 e-08	8.4017 e-09
1	$2^{11}$	187.3001	2.2031 e-09	7.4148 e-09	8.4040 e-09

Tabela 2.7.13. Greška i vreme izračunavanja; nasumične singularne matrice.

Osnovna motivacija rada [81] je reprezentacija Moore–Penrose-ovog inverza  $A^\dagger$  iz [44]. Metod uveden u [44] je dobijen iz  $QR$  dekompozicije matrice  $A$ . U radu [81] dobijene su dve ekstenzije ovog algoritma. Prva generalizacija je u tome što je naš algoritam primenljiv za izračunavanje šire klase  $A_{T,S}^{(2)}$  inverza, a ne samo za izračunavanje Moore–Penrose inverza.

## 2.8 Simboličko izračunavanje $A_{T,S}^{(2)}$ inverza QDR faktorizacijom

Umesto  $QR$  dekompozicije matrice  $A$  korišćena je  $QDR$  dekompozicija proizvoljno odabране matrice  $W$  i na taj način dobijene faktorizacije potpunog ranga. Odabir  $QDR$  dekompozicije je kritičan u eliminaciji pojavljivanja kvadratnih korena u elementima iz  $QR$  dekompozicije.

Pritom, kvadratni koreni nekih matričnih polinoma se često javljaju pri generisanju  $QR$  faktorizacije. Generisanje izraza koji sadrže kvadratne korene se može izbeći korišćenjem  $QDR$  dekompozicije. Ova tehnika je od esencijalne važnosti pri simboličkom polinomijalnom izračunavanju. Slično kao u radu [77], motivacija u radu [81] bila je iskoristiti prednosti bez-korene dekompozicije u simboličkim izračunavanjima i proširiti Algoritam QRATS2 na skup polinomijalnih matrica. Postavlja se pitanje: koji je glavni razlog za zamenu  $LDL^*$  sa  $QR$  dekompozicijom? Glavna nepogodnost  $LDL^*$  dekompozicije je što je ona primenljiva samo na simetrične pozitivno definitne matrice. Ova prepreka postavlja granicu rezultata iz [77] na skup  $\{1, 2, 3\}, \{1, 2, 4\}$ -inverza i na Moore–Penrose-ov inverz. Reprezentacije navedene u radu [81] su primenljive na širi skup spoljnih inverza definisane jezgra i slike. Iz ovih razloga, umesto  $LDL^*$  dekompozicije (korišćene u [77]), koristićemo  $QDR$  faktorizaciju racionalne matrice za izbegavanje dobijanja elemenata sa korenima. Evidentno, dobijeni algoritam je vrlo pogodan za implementaciju u proceduralnim programskim jezicima, zbog bez-korenih elemenata matrica i osnovnog metoda simplifikacije koji podrazumeva izračunavanje  $nzd$ -a dva polinoma.

Osnovna  $QDR$  faktorizacija matrice  $A$  generiše tri matrice: matricu  $Q$  ranga jednakog rangu od  $A$ , dijagonalnu matricu  $D$  i matricu  $R$ , u nekoliko etapa. Ovde je predložen algoritam koji smo uveli u [81] za direktno izračunavanje  $QDR$  dekompozicije potpunog ranga, gde je matrica  $Q$  formirana bez nula kolona,  $R$  je generisana bez nula vrsta, a dijagonalna matrica  $D$  je bez nula vrsta i nula kolona.  $QDR$  dekompozicija proizvodi jednu dodatnu dijagonalnu matricu u odnosu na  $QR$  dekompoziciju, ali vraća kao rezultat matrice sa elementima bez korena, pogodnije za dalje simboličko izračunavanje.

---

### Algoritam 2.9 $QDR$ dekompozicija potpunog ranga racionalne matrice $A$

---

**Ulaz:** Racionalna matrica  $A \in \mathbb{C}(x)_s^{n \times m}$ .

- 1: Konstruisati tri nula matrice:  $Q \in \mathbb{C}(x)^{n \times s}$ ,  $D \in \mathbb{C}(x)^{s \times s}$ ,  $R \in \mathbb{C}(x)^{s \times m}$ .
  - 2: Za  $i = \overline{1, s}$  ponavljati korake
    - 2.1: Postaviti matricu  $B$  na vrednost  $A - QDR$ .
    - 2.2: Odrediti prvu sledeću ne-nula kolonu matrice  $B$  i označiti je sa  $c$ .
    - 2.3: Postaviti  $i$ -tu kolonu matrice  $Q$  na vrednosti kolone  $c$ .
    - 2.4: Za  $j = \overline{i, m}$  postaviti  $R_{i,j}$  na vrednost unutrašnjeg proizvoda vektora  $c$  i  $j$ -te kolone od  $B$ .
    - 2.5: Postaviti element  $D_{i,i}$  na recipročnu vrednost kvadrata 2-norme kolone  $c$ .
- 

Primetimo da je matrična jednačina  $A = QDR + B$  rešavana u svakom koraku algoritma, gde se počinje od  $B = A$ . Na kraju je  $B = 0$  i  $A = QDR$ . Takođe je matrica  $R$

gornja trougaona, a kolone matrice  $Q$  sadrže orthogonalnu bazu prostora kolona matrice  $A$ . Implementacija ovog algoritma u **MATHEMATICA** je navedena u dodatku.

U mnogo slučajeva, Gram-Schmidt-ov algoritam sa pivotiranjem kolona je neophodan. U svakom koraku kolona matrice  $B$  sa najvećom 2-normom je odabrana, umesto odabira prve ne-nula kolone. Tada je matrica  $R$  sgora trougaona matrica sa permutovanim kolonama, dok kolone matrice  $Q$  ponovo again contain orthogonalnu bazu prostora kolona matrice  $A$ .

**Primer 2.8.1.** Posmatrajmo sledeće dve matrice:

$$F = \begin{bmatrix} 3 & -2 \\ -1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Odaberimo matricu  $W$  kao proizvod ove dve matrice

$$W = FG = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$QDR$  dekompozicija matrice  $W$  je određena sa

$$\begin{bmatrix} 3 & -2 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & \frac{1}{10} \\ -1 & \frac{3}{10} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & 10 \end{bmatrix} \begin{bmatrix} 10 & -7 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{10} & 0 & 0 & 0 & 0 \end{bmatrix}.$$

**Primer 2.8.2.** Neka su date sledeće polinomijalne matrice:

$$W = \begin{bmatrix} -3 - 4x^2 & 2 - 7x & 4 \\ -9x & -3 + 3x^2 & -5 \\ -2x + 9x^2 & 9x^2 & -5 \end{bmatrix}.$$

Primenivši Algoritam 2.9 dobijamo sledeće matrice iz  $QDR$  dekompozicije matrice  $W$ :

$$Q = \begin{bmatrix} -3 - 4x^2 & \frac{x(81+170x-694x^2+657x^3-747x^4+324x^5)}{9+109x^2-36x^3+97x^4} \\ -9x & -3 + 3x^2 + \frac{9x(-6+48x-8x^2-17x^3+81x^4)}{9+109x^2-36x^3+97x^4} \\ x(-2 + 9x) & \frac{x(-12+231x-448x^2+1019x^3-9x^4+144x^5)}{9+109x^2-36x^3+97x^4} \\ \frac{3x(90-217x+53x^2+3742x^3-1388x^4-3444x^5+5565x^6+2052x^7)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \\ -\frac{x(-180+2279x+3163x^2-10909x^3+8706x^4+10329x^5-14904x^6+8208x^7)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \\ \frac{3(-135-552x+675x^2+2603x^3-2674x^4-1292x^5+4108x^6-60x^7+912x^8)}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} \end{bmatrix},$$

$$D = \begin{bmatrix} \frac{1}{9+109x^2-36x^3+97x^4} & 0 & 0 \\ 0 & \frac{9+109x^2-36x^3+97x^4}{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8} & 0 \\ 0 & \frac{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8}{(45+94x-113x^2-15x^3+228x^4)^2} & 0 \end{bmatrix},$$

$$R = \begin{bmatrix} 9 + 109x^2 - 36x^3 + 97x^4 & -6 + 48x - 8x^2 - 17x^3 + 81x^4 \\ 0 & \frac{81+324x-704x^2-2956x^3+8998x^4-11880x^5+13824x^6-486x^7+2169x^8}{9+109x^2-36x^3+97x^4} \\ 0 & \frac{-12 + 55x - 61x^2}{\frac{-135-654x+1135x^2+716x^3+1882x^4+6048x^5+879x^6}{(45+94x-113x^2-15x^3+228x^4)^2}} \end{bmatrix}.$$

Mnogo je poznatih reprezentacija za različite uopštene inverze definisanog ranga kao i za uopštene inverze definisanog jezgra i slike. Najkorisnija reprezentacija za naše istraživanje je sledeća "full-rank" reprezentacija spoljnih inverza unapred navedenog jezgra i slike iz [70].

U specijalnom slučaju, reprezentacija potpunog ranga Drazinovog inverza  $A^D$ , zasnovana na proizvoljnoj dekompoziciji potpunog ranga  $A^l$ ,  $l \geq \text{ind}(A)$ , je uvedena u radu [89].

Alternativni eksplicitni izraz za uopšteni inverz  $A_{T,S}^{(2)}$ , koji je zasnovan na korišćenju grupnog inverza, je uveden u radu [110]. Karakteracija, teorema o reprezentaciji i izraz ograničenja za  $A_{T,S}^{(2)}$  su dobijeni u [110] primenom ove reprezentacije.

Autori rada [12] su izveli osnovnu reprezentaciju i uopštenu teoremu reprezentacije spoljašnjeg inverza  $A_{T,S}^{(2)}$ . Na osnovu ove reprezentacije, nekoliko specifičnih reprezentacija i iterativnih metoda za izračunavanje  $A_{T,S}^{(2)}$  je prezentovano u radu [12].

Sledeće tvrđenje određuje reprezentaciju potpunog ranga spoljnih inverza sa zadatim opsegom, nula prostorom i rangom, iste opšte forme kao i u Stavu 2.6.1. Tvrđenje je validno za racionalne matrice i zasnovano je na faktorizaciji potpunog ranga matrice  $W$  dobijenu iz  $QDR$  dekompozicije definisane u Algoritmu 2.9.

**Lema 2.8.1.** *Neka je  $A \in \mathbb{C}(x)_r^{m \times n}$  data matrica. Za proizvoljnu matricu  $W \in \mathbb{C}(x)_s^{n \times m}$ ,  $s \leq r$ , posmatrajmo njenu  $QDR$  dekompoziciju dobijenu Algoritmom 2.9, oblika*

$$W = QDR,$$

gde je  $Q \in \mathbb{C}(x)_s^{n \times s}$ ,  $D \in \mathbb{C}(x)_s^{s \times s}$  je dijagonalna matrica a  $R \in \mathbb{C}(x)_s^{s \times m}$  je gornja trougaona matrica. Neka je zadovoljen uslov

$$\text{rang}(W) = \text{rang}(RAQ) = s. \quad (2.8.1)$$

Definišimo skup

$$\mathbb{C}_s(W) = \{x_c \in \mathbb{C} \mid \text{rang}(W) = \text{rank}(W(x_c)) = \text{rank}(R(x_c)A(x_c)Q(x_c)) = s\}. \quad (2.8.2)$$

Tada je sledeće tvrđenje zadovoljeno na skupu  $\mathbb{C}_s(W)$ :

$$\begin{aligned} A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} &= Q(RAQ)^{-1}R \\ &= A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}. \end{aligned} \quad (2.8.3)$$

**Dokaz.** Očigledno, faktorizacija

$$W = QDR = (QD)(R), \quad (2.8.4)$$

predstavlja faktorizaciju potpunog ranga matrice  $W$  na skupu  $\mathbb{C}_s(W)$ . S obzirom da su  $D$  i  $RAQ$  invertibilne matrice, one zadovoljavaju svojstvo obrnutog poretka, tj.  $(RAQD)^{-1} = D^{-1}(RAQ)^{-1}$ . Sada, prvi identitet u (2.8.3) sledi iz Stava 2.6.1 i identiteta

$$\begin{aligned} QD(RAQD)^{-1}R &= Q(RAQ)^{-1}R \\ &= A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}. \end{aligned}$$

Evidentno je sada

$$A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)} = A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}$$

zadovoljeno na skupu  $\mathbb{C}_s(W)$  iz (2.8.4).  $\square$

**Napomena 2.8.1.** Primetimo da za datu matricu  $A \in \mathbb{C}(x)_r^{m \times n}$  proizvoljno odabrana matrica  $W \in \mathbb{C}(x)_s^{n \times m}$ ,  $s \leq r$ , proizvodi odgovarajući spoljni inverz zadatog jezgra i slike oblika (2.8.3), gde je (2.8.4)  $QDR$  dekompozicija matrice  $W$ . Spoljni inverz  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}$  predstavlja funkciju na skupu  $\mathbb{C}(x)$ . Elementi spoljnašnjeg inverza, označeni sa  $g_{ij}$ , su takođe funkcije na skupu  $\mathbb{C}(x)$ . Tada je domen funkcije  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}$  određen kao  $C_s(W) \bigcap_{i,j} \text{Dom}(g_{ij})$ , gde  $\text{Dom}(g_{ij})$  označava domen funkcije  $g_{ij}$ .

Uvezši u obzir reprezentacije (1.1.1)–(1.1.3) za glavne spoljne inverze, dobijamo sledeće reprezentacije.

**Posledica 2.8.2.** Za datu matricu  $A \in \mathbb{C}(x)_r^{m \times n}$  i proizvoljno odabranu matricu  $W \in \mathbb{C}(x)_s^{n \times m}$  sa  $QDR$  dekompozicijom definisanom u (2.8.4) Sledеća tvrđenja su validna na skupu  $\mathbb{C}_s(W) \bigcap_{i,j} \text{Dom}(g_{ij})$ :

- (a)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A^\dagger$  za slučaj  $W = A^*$ ;
- (b)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A_{M,N}^\dagger$  za slučaj  $W = A^\sharp$ ;
- (c)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A^\#$  za slučaj  $W = A$ ;
- (d)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A^D$  za slučaj  $W = A^k$ ,  $k \geq \text{ind}(A)$ ;
- (e)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A_{(L)}^{(-1)}$  za slučaj  $\mathcal{R}(W) = L$ ,  $\mathcal{N}(W) = L^\perp$ ;
- (f)  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A_{(L)}^{(\dagger)}$  za slučaj  $\mathcal{R}(W) = S$ ,  $\mathcal{N}(W) = S^\perp$ .

Numerički vrlo stavljeni pristup za izračunavanje (2.8.3) je rešavanje skupa jednačina

$$RAQX = R \quad (2.8.6)$$

i potom izračunavanje proizvoda matrica

$$A_{\mathcal{R}(Q),\mathcal{N}(R)}^{(2)} = QX. \quad (2.8.7)$$

---

**Algoritam 2.10** Izračunavanje inverzne matrice  $A_{T,S}^{(2)}$  korišćenjem QDR faktorizacije  
**(Algoritam QDRATS2)**

---

**Ulaz:** Racionalna matrica  $A(x) \in \mathbb{C}(x)_r^{m \times n}$ .

- 1: Odabratи proizvoljnu fiksiranu  $n \times m$  matricu  $W$  normalnog ranga  $s \leq r$ .
- 2: Generisati QDR dekompoziciju potpunog ranga matrice  $W$  primenom Algoritma 2.9.  
Transformisati racionalne matrice  $Q, R$  u opšti oblik (2.7.1).
- 3: Transformisati racionalnu matricu  $M = RAQ$  u oblik:

$$M = \frac{1}{p(x)} M_1,$$

gde je  $p(x)$  odgovarajući polinom a  $M_1$  polinomijalna matrica.

- 4: Naći inverz matrice  $M_1$  koristeći Algoritam 3.2 iz [98]. Generisati inverz matrice  $N = M_1^{-1} = p(x)M_1^{-1}$ , i transformisati ga u oblik:

$$N_{ij}(x) = \frac{\sum_{k=0}^{\bar{n}_q} \bar{n}_{k,i,j} x^k}{\sum_{k=0}^{\bar{\bar{n}}_q} \bar{\bar{n}}_{k,i,j} x^k}.$$

- 5: Uvesti smene  $\bar{\alpha}_q = \bar{q}_q + \bar{n}_q + \bar{r}_q$ ,  $\bar{\bar{\alpha}}_q = \bar{\bar{q}}_q + \bar{\bar{n}}_q + \bar{\bar{r}}_q$ , i za  $i = \overline{1,n}$ ,  $j = \overline{1,m}$  ponoviti Korake 5.1 – 5.5.

5.1: Za  $k = \overline{1, \min\{j, s\}}$ ,  $l = \overline{1, s}$  izvršiti sledeća izračunavanja:

$$\begin{aligned} \bar{\alpha}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{q}_{t_2,i,l} \bar{n}_{t_1-t_2-t_3,l,k} \bar{r}_{t_3,k,j}, \quad 0 \leq t \leq \bar{\alpha}_q, \\ \bar{\bar{\alpha}}_{t,i,j,k,l} &= \sum_{t_2=0}^{t_1} \sum_{t_3=0}^{t_1-t_2} \bar{\bar{q}}_{t_2,i,l} \bar{\bar{n}}_{t_1-t_2-t_3,l,k} \bar{\bar{r}}_{t_3,k,j}, \quad 0 \leq t \leq \bar{\bar{\alpha}}_q. \end{aligned}$$

5.2: Odrediti polinom imenioca  $(i, j)$ -tog elementa  $A_{\mathcal{R}(Q),\mathcal{N}(R)}^{(2)}$  kao

$$\text{PolinomijalniNZS} \left\{ \sum_{t=0}^{\bar{\alpha}_q} \bar{\alpha}_{t,i,j,k,l} x^t \mid k = \overline{1, \min\{j, s\}}, l = \overline{1, s} \right\}, \quad (2.8.8)$$

i označiti ga sa  $\bar{\Gamma}_{ij}(x) = \sum_{t=0}^{\bar{\alpha}_q} \bar{\alpha}_{t,i,j,k,l} x^t$ .

5.3: Za svako  $k = \overline{1, \min\{j, s\}}$ ,  $l = \overline{1, s}$  izračunati polinom  $\bar{\Gamma}_{ij}(x) / \sum_{t=0}^{\bar{\alpha}_q} \bar{\alpha}_{t,i,j,k,l} x^t$  i označiti ga kao  $\Gamma_{i,j,k,l}(x) = \sum_{t=0}^{\bar{\Gamma}_q - \bar{\alpha}_q} \gamma_{t,i,j,k,l} x^t$ .

5.4: Izračunati polinom brojioca  $(i, j)$ -tog elementa inverza  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}$  na sledeći način

$$\bar{\Gamma}_{ij}(x) = \sum_{t=0}^{\bar{\Gamma}_q - \bar{\alpha}_q + \bar{\alpha}_q} \left( \sum_{k=1}^{\min\{j,s\}} \sum_{l=1}^s \sum_{t_1=0}^t \bar{\alpha}_{t_1, i, j, k, l} \gamma_{t-t_1, i, j, k, l} \right) x^t,$$

5.5: Postaviti  $(i, j)$ -ti element matrice  $A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)}$  na vrednost  $\bar{\Gamma}_{ij}(x) / \bar{\Gamma}_{ij}(x)$ .

### 2.8.1 Eksperimenti sa polinomijalnim i racionalnim matricama

U sledećih nekoliko primera ćemo ilustrovati naš algoritam i onda testirati neke implementacije radi upoređivanja vremena izvršavanje na različitim test matricama.

**Primer 2.8.3.** Posmatrajmo sledeće polinomijalne matrice

$$A = \begin{bmatrix} -4x^2 - 3 & 2 - 7x & 4 \\ -9x & 3x^2 - 3 & -5 \\ 9x^2 - 2x & 9x^2 & -5 \\ -4x^2 - 3 & 2 - 7x & 4 \end{bmatrix}, \quad W = \begin{bmatrix} 3 & 7x & 4 & 5 \\ -9x & 3x^2 - 3 & 5 & x + 5 \\ -6 & -14x & -8 & -10 \end{bmatrix}.$$

Matrica  $W$  je odabrana sasvim proizvoljno, ali sa odgovarajućim dimenzijama. Imamo da je  $r = \text{rang}(A) = 3$ ,  $s = \text{rang}(W) = 2$ . Algoritam 2.9 daje sledeću  $QDR$  faktorizaciju od  $W$ :

$$Q = \begin{bmatrix} 3 & \frac{9x(8x^2-1)}{9x^2+5} \\ -9x & \frac{15(8x^2-1)}{9x^2+5} \\ -6 & -\frac{18x(8x^2-1)}{9x^2+5} \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{81x^2+45} & 0 \\ 0 & \frac{9x^2+5}{45(1-8x^2)^2} \end{bmatrix},$$

$$R = \begin{bmatrix} 9(9x^2+5) & 3x(44-9x^2) & 60-45x & 75-9x(x+5) \\ 0 & \frac{45(1-8x^2)^2}{9x^2+5} & \frac{15(12x+5)(8x^2-1)}{9x^2+5} & \frac{15(16x+5)(8x^2-1)}{9x^2+5} \end{bmatrix}.$$

Izraz `X = QDRAlgorthm[A, W]` dovodi do sledećeg spoljašnjeg inverza matrice  $A$ :

$$X = \begin{bmatrix} \frac{3(72x^4+108x^3-148x^2-3x+19)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{108x^4-875x^3+297x^2+98x-48}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{3(172x^3-241x^2+39x+35)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{212x^4+199x^3+702x^2-59x-144}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{6(72x^4+108x^3-148x^2-3x+19)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{2(108x^4-875x^3+297x^2+98x-48)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{3(72x^4+108x^3-148x^2-3x+19)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{108x^4-875x^3+297x^2+98x-48}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{3(172x^3-241x^2+39x+35)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{212x^4+199x^3+702x^2-59x-144}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \\ \frac{6(72x^4+108x^3-148x^2-3x+19)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} & \frac{2(108x^4-875x^3+297x^2+98x-48)}{636x^6+777x^5+9129x^4-9265x^3-198x^2+749x+352} \end{bmatrix}.$$

Na osnovu Leme 2.8.1 dobijamo

$$X = A_{\mathcal{R}(Q), \mathcal{N}(R)}^{(2)} = A_{\mathcal{R}(W), \mathcal{N}(W)}^{(2)}.$$

Koristeći MATHEMATICA funkciju `NullSpace` dobijamo

$$\mathcal{N}(R) = \begin{bmatrix} -\frac{8x^2-35x-15}{9(8x^2-1)} & -\frac{16x+5}{3(8x^2-1)} & 0 & 1 \\ -\frac{12x^2-35x-12}{9(8x^2-1)} & -\frac{12x+5}{3(8x^2-1)} & 1 & 0 \end{bmatrix}.$$

Takođe, jednostavno se proverava da je

$$\mathcal{R}(Q) = \left\{ \frac{9x(8x^2-1)z}{9x^2+5} + 3y, \frac{15(8x^2-1)z}{9x^2+5} - 9xy, -\frac{18x(8x^2-1)z}{9x^2+5} - 6y \right\},$$

pri čemu su  $y, z$  proizvoljni kompleksni brojevi.

Sa druge strane, izraz `QDRAlgorthm[A, Transpose[A]]` povlači izračunavanja koja odgovaraju slučaju  $W = A^T$ , i generiše Moore–Penrose inverz matrice  $A$ :

$$A^\dagger = \begin{bmatrix} -\frac{15(2x^2+1)}{456x^4-30x^3-226x^2+188x+90} & \frac{-36x^2+35x-10}{228x^4-15x^3-113x^2+94x+45} & \frac{12x^2-35x-2}{228x^4-15x^3-113x^2+94x+45} \\ -\frac{5x(9x+7)}{456x^4-30x^3-226x^2+188x+90} & \frac{16x^2-8x-15}{228x^4-15x^3-113x^2+94x+45} & \frac{20x^2+36x+15}{228x^4-15x^3-113x^2+94x+45} \\ -\frac{3x(9x^3+25x^2-9x+2)}{456x^4-30x^3-226x^2+188x+90} & \frac{x(36x^3-63x^2+59x-4)}{228x^4-15x^3-113x^2+94x+45} & \frac{3(4x^4+20x^2-6x-3)}{228x^4-15x^3-113x^2+94x+45} \\ -\frac{15(2x^2+1)}{456x^4-30x^3-226x^2+188x+90} & \frac{456x^4-30x^3-226x^2+188x+90}{5x(9x+7)} & \frac{456x^4-30x^3-226x^2+188x+90}{5x(9x+7)} \\ -\frac{5x(9x+7)}{456x^4-30x^3-226x^2+188x+90} & -\frac{3x(9x^3+25x^2-9x+2)}{456x^4-30x^3-226x^2+188x+90} & -\frac{3x(9x^3+25x^2-9x+2)}{456x^4-30x^3-226x^2+188x+90} \end{bmatrix}.$$

Isti rezultat se dobija i za  $W = A$  i  $W = A^k$ ,  $k \geq 2$ , s obzirom da su Drazinov i grupni inverz jednaki sa to  $A^\dagger$ .

U sledećem primeru ćemo posmatrati racionalnu matricu čiji su Moore–Penrose-ov inverz i grupni inverz različiti.

**Primer 2.8.4.** Sledеća racionalna matrica

$$A_1 = \begin{bmatrix} \frac{144(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 0 & \frac{108+175x+4x^2}{252+175x-1148x^2} \\ \frac{75+372x}{252+175x-1148x^2} & 1 & 0 & \frac{3(25+124x)}{7(-36-25x+164x^2)} \\ -\frac{99(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 1 & \frac{99(-1+8x^2)}{7(-36-25x+164x^2)} \\ \frac{144(-1+8x^2)}{7(-36-25x+164x^2)} & 0 & 0 & \frac{108+175x+4x^2}{252+175x-1148x^2} \end{bmatrix}$$

je jednaka proizvodu  $AX_1$  matrice  $A$  iz Primera 2.8.3 i njenog spoljnog inverza  $X_1$ :

$$X_1 = \begin{bmatrix} -\frac{6(-452+395x+3418x^2-6852x^3+7344x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{-10+35x-36x^2}{45+94x-113x^2-15x^3+228x^4} \\ \frac{6(435+784x-2434x^2+976x^3+6000x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{-15-8x+16x^2}{45+94x-113x^2-15x^3+228x^4} \\ \frac{3(-297-982x+6631x^2+9197x^3-34020x^4+33264x^5+7200x^6)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} & \frac{x(4-59x+63x^2-36x^3)}{45+94x-113x^2-15x^3+228x^4} \\ \frac{-2-35x+12x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{3(356+1665x+3616x^2-11954x^3+3208x^4)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \\ \frac{15+36x+20x^2}{45+94x-113x^2-15x^3+228x^4} & \frac{-2610-13524x-2861x^2+26449x^3+15660x^4}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \\ \frac{3(-3-6x+20x^2+4x^4)}{45+94x-113x^2-15x^3+228x^4} & \frac{3(297+478x-4713x^2-11626x^3+17045x^4-6139x^5+3132x^6)}{7(-1620-4509x+9098x^2+18781x^3-26365x^4-8160x^5+37392x^6)} \end{bmatrix}.$$

Matrica  $A_1$  je idempotentna, i očigledno je  $\text{ind}(A_1) = 1$ . Moore-Penrose inverz  $A_1$  je generisan za slučaj  $W = A_1^T$ , i jednak je sledećem

$$A_1^\dagger = \left[ \begin{array}{c} \frac{9(2873+4500x-26336x^2-11200x^3+108320x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} \\ \frac{3(-900-89x+50600x^2+143344x^3)}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{99(36-175x-1444x^2+1400x^3+9248x^4)}{42642+118800x-110783x^2-200200x^3+622672x^4} \\ \frac{95652+187200x-637438x^2+2800x^3+3908768x^4}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{99(-108-175x+860x^2+1400x^3+32x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} \\ -\frac{297(-25-124x+200x^2+992x^3)}{47826+93600x-318719x^2+1400x^3+1954384x^4} \\ -\frac{47826+93600x-318719x^2+1400x^3+1954384x^4}{38025+93600x-161903x^2+1400x^3+1327120x^4} \\ \frac{47826+93600x-318719x^2+1400x^3+1954384x^4}{14256(1-8x^2)^2} \\ \frac{47826+93600x-318719x^2+1400x^3+1954384x^4}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{9(2873+4500x-26336x^2-11200x^3+108320x^4)}{47826+93600x-318719x^2+1400x^3+1954384x^4} \\ \frac{3(-900-89x+50600x^2+143344x^3)}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{99(36-175x-1444x^2+1400x^3+9248x^4)}{95652+187200x-637438x^2+2800x^3+3908768x^4} \\ \frac{42642+118800x-110783x^2-200200x^3+622672x^4}{95652+187200x-637438x^2+2800x^3+3908768x^4} \end{array} \right].$$

Za  $W = A_1$ , dobijamo grupni inverz koji je jednak početnoj matrici  $A_1$ .

**Primer 2.8.5.** Neka je data matrica

$$A = \begin{bmatrix} x-1 & x-1 & 2x-2 \\ x & x & x \end{bmatrix}$$

i odaberimo matricu  $W = A^*$ . Tada važi da je  $\text{rang}(W) = 2$  i  $C_2(W) = \mathbb{C} \setminus \{1, 0\}$ , s obzirom da je

$$\text{rank}(W(1)) = \text{rank}(W(0)) = 1 < 2.$$

Notacije  $W(1)$  i  $W(0)$  označavaju konstantne matrice dobijene zamenom simbola  $x$  vrednostima  $x = 1$  i  $x = 0$ , respektivno. Moore-Penrose-ov inverz matrice  $A$  je dat sa

$$A^\dagger = \begin{bmatrix} \frac{1}{2-2x} & \frac{1}{x} \\ \frac{1}{2-2x} & \frac{1}{x} \\ \frac{1}{x-1} & -\frac{1}{x} \end{bmatrix}.$$

Očito  $A^\dagger$  nije definisano za slučaj  $x \in \{1, 0\}$  (ili ekvivalentno, za slučaj  $x \notin C_2(W)$ ).

## 2.8.2 Implementacioni detalji i rezultati testiranja

Složenost  $QDR$  dekompozicije je jednaka složenosti  $QR$  dekompozicije. Implementacija Algoritma  $QDRATS2$  je urađena u paketu MATHEMATICA. Kreirane su dve glavne funkcije, nazvane `QDRDecomposition[A_List]` i `QDRAlgortithm[A_List]`, za testiranje i proveru.

Normalni rang zadate racionalne matrice je određen standardnom MATHEMATICA funkcijom `MatrixRank`, koja radi sa numeričkim, kao i sa simboličkim matricama [113]. Algoritam 2.9 je implementiran sledećom MATHEMATICA funkcijom.

```

QDRDecomposition[A_List] :=

Module[{i, j, n = Length[A[[1]]], m = Length[A], Q, c = 0, Diag, R, B, A1}

Q = Table[0, {n}, {m}]; Diag = R = Table[0, {n}, {n}]; B = A;
For[i = 1, i ≤ n, i++,
A1 = Transpose[B];
While[(i ≤ n) && (Norm[A1[[i]]] == 0), c++; i++];
c = 0;
If[i > n, Break[]];
NextCol = A1[[i]];
Q[[i]] = Simplify[A1[[i]]];
For[j = i, j ≤ n, j++,
R[[i]][[j]] = Simplify[NextCol.A1[[j]]];
Diag[[i]][[i]] = Simplify[1 / (Norm[NextCol]^2), Element[x, Reals]];
B = A - Transpose[Q].Diag.R;
];
Q = Drop[Q, -c]; Diag = Drop[Diag, -c];
Diag = Drop[Transpose[Diag], -c]; R = Drop[R, -c];
Return[{ExpandNumerator[Together[Transpose[Q]]],
ExpandNumerator[ExpandDenominator[Diag]], ExpandNumerator[R]}];
];

```

Ovde navodimo i implementaciju Algoritma 2.10, nazvanog *QDRATS2*, u MATHEMATICA. Primetimo da funkcija ima samo jedan parametar  $A$ . Pritom se  $W$  uzima kao transponovana matrica od  $A$ , generišući odgovarajući uopšteni inverz.

Implementacija Algoritma *QDRATS* je data sledećim prostim kodom, pri čemu se matrična jednačina (2.8.6) rešava standardnom **MATHEMATICA** funkcijom **Inverse**:

```
QDRATS[A_List, W_List] :=
Module[{N1, Q, Diag, R},
{Q, Diag, R} = QDRDecomposition[W];
N1 = Inverse[R.A.Q] // Simplify;
Return[Simplify[Q.N1.R]];
];
```

**Primer 2.8.6.** Upoređivanje različitih algoritama za simboličko izračunavanje Moore–Penrose-ovog inverza je navedeno u Tabeli 2.8.1. CPU vreme je uzeto kao kriterijum za upoređivanje navedenih algoritama. Algoritam *QDRATS2* je testiran za poseban slučaj  $W = A^*$ . Procesorska vremena su dobijena primenom **MATHEMATICA** implementacije različitih algoritama za izračunavanje pseudoinverse na nekim test matricama iz [117].

Prva vrsta tabele sadrži naziv test matrice iz [117], koje je generisao Zielke, pri čemu su tri grupe test matrica ( $A$ ,  $S$  and  $F$ ) uzete u razmatranje. Poslednja vrsta sadrži vremena dobijena *QDRATS2* algoritmom. Primetimo da je *QDRATS2* algoritam manje efikasan nego partitioning metod ili Leverier-Faddeev algoritam, zbog nekoliko matričnih množenja, gde među-rezultati i koeficijenti mogu veoma narasti. U odnosu na algoritme bazirane na Cholesky i  $LDL^*$  dekompoziciji, naš algoritam je superioran.

Test matrica [117]	$A_5$	$A_6$	$A_7$	$S_5$	$S_6$	$S_7$	$F_5$	$F_6$	$F_7$
PseudoInverse [113]	0.3	0.6	1.1	0.1	0.5	0.9	0.5	0.9	1.4
Partitioning [87]	0.1	0.2	0.5	0.1	0.2	0.4	0.1	0.2	0.6
Lev.-Faddeev [41, 88]	0.0	0.1	0.3	0.0	0.1	0.2	0.0	0.1	0.4
LDLGInverse [77]	8.8	73.5	1535.4	6.4	300.2	—	12.1	200.4	—
ModCholesky [91]	9.0	79.8	1588.8	8.7	323.5	—	13.2	212.1	—
<b>QDRATS</b>	1.2	13.9	197.2	0.1	1.1	73.0	1.5	16.4	235.8
<b>QDRATS2</b>	1.1	13.3	194.3	0.1	0.9	70.2	1.4	15.2	231.2

Tabela 2.8.1. Prosečna vremena izračunavanja (u sekundama) nekoliko algoritama i Algoritma 2.10.

## 2.9 Uopšteni inverzi blokovskih matrica

**Teorema 2.9.1.** Za zadatu matricu  $A \in \mathbb{C}_r^{m \times n}$  postoje regularne matrice  $R$  i  $G$ , permutacione matrice  $E$  i  $F$  i unitarne matrice  $U$  i  $V$ , odgovarajućih dimenzija, tako da važi:

$$(T_1) \quad RAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1 \quad (T_2) \quad RAG = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2$$

$$(T_3) \quad RAF = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_3 \quad (T_4) \quad EAG = \begin{bmatrix} I_r & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_4$$

$$(T_5) \quad UAG = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1 \quad (T_6) \quad RAV = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1$$

$$(T_7) \quad UAV = \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_2 \quad (T_8) \quad UAF = \begin{bmatrix} B & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_5$$

$$(T_9) \quad EAV = \begin{bmatrix} B & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} = N_6$$

$$(T_{10a}) EAF = \begin{bmatrix} A_{11} & A_{11}T \\ SA_{11} & SA_{11}T \end{bmatrix},$$

gde su  $S$  i  $T$  multiplikatori dobijeni iz uslova [58, 101]

$$T = A_{11}^{-1}A_{12}, \quad S = A_{21}A_{11}^{-1};$$

$$(T_{10b}) \quad EAF = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{21}A_{11}^{-1}A_{12} \end{bmatrix}.$$

$T_{11}$  Transformacija sličnosti kvadratnih matrica [66]

$$RAR^{-1} = RAEE^*R^{-1} = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} E^*R^{-1} = \begin{bmatrix} T_1 & T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}.$$

Regularna matrica  $R$  i permutaciona matrica  $E$  mogu se dobiti izvršavanjem transformacija koje su analogne transformacijama nad vrstama matrice  $A$ , nad jediničnom matricom  $I_m$ , dok se regularna matrica  $G$  i permutaciona matrica  $F$  mogu se dobiti sprovođenjem analognih transformacija koje su sprovedene nad kolonama matrice  $A$ , na jediničnoj matrici  $I_n$ .

Implementacija blokovske reprezentacije ( $T_3$ ) u MATHEMATICA data je u sledećoj proceduri.

```
t3[a_]:=Block[{m,n,r,f,b=a,i=1,j,max=1,g,h,p1,p2},
{m,n}=Dimensions[b];
r=IdentityMatrix[m]; f=IdentityMatrix[n];
While[max != 0,
max=Abs[b[[i,i]]]; p1=i; p2=i;
Do[Which[max<Abs[b[[g,h]]],
max=Abs[b[[g,h]]];p1=g;p2=h
], {g,i+1,m},{h,i+1,n}];
If[max!=0,
r=ChRow[r,i,p1];f=ChColumn[f,i,p2];
b=ChRow[b,i,p1];b=ChColumn[b,i,p2];
r=MultRow[r,i,1/b[[i,i]]];
b=MultRow[b,i,1/b[[i,i]]];
Do[Which[j!=i,r=AddRow[r,j,i,-b[[j,i]]];
b=AddRow[b,j,i,-b[[j,i]]]], {j,m}];
i++ ];
{r,f}] ;MatrixQ[a]
```

Blokovska dekompozicija ( $T_1$ ) može se generisati primenom transformacije ( $T_3$ ) dva puta:

$$R_1 A F_1 = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_3, \quad R_2 N_3^T F_2 = \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} = N_1.$$

Tada, regularne matrice  $R, G$  mogu da se izračunaju kako sledi:

$$N_1 = N_1^T = F_2^T N_3 R_2^T = F_2^T R_1 A F_1 R_2^T \Rightarrow R = F_2^T R_1, \quad G = F_1 R_2^T.$$

Implementacija reprezentacije ( $T_1$ ) se time može realizovati na sledeći način

```
t1[a_]:=  
  Block[{r1,f1,r2,f2,b=a},  
    {r1,f1}=t3[b];  
    {r2,f2}=t3[Transpose[r1.b.f1]];  
    {Transpose[f2].r1,f1.Transpose[r2]}  
  ];MatrixQ[a]
```

Blokovska dekompozicija ( $T_4$ ) može se implementirati pomoću transformacije ( $T_3$ ) na matrici  $A^T$ :

$$R_1 A^T F_1 = \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix}.$$

Prema tome,

$$F_1^T A R_1^T = \begin{bmatrix} I_r & \mathbb{O} \\ K_1^T & \mathbb{O} \end{bmatrix},$$

što povlači  $E = F_1^T$ ,  $G = R_1^T$ .

```
t4[a_]:=  
  Block[{r1,f1,b=a},  
    {r1,f1}=t3[Transpose[b]];  
    {Transpose[f1],Transpose[r1]}  
  ];MatrixQ[a]
```

**Primer 2.9.1.** Neka je data matrica

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 6 & 3 \\ -1 & 4 & 2 \end{bmatrix}$$

Niz transformacija je sledeći:

$$\begin{bmatrix} 1 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 6 & 3 & 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & 4 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 0 & 3 & 0 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 4 & -1 & 2 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & -\frac{1}{3} & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -\frac{2}{3} & 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & -\frac{1}{3} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Iz druge i treće matrice čitamo  $R$  i  $F$ :

$$R = \begin{bmatrix} 1 & \frac{1}{6} & 0 \\ 1 & -\frac{1}{3} & 0 \\ 1 & -1 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Sledeći metod za izračunavanje matrica  $S, T$  i  $A_{11}^{-1}$  dat je u [58]:

$$R = \begin{bmatrix} A_{11} & A_{12} & I \\ A_{21} & A_{22} & \mathbb{O} \end{bmatrix} \rightarrow \begin{bmatrix} I_r & A_{11}^{-1}A_{12} & A_{11}^{-1} \\ \mathbb{O} & A_{22} - SA_{12} & -S \end{bmatrix} = \begin{bmatrix} I_r & T & A_{11}^{-1} \\ \mathbb{O} & \mathbb{O} & -S \end{bmatrix}.$$

Očigledno je  $T = K$ .

**Primer 2.9.2.** Za matricu

$$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$$

može se dobiti

$$A_{11}^{-1} = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix}.$$

### 2.9.1 Blokovske reprezentacije generalisanih inverza

Iz blokovskih dekompozicija matrica sledi čitav niz reprezentacija za različite klase generalisanih inverza. Dalje navodimo prikaz nekih poznatih blokovskih reprezentacija za različite klase generalisanih inverza, koje su razvijene u [71, 118].

**Teorema 2.9.2.** Ako je  $A \in \mathbb{C}_r^{m \times n}$ , i ako su  $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$  i  $G = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$ , matrice kojima se  $A$  transformiše u normalnu formu  $A = R^{-1} \begin{bmatrix} I_r & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} G^{-1}$  tada je:

$$\begin{aligned} A^{(1)} &= G \begin{bmatrix} I_r & X \\ Y & Z \end{bmatrix} R, \\ A^{(1,2)} &= G \begin{bmatrix} I_r & X \\ Y & YX \end{bmatrix} R = G \begin{bmatrix} I_r \\ Y \end{bmatrix} \begin{bmatrix} I_r & X \end{bmatrix} R, \end{aligned}$$

$$\begin{aligned}
A^{(1,3)} &= G \begin{bmatrix} I_r & -R_1 R_2^\dagger \\ Y & Z \end{bmatrix} R, \\
A^{(1,4)} &= G \begin{bmatrix} I_r & X \\ -G_2^\dagger G_1 & Z \end{bmatrix} R, \\
A^{(1,2,3)} &= G \begin{bmatrix} I_r & -R_1 R_2^\dagger \\ Y & -Y R_1 R_2^\dagger \end{bmatrix} R = G \begin{bmatrix} I_r \\ Y \end{bmatrix} [ I_r, -R_1 R_2^\dagger ] R, \\
A^{(1,2,4)} &= G \begin{bmatrix} I_r & X \\ -G_2^\dagger G_1 & -G_2^\dagger G_1 X \end{bmatrix} R = G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} [ I_r, X ] R, \\
A^{(1,3,4)} &= G \begin{bmatrix} I_r & -R_1 R_2^\dagger \\ -G_2^\dagger G_1 & Z \end{bmatrix} R, \\
A^\dagger &= G \begin{bmatrix} I_r & -R_1 R_2^\dagger \\ -G_2^\dagger G_1 & G_2^\dagger G_1 R_1 R_2^\dagger \end{bmatrix} R = G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} [ I_r, -R_1 R_2^\dagger ] R \\
&= (G_1 - G_2 G_2^\dagger G_1)(R_1 - R_1 R_2^\dagger R_2).
\end{aligned}$$

**Teorema 2.9.3.** Ako je  $A \in \mathbb{C}_r^{m \times n}$ , i ako su  $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$  i  $G = [ G_1 \ G_2 ]$ , matrice kojima se  $A$  transformiše u normalnu formu  $A = R^{-1} \begin{bmatrix} B & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} G^{-1}$  tada je:

$$\begin{aligned}
A^{(1)} &= G \begin{bmatrix} B^{-1} & X \\ Y & Z \end{bmatrix} R, \\
A^{(1,2)} &= G \begin{bmatrix} B^{-1} & X \\ Y & YBX \end{bmatrix} R = G \begin{bmatrix} B^{-1} \\ Y \end{bmatrix} [ I_r, BX ] R, \\
A^{(1,3)} &= G \begin{bmatrix} B^{-1} & -B^{-1} R_1 R_2^\dagger \\ Y & Z \end{bmatrix} R, \\
A^{(1,4)} &= G \begin{bmatrix} B^{-1} & X \\ -G_2^\dagger G_1 B^{-1} & Z \end{bmatrix} R, \\
A^{(1,2,3)} &= G \begin{bmatrix} B^{-1} & -B^{-1} R_1 R_2^\dagger \\ Y & -Y R_1 R_2^\dagger \end{bmatrix} R = G \begin{bmatrix} B^{-1} \\ Y \end{bmatrix} [ I_r, -R_1 R_2^\dagger ] R, \\
A^{(1,2,4)} &= G \begin{bmatrix} B^{-1} & X \\ -G_2^\dagger G_1 & -G_2^\dagger G_1 X \end{bmatrix} R = G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} [ B^{-1}, X ] R, \\
A^{(1,3,4)} &= G \begin{bmatrix} B^{-1} & -B^{-1} R_1 R_2^\dagger \\ -G_2^\dagger G_1 B^{-1} & Z \end{bmatrix} R, \\
A^\dagger &= G \begin{bmatrix} B^{-1} & -B^{-1} R_1 R_2^\dagger \\ -G_2^\dagger G_1 B^{-1} & G_2^\dagger G_1 R_1 B^{-1} R_2^\dagger \end{bmatrix} R = G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} B^{-1} [ I_r, -R_1 R_2^\dagger ] R, \\
&= (G_1 - G_2 G_2^\dagger G_1) B^{-1} (R_1 - R_1 R_2^\dagger R_2).
\end{aligned}$$

**Teorema 2.9.4.** Ako je  $A \in \mathbb{C}_r^{m \times n}$ , i ako je  $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$  regularna matrica i  $F$  permuta-

cionalna matrica, tako da važi transformacija  $A = R^{-1} \begin{bmatrix} I_r & K \\ \mathbb{O} & \mathbb{O} \end{bmatrix} F^*$  tada je:

$$\begin{aligned} A^{(1)} &= F \begin{bmatrix} I_r - KY & X - KZ \\ Y & Z \end{bmatrix} R, \\ A^{(1,2)} &= F \begin{bmatrix} I_r - KY & X - KYX \\ Y & YX \end{bmatrix} R = F \begin{bmatrix} I_r - KY \\ Y \end{bmatrix} \begin{bmatrix} I_r & X \end{bmatrix} R, \\ A^{(1,3)} &= F \begin{bmatrix} I_r - KY & -R_1 R_2^\dagger - KZ \\ Y & Z \end{bmatrix} R, \\ A^{(1,4)} &= F \begin{bmatrix} (I_r + KK^*)^{-1} & X - KZ \\ K^*(I_r + KK^*)^{-1} & Z \end{bmatrix} R, \\ A^{(1,2,3)} &= F \begin{bmatrix} I_r - KY & -(I_r - KY)B^{-1}R_1R_2^\dagger \\ Y & -YR_1R_2^\dagger \end{bmatrix} R, \\ &= F \begin{bmatrix} I_r - KY \\ Y \end{bmatrix} \begin{bmatrix} I_r, & -R_1R_2^\dagger \end{bmatrix} R, \\ A^{(1,2,4)} &= F \begin{bmatrix} (I_r + KK^*)^{-1} & (I_r + KK^*)^{-1}X \\ K^*(I_r + KK^*)^{-1} & K^*(I_r + KK^*)^{-1}X \end{bmatrix} R \\ &= F \begin{bmatrix} I_r \\ K^* \end{bmatrix} (I_r + KK^*)^{-1} \begin{bmatrix} I_r & X \end{bmatrix} R, \\ A^{(1,3,4)} &= F \begin{bmatrix} (I_r + KK^*)^{-1} & -R_1R_2^\dagger - KZ \\ K^*(I_r + KK^*)^{-1} & Z \end{bmatrix} R, \\ A^\dagger &= F \begin{bmatrix} (I_r + KK^*)^{-1} & -(I_r + KK^*)^{-1}R_1R_2^\dagger \\ K^*(I_r + KK^*)^{-1} & -K^*(I_r + KK^*)^{-1}R_1R_2^\dagger \end{bmatrix} R \\ &= F \begin{bmatrix} I_r \\ K^* \end{bmatrix} (I_r + KK^*)^{-1} \begin{bmatrix} I_r & -R_1R_2^\dagger \end{bmatrix} R. \end{aligned}$$

**Teorema 2.9.5.** Ako je  $A \in \mathbb{C}_r^{m \times n}$ , i ako je  $G = [ G_1 \ G_2 ]$  regularna matrica i  $E$  permutaciona matrica, tako da važi  $A = E^* \begin{bmatrix} I_r & \mathbb{O} \\ K & \mathbb{O} \end{bmatrix} G^{-1}$  tada je:

$$\begin{aligned} A^{(1)} &= G \begin{bmatrix} I_r - XK & X \\ Y - ZK & Z \end{bmatrix} E, \\ A^{(1,2)} &= G \begin{bmatrix} I_r - XK & X \\ Y - YXK & YX \end{bmatrix} E = G \begin{bmatrix} I_r \\ Y \end{bmatrix} \begin{bmatrix} I_r - XK, & X \end{bmatrix} E, \\ A^{(1,3)} &= G \begin{bmatrix} (I_r + K^*K)^{-1} & (I_r + K^*K)^{-1}K^* \\ Y - ZK & Z \end{bmatrix} E, \\ A^{(1,4)} &= G \begin{bmatrix} I_r - XK & X \\ -G_2^\dagger G_1 - ZK & Z \end{bmatrix} E, \\ A^{(1,2,3)} &= G \begin{bmatrix} (I_r + K^*K)^{-1} & (I_r + K^*K)^{-1}K^* \\ Y(I_r + K^*K)^{-1} & -Y(I_r + K^*K)^{-1}K^* \end{bmatrix} E \\ &= G \begin{bmatrix} I_r \\ Y \end{bmatrix} (I_r + K^*K)^{-1} \begin{bmatrix} I_r, & K^* \end{bmatrix} E \end{aligned}$$

$$\begin{aligned}
A^{(1,2,4)} &= G \begin{bmatrix} I_r - XK & X \\ -G_2^\dagger G_1(I_r - XK) & -G_2^\dagger G_1 X \end{bmatrix} E \\
&= G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} \begin{bmatrix} I_r - XK & X \end{bmatrix} E, \\
A^{(1,3,4)} &= G \begin{bmatrix} (I_r + K^*K)^{-1} & (I_r + K^*K)^{-1}K^* \\ -G_2^\dagger G_1 - ZK & Z \end{bmatrix} E, \\
A^\dagger &= G \begin{bmatrix} (I_r + K^*K)^{-1} & (I_r + K^*K)^{-1}K^* \\ -G_2^\dagger G_1(I_r + K^*K)^{-1} & G_2^\dagger G_1(I_r + K^*K)^{-1}K^* \end{bmatrix} R \\
&= G \begin{bmatrix} I_r \\ -G_2^\dagger G_1 \end{bmatrix} (I_r + K^*K)^{-1} \begin{bmatrix} I_r & K^* \end{bmatrix} E.
\end{aligned}$$

**Teorema 2.9.6.** Neka je  $A \in \mathbb{C}_r^{m \times n}$ , i neka su  $E$  i  $F$  permutaciona matrica, tako da važi  
 $A = E^* \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} F^*$ ,  $A_{11} \in \mathbb{C}_r^{r \times r}$ . Koristeći oznake

$$L = A_{21}A_{11}^{-1}, \quad R = A_{11}^{-1}A_{12}, \quad M = I_r + RR^*, \quad N = I_r + LL^*$$

dobija se:

$$\begin{aligned}
A^{(1)} &= F \begin{bmatrix} A_{11}^{-1} - XL - DY - DZL & X \\ Y & Z \end{bmatrix} E, \\
A^{(1,2)} &= F \begin{bmatrix} A_{11}^{-1} - XL - DY + DY A_{11}XL & X - DY A_{11}X \\ Y - YA_{11}XL & YA_{11}X \end{bmatrix} E \\
&= F \begin{bmatrix} A_{11}^{-1} - DY \\ Y \end{bmatrix} \begin{bmatrix} I_r - A_{11}XL, & A_{11}X \end{bmatrix} E, \\
A^{(1,3)} &= F \begin{bmatrix} (NA_{11})^{-1} - DY & (NA_{11})^{-1}L^* - DZ \\ Y & Z \end{bmatrix} E, \\
A^{(1,4)} &= F \begin{bmatrix} (A_{11}M)^{-1} - XL & X \\ R^*(A_{11}M)^{-1} - ZL & Z \end{bmatrix} E, \\
A^{(1,2,3)} &= F \begin{bmatrix} (NA_{11})^{-1} - DY & (NA_{11})^{-1}L^* - DY L^* \\ Y & YL^* \end{bmatrix} E \\
&= F \begin{bmatrix} (NA_{11})^{-1} - DY \\ Y \end{bmatrix} \begin{bmatrix} I_r, & L^* \end{bmatrix} E, \\
A^{(1,2,4)} &= F \begin{bmatrix} (A_{11}M)^{-1} - XL & X \\ D^*(A_{11}M)^{-1} - D^*XL & R^*X \end{bmatrix} E \\
&= F \begin{bmatrix} I_r \\ D^* \end{bmatrix} \begin{bmatrix} (A_{11}M)^{-1} - XL & X \end{bmatrix} E, \\
A^{(1,3,4)} &= F \begin{bmatrix} (NA_{11})^{-1} + (A_{11}M)^{-1} - A_{11}^{-1} + DZL & (NA_{11})^{-1}L^* - DZ \\ D^*(A_{11}M)^{-1} - ZL & Z \end{bmatrix} E, \\
A^\dagger &= F \begin{bmatrix} (NA_{11}M)^{-1} & (NA_{11}M)^{-1}L^* \\ D^*(NA_{11}M)^{-1} & D^*(NA_{11}M)^{-1}L^* \end{bmatrix} E \\
&= F \begin{bmatrix} I_r \\ D^* \end{bmatrix} (NA_{11}M)^{-1} \begin{bmatrix} I_r & L^* \end{bmatrix} E.
\end{aligned}$$

Sledeće dve teoreme za blokovske reprezentacije Moore-Penrose-ovog inverza je uveo S. Zlobec u [120].

**Teorema 2.9.7.** *Moore-Penrose-ov inverz matrice  $A \in \mathbb{C}^{m \times n}$  jednak je*

$$A^\dagger = A^* (A^* A A^*)^{(1)} A^*.$$

**Teorema 2.9.8.** *Neka je matrica  $A \in \mathbb{C}_r^{m \times n}$  predstavljena u obliku  $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ , gde je  $A_{11} \in \mathbb{C}_r^{r \times r}$ . Tada je*

$$A^\dagger = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix}^* T_{11}^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}^*,$$

gde je

$$T_{11} = \left( \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} A^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \right)^{-1}.$$

Može se koristiti i Hermiteov algoritam, koji se zasniva na sledećoj formuli

$$A^\dagger = A^* (A A^* A A^*)^{(1,2)} A^* = ((A A^*)^2)^{(1,2)}.$$

Pri tome, za izračunavanje inverza  $((A A^*)^2)^{(1,2)}$  može se iskoristiti, na primer njegova blokovska reprezentacija iz Teoreme 2.10.2, za  $X = Y = \mathbb{O}$ . Ako je  $(A A^*)^2 = R^T I_r G$ , tada je  $((A A^*)^2)^{(1,2)} = G^T I_r R = (G^T)_{|r}^r R_{|r}$ , gde  $(G^T)_{|r}^r$  i  $R_{|r}$  predstavljaju prvih  $r$  kolona matrice  $G^T$  i prvih  $r$  vrsta matrice  $R$ , respektivno.

Takođe, poznate su i blokovske reprezentacije grupnog i Drazinovog inverza. Sledeća teorema data je u radu [66].

**Teorema 2.9.9.** *Neka je  $A \in \mathbb{C}_r^{n \times n}$ , i  $R$  regularna matrica, tako da važi  $R A R^{-1} = \begin{bmatrix} T_1 & T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix}$ . Tada je*

$$A^\# = R^{-1} \begin{bmatrix} T_1^{-1} & T_1^{-2} T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R.$$

Ovaj rezultat je uopštio Hartwig, čime je razvio metod za izračunavanje Drazinovog inverza [30], opisan sledećom teoremom.

**Teorema 2.9.10.** *Ako je  $A \in \mathbb{C}_r^{n \times n}$ , i  $R$  nesingularna matrica, tako da je  $R A R^{-1} = \begin{bmatrix} U & V \\ \mathbb{O} & \mathbb{O} \end{bmatrix}$ . Tada je*

$$A^D = R^{-1} \begin{bmatrix} U_d & U_d^2 V_d \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R.$$

Ako  $X$  zadovoljava jednakost  $X A X = X$ , sledi da postoji celobrojno  $t \in \{1, \dots, r\}$ , gde je  $r = \text{rang}(A)$ , takvo da se  $X$  može pretstaviti u obliku

$$X = W_1 (W_2 A W_1)^{-1} W_2, \quad W_1 \in \mathbb{C}^{n \times t}, \quad W_2 \in \mathbb{C}^{t \times m}, \quad \text{rang}(W_2 A W_1) = t. \quad (2.9.1)$$

Na osnovu ovoga, imamo sledeće tvrdjenje za blokovsku reprezentaciju  $\{1, 2, 3\}$  i  $\{1, 2, 4\}$  inverza iz [82].

**Teorema 2.9.11.** Neka je  $A \in \mathbb{C}_r^{m \times n}$  i neka matrice  $W_1$  i  $W_2$  ispunjavaju uslove iz (2.9.1). Tada se može dobiti sledeće blokovske reprezentacije  $\{1, 2, 3\}$  i  $\{1, 2, 4\}$  generalisanih inverza za  $A$ , pod uslovom da  $(G_{1,2,3}^i)$  i  $(G_{1,2,4}^i)$  odgovaraju  $(T_i)$ ,  $i \in \{1, \dots, 11\}$ :

$$(G_{1,2,3}^1) \quad A\{1, 2, 3\} = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1} \left( (RR^*)^{-1}|_r \right)^{-1} \begin{bmatrix} I_r & \mathbb{O} \end{bmatrix} (R^*)^{-1} \\ = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \left( (RR^*)^{-1}|_r U_1 \right)^{-1} (R^*)^{-1}|_r,$$

$$(G_{1,2,4}^1) \quad A\{1, 2, 4\} = (G^*)^{-1} \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} \left( (G^*G)^{-1}|_r \right)^{-1} V_1^{-1} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} R \\ = (G^*)^{-1}|_r \left( V_1 (G^*G)^{-1}|_r \right)^{-1} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} R,$$

$$(G_{1,2,3}^2) \quad A\{1, 2, 3\} = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1} \left( B^* (RR^*)^{-1}|_r B \right)^{-1} \begin{bmatrix} B^* & \mathbb{O} \end{bmatrix} (R^*)^{-1} \\ = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \left( (RR^*)^{-1}|_r B U_1 \right)^{-1} (R^*)^{-1}|_r,$$

$$(G_{1,2,4}^2) \quad A\{1, 2, 4\} = (G^*)^{-1} \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} \left( (G^*G)^{-1}|_r \right)^{-1} (V_1 B)^{-1} \begin{bmatrix} V_1 & B \end{bmatrix} R \\ = (G^*)^{-1}|_r \left( V_1 B (G^*G)^{-1}|_r \right)^{-1} \begin{bmatrix} V_1 & B \end{bmatrix} R,$$

$$(G_{1,2,3}^3) \quad A\{1, 2, 3\} = F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} (S_1 + KS_2)^{-1} \left( (RR^*)^{-1}|_r \right)^{-1} \begin{bmatrix} I_r & \mathbb{O} \end{bmatrix} (R^*)^{-1} \\ = F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \left( (RR^*)^{-1}|_r (S_1 + KS_2) \right)^{-1} (R^*)^{-1}|_r,$$

$$(G_{1,2,4}^3) \quad A\{1, 2, 4\} = F \begin{bmatrix} I_r \\ K^* \end{bmatrix} (I_r + KK^*)^{-1} V_1^{-1} \begin{bmatrix} V_1 & V_2 \end{bmatrix} R \\ = (F^{|r|} + F^{n-r}|K^*|) (V_1 (I_r + KK^*))^{-1} \begin{bmatrix} V_1 & V_2 \end{bmatrix} R,$$

$$(G_{1,2,3}^4) \quad A\{1, 2, 3\} = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1} (I_r + KK^*)^{-1} \begin{bmatrix} I_r & K^* \end{bmatrix} E \\ = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} ((I_r + KK^*)U_1)^{-1} (E|_r + K^*E_{n-r|}),$$

$$(G_{1,2,4}^4) \quad A\{1, 2, 4\} = (G^*)^{-1} \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} \left( (G^*G)^{-1}|_r \right)^{-1} (T_1 + T_2 K)^{-1} \begin{bmatrix} T_1 & T_2 \end{bmatrix} E \\ = (G^*)^{-1}|_r \left( (T_1 + T_2 K) (G^*G)^{-1}|_r \right)^{-1} \begin{bmatrix} T_1 & T_2 \end{bmatrix} E,$$

$$(G_{1,2,3}^5) \quad A\{1, 2, 3\} = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1} \begin{bmatrix} I_r & \mathbb{O} \end{bmatrix} U = G \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} U_1^{-1} U|_r,$$

$$(G_{1,2,4}^5) \quad A\{1, 2, 4\} = (G^*)^{-1} \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} \left( (G^*G)^{-1}|_r \right)^{-1} D_1^{-1} \begin{bmatrix} D_1 & D_2 \end{bmatrix} U \\ = (G^*)^{-1}|_r \left( D_1 (G^*G)^{-1}|_r \right)^{-1} \begin{bmatrix} D_1 & D_2 \end{bmatrix} U,$$

$$\begin{aligned}
(G_{1,2,3}^6) \quad A\{1, 2, 3\} &= V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} C_1^{-1} \left( (RR^*)^{-1} \Big|_r^r \right)^{-1} \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} (R^*)^{-1} \\
&= V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \left( (RR^*)^{-1} \Big|_r^r C_1 \right)^{-1} (R^*)^{-1} \Big|_r, \\
(G_{1,2,4}^6) \quad A\{1, 2, 4\} &= V \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} V_1^{-1} \begin{bmatrix} V_1, & V_2 \end{bmatrix} R = V^{|r} V_1^{-1} \begin{bmatrix} V_1, & V_2 \end{bmatrix} R, \\
(G_{1,2,3}^7) \quad A\{1, 2, 3\} &= V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} C_1^{-1} (B^* B)^{-1} \begin{bmatrix} B^*, & \mathbb{O} \end{bmatrix} U \\
&= V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (BC_1)^{-1} U \Big|_r, \\
(G_{1,2,4}^7) \quad A\{1, 2, 4\} &= V \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} (D_1 B)^{-1} \begin{bmatrix} D_1 & D_2 \end{bmatrix} U = V^{|r} (D_1 B)^{-1} \begin{bmatrix} D_1 & D_2 \end{bmatrix} U, \\
(G_{1,2,3}^8) \quad A\{1, 2, 3\} &= F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} (BS_1 + KS_2)^{-1} \begin{bmatrix} I_r, & \mathbb{O} \end{bmatrix} U \\
&= F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} (BS_1 + KS_2)^{-1} U \Big|_r, \\
(G_{1,2,4}^8) \quad A\{1, 2, 4\} &= F \begin{bmatrix} B^* \\ K^* \end{bmatrix} (D_1(BB^* + KK^*))^{-1} \begin{bmatrix} D_1, & D_2 \end{bmatrix} U, \\
(G_{1,2,3}^9) \quad A\{1, 2, 3\} &= V \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} ((B^* B + K^* K)C_1)^{-1} \begin{bmatrix} B^*, & K^* \end{bmatrix} E, \\
(G_{1,2,4}^9) \quad A\{1, 2, 4\} &= V \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} (T_1 B + T_2 K)^{-1} \begin{bmatrix} T_1, & T_2 \end{bmatrix} E \\
&= V^{|r} (T_1 B + T_2 K)^{-1} \begin{bmatrix} T_1, & T_2 \end{bmatrix} E, \\
(G_{1,2,3}^{10a}) \quad A\{1, 2, 3\} &= F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} (S_1 + TS_2)^{-1} A_{11}^{-1} (I_r + S^* S)^{-1} \begin{bmatrix} I_r, & S^* \end{bmatrix} E \\
&= F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} ((I_r + S^* S)A_{11}(S_1 + TS_2))^{-1} (E \Big|_r + S^* E_{n-r}), \\
(G_{1,2,4}^{10a}) \quad A\{1, 2, 4\} &= F \begin{bmatrix} I_r \\ T^* \end{bmatrix} (I_r + TT^*)^{-1} A_{11}^{-1} (T_1 + T_2 S)^{-1} \begin{bmatrix} T_1, & T_2 \end{bmatrix} E \\
&= (F^{|r} + F^{n-r} | T^*) ((T_1 + T_2 S)A_{11}(I_r + TT^*))^{-1} \begin{bmatrix} T_1, & T_2 \end{bmatrix} E, \\
(G_{1,2,3}^{10b}) \quad A\{1, 2, 3\} &= F \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} (A_{11}S_1 + A_{12}S_2)^{-1} A_{11}(A_{11}^* A_{11} + A_{21}^* A_{21})^{-1} \begin{bmatrix} A_{11}^*, & A_{21}^* \end{bmatrix} E, \\
(G_{1,2,4}^{10b}) \quad A\{1, 2, 4\} &= F \begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix} (A_{11}A_{11}^* + A_{12}A_{12}^*)^{-1} A_{11}(T_1 A_{11} + T_2 A_{21})^{-1} \begin{bmatrix} T_1, & T_2 \end{bmatrix} E, \\
(G_{1,2,3}^{11}) \quad A\{1, 2, 3\} &= R^{-1} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} (Q_1 + T_1^{-1} T_2 Q_2)^{-1} \left( T_1^*(RR^*)^{-1} \Big|_r^r T_1 \right)^{-1} \begin{bmatrix} T_1^* \\ \mathbb{O} \end{bmatrix} (R^{-1})^* \\
&= R^{-1} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \left( T_1^*(RR^*)^{-1} \Big|_r^r (T_1 Q_1 + T_2 Q_2) \right)^{-1} (R^{-1} \Big|_r^r T_1)^*, \\
(G_{1,2,4}^{11}) \quad A\{1, 2, 4\} &= R^* \left[ \begin{array}{c} I_r \\ (T_1^{-1} T_2)^* \end{array} \right] \left( V_1 \begin{bmatrix} T_1, & T_2 \end{bmatrix} RR^* \left[ \begin{array}{c} I_r \\ (T_1^{-1} T_2)^* \end{array} \right] \right)^{-1} \begin{bmatrix} V_1, & V_2 \end{bmatrix} R.
\end{aligned}$$

Sledi implementacija navedenih blokovskih reprezentacija uopštenih inverza. Ovde navodimo samo funkcije za određivanje reprezentacija  $(G_{1,2,3}^i)$  i  $(G_{1,2,4}^i)$ ,  $i \in \{1, 3, 4\}$ .

```

G1231[A_List,R1_,R2_]:= 
Block[A=b,W1=R1,W2=R2, ran,r,g,u1,u2,rz,pom1,pom2,
{r,g}=t1[A]; ran=rang[A];
pom1=Inverse[g].W1; pom2=W2.Inverse[r];
u1=Take[pom1,ran];
u2=Drop[pom1,ran];
rz=Inverse[r.Hermit[r]];
rz=Take[rz,ran];
rz=Transpose[Take[Transpose[rz],ran]];
Return[g.pom1.Inverse[rz.u1].Take[Inverse[Hermit[r]],ran]];
]

G1241[A_List,R1_,R2_]:= 
Block[W1=R1,W2=R2, ran,r,g,v1,v2,gz,gz1,pom2,
{r,g}=t1[A]; ran=rang[A];
pom2=W2.Inverse[r];
v1=Transpose[Take[Transpose[pom2],ran]];
v2=Transpose[Drop[Transpose[pom2],ran]];
gz=Inverse[Hermit[g].g];
gz=Take[gz,ran];
gz=Transpose[Take[Transpose[gz],ran]];
gz1=Transpose[Take[Transpose[Inverse[Hermit[g]]],ran]];
Return[gz1.Inverse[v1.gz].pom2.r];
]

G1233[A_List,R1_,R2_]:= 
Block[{W1=R1,W2=R2, ran,r,f,s1,s2,k,v1,v2,pom1,pom3},
{r,f}=t3[A]; ran=rang[A];
pom1=Hermit[f].W1;
s1=Take[pom1,ran];
s2=Drop[pom1,ran];
pom3=Take[r.A.f,ran];
k=Transpose[Drop[Transpose[pom3],ran]];
rz=Inverse[r.Hermit[r]];
rz=Take[rz,ran];
rz=Transpose[Take[Transpose[rz],ran]];
Return[f.pom1.Inverse[rz.(s1+k.s2)].Take[Inverse[Hermit[r]],ran]];
]

G1243[A_List,R1_,R2_]:= 
Block[{W1=R1,W2=R2, ran,r,f,v1,v2,k,pom2,pom3},
{r,f}=t3[A]; ran=rang[A];
pom2=W2.Inverse[r];
v1=Transpose[Take[Transpose[pom2],ran]];
v2=Transpose[Drop[Transpose[pom2],ran]];
pom3=Take[r.A.f,ran];
k=Transpose[Drop[Transpose[pom3],ran]];
Return[f.Hermit[pom3].Inverse[IdentityMatrix[ran]+k.Hermit[k]].
Inverse[v1].pom2.r];
]

```

```

G1234[A>List,R1_,R2_]:= 
Block[{W1=R1,W2=R2, ran,e,g,u1,u2,k,pom1,pom3},
{e,g}=t4[A]; ran=rang[A];
pom1=Inverse[g].W1;
u1=Take[pom1,ran];
u2=Drop[pom1,ran];
pom3=Drop[e.A.g,ran];
k=Transpose[Take[Transpose[pom3],ran]];
pom3=Conjugate[Take[Transpose[e.A.g],ran]];
Return[g.pom1.Inverse[(IdentityMatrix[ran]+k.Hermit[k]).u1].pom3.e];
]

G1244[A>List,R1_,R2_]:= 
Block[{W1=R1,W2=R2, ran,e,g,u1,u2,k,gz,t1,t2,pom2,pom3},
{e,g}=t4[A]; ran=rang[A];
pom2=W2.Hermit[e];
t1=Transpose[Take[Transpose[pom2],ran]];
t2=Transpose[Drop[Transpose[pom2],ran]];
pom3=Drop[e.A.g,ran];
k=Transpose[Take[Transpose[pom3],ran]];
gz=Inverse[Hermit[g].g];
gz=Take[gz,ran];
gz=Transpose[Take[Transpose[gz],ran]];
gz1=Transpose[Take[Transpose[Inverse[Hermit[g]]],ran]];
Return[gz1.Inverse[(t1+t2.k).gz].pom2.e];
]

```

**Primer 2.9.3.** Koristeći

$$A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix} \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & -1 \end{bmatrix}$$

izraz  $x=G1231[a,w1,w2]$  dobija vrednost

$$X = \begin{bmatrix} -\frac{2}{17} & -\frac{3}{17} & \frac{1}{17} & -\frac{1}{17} & \frac{3}{17} & \frac{2}{17} \\ \frac{19}{102} & \frac{10}{51} & -\frac{1}{102} & \frac{1}{102} & -\frac{10}{51} & -\frac{19}{102} \\ -\frac{1}{17} & -\frac{3}{34} & \frac{34}{51} & -\frac{34}{51} & \frac{3}{34} & -\frac{1}{17} \\ \frac{7}{51} & \frac{2}{51} & \frac{5}{51} & -\frac{5}{51} & -\frac{2}{51} & -\frac{7}{51} \end{bmatrix}$$

U slučaju  $W_1 = Q^*$ ,  $W_2 = P^*$  dobija se ovakva reprezentacija Moore-Penroseovog inverza.

**Posledica 2.9.12.** *Moore-Penroseov inverza date matrice  $A \in \mathbb{C}_r^{m \times n}$  se može predstaviti kako sledi, pri čemu reprezentacije  $(A_{\dagger}^i)$  odgovaraju blokovskim dekompozicijama  $(T_i)$ ,  $i \in \{1, \dots, 9, 10a, 10b, 11\}$*

- $$(A_{\dagger}^1) \quad A^{\dagger} = (G^{-1}|_r)^* \left( \left( R^{-1|_r} \right)^* A (G^{-1}|_r)^* \right)^{-1} \left( R^{-1|_r} \right)^*$$
- $$= (G^{-1}|_r)^* \left( (RR^*)^{-1}|_r (G^*G)^{-1}|_r \right)^{-1} \left( R^{-1|_r} \right)^*,$$
- $$(A_{\dagger}^2) \quad A^{\dagger} = (G^{-1}|_r)^* \left( \left( R^{-1|_r} B \right)^* A (G^{-1}|_r)^* \right)^{-1} \left( R^{-1|_r} B \right)^*$$
- $$= (G^{-1}|_r)^* \left( B^* (RR^*)^{-1}|_r B (G^*G)^{-1}|_r \right)^{-1} B^* \left( R^{-1|_r} \right)^*,$$
- $$(A_{\dagger}^3) \quad A^{\dagger} = F \begin{bmatrix} I_r \\ K^* \end{bmatrix} \left( \left( R^{-1|_r} \right)^* A F \begin{bmatrix} I_r \\ K^* \end{bmatrix} \right)^{-1} \left( R^{-1|_r} \right)^*$$
- $$= (F^{|r} + F^{n-r}|K) \left( (RR^*)^{-1}|_r (I_r + KK^*) \right)^{-1} \left( R^{-1|_r} \right)^*,$$
- $$(A_{\dagger}^4) \quad A^{\dagger} = (G^{-1}|_r)^* \left( [ I_r, \ K^* ] EA (G^{-1}|_r)^* \right)^{-1} [ I_r, \ K^* ] E$$
- $$= (G^{-1}|_r)^* \left( (I_r + K^*K)(G^*G)^{-1}|_r \right)^{-1} (E|_r + K^*E_{n-r|}),$$
- $$(A_{\dagger}^5) \quad A^{\dagger} = (G^{-1}|_r)^* (U|_r A (G^{-1}|_r)^*)^{-1} U|_r = (G^{-1}|_r)^* \left( (G^*G)^{-1}|_r \right)^{-1} U|_r,$$
- $$(A_{\dagger}^6) \quad A^{\dagger} = V^{|r} \left( \left( R^{-1|_r} \right)^* AV^{|r} \right)^{-1} \left( R^{-1|_r} \right)^* = V^{|r} \left( (RR^*)^{-1}|_r \right)^{-1} \left( R^{-1|_r} \right)^*,$$
- $$(A_{\dagger}^7) \quad A^{\dagger} = V^{|r} (B^*U|_r AV^{|r})^{-1} B^*U|_r = V^{|r} B^{-1}U|_r,$$
- $$(A_{\dagger}^8) \quad A^{\dagger} = F \begin{bmatrix} B^* \\ K^* \end{bmatrix} \left( U|_r A F \begin{bmatrix} B^* \\ K^* \end{bmatrix} \right)^{-1} U|_r = F \begin{bmatrix} B^* \\ K^* \end{bmatrix} (BB^* + KK^*)^{-1} U|_r,$$
- $$(A_{\dagger}^9) \quad A^{\dagger} = V^{|r} \left( [ B^*, \ K^* ] EA V^{|r} \right)^{-1} [ B^*, \ K^* ] E$$
- $$= V^{|r} (B^*B + K^*K)^{-1} [ B^*, \ K^* ] E,$$
- $$(A_{\dagger}^{10a}) \quad A^{\dagger} = F \begin{bmatrix} I_r \\ T^* \end{bmatrix} \left( A_{11}^* [ I_r, \ S^* ] EA F \begin{bmatrix} I_r \\ T^* \end{bmatrix} \right)^{-1} A_{11}^* [ I_r, \ S^* ] E$$
- $$= (F^{|r} + F^{n-r}|T^*) ((I_r + S^*S) A_{11} (I_r + TT^*))^{-1} (E|_r + S^*E_{n-r|}),$$
- $$(A_{\dagger}^{10b}) \quad A^{\dagger} = F \begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix} \left( (A_{11}^*)^{-1} [ A_{11}^*, \ A_{21}^* ] EA F \begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix} \right)^{-1} (A_{11}^*)^{-1} [ A_{11}^*, \ A_{21}^* ] E$$
- $$= F \begin{bmatrix} A_{11}^* \\ A_{12}^* \end{bmatrix} (A_{11} A_{11}^* + A_{12} A_{12}^*)^{-1} A_{11} (A_{11}^* A_{11} + A_{21}^* A_{21})^{-1} [ A_{11}^*, \ A_{21}^* ] E,$$
- $$(A_{\dagger}^{11}) \quad A^{\dagger} = R^* \begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix} \left( \left( R^{-1|_r} T_1 \right)^* A R^* \begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix} \right)^{-1} \left( R^{-1|_r} T_1 \right)^*$$
- $$= R^* \begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix} \left( T_1^* (RR^*)^{-1}|_r [ T_1, \ T_2 ] RR^* \begin{bmatrix} I_r \\ (T_1^{-1}T_2)^* \end{bmatrix} \right)^{-1} \left( R^{-1|_r} T_1 \right)^*.$$

Implementacione procedure za izračunavanje Moore-Penroseovog inverza, bazirane na reprezentacijama  $(A_{\dagger}^1)$ ,  $(A_{\dagger}^3)$  i  $(A_{\dagger}^4)$  date su sledećim kodovima:

```
MPt1[A_List]:= 
  Block[{r,g,ran,pt,qt},
    {r,g}=t1[A]; ran=rang[A];
    qt=Hermit[Take[Inverse[g],ran]];
    pt=Conjugate[Take[Transpose[Inverse[r]],ran]];
    qt.Inverse[pt.A.qt].pt
  ];MatrixQ[A]
```

```

MPt3[A_List]:= 
  Block[{r,f,ran,pt,qt},
    {r,f}=t3[A]; ran=rang[A];
    qt=Hermit[Take[r.A.f,ran]];
    pt=Conjugate[Take[Transpose[Inverse[r]],ran]];
    qt.Inverse[pt.A.qt].pt
  ];MatrixQ[A]

MPt4[A_List]:= 
  Block[{e,g,ran,pt,qt},
    {e,g}=t4[A]; ran=rang[A];
    qt=Hermit[Take[Inverse[g],ran]];
    pt=Conjugate[Take[Transpose[e.A.g],ran]].e;
    qt.Inverse[pt.A.qt].pt
  ];MatrixQ[A]

```

**Teorema 2.9.13.** Neka je  $A$  kvadratna matrica reda  $n$  i ranga  $r$ . Tada grupni inverz postoji ako i samo ako su sledeći uslovi ( $E_i$ ), zavisni od odgovarajućih blokovskih dekompozicija ( $T_i$ ), ispunjeni:

$$(E_1) \quad G^{-1}|_r R^{-1}|^r = (RG)^{-1}|_r^r \text{ je invertibilno,}$$

$$(E_2) \quad G^{-1}|_r R^{-1}|^r B = (RG)^{-1}|_r^r B \text{ je invertibilno,}$$

$$(E_3) \quad [ \begin{matrix} I_r & K \end{matrix} ] (RF)^{-1}|^r = (RF)^{-1}|_r^r + K(RF)^{-1}|_{n-r}^r \quad \text{je invertibilno,}$$

$$(E_4) \quad (EG)^{-1}|_r \begin{bmatrix} I_r \\ K \end{bmatrix} = (EG)^{-1}|_r^r + (EG)^{-1}|_{n-r}^r K \quad \text{je invertibilno,}$$

$$(E_5) \quad U^*|_r G^{-1}|_r = (UG)^{-1}|_r^r \quad \text{je invertibilno,}$$

$$(E_6) \quad V^*|_r R^{-1}|^r = (RV)^{-1}|_r^r \quad \text{je invertibilno,}$$

$$(E_7) \quad V^*|_r U^*|_r B = (UV)^*|_r^r B \quad \text{je invertibilno,}$$

$$(E_8) \quad [ \begin{matrix} B, & K \end{matrix} ] (UF)^* \begin{bmatrix} I_r \\ \mathbb{O} \end{bmatrix} = [ \begin{matrix} B, & K \end{matrix} ] (UF)^*|_r^r \quad \text{je invertibilno,}$$

$$(E_9) \quad [ \begin{matrix} I_r, & \mathbb{O} \end{matrix} ] (EV)^* \begin{bmatrix} B \\ K \end{bmatrix} = (EV)^*|_r \begin{bmatrix} B \\ K \end{bmatrix} \quad \text{je invertibilno,}$$

$$(E_{10a}) \quad [ \begin{matrix} I_r, & T \end{matrix} ] (EF)^* \begin{bmatrix} I_r \\ S \end{bmatrix} \quad \text{je invertibilno,}$$

$$(E_{10b}) \quad [ \begin{matrix} A_{11}, & A_{12} \end{matrix} ] (EF)^* \begin{bmatrix} A_{11} \\ A_{12} \end{bmatrix} \quad \text{je invertibilno,}$$

$$(E_{11}) \quad [ \begin{matrix} I_r, & T_1^{-1}T_2 \end{matrix} ] \begin{bmatrix} T_1 \\ \mathbb{O} \end{bmatrix} = T_1 \quad \text{je invertibilno.}$$

Takođe, grupni inverz, ako postoji, poseduje sledeće blokovske reprezentacije:

- $$(A_{\#}^1) \quad A^{\#} = R^{-1|r} \left( (RG)^{-1|_r} \right)^{-2} G^{-1|_r},$$
- $$(A_{\#}^2) \quad A^{\#} = R^{-1|r} B^{-1} \left( (RG)^{-1|_r} \right)^{-2} G^{-1|_r},$$
- $$(A_{\#}^3) \quad A^{\#} = R^{-1|r} \left( (RF)^{-1|_r} + K(RF)^{-1|_{n-r}} \right)^{-2} \left( F^*|_r + KF^*_{n-r} \right),$$
- $$(A_{\#}^4) \quad A^{\#} = (E^*|_r + E^{*n-r}|_r K) \left( (EG)^{-1|_r} + (EG)^{-1|_{n-r}} K \right)^{-2} G^{-1|_r},$$
- $$(A_{\#}^5) \quad A^{\#} = U^*|_r \left( (UG)^{-1|_r} \right)^{-2} G^{-1|_r},$$
- $$(A_{\#}^6) \quad A^{\#} = R^{-1|r} \left( (RV)^{-1|_r} \right)^{-2} V^*|_r,$$
- $$(A_{\#}^7) \quad A^{\#} = U^*|_r B^{-1} \left( (UV)^{*|_r} \right)^{-2} V^*|_r,$$
- $$(A_{\#}^8) \quad A^{\#} = U^*|_r \left( [B, K] (UF)^* \right)^{-2} [B, K] F^*,$$
- $$(A_{\#}^9) \quad A^{\#} = E^* \begin{bmatrix} B \\ K \end{bmatrix} \left( (EV)^{*|_r} \begin{bmatrix} B \\ K \end{bmatrix} \right)^{-2} V^*|_r,$$
- $$(A_{\#}^{10a}) \quad A^{\#} = E^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11} \left( [I_r, T] (EF)^* \begin{bmatrix} I_r \\ S \end{bmatrix} A_{11} \right)^{-2} [I_r, T] F^*,$$
- $$(A_{\#}^{10b}) \quad A^{\#} = E^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} A_{11} \left( [A_{11}, A_{12}] (EF)^* \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} \right)^{-2} [A_{11}, A_{12}] F^*,$$
- $$(A_{\#}^{11}) \quad A^{\#} = R^{-1} \begin{bmatrix} T_1^{-1} & T_1^{-2}T_2 \\ \mathbb{O} & \mathbb{O} \end{bmatrix} R = R^{-1}|_r [T_1^{-1} \ T_1^{-2}T_2] R.$$

**Dokaz.** Sledi neposredno iz sledećeg: Ako je  $A = PQ$  potpunarang faktorizacija za  $A$ , tada  $A^{\#}$  postoji ako i samo ako je  $QP$  invertibilna matrica, i  $A^{\#} = P(QP)^{-2}Q$ .  $\square$

Metodi za izračunavanje grupnog inverza se mogu implementirati na sledeći način:

```

Group1[a_List]:=Block[{ran,r,g,rg,pom1,pom2},
{r,g}=t1[a]; ran=rang[a];
rg=r.g;
rg=Inverse[rg];
rg=Take[rg,ran];
rg=Transpose[Take[Transpose[rg],ran]];
If[rang[rg]==ran,
pom1=Transpose[Take[Transpose[Inverse[r]],ran]];
pom2=Take[Inverse[g],ran];
Return[pom1.MatrixPower[Inverse[rg],2].pom2],
Return["Grupni inverz ne postoji"];
];
];

Group3[a_List]:=Block[{ran,r,g,pom1,pom2},
{r,f}=t3[a]; ran=rang[a];
pom1=Transpose[Take[Transpose[Inverse[r]],ran]];
pom2=Take[r.a.f,ran].Hermit[f];
pom3=pom2.pom1;
If[rang[pom3]==ran,
Return[pom1.MatrixPower[Inverse[pom3],2].pom2],
Return["Grupni inverz ne postoji"];
];
];

```

```

Group4[a_List]:=Block[{ran,r,g,k,pom1,pom2},
{e,g}=t3[a]; ran=rang[a];
pom1=Take[Inverse[g],ran];
pom2=Hermit[e].Transpose[Take[Transpose[e.a.g],ran]];
pom3=pom2.pom1;
If[rang[pom3]==ran,
Return[pom2.MatrixPower[Inverse[pom3],2].pom1],
Return["Grupni inverz ne postoji"];
];
]

```

**Primer 2.9.4.** Za  $PQ$  faktorizaciju matrice  $A = \begin{bmatrix} 9 & 6 & 9 \\ 6 & 6 & 6 \\ 9 & 6 & 9 \end{bmatrix}$  imamo redom

$$R = \begin{bmatrix} \frac{1}{9} & 0 & 0 \\ -\frac{1}{3} & \frac{1}{2} & 0 \\ -1 & 0 & 1 \end{bmatrix}, G = \begin{bmatrix} 1 & -\frac{2}{3} & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^{-1} = \begin{bmatrix} 9 & 0 & 0 \\ 6 & 2 & 0 \\ 9 & 0 & 1 \end{bmatrix}, G^{-1} = \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P = R^{-1} \begin{bmatrix} I_2 \\ \mathbb{O} \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 \\ 6 & 2 & 0 \\ 9 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 6 & 2 \\ 9 & 0 \end{bmatrix}$$

$$Q = [ I_2 \ \mathbb{O} ] G^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{2}{3} & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

**Primer 2.9.5.** Data je matrica  $A = \begin{bmatrix} -1 & 1 & 3 & 5 & 7 \\ 1 & 0 & -2 & 0 & 4 \\ 1 & 1 & -1 & 5 & 15 \\ -1 & 2 & 4 & 10 & 18 \end{bmatrix}$ .

Iz  $\{r1, f1\} = t3[a]$  dobijamo

$$R_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & -2 & 1 & 0 \\ -2 & -1 & 0 & 1 \end{bmatrix}, \quad F_1 = I_5$$

Primenom  $\{r2, f2\} = t3[Transpose[r1, a, f1]]$  dobijamo

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & -1 & 1 & 0 & 0 \\ 0 & -5 & 0 & 1 & 0 \\ -4 & -11 & 0 & 0 & 1 \end{bmatrix}, \quad F_2 = I_4.$$

Sada je  $R = F_2^T R_1, G = F_1 R_2^T$ , i

$$R_{|2}^{-1} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 1 & 1 \\ -1 & 2 \end{bmatrix}, \quad G_{|2}^{-1} = \begin{bmatrix} 1 & 0 & -2 & 0 & 4 \\ 0 & 1 & 1 & 5 & 11 \end{bmatrix}.$$

Primenom formule  $\{A_1^\dagger\}$  dobijamo

$$A^\dagger = \begin{bmatrix} -\frac{169}{6720} & \frac{67}{2240} & \frac{233}{6720} & -\frac{137}{6720} \\ \frac{1}{128} & -\frac{1}{128} & -\frac{1}{128} & \frac{1}{128} \\ \frac{781}{13440} & -\frac{330}{4480} & -\frac{1037}{13440} & \frac{653}{13440} \\ \frac{5}{128} & -\frac{5}{128} & -\frac{5}{128} & \frac{5}{128} \\ \frac{128}{197} & -\frac{128}{151} & -\frac{128}{709} & \frac{128}{59} \\ -\frac{128}{13440} & \frac{128}{4480} & \frac{128}{13440} & \frac{128}{13440} \end{bmatrix}$$

**Primer 2.9.6.** Za  $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$  dobijamo

$$A_{11}^{-1} = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & -1 \\ -1 & 0 \end{bmatrix}, \quad T = \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix}.$$

Primenom tvrđenja Teoreme 2.9.11, Teoreme 2.9.12 za

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 & 2 & -1 \\ 0 & -1 & 0 & 0 & -2 & 1 \end{bmatrix}.$$

dobijamo:

$$A^{(1,2,3)} = \begin{bmatrix} -\frac{2}{19} & -\frac{3}{17} & \frac{1}{17} & -\frac{1}{17} & \frac{3}{17} & \frac{2}{17} \\ \frac{102}{19} & \frac{10}{51} & -\frac{1}{102} & \frac{1}{102} & -\frac{10}{51} & -\frac{19}{102} \\ -\frac{1}{17} & -\frac{3}{2} & \frac{34}{51} & -\frac{1}{5} & \frac{3}{34} & -\frac{1}{17} \\ \frac{7}{51} & \frac{51}{51} & \frac{5}{51} & -\frac{5}{51} & -\frac{2}{51} & -\frac{7}{51} \end{bmatrix},$$

$$A^{(1,2,4)} = \begin{bmatrix} -\frac{11}{17} & \frac{43}{51} & -\frac{11}{17} & -\frac{11}{51} & \frac{86}{51} & -\frac{43}{51} \\ \frac{7}{17} & -\frac{32}{51} & \frac{7}{17} & \frac{7}{51} & -\frac{64}{51} & \frac{32}{51} \\ \frac{4}{17} & -\frac{11}{51} & \frac{4}{17} & \frac{4}{51} & -\frac{22}{51} & \frac{11}{51} \\ \frac{1}{17} & \frac{10}{51} & \frac{1}{17} & \frac{1}{51} & \frac{20}{51} & -\frac{10}{51} \end{bmatrix},$$

$$A^\dagger = \begin{bmatrix} -\frac{5}{34} & -\frac{3}{17} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{17} & \frac{5}{34} \\ \frac{51}{4} & \frac{13}{17} & -\frac{5}{34} & -\frac{34}{5} & -\frac{13}{17} & -\frac{4}{34} \\ \frac{51}{102} & \frac{102}{5} & -\frac{1}{102} & \frac{1}{5} & -\frac{102}{5} & -\frac{51}{102} \\ \frac{1}{17} & -\frac{1}{34} & \frac{3}{34} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{17} \end{bmatrix}.$$

**Primer 2.9.7.** Za matricu  $R = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 1 & 1 & -2 & -5 \end{bmatrix}$  dobijamo

$$R = \begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 2 & -1 & 0 & 1 \end{bmatrix}, \quad T_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} -1 & -2 \\ -1 & -3 \end{bmatrix}, \quad E = I_4.$$

Koristeći matrice

$$W_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \\ 4 & 2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & 1 & 3 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix},$$

dobija se:

$$A^{(1,2,3)} = \begin{bmatrix} -\frac{10}{51} & -\frac{6}{51} & \frac{8}{51} & \frac{2}{51} \\ -\frac{13}{51} & \frac{20}{51} & -\frac{7}{51} & -\frac{2}{51} \\ -\frac{5}{51} & -\frac{3}{51} & \frac{4}{51} & \frac{1}{51} \\ -\frac{2}{17} & \frac{4}{51} & \frac{2}{51} & -\frac{8}{51} \end{bmatrix}, \quad A^{(1,2,4)} = \begin{bmatrix} -\frac{120}{221} & -\frac{40}{221} & \frac{2}{13} & -\frac{1}{13} \\ \frac{81}{221} & \frac{27}{221} & -\frac{2}{13} & \frac{1}{13} \\ \frac{3}{221} & \frac{1}{221} & 0 & 0 \\ -\frac{3}{221} & -\frac{1}{221} & \frac{2}{13} & -\frac{1}{13} \end{bmatrix},$$

$$A^\dagger = \begin{bmatrix} -\frac{11}{51} & -\frac{6}{51} & \frac{7}{51} & \frac{4}{51} \\ -\frac{7}{51} & \frac{13}{51} & -\frac{2}{51} & -\frac{1}{51} \\ \frac{4}{51} & \frac{51}{51} & -\frac{1}{51} & -\frac{17}{51} \\ \frac{1}{51} & -\frac{1}{17} & \frac{4}{51} & -\frac{5}{51} \end{bmatrix}, \quad A^\# = \begin{bmatrix} -5 & 4 & 1 & -2 \\ -21 & 17 & 4 & -9 \\ 16 & -13 & -3 & 7 \\ -11 & 9 & 2 & -5 \end{bmatrix}.$$

**Posledica 2.9.14.** Neka je  $A \in \mathbb{C}_r^{m \times n}$  i  $T, S$  podskupovi od  $\mathbb{C}^n$  i  $\mathbb{C}^m$ , respektivno, takvi da je  $\dim T = \dim S^\perp = t \leq r$  i  $AT \oplus S = \mathbb{C}^m$ . Onda postoji  $\{2\}$ -inverz od  $A, X = A_{T,S}^{(2)}$ , takav da je  $R(X) = T, N(X) = S$ . Šta više, ako  $D \in \mathbb{C}^{n \times m}, C \in \mathbb{C}^{n \times t}$  zadovoljava  $R(D) = T, N(D) = S, R(C) = T$ , onda blokovsko predstavljanje za  $A_{T,S}^{(2)}$  može biti izvršeno na isti način kao blok predstavljanje  $(G_2^i), i = 1, \dots, n$  koristeći sledeću smenu:

$$W_1 = C, \quad W_2 = C^*D.$$

## 2.10 Blokovska LDL\* dekompozicija potpunog ranga

Procedura za blok Cholesky dekompoziciju podrazumeva izračunavanje kvadratnog korena dijagonalne bloka i potom množenje inverza trougaonog bloka i kvadratnog bloka. Određivanje kvadratnog korena blok matrice može biti težak i dug zadatak. Poznato je nekoliko metoda za nalaženje aproksimacije blok LU i blok Cholesky faktorizacije (videti [13]). Pritom, množenje inverza trougaonog bloka sa blok matricom se može postići tzv. "back substitution" procesom.

Zamena LL\* dekompozicije sa LDL\* radi izbegavanja elemenata sa kvadratnim ko- renima je dobro objašnjena tehnika iz linearne algebri (videti na primer rad Goluba i

Van Loana [25]). Pritom je generisana jedna dodatna dijagonalna matrica  $D$ , ali totalna količina izračunavanja je ista kao kod Cholesky dekompozicije. Primetimo još da za datu Hermitsku matricu  $A$ , matrica  $D$  mora imati ne-nula pozitivne elemente.

Posmatrajmo sada proizvoljan skup normalnih matričnih jednačina. On se može izraziti u sledećem matričnom obliku:

$$Ax = B,$$

gde je  $x$   $n$ -dimenzionalni vektor rešenja. Ovde navodimo neke preliminarne rezultate iz rada [75], za izračunavanje vektora  $x$ , zasnovane na  $LDL^*$  faktorizaciji matrice  $A$ . Neka je matrica  $A$  datog normalnog sistema jednačina Hermitska i pozitivno definitna. Tada postoje nesingularna donja trougaona matrica  $L$  i nesingularna dijagonalna matrica  $D$  takve da je

$$A = LDL^*.$$

Inverzna matrica  $A^{-1}$  i vektor rešenja  $x$  se mogu odrediti na sledeće načine:

$$\begin{aligned} A^{-1} &= L^{*-1} \cdot D^{-1} \cdot L^{-1}, \\ x &= L^{*-1} \cdot D^{-1} \cdot L^{-1} \cdot B. \end{aligned} \tag{2.10.1}$$

Ove jednačine podrazumevaju inverz trougaone i dijagonalne matrice. Međutim, to nije neophodno, jer je moguće podeliti matricu trougaonom matricom bez računanja inverza. Stoga, posmatrajmo sledeće pomeranje:

$$C = DL^*x.$$

Tada  $L \cdot C = B$  i važe sledeći izrazi:

$$c_i = \frac{b_i - \sum_{k=1}^{i-1} l_{i,k} \cdot c_k}{l_{i,i}}, \quad \text{for } i = \overline{1, n}. \tag{2.10.2}$$

Očigledno je proizvod matrica  $DL^*$  gornja trougaona matrica matrica, i samim važe sledeće jednakosti:

$$x_n = \frac{c_n}{d_{n,n}}, \tag{2.10.3}$$

$$x_i = \frac{c_i - \sum_{k=i+1}^n d_{i,i} l_{k,i} \cdot x_k}{d_{i,i}}, \quad \text{for } i = \overline{n-1, 1} \tag{2.10.4}$$

Glavna motivacija je odrediti proceduru za blokovsku  $LDL^*$  dekompoziciju, za opšti slučaj  $n \times n$  blok Hermitske matrice  $A \in \mathbf{C}^{m \times m}$ . Ovim se izbegavaju poteškoće kod izračunavanja korenih elemenata submatrica (često zvanih blokovima).

Neka je Hermitska matrica  $A$  podeljena u  $n \times n$  blockova, tako da je  $A = [A_{ij}]$ ,  $1 \leq i, j \leq n$ , pri čemu su dijagonalni blokovi kvadratne matrice. Potrebno je odrediti donju blok trougaonu matricu  $L$  i blok dijagonalnu matricu  $D$ , takve da je jednakost  $A = LDL^*$  zadovoljena, pri čemu su dimenzijsi sub-matrica  $L_{i,j}$  i  $D_{i,j}$  jednake  $d_i \times d_j$ . Jednostavna modifikacija iterativne procedure za  $LDL^*$  dekompoziciju koja važi za blokovske elemente matrica  $D$  i  $L$ , data je u [75].

**Teorema 2.10.1.** [75] Neka je data Hermitova blok matrica  $A = [A_{ij}]_{n \times n}$  ranga  $r$ , podeljena na takav način da su dimenzije svake sub-matrice  $A_{i,j}$  jednake  $d_i \times d_j$ , i neka postoji prirodan broj  $m \leq n$  takav da važi izraz  $\sum_{k=1}^m d_k = r$ . Neka su  $L = [L_{ij}]_{n \times m}$  i  $D = [D_{ij}]_{m \times m}$  blokovske matrice, gde je dimenzija svake sub-matrice  $L_{ij}$  i  $D_{ij}$  jednaka  $d_i \times d_j$ . Ako važe sledeće matrične jednačine za  $j = \overline{1, m}$ :

$$D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk} \cdot D_{kk} \cdot L_{jk}^*, \quad (2.10.5)$$

$$L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik} \cdot D_{kk} \cdot L_{jk}^*) \cdot D_{jj}^{-1}, \quad i = \overline{j+1, n}, \quad (2.10.6)$$

pri čemu su sub-matrice  $D_{jj}$ ,  $j = \overline{1, m}$  nesingularne, tada je  $LDL^*$  dekompozicija potpunog ranga matrice  $A$ .

**Dokaz.** Kako je  $A = LDL^*$ , to za proizvoljne indekse  $1 \leq i, j \leq n$  važi sledeći identitet:

$$A_{ij} = \sum_{k=1}^m L_{ik} D_{kk} L_{jk}^* = \sum_{k=1}^{\min\{i,j\}} L_{ik} D_{kk} L_{jk}^*,$$

s obzirom da su sledeće matrične jednakosti zadovoljene za svako  $i = \overline{1, m}$ :

$$\begin{aligned} L_{ii} &= \mathbf{I}_{d_i}, \\ L_{ij} &= \mathbf{0}_{d_i \times d_j}, \quad D_{ij} = \mathbf{0}_{d_i \times d_j}, \quad D_{ji} = \mathbf{0}_{d_j \times d_i}, \quad j = \overline{i+1, m}. \end{aligned}$$

Stoga je, za svaki indeks  $j = \overline{1, m}$ :

$$A_{jj} = \sum_{k=1}^j L_{jk} D_{kk} L_{jk}^* = D_{jj} + \sum_{k=1}^{j-1} L_{jk} D_{kk} L_{jk}^*$$

tačna jednakost, iz koje sledi (2.10.5).

Dalje, za slučaj  $i > j$  imamo

$$A_{ij} = \sum_{k=1}^j L_{ik} D_{kk} L_{jk}^* = L_{ij} D_{jj} L_{jj}^* + \sum_{k=1}^{j-1} L_{ik} D_{kk} L_{jk}^*.$$

Dakle, zadovoljeno je sledeće:

$$L_{ij} D_{jj} = A_{ij} - \sum_{k=1}^{j-1} L_{ik} D_{kk} L_{jk}^*,$$

i time je jednačina (2.10.6) validna, jer je sub-matrica  $D_{jj}$  nesingularna.

S obzirom da je izraz  $\sum_{k=1}^m d_k = r$  zadovoljen, matrice  $L = [L_{ij}]_{n \times m}$  i  $D = [D_{ij}]_{m \times m}$  su obe ranga  $r$ , na osnovu jednakosti (2.10.5) i (2.10.6). Dakle,  $LDL^*$  je dekompozicija potpunog ranga matrice  $A$ .  $\square$

Očigledno, iterativna procedura za blok  $LDL^*$  dekompoziciju Hermitske matrice određena je iz jednačina (2.10.5) i (2.10.6). U svakoj iteraciji, sub-matrice  $D_{jj}$  i  $L_{ij}$ ,  $i > j$  se mogu izračunati na osnovu prethodnih rezultata i poznatih blokova  $A_{ij}$ .

Primetimo da uslov da sub-matrice  $D_{jj}$ ,  $j = \overline{1, n}$  budu ne-singularne može biti prejak u nekim slučajevima. Ovi se uslovi mogu izbeći korišćenjem Moore-Penrose-ovih inverza matrica  $D_{jj}$ , u kom slučaju jednačina (2.10.6) postaje:

$$L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik} \cdot D_{kk} \cdot L_{jk}^*) \cdot D_{jj}^\dagger, \quad i = \overline{j+1, n}. \quad (2.10.7)$$

Iterativna procedura za  $LDL^*$  dekompoziciju blok vektora rešenja je slična kao u navedenom pristupu.

Dakle, kao produžetak Teoreme 2.10.1, sledeći algoritam, kojim se generišu blokovske matrice potpunog ranga  $L$ ,  $D$  dimenzija manjih ili jednakih dimenzijama matrice  $A$ , se može formulisati.

---

**Algoritam 2.11** Blokovska  $LDL^*$  dekompozicija potpunog ranga podeljene matrice

---

**Ulaz:** Blokovska matrica  $A = [A_{ij}]_{i,j=1}^n$  ranga  $r$ , gde su dimenzijsi proizvoljnog bloka  $A_{i,j}$  jednake  $d_i \times d_j$ , a za neko  $m \in \{1, \dots, n\}$  je zadovoljen identitet  $\sum_{k=1}^m d_k = r$ .

1: **Inicijalizacija:** za  $i = \overline{1, m}$  izvršiti sledeće korake:

- 1.1: Postaviti  $L_{ii} = \mathbf{I}_{d_i}$ .
- 1.2: Za  $j = \overline{i+1, n}$  postaviti  $L_{ij} = \mathbf{0}_{d_i \times d_j}$ .
- 1.2: Za  $j = \overline{i+1, m}$  postaviti  $D_{ij} = \mathbf{0}_{d_i \times d_j}$ ,  $D_{ji} = \mathbf{0}_{d_j \times d_i}$ .

2: **Evaluacija:** za svako  $j = \overline{1, m}$  izvršiti korake 2.1 i 2.2:

- 2.1: Postaviti  $D_{jj} = A_{jj} - \sum_{k=1}^{j-1} L_{jk} \cdot D_{kk} \cdot L_{jk}^*$ .
  - 2.2: Za  $i = \overline{j+1, n}$  postaviti  $L_{ij} = (A_{ij} - \sum_{k=1}^{j-1} L_{ik} \cdot D_{kk} \cdot L_{jk}^*) \cdot D_{jj}^{-1}$ .
- 

**Primer 2.10.1.** Posmatrajmo sledeću  $3 \times 3$  blokovsku matricu  $A$ :

$$A = \left[ \begin{array}{ccc|ccc|c} 5 & 6 & 7 & 5 & 6 & 7 \\ 6 & 9 & 10 & -6 & 9 & 10 \\ 7 & 10 & 13 & 7 & 10 & 13 \\ \hline 5 & -6 & 7 & 5 & -6 & -7 \\ 6 & 9 & 10 & -6 & -9 & 12 \\ \hline 7 & 10 & 13 & -7 & 12 & -13 \end{array} \right].$$

Nakon inicijalizacije sub-matrica  $D_{12}, D_{13}, D_{21}, D_{23}, D_{31}, D_{32}, L_{12}, L_{13}, L_{23}$ , izvršavamo Korak 2 Algoritma 2.11. Za  $j = 1$  sub-matrica  $D_{11}$  je jednaka  $A_{11}$ , pa je dakle, na osnovu Koraka 2.2:

$$\begin{aligned} L_{21} &= A_{21} \cdot D_{11}^{-1} = \begin{bmatrix} 7 & -12 & 6 \\ 0 & 1 & 0 \end{bmatrix}, \\ L_{31} &= A_{31} \cdot D_{11}^{-1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

U drugoj iteraciji, za  $j = 2$  izračunavaju se sub-matrici  $D_{22}$  i  $L_{32}$ :

$$\begin{aligned} D_{22} &= A_{22} - L_{21}D_{11}L_{21}^* = \begin{bmatrix} -144 & 0 \\ 0 & -18 \end{bmatrix}, \\ L_{32} &= (A_{32} - L_{31} \cdot D_{11} \cdot L_{21}^*) \cdot D_{22}^{-1} = \begin{bmatrix} \frac{7}{72} & \frac{11}{9} \end{bmatrix}. \end{aligned}$$

Na kraju, za slučaj  $j = 3$ , imamo da je

$$D_{33} = A_{33} - L_{31} \cdot D_{11} \cdot L_{31}^* - L_{32} \cdot D_{22} \cdot L_{32}^* = \begin{bmatrix} \frac{9}{4} \end{bmatrix},$$

pa je  $A = LDL^*$  blokovska dekompozicija potpunog ranga matrice  $A$ , gde je

$$L = \left[ \begin{array}{ccc|ccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 7 & -12 & 6 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & \frac{7}{72} & \frac{11}{9} & 1 \end{array} \right], \quad D = \left[ \begin{array}{ccc|cc|c} 5 & 6 & 7 & 0 & 0 & 0 \\ 6 & 9 & 10 & 0 & 0 & 0 \\ 7 & 10 & 13 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -144 & 0 & 0 \\ 0 & 0 & 0 & 0 & -18 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & \frac{9}{4} \end{array} \right]$$

Očito da ukupan broj ne-nula elemenata u matricama  $L$  i  $D$  opada pri daljem deljenju matrice  $A$ . Matrice  $L$  i  $D$  se u nekoj tački deljenja mogu smatrati retkim.

**Primer 2.10.2.** Glavna razlika između blokovske  $LDL^*$  dekompozicija potpunog ranga i njene standardne varijante je vidljiva na klasi rang-deficijentnih matrica. U tom smislu, uočimo matricu  $A_2$  ranga  $r = 2$  iz [117], podeljenu na sledeći način:

$$A_2 = \left[ \begin{array}{cc|cc|cc} a+10 & a+9 & a+8 & a+7 & a+6 & a+5 \\ a+9 & a+8 & a+7 & a+6 & a+5 & a+4 \\ \hline a+8 & a+7 & a+6 & a+5 & a+4 & a+3 \\ a+7 & a+6 & a+5 & a+4 & a+3 & a+2 \\ \hline a+6 & a+5 & a+4 & a+3 & a+2 & a+1 \\ a+5 & a+4 & a+3 & a+2 & a+1 & a \end{array} \right]$$

Primenom Algoritma 2.11 sledeće matrice potpunog ranga su dobijene kao rezultat, za  $m = 1$ :

$$L = \left[ \begin{array}{c|c} 1 & 0 \\ 0 & 1 \\ \hline -1 & 2 \\ -2 & 3 \\ \hline -3 & 4 \\ -4 & 5 \end{array} \right], \quad D = \left[ \begin{array}{cc} 10+a & 9+a \\ 9+a & 8+a \end{array} \right].$$

**Primer 2.10.3.** Posmatrajmo rang-deficijentnu simboličku matricu  $F_6$  dobijenu u radu [117], podeljenu na sledeći način:

$$F_6 = \left[ \begin{array}{cc|cc|c} a+6 & a+5 & a+4 & a+3 & a+2 & a+1 \\ a+5 & a+5 & a+4 & a+3 & a+2 & a+1 \\ \hline a+4 & a+4 & a+4 & a+3 & a+2 & a+1 \\ a+3 & a+3 & a+3 & a+3 & a+2 & a+1 \\ a+2 & a+2 & a+2 & a+2 & a+1 & a \\ \hline a+1 & a+1 & a+1 & a+1 & a & a-1 \end{array} \right].$$

Primetimo da je  $\text{rang}(F_6) = 5$ , i zato iz blokovske  $LDL^*$  faktorizacije potpunog ranga proizilaze matrice  $L \in \mathbb{C}(a)_5^{6 \times 5}$  i  $D \in \mathbb{C}(a)_5^{5 \times 5}$ . Primenom Algoritma 2.11 na matricu  $F_6$ , za slučaj  $n = 3$ ,  $m = 2$ , generišu se sledeće racionalne matrice kao rezultat:

$$L = \left[ \begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & \frac{4+a}{5+a} & 1 & 0 & 0 \\ 0 & \frac{3+a}{5+a} & 0 & 1 & 0 \\ 0 & \frac{2+a}{5+a} & 0 & 0 & 1 \\ 0 & \frac{1+a}{5+a} & 0 & -1 & 2 \end{array} \right], \quad D = \left[ \begin{array}{cc|ccc} a+6 & a+5 & 0 & 0 & 0 \\ a+5 & a+5 & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{4+a}{5+a} & \frac{3+a}{5+a} & \frac{2+a}{5+a} \\ 0 & 0 & \frac{3+a}{5+a} & \frac{2(3+a)}{5+a} & \frac{2(2+a)}{5+a} \\ 0 & 0 & \frac{2+a}{5+a} & \frac{2(2+a)}{5+a} & \frac{1+2a}{5+a} \end{array} \right].$$

### 2.10.1 Inverzi $n \times n$ blokovskih matrica

Matrice  $L$  i  $D$ , uključene u blokovskoj  $LDL^*$  dekompoziciji, su donja blok trougaona i blok dijagonalna matrica, respektivno. Ispitajmo najpre inverze ovih tipova blokovskih matrica.

**Lema 2.10.2.** *Neka je data blok dijagonalna matrica  $D$  sa  $n$  sub-matrica na dijagonalni. Tada je  $D$  invertibilna matrica ako i samo ako je  $D_i$  kvadratna i nesingularna matrica za svako  $1 \leq i \leq n$ , u kojem slučaju je  $D^{-1}$  takođe blok dijagonalna matrica oblika*

$$D^{-1} = \begin{bmatrix} D_1^{-1} & 0 & \cdots & 0 \\ 0 & D_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_n^{-1} \end{bmatrix}.$$

**Lema 2.10.3.** [75] *Inverz nesingularne donje blok trougaone matrice*

$$L = \begin{bmatrix} I & 0 & \cdots & 0 \\ L_{21} & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & I \end{bmatrix},$$

je donja blok trougaona matrica oblika

$$L^{-1} = \begin{bmatrix} I & 0 & \cdots & 0 \\ (L^{-1})_{21} & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (L^{-1})_{n1} & (L^{-1})_{n2} & \cdots & I \end{bmatrix},$$

pri čemu je  $(L^{-1})_{i+1,i} = -L_{i+1,i}$  za svaki indeks  $i = \overline{1, n-1}$ .

**Dokaz.** Očigledno, matrica  $L^{-1}$  je donja blok dijagonalna matrica sa jediničnim maticama na dijagonalni.

Na osnovu jednačine  $LL^{-1} = I$ , imamo sistem od  $n^2$  matričnih jednačina. Za proizvoljno  $1 \leq i < n$ , množeći  $(i+1)$ -tu vrstu matrice  $L$  sa  $i$ -tom kolonom inverza  $L^{-1}$  imamo

$$L_{i+1,i}I + I(L^{-1})_{i+1,i} = 0,$$

na osnovu čega je  $(L^{-1})_{i+1,i} = -L_{i+1,i}$ .  $\square$

Kao što smo već ustanovili, LDL\* dekompozicija matrice  $A$  se može upotrebiti za određivanje njenog inverza:

$$A^{-1} = (L^{-1})^* \cdot D^{-1} \cdot L^{-1}. \quad (2.10.8)$$

Metod za kalkulaciju Hermitske inverzne matrice u blok  $LDL^*$  dekompoziciji je baziran na ovoj jednačini. Dakle, iz ovoga sledi naredna teorema uvedena u radu [75].

**Teorema 2.10.4.** [75] *Posmatrajmo blokovsku  $LDL^*$  dekompoziciju nesingularne Hermitske blok matrice  $A = [A_{ij}]_{i,j=1}^n$ , pri čemu matrice  $L$  i  $D$  imaju istu podelu kao i  $A$ . Ako su matrica  $L$  i sub-matrice  $D_{jj}$ ,  $j = \overline{1, n}$  nesingularne, tada važe sledeće matrične jednačine:*

$$(A^{-1})_{ij} = \sum_{k=j}^n (L^{-1})_{ki}^* D_{kk}^{-1} (L^{-1})_{kj}, \quad 1 \leq i \leq j \leq n. \quad (2.10.9)$$

**Dokaz.** Sledi na osnovu Leme 2.10.2 i jednačine (2.10.8) posmatranjem podele inverzne matrice  $L^{-1}$  iste kao podele matrice  $A$ .  $\square$

**Posledica 2.10.5.** [75] *Neka je  $A \in \mathbf{C}^{m \times m}$  Hermitska  $2 \times 2$  blokovska matrica, podeljena na sledeći način:*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}.$$

*Posmatrajmo blokovsku  $LDL^*$  dekompoziciju matrice  $A$ , gde matrice  $L$  i  $D$  imaju istu podelu kao i  $A$ . Ako su sub-matrice  $D_{11}$ ,  $D_{22}$  nesingularne, tada inverzna matrica  $A^{-1}$  ima sledeću formu:*

$$A^{-1} = \begin{bmatrix} D_{11}^{-1} + L_{21}^* D_{22}^{-1} L_{21} & -L_{21}^* D_{22}^{-1} \\ -D_{22}^{-1} L_{21} & D_{22}^{-1} \end{bmatrix}. \quad (2.10.10)$$

**Dokaz.** Sledi iz Teoreme 2.10.4 i Leme 2.10.3, s obzirom da su validni sledeći identiteti:

$$\begin{aligned} A^{-1} &= \begin{bmatrix} L_{11}^* D_{11}^{-1} L_{11} + L_{21}^* D_{22}^{-1} L_{21} & -L_{21}^* D_{22}^{-1} L_{22} \\ -L_{22}^* D_{22}^{-1} L_{21} & L_{22}^* D_{22}^{-1} L_{22} \end{bmatrix} \\ &= \begin{bmatrix} D_{11}^{-1} + L_{21}^* D_{22}^{-1} L_{21} & -L_{21}^* D_{22}^{-1} \\ -D_{22}^{-1} L_{21} & D_{22}^{-1} \end{bmatrix}. \end{aligned} \quad (2.10.11)$$

$\square$

Očito je ova reprezentacija inverza  $2 \times 2$  blok matrice jednostavnija nego njoj slična reprezentacija iz rada [45], koja je bazirana na Schurovom komplementu. Zapravo, reprezentacija (2.10.10) je posledica Teoreme 2.1 iz [45], za slučaj  $D_{11} = A_{11}$ ,  $L_{21} = A_{12}^* D_{11}^{-1}$  i Schurovog komplementa  $D_{22} = A_{22} - A_{12}^* D_{11}^{-1} A_{12}$ . Takođe, uslovi da su  $D_{11}$ ,  $D_{22}$  nesingularne matrice su ekvivalentni uslovima da su matrice  $A_{11}$  i Schurov komplement  $A_{22} - A_{12}^* A_{11}^{-1} A_{12}$  nesingularni.

**Primer 2.10.4.** Uočimo sledeću podeljenu matricu:

$$A = \left[ \begin{array}{cc|ccc} -1 & 6 & 5 & 6 & 7 \\ 6 & 9 & -6 & 0 & 10 \\ \hline 5 & -6 & 5 & -6 & -7 \\ 6 & 0 & -6 & -9 & 12 \\ 7 & 10 & -7 & 12 & -13 \end{array} \right].$$

Tada su matrice  $L$  i  $D$  iz blokovske  $LDL^*$  faktorizacije matrice  $A$  jednake:

$$L = \left[ \begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline -\frac{9}{5} & \frac{8}{15} & 1 & 0 & 0 \\ -\frac{6}{5} & \frac{4}{5} & 0 & 1 & 0 \\ -\frac{1}{15} & \frac{52}{45} & 0 & 0 & 1 \end{array} \right], \quad D = \left[ \begin{array}{cc|ccc} -1 & 6 & 0 & 0 & 0 \\ 6 & 9 & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{86}{5} & \frac{24}{5} & \frac{4}{15} \\ 0 & 0 & \frac{24}{5} & -\frac{9}{5} & \frac{62}{5} \\ 0 & 0 & \frac{4}{15} & \frac{62}{5} & -\frac{1084}{45} \end{array} \right].$$

Da bi izračunali  $A^{-1}$ , imamo sledeću sekvencu izraza:

$$\begin{aligned} (A^{-1})_{11} &= \begin{bmatrix} -1 & 6 \\ 6 & 9 \end{bmatrix}^{-1} + \begin{bmatrix} -\frac{9}{5} & \frac{8}{15} \\ -\frac{6}{5} & \frac{4}{5} \\ -\frac{1}{15} & \frac{52}{45} \end{bmatrix}^T \cdot \begin{bmatrix} \frac{86}{5} & \frac{24}{5} & \frac{4}{15} \\ \frac{24}{5} & -\frac{9}{5} & \frac{62}{5} \\ \frac{4}{15} & \frac{62}{5} & -\frac{1084}{45} \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\frac{9}{5} & \frac{8}{15} \\ -\frac{6}{5} & \frac{4}{5} \\ -\frac{1}{15} & \frac{52}{45} \end{bmatrix} \\ &= \begin{bmatrix} \frac{99}{656} & -\frac{33}{164} \\ -\frac{33}{164} & \frac{96}{205} \end{bmatrix}, \\ (A^{-1})_{12} &= (A^{-1})_{21}^* = - \begin{bmatrix} -\frac{9}{5} & \frac{8}{15} \\ -\frac{6}{5} & \frac{4}{5} \\ -\frac{1}{15} & \frac{52}{45} \end{bmatrix}^T \cdot \begin{bmatrix} \frac{86}{5} & \frac{24}{5} & \frac{4}{15} \\ \frac{24}{5} & -\frac{9}{5} & \frac{62}{5} \\ \frac{4}{15} & \frac{62}{5} & -\frac{1084}{45} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{13}{328} & \frac{223}{984} & \frac{75}{656} \\ \frac{33}{410} & -\frac{481}{1230} & -\frac{25}{164} \end{bmatrix}, \\ (A^{-1})_{22} &= \begin{bmatrix} \frac{86}{5} & \frac{24}{5} & \frac{4}{15} \\ \frac{24}{5} & -\frac{9}{5} & \frac{62}{5} \\ \frac{4}{15} & \frac{62}{5} & -\frac{1084}{45} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{69}{820} & -\frac{223}{2460} & -\frac{15}{328} \\ -\frac{820}{2460} & \frac{259}{820} & \frac{53}{328} \\ -\frac{15}{328} & \frac{53}{328} & \frac{27}{656} \end{bmatrix}, \end{aligned}$$

na osnovu čega je inverzna matrica od  $A$  jednaka

$$A^{-1} = \left[ \begin{array}{cc|cc|c} \frac{99}{656} & -\frac{33}{164} & \frac{13}{328} & \frac{223}{984} & \frac{75}{656} \\ -\frac{33}{164} & \frac{96}{205} & \frac{33}{410} & -\frac{481}{1230} & -\frac{164}{25} \\ \hline \frac{13}{328} & \frac{33}{410} & \frac{69}{820} & -\frac{223}{2460} & -\frac{15}{328} \\ \frac{223}{984} & -\frac{481}{1230} & -\frac{223}{820} & \frac{259}{2460} & \frac{53}{328} \\ \frac{75}{656} & -\frac{25}{164} & -\frac{15}{328} & \frac{53}{328} & \frac{27}{656} \end{array} \right].$$

## 2.10.2 Moore-Penrose-ovi inverzi $2 \times 2$ blokovskih matrica

Inverzi  $2 \times 2$  blok matrica su dobro istraženi (videti npr. [45, 55]), i često se pojavljuju u mnogim izračunavanjima. Neke eksplisitne inverzne formule za  $2 \times 2$  blok matrice su uvedene u radu [45], a bazirane su na Schurovom komplementu  $D - CA^{-1}B$  matrice  $A$ .

**Lema 2.10.6.** *Neka je data  $2 \times 2$  nesingulararna podeljena matrica*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

*pri čemu je  $A \in \mathbf{C}^{k \times k}$ ,  $B \in \mathbf{C}^{k \times l}$ ,  $C \in \mathbf{C}^{l \times k}$  i  $D \in \mathbf{C}^{l \times l}$ . Ako je submatrica  $A$  nesingularna, tada Schurov komplement  $S_A = D - CA^{-1}B$  matrice  $A$  u  $M$  je takođe nesingularan, a inverz matrice  $M$  ima sledeću formu*

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix}. \quad (2.10.12)$$

Mnogo radova navodi metode za izračunavanje Moore-Penrose-ovog inverza  $2 \times 2$  blokovskih matrica (videti npr. [34]). Takođe, neke reprezentacije Drazinovog inverza  $2 \times 2$  blok matrica su dobijene u [18]. Ipak, opšti slučaj  $n \times n$  blokovskih matrica,  $n > 2$ , nije toliko opisan. Dakle, u radu [75] smo izveli metode za  $n \times n$  blok matrice i onda posmatrali specijalan slučaj  $2 \times 2$  blok matrica.

Tvrđenje analogno Teoremi 2.10.4 se može uvesti radi izračunavanja Moore-Penrose-ovog inverza Hermitske  $n \times n$  blok matrice, posmatrajući njenu blokovsku  $LDL^*$  dekompoziciju potpunog ranga.

**Teorema 2.10.7.** [75] *Neka je data Hermitska blok matrica  $A = [A_{i,j}]_{n \times n}$  ranga  $r$ , gde su dimenzije proizvoljne matrice  $A_{i,j}$  date sa  $d_i \times d_j$ , i neka postoji prirodan broj  $m \leq n$  takav da je zadovoljen izraz  $\sum_{k=1}^m d_k = r$ . Prepostavimo da je  $LDL^*$  blokovska dekompozicija potpunog ranga matrice  $A$ , pri čemu su sub-matrice  $D_{jj}$ ,  $j = \overline{1, m}$  nesingularne. Tada važe sledeće matrične jednačine:*

$$(A^\dagger)_{ij} = \sum_{k=j}^m (L^\dagger)_{ki}^* D_{kk}^{-1} (L^\dagger)_{kj}, \quad 1 \leq i \leq j \leq n. \quad (2.10.13)$$

Sada smo u mogućnosti da razvijemo metod za izračunavanje Moore-Penrose-ovog inverza  $2 \times 2$  blokovske matrice, dat u sledećoj posledici Teoreme 3.2 i Leme 2.10.6.

**Teorema 2.10.8.** [75] *Neka su  $A \in \mathbf{C}^{m \times n}$  i  $(A^*A)^2 \in \mathbf{C}^{n \times n}$   $2 \times 2$  blokovske matrice formi*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (A^*A)^2 = \begin{bmatrix} B & C \\ C^* & E \end{bmatrix},$$

*sa istom podelom. Uvedimo sledeće smene, za slučaj nesingularne matrice  $B$ :*

$$\begin{aligned} X_{11} &= B + N^* + N + M^*EM, \\ X_{12} &= C + ME, \\ X_{22} &= E, \end{aligned} \quad (2.10.14)$$

*pri čemu su matrice  $M$ ,  $N$  određene kao*

$$\begin{aligned} M &= C^*B^{-1}, \\ N &= CM. \end{aligned} \quad (2.10.15)$$

Ako su matrica  $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^* & X_{22} \end{bmatrix}$  i sub-matrice  $X_{11}, B$  nesingularne, tada je Moore-Penrose-ov inverz matrice  $A$  dat u sledećem obliku:

$$A^\dagger = \begin{bmatrix} Y_{11}\Sigma_{11} + (Y_{11}M^* + Y_{12})\Sigma_{21} \\ (MY_{11} + Y_{21})(\Sigma_{11} + M^*\Sigma_{21}) + (MY_{12} + Y_{22})\Sigma_{21} \\ Y_{11}\Sigma_{12} + (Y_{11}M^* + Y_{12})\Sigma_{22} \\ (MY_{11} + Y_{21})(\Sigma_{12} + M^*\Sigma_{22}) + (MY_{12} + Y_{22})\Sigma_{22} \end{bmatrix}, \quad (2.10.16)$$

gde je

$$\begin{aligned} Y_{11} &= X_{11}^{-1} + X_{11}^{-1}X_{12}Y_{22}X_{12}^*X_{11}^{-1}, \\ Y_{12} &= -X_{11}^{-1}X_{12}Y_{22}, \\ Y_{21} &= -Y_{22}X_{12}^*X_{11}^{-1}, \\ Y_{22} &= (X_{22} - X_{12}^*X_{11}^{-1}X_{12})^{-1}, \\ \Sigma_{11} &= (A_{11}^*A_{11} + A_{21}^*A_{21})A_{11}^* + (A_{11}^*A_{12} + A_{21}^*A_{22})A_{12}^*, \\ \Sigma_{12} &= (A_{11}^*A_{11} + A_{21}^*A_{21})A_{21}^* + (A_{11}^*A_{12} + A_{21}^*A_{22})A_{22}^*, \\ \Sigma_{21} &= (A_{12}^*A_{11} + A_{22}^*A_{21})A_{11}^* + (A_{12}^*A_{12} + A_{22}^*A_{22})A_{12}^*, \\ \Sigma_{22} &= (A_{12}^*A_{11} + A_{22}^*A_{21})A_{21}^* + (A_{12}^*A_{12} + A_{22}^*A_{22})A_{22}^*. \end{aligned}$$

**Dokaz.** Posmatrajmo blokovsku  $LDL^*$  dekompoziciju potpunog ranga Hermitske matrice  $(A^*A)^2$ , i uvedimo odgovarajuće unifikacije nekih sub-matrica iz  $L$  i  $D$ , ako je potrebno, tako da su  $L$  i  $D$   $2 \times 2$  blokovske matrice. Posmatranjem proizvoda matrica  $L^*LDL^*L$ , sa istom podelom, zadovoljene su sledeće matrične jednačine:

$$\begin{aligned} (L^*LDL^*L)_{11} &= L_{11}^*L_{11}D_{11}L_{11}^*L_{11} + L_{21}^*L_{21}D_{11}L_{11}^*L_{11} + L_{11}^*L_{11}D_{11}L_{21}^*L_{21} \\ &\quad + L_{21}^*L_{21}D_{11}L_{21}^*L_{21} + L_{21}^*L_{22}D_{22}L_{22}^*L_{21} \\ &= D_{11} + L_{21}^*L_{21}D_{11} + D_{11}L_{21}^*L_{21} + L_{21}^*L_{21}D_{11}L_{21}^*L_{21} + L_{21}^*D_{22}L_{21} \\ &= B + B^{-1}CC^*B^{-1}B + BB^{-1}CC^*B^{-1} \\ &\quad + B^{-1}CC^*B^{-1}BB^{-1}CC^*B^{-1} + B^{-1}C(E - C^*B^{-1}C)C^*B^{-1} \\ &= B + B^{-1}CC^* + CC^*B^{-1} + B^{-1}(CC^*B^{-1})^2 + B^{-1}C(E - C^*B^{-1}C)C^*B^{-1} \\ &= B + B^{-1}CC^* + CC^*B^{-1} + B^{-1}CEC^*B^{-1} \\ &= X_{11}, \\ (L^*LDL^*L)_{12} &= L_{11}^*L_{11}D_{11}L_{21}^*L_{22} + L_{21}^*L_{21}D_{11}L_{21}^*L_{22} + L_{21}^*L_{22}D_{22}L_{22}^*L_{22} \\ &= D_{11}L_{21}^* + L_{21}^*L_{21}D_{11}L_{21}^* + L_{21}^*D_{22} \\ &= BB^{-1}C + B^{-1}CC^*B^{-1}BB^{-1}C + B^{-1}C(E - C^*B^{-1}C) \\ &= C + B^{-1}CC^*B^{-1}C + B^{-1}CE - B^{-1}CC^*B^{-1}C \\ &= C + B^{-1}CE = X_{12}, \\ (L^*LDL^*L)_{21} &= (L^*LDL^*L)_{12}^* = X_{12}^*, \\ (L^*LDL^*L)_{22} &= L_{22}^*L_{21}D_{11}L_{21}^*L_{22} + L_{22}^*L_{22}D_{22}L_{22}^*L_{22} \\ &= L_{21}D_{11}L_{21}^* + D_{22} \\ &= C^*B^{-1}BB^{-1}C + E - C^*B^{-1}C \\ &= E = X_{22}. \end{aligned}$$

Na osnovu Leme 2.10.6, inverzna matrica  $X^{-1}$  je jednaka blokovskoj matrici

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}.$$

Tada su sledeće jednakosti zadovoljene:

$$\begin{aligned} LX^{-1}L^* &= \begin{bmatrix} L_{11}Y_{11}L_{11}^* & L_{11}Y_{11}L_{21}^* + L_{11}Y_{12}L_{22}^* \\ L_{21}Y_{11}L_{11}^* + L_{22}Y_{21}L_{21}^* & (L_{21}Y_{21} + L_{22}Y_{21})L_{21}^* + (L_{21}Y_{12} + L_{22}Y_{22})L_{22}^* \end{bmatrix} \\ &= \begin{bmatrix} Y_{11} & Y_{11}L_{21}^* + Y_{12} \\ L_{21}Y_{11} + Y_{21} & (L_{21}Y_{11} + Y_{21})L_{21}^* + L_{21}Y_{12} + Y_{22} \end{bmatrix}, \\ A^*AA^* &= \begin{bmatrix} (A_{11}^*A_{11} + A_{21}^*A_{21})A_{11}^* + (A_{11}^*A_{12} + A_{21}^*A_{22})A_{12}^* \\ (A_{12}^*A_{11} + A_{22}^*A_{21})A_{11}^* + (A_{12}^*A_{12} + A_{22}^*A_{22})A_{12}^* \\ (A_{11}^*A_{11} + A_{21}^*A_{21})A_{21}^* + (A_{11}^*A_{12} + A_{21}^*A_{22})A_{22}^* \\ (A_{12}^*A_{11} + A_{22}^*A_{21})A_{21}^* + (A_{12}^*A_{12} + A_{22}^*A_{22})A_{22}^* \end{bmatrix} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}. \end{aligned}$$

Na osnovu prvog tvrđenja Teoreme 2.1.6 imamo da je  $A^\dagger = LX^{-1}L^* \cdot A^*AA^*$ . Kako važi jednakost  $L_{21} = M$ , to važi izraz (2.10.16) na osnovu proizvoda blok matrica  $LX^{-1}L^*$  i  $A^*AA^*$ .  $\square$

Za datu blok matricu  $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ , blokovi podeljene matrice  $(A^*A)^2 = \begin{bmatrix} B & C \\ C^* & E \end{bmatrix}$  se mogu jednostavno izračunati primenom notacije

$$\alpha_{ij} = A_{1i}^*A_{1j} + A_{2i}^*A_{2j}, \quad i, j = \overline{1, 2}.$$

Dakle, blokovi  $B, C, E$  se mogu dobiti na sledeći način:

$$\begin{aligned} B &= \alpha_{11}^2 + \alpha_{12}\alpha_{21}, \\ C &= \alpha_{11}\alpha_{12} + \alpha_{12}\alpha_{22}, \\ E &= \alpha_{21}\alpha_{12} + \alpha_{22}^2. \end{aligned}$$

Izračunavanje osnovnih matrica  $M = C^*B^{-1}$  i  $N = CM$  dovodi nas do evaluacije blokova matrice  $X$ , koristeći jednačine (2.10.14). Inverznu matricu  $X^{-1}$  je potrebno izračunati. Primetimo još da se blokovi matrice  $\Sigma$  mogu odrediti na sledeći način:

$$\begin{aligned} \Sigma_{11} &= \alpha_{11}A_{11}^* + \alpha_{12}A_{12}^*, \\ \Sigma_{12} &= \alpha_{11}A_{21}^* + \alpha_{12}A_{22}^*, \\ \Sigma_{21} &= \alpha_{21}A_{11}^* + \alpha_{22}A_{12}^*, \\ \Sigma_{22} &= \alpha_{21}A_{21}^* + \alpha_{22}A_{22}^*. \end{aligned}$$

Konačno, Moore-Penrose-ov inverz  $A^\dagger$  se može dobiti na osnovu (2.10.16).

**Primer 2.10.5.** Neka je data sledeća  $2 \times 2$  blokovska matrica:

$$A = \left[ \begin{array}{cc|cc} 5 & 6 & 15 & -6 \\ 16 & 9 & 6 & 9 \\ \hline 7 & 10 & 7 & 13 \\ \hline -5 & 6 & 1 & -6 \\ -6 & -9 & -6 & -19 \\ \hline 7 & 3 & -7 & 12 \end{array} \right].$$

Na osnovu Teoreme 2.11.8, blokovi  $B, C$  i  $E$  matrice  $(A^*A)^2$  su jednaki:

$$B = \begin{bmatrix} 505414 & 427211 \\ 427211 & 384895 \end{bmatrix}, \quad C = \begin{bmatrix} 276196 & 664563 \\ 272679 & 549944 \end{bmatrix}, \quad E = \begin{bmatrix} 267870 & 271621 \\ 271621 & 997375 \end{bmatrix},$$

a blokovi  $X_{11}, X_{12}$  i  $X_{22}$  imaju oblik:

$$\begin{aligned} X_{11} &= \begin{bmatrix} \frac{684308607306455887247574383}{144530479875286494081} & \frac{135121448833328319117306197}{183286313055058544078250686} \\ \frac{135121448833328319117306197}{144530479875286494081} & \frac{144530479875286494081}{144530479875286494081} \end{bmatrix}, \\ X_{12} &= \begin{bmatrix} \frac{6254091839007455}{12022083009} & \frac{2890344581407857}{1335787001} \\ \frac{6969187899173894}{12022083009} & \frac{672434395147319}{1335787001} \end{bmatrix}, \quad X_{22} = \begin{bmatrix} 267870 & 271621 \\ 271621 & 997375 \end{bmatrix}. \end{aligned}$$

Inverz podeljene matrice  $X$  se jednostavno izračunava na osnovu Lemma 2.10.6 i ima sledeći oblik:

$$Y = \left[ \begin{array}{cc|c} \frac{90644855931319}{1432836657050760392} & -\frac{10229238501285}{1432836657050760392} \\ \hline -\frac{1432836657050760392}{10229238501285} & \frac{191635092581707}{716418328525380196} \\ \hline \frac{668828031951276839297303}{17225681229402306559193379528} & -\frac{5280839149740617102018467}{17225681229402306559193379528} \\ \hline -\frac{17225681229402306559193379528}{2484236246335659119136665} & \frac{8612840614701153279596689764}{817497272144580142240559} \\ \hline -\frac{2484236246335659119136665}{17225681229402306559193379528} & \frac{2484236246335659119136665}{17225681229402306559193379528} \\ \hline \frac{668828031951276839297303}{17225681229402306559193379528} & -\frac{17225681229402306559193379528}{817497272144580142240559} \\ \hline -\frac{17225681229402306559193379528}{8612840614701153279596689764} & \frac{17225681229402306559193379528}{8612840614701153279596689764} \\ \hline -\frac{8612840614701153279596689764}{5440190027215884480648068832588025} & -\frac{3706857351834826612842527539209653}{3706857351834826612842527539209653} \\ \hline -\frac{5440190027215884480648068832588025}{3834973511600883350197925569793652588} & -\frac{23009841069605300101187553418761915528}{23009841069605300101187553418761915528} \\ \hline -\frac{3834973511600883350197925569793652588}{23009841069605300101187553418761915528} & \frac{7680617251922772621309028290323071}{23009841069605300101187553418761915528} \end{array} \right].$$

Nakon još nekoliko izračunavanja, Moore-Penrose-ov inverz matrice  $A$  je:

$$A^\dagger = \frac{1}{846414986} \begin{bmatrix} 10488670 & 51081296 & -14713647 & -25185490 & 31824819 & 20669699 \\ -5934145 & 14986696 & 18308660 & 93274281 & 7854044 & 25031234 \\ 32955519 & -18409038 & 8264571 & -38970491 & -26726457 & -40470883 \\ -12297891 & -22045404 & 12559500 & -28255629 & -36841766 & -5147046 \end{bmatrix}.$$

**Primer 2.10.6.** Neka je sada zadata rang-deficijentna matrica  $A_2$  iz Primera 2.11.2 i njena blokovska  $LDL^*$  dekompozicija potpunog ranga. Moore-Penrose-ov inverz konstantne matrice  $L$  se može jednostavno dobiti:

$$L^\dagger = \left[ \begin{array}{cc|cc|cc} \frac{11}{21} & \frac{8}{21} & \frac{5}{21} & \frac{2}{21} & -\frac{1}{21} & -\frac{4}{21} \\ \frac{8}{21} & \frac{31}{105} & \frac{21}{105} & \frac{13}{105} & \frac{4}{105} & -\frac{1}{21} \\ \hline \end{array} \right],$$

i kako je  $m = 1$  imamo da važi  $D^{-1} = D_{11}^{-1} = \begin{bmatrix} -8-x & 9+x \\ 9+x & -10-x \end{bmatrix}$ . Na osnovu Teoreme 2.11.8 se uopšteni inverz matrice  $A_2$  može izraziti kao:

$$A_2^\dagger = \left[ \begin{array}{cc|cc|cc} \frac{1}{147}(-8-3x) & \frac{1}{735}(-17-9x) & \frac{2-x}{245} & \frac{1}{735}(29+3x) & \frac{1}{735}(52+9x) & \frac{5+x}{49} \\ \frac{1}{735}(-17-9x) & \frac{-10-9x}{1225} & \frac{25-9x}{3675} & \frac{80+9x}{3675} & \frac{9(5+x)}{1225} & \frac{1}{735}(38+9x) \\ \hline \frac{2-x}{245} & \frac{25-9x}{3675} & \frac{20-3x}{3675} & \frac{3+x}{3675} & \frac{10+9x}{3675} & \frac{1}{735}(1+3x) \\ \frac{1}{735}(29+3x) & \frac{80+9x}{3675} & \frac{3+x}{3675} & \frac{-50-3x}{3675} & \frac{-115-9x}{3675} & \frac{1}{245}(-12-x) \\ \hline \frac{1}{735}(52+9x) & \frac{9(5+x)}{1225} & \frac{10+9x}{3675} & \frac{-115-9x}{3675} & \frac{-80-9x}{1225} & \frac{1}{735}(-73-9x) \\ \frac{5+x}{49} & \frac{1}{735}(38+9x) & \frac{1}{735}(1+3x) & \frac{1}{245}(-12-x) & \frac{1}{735}(-73-9x) & \frac{1}{147}(-22-3x) \end{array} \right].$$

Očito je ovaj pristup za Hermitske matrice vrlo pogodan i efikasan, s obzirom da se inverz blok dijagonalne matrice  $D$  može jednostavno odrediti prostim nalaženjem inverza dijagonalnih matrica  $D_{ii}$ ,  $i = 1, m$ .

**Primer 2.10.7.** Za blok Hermitsku matricu  $F_6$  iz Primera 2.11.3, izraz dobijen u Teoremi 2.10.7 se može primeniti za određivanje Moore-Penrose-ovog inverza  $F_6^\dagger$ . Blokovske matrice  $L$  i  $D$  dobijene ranije se mogu uzeti za evaluaciju matrice  $F_6^\dagger$ , kao

$$\begin{aligned}
 F_6^\dagger &= (L^\dagger)^* D^{-1} L^\dagger \\
 &= \left[ \begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{-4-a}{5+a} & \frac{-3-a}{5+a} & \frac{-2-a}{5+a} \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{6} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ \hline 0 & 0 & 0 & -\frac{1}{6} & \frac{1}{3} \end{array} \right] \cdot \left[ \begin{array}{cc|ccc} 1 & -1 & 0 & 0 & 0 \\ -1 & \frac{6+a}{5+a} & 0 & 0 & 0 \\ \hline 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & -a & 2+a \\ 0 & 0 & 0 & 2+a & -3-a \end{array} \right]. \\
 &\quad \left[ \begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & \frac{-4-a}{5+a} & 1 & 0 & 0 \\ 0 & \frac{-3-a}{5+a} & 0 & \frac{5}{6} & \frac{1}{3} \\ 0 & \frac{-2-a}{5+a} & 0 & \frac{1}{3} & \frac{1}{3} \end{array} \right] \\
 &= \left[ \begin{array}{cc|ccc} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ \hline 0 & -1 & 2 & -\frac{5}{6} & -\frac{1}{3} & \frac{1}{6} \\ 0 & 0 & -\frac{5}{6} & \frac{7}{9} - \frac{a}{4} & \frac{4}{9} & \frac{1}{9} + \frac{a}{4} \\ 0 & 0 & -\frac{1}{3} & \frac{4}{9} & \frac{1}{9} & -\frac{2}{9} \\ \hline 0 & 0 & \frac{1}{6} & \frac{1}{9} + \frac{a}{4} & -\frac{2}{9} & -\frac{5}{9} - \frac{a}{4} \end{array} \right].
 \end{aligned}$$

## 2.11 Grevilov metod pregrađivanja

Trenutno, Leverrier–Faddeev algoritam [41] i metod pregrađivanja (partitioning metod [27]) su najčešće uzimani za izračunavanje Moore-Penrose-ovog inverza. Različite modifikacije "partitioning" algoritma na skupovima polinomijalnih i racionalnih matrica su napravljene i testirane u radu [87].

Prvi metod pregrađivanja (eng. partitioning metod) potiče od Grevilea [27]. Ovaj metod predstavlja rekurzivni algoritam za izračunavanje Moore-Penroseovog inverza. U ovom metodu se koristi izraz koji povezuje Moore-Penroseov inverz matrice  $[A|a]$  (matrice  $A$  sa dodatom kolonom  $a$ ) i Moore-Penroseov inverz matrice  $A$ . Polazni slučaj je Moore-Penroseov inverz prve kolone  $a_1$  matrice  $A$ , koji se izračunava prema formuli

$$A_1^\dagger = a_1^\dagger = \begin{cases} \frac{1}{a_1^* a_1} a_1^*, & a_1 \neq 0 \\ a_1^*, & a_1 = 0. \end{cases} \quad (2.11.1)$$

Dalje, za svaku kolonu  $a_i$ ,  $2 \leq i \leq n$  matrice  $A \in \mathbb{C}_r^{m \times n}$  važi sledeće. Ako je

$$A_i = [A_{i-1}|a_i],$$

tada je

$$A_i^\dagger = \begin{bmatrix} A_{i-1}^\dagger - d_i b_i^* \\ b_i^* \end{bmatrix}, \quad 2 \leq i \leq n$$

gde je

$$\begin{aligned} d_i &= A_{i-1}^\dagger a_i, \\ c_i &= a_i - A_{i-1} d_i = (I - A_{i-1} A_{i-1}^\dagger) a_i, \\ b_i &= \begin{cases} \frac{1}{c_i c_i^*} c_i & c_i \neq 0 \\ \frac{1}{1 + d_i^* d_i} (A_{i-1}^\dagger)^* d_i & c_i = 0. \end{cases} \end{aligned} \quad (2.11.2)$$

Konačno  $A^\dagger = A_n^\dagger$ . U [87] je opisan algoritam za tačno izračunavanje Moore-Penroseovog inverza, koji se zasniva na primjeni rezidualne aritmetike i ovakvom metodu pregrađivanja.

**Primer 2.11.1.** Data je matrica

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{bmatrix}.$$

Tada je Moore-Penrose-ov inverz sub-matrice  $A_1$  jednak

$$A_1^\dagger = a_1^\dagger = \frac{1}{a_1^* a_1} a_1^* = \frac{1}{14} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}.$$

Za indeks  $i = 2$  dobijamo sledeće elemente:

$$\begin{aligned} d_2 &= A_1^\dagger a_2 = \frac{1}{14} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \frac{5}{7}, \\ c_2 &= a_2 - A_1 d_2 = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \frac{5}{7} = \frac{4}{7} \cdot \begin{bmatrix} 4 \\ 1 \\ -2 \end{bmatrix}, \\ b_2 &= \frac{1}{c_2^* c_2} c_2 = \frac{1}{12} \begin{bmatrix} 4 \\ 1 \\ -2 \end{bmatrix}. \end{aligned}$$

Tada je Moore-Penrose-ov inverz matrice  $A$  jednak  $A_2^\dagger$ .

$$A^\dagger = A_2^\dagger = \begin{bmatrix} A_1^\dagger - d_2 b_2^* \\ b_2^* \end{bmatrix} = \begin{bmatrix} -\frac{1}{6} & \frac{1}{12} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{12} & -\frac{1}{6} \end{bmatrix}.$$

Greville je predložio rekurzivan algoritam koji povezuje pseudoinverze matrice  $R$  podržane odgovarajućim vektorom  $r$  pseudoinverza  $R^\dagger$  od  $R$ . Neka je  $A$  bilo koja matrica dimenzija  $m \times n$ . Neka je  $A_i$  podmatrica od  $A$  koja sadrži prve  $i$  kolone matrice  $A$ . Ako  $i$ -tu kolonu matrice  $A$  označimo sa  $a_i$ , onda je  $A_i$  podeljena kao  $A_i = [A_{i-1} \mid a_i]$ .

**Algoritam 2.12** Grevilleov rekurzivni algoritam pregrađivanja

1: Početne vrednosti:

$$A_1^\dagger = a_1^\dagger = \begin{cases} \frac{1}{a_1^* a_1} a_1^*, & a_1 \neq 0 \\ a_1^* & a_1 = 0. \end{cases} \quad (2.11.3)$$

2: Rekurzivni korak: Za svaku kolonu  $a_i$ ,  $2 \leq i \leq n$  od  $A$  izračunamo

$$A_i^\dagger = \begin{bmatrix} A_{i-1}^\dagger - d_i b_i^* \\ b_i^* \end{bmatrix}, \quad 2 \leq i \leq n$$

gde je

$$2.1: d_i = A_{i-1}^\dagger a_i,$$

$$2.2: c_i = a_i - A_{i-1} d_i = (I - A_{i-1} A_{i-1}^\dagger) a_i,$$

2.3:

$$b_i = \begin{cases} \frac{1}{c_i c_i^*} c_i & c_i \neq 0 \\ \frac{1}{1+d_i^* d_i} (A_{i-1}^\dagger)^* d_i & c_i = 0. \end{cases} \quad (2.11.4)$$

3: Uslov zaustavljanja:  $A^\dagger = A_n^\dagger$ .

**Implementacija Algoritma 2.12**

Na početku opisaćemo nekoliko pomoćnih procedura.

A. Generisanje  $i$ -te kolone  $a_i$  od  $A$ :

```
Col[a_List, i_] :=
  Return[Transpose[{Transpose[a][[i]]}]]
```

];

B. Podmatrica  $A_j = [a_1, \dots, a_j]$  koja sadrži prvih  $j \leq n$  kolona matrice  $A = A_n = [a_1, \dots, a_n]$  je generisana na sledeći način:

```
Adopuna[a_List, j_] :=
  Block[{},
    {m, n} = Dimensions[a];
    izlaz = Drop[Transpose[a], -(n-j)];
    Return[Transpose[Take[Transpose[a], j]]];
  ];
```

C. Konjugovana i transponovana matrica se formira sledećom funkcijom:

```
Hermit[a_] := Conjugate[Transpose[a]];
```

*Korak 2.* algoritma 2.12 se implementira sledećom funkcijom koja čuva prethodne izračunate vrednosti.

Implementacija *Koraka 2.1.*

```
Block[{s = {}},
  s = A[[a, i-1]].Col[a, i];
  If[Length[s] == 1, Return[s[[1]]], Return[s]];
];
```

Implementacija *Koraka 2.2.*

```
CC[a>List,i_]:=CC[a,i]=
  If[Length[DD[a,i]]==0,
    Col[a,i]-Adopuna[a,i-1]DD[a,i],
    Col[a,i]-Adopuna[a,i-1].DD[a,i] ];
```

Implementacija *Koraka 2.3.*

```
B[a>List,i_]:=B[a,i]=
  Block[{nula,m1,j,k,n1},
    {m1,n1}=Dimensions[CC[a,i]];
    nula=Tabla[0,{j,1,m1},{k,1,n1}];
    If[CC[a,i]==nula,
      If[Length[DD[a,i]]==0,(1/(1+Hermit[({{DD[a,i]}})]).
        ({DD[a,i]}))[[1,1]])Hermit[{A[a,i-1]}]. ({DD[a,i]}),
        (1/(1+Hermit[DD[a,i]].DD[a,i]))[[1,1]]]
        Hermit[A[a,i-1]].DD[a,i],
      (1/(Hermit[CC[a,i]].CC[a,i]))[[1,1]])CC[a,i] ]];
```

Implementacija *Koraka 1.,Koraka 2.,Koraka 3.*

```
A[a>List,i_]:=A[a,i]=
  Block[{b=a},
    If[i==1,
      (1/(Hermit[a][[i]].Col[a,i]))[[1]])Hermit[a][[1]],
    If[Length[DD[a,i]]==0,
      b={A[a,i-1]}-DD[a,i]Hermit[B[a,i]],
      b=A[a,i-1]-DD[a,i].Hermit[B[a,i]]];
    b=Append[b,Hermit[B[a,i]][[1]]] ]];
```

Iterativna implementacija je predstavljena na sledeći način.

```
DD[a>List,i_,a0>List]:= 
  Block[{s={}},
    s=a0.Col[a,i];
    If[Length[s]==1, Return[s[[1]]], Return[s] ];
  ]
CC[a>List,i_,a0>List]:= 
  If[Length[DD[a,i,a0]]==0,
    Col[a,i]-Adopuna[a,i-1]DD[a,i,a0],
    Col[a,i]-Adopuna[a,i-1].DD[a,i,a0]]
B[a>List,i_,a0>List]:= 
  Block[{nula,m1,j,k,n1}
    {m1,n1}=Dimensions[CC[a,i,a0]];
    nula=Table[0,{j,1,m1},{k,1,n1}];
    If[CC[a,i,a0]==nula,
      If[Length[DD[a,i,a0]]==0,
        (1/(1+Hermit[({{DD[a,i,a0]}})]).
          ({DD[a,i,a0]}))[[1,1]])Hermit[{a0}]. ({DD[a,i,a0]}),
        (1/(1+Hermit[DD[a,i,a0]].DD[a,i,a0]))[[1,1]]]
        Hermit[a0].DD[a,i,a0],
      (1/(Hermit[CC[a,i,a0]].CC[a,i,a0]))[[1,1]])CC[a,i,a0]
    ] ]
```

```

A[a_List]:= 
  Block[{b=a,m,n,i},
    {m,n}=Dimensions[a];
    za[i=1,i<=n,i++,
     If[i==1,
      a0=(1/(Hermit[a][[i]].Col[a,i])[1])Hermit[a][[1]],
      If[Length[DD[a,i,a0]]==0,
       b={a0}-DD[a,i,a0]Hermit[B[a,i,a0]],
       b=a0-DD[a,i,a0].Hermit[B[a,i,a0]]
      ];
      b=Append[b,Hermit[B[a,i,a0]][1]];
      a0=b;
     ]];
    Return[a0];
  ]
Partitioning[a_List]:= 
  Block[{}, A[a] ]

```

**Primer 2.11.2.** Za matricu  $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 0 & 1 & -1 & -3 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & -2 \end{bmatrix}$  dobija se

$$A^\dagger = \begin{bmatrix} -\frac{5}{34} & -\frac{3}{17} & \frac{1}{34} & -\frac{1}{34} & \frac{3}{17} & \frac{5}{34} \\ \frac{4}{51} & \frac{13}{102} & -\frac{5}{102} & \frac{5}{102} & -\frac{13}{102} & -\frac{4}{51} \\ \frac{7}{102} & \frac{5}{102} & \frac{1}{51} & -\frac{1}{51} & -\frac{5}{102} & -\frac{7}{102} \\ \frac{1}{17} & -\frac{1}{34} & \frac{3}{34} & -\frac{3}{34} & \frac{1}{34} & -\frac{1}{17} \end{bmatrix}.$$

**Primer 2.11.3.** Za matricu  $A = \begin{bmatrix} -1 & 0 & 1 & 2 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 3 \\ 1 & 1 & -2 & -5 \end{bmatrix}$  dobijamo

$$A^\dagger = \begin{bmatrix} -\frac{11}{51} & -\frac{6}{17} & \frac{7}{51} & \frac{4}{51} \\ \frac{7}{51} & \frac{13}{51} & -\frac{2}{17} & -\frac{1}{51} \\ \frac{4}{51} & \frac{5}{51} & -\frac{1}{17} & -\frac{1}{51} \\ \frac{1}{51} & -\frac{1}{17} & \frac{4}{51} & -\frac{5}{51} \end{bmatrix}.$$

### 3. Višekriterijumska optimizacija

Iako je linearno programiranje veoma primenljivo u praksi, mnoge probleme iz prakse je nemoguće adekvatno linearizovati a da se pritom drastično ne izgubi na tačnosti. U tom slučaju primenjuju se metodi nelinearnog programiranja. Osim nelinearnosti, u mnogim problemima je potrebno naći optimum više od jedne funkcije cilja. U tom slučaju, moramo rešavati **problem višekriterijumske optimizacije**. Ukoliko sve funkcije cilja imaju optimum u istoj tački, problem je trivijalan i direktno se svodi na problem nelinearnog ili linearog programiranja. U praksi je ova situacija veoma retka. Postoji više metoda za rešavanje problema višekriterijumske optimizacije [46]. Zajedničko svim tim metodama je da se polazni problem na odgovarajući način svodi na problem linearog i nelinearnog programiranja.

Višekriterijumska optimizacija se može posmatrati kao nastavak istraživanja u klasičnoj (jednokriterijumskoj) optimizaciji, uz izvesna proširenja. Formalno, osnovno proširenje je uvođenje vektorske kriterijumske funkcije, što dovodi do problema vektorskog maksimuma. Suštinski, potrebno je da se proširi koncept optimalnosti. Razmatrajući problem vektorskog maksimuma koncept optimalnosti se zamjenjuje konceptom neinferiornosti (Pareto optimalnosti). Može se uvesti pojam opšteg (jedinstvenog) kriterijuma optimizacije, koji uključuje kriterijumske funkcije i donosioca odluke. Rešenje zadatka višekriterijumske optimizacije koje se dobija prema takvom kriterijumu je optimalno. U tom slučaju pojam optimalnog rešenja iz klasične optimizacije može se zadržati i u višekriterijumskoj. Međutim, teškoće se upravo javljaju pri pokušaju formalizacije takvog jedinstvenog kriterijuma. Zato se u višekriterijumskoj optimizaciji koriste dve faze ili etape. U prvoj fazi se određuje skup "boljih" rešenja na osnovu vektorske kriterijumske funkcije, a u drugoj se na osnovu preferencije donosioca odluke usvaja konačno rešenje, koje se može nazvati optimalnim. Skup rešenja koji se prezentira donosocu odluke trebalo bi da sadrži mali broj rešenja, koja su neinferiorna prema datim kriterijumskim funkcijama. Problem višekriterijumske optimizacije se najčešće javlja u planiranju složenih sistema; na primer, regionalni razvoj, razvoj vodoprivrednih ili elektroprivrednih sistema, urbano planiranje i očuvanje prirodne okoline [59]. Višekriterijumski problem se javlja u ekonomiji kao problem određivanja tržišne ravnoteže [59]. Sličan problem se javlja i kao problem ravnoteže u teoriji igara [59]. U teoriji igara razmatraju se igre sa više igrača, što se u teoriji odlučivanja javlja kao "grupno odlučivanje" ili odlučivanje sa više donosioca odluke.

U ovoj disertaciji izložićemo problem višekriterijumske optimizacije, kao i metode za njeno rešavanje. Najpre ćemo dati definiciju problema višekriterijumske optimizacije kao i neophodnih pojmove za kasnija razmatranja. Teorijski ćemo obraditi i dati implementaciju nekoliko klasičnih metoda višekriterijumske optimizacije. Svaki od opisanih

metoda biće ilustrovan na jednom ili više primera. Razmatranja vezana za implementaciju metoda su originalna i preuzeta su iz radova [83], [84] kao i iz monografije [92].

### 3.1 Osnovni pojmovi

Opšta formulacija višekriterijumske optimizacije (VKO) poseduje opšti oblik

$$\begin{aligned} \max & \quad Q(\mathbf{x}) = Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \\ \text{p.o.} & \quad f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & \quad h_i(\mathbf{x}) = 0, \quad i = 1, \dots, k. \end{aligned} \quad (3.1.1)$$

gde su:  $Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})$ ,  $f_1(x), \dots, f_m(x)$ ,  $g_1(x), \dots, g_m(x)$  realne funkcije od  $n$  promenljivih koje su sadržane u vektoru  $\mathbf{x} = (x_1, \dots, x_n)$ .

U ovom zadatku traži se rešenje  $\mathbf{x}$  koje maksimizira svih  $l$  funkcija cilja. Zato se zadatak višekriterijumske optimizacije (VKO) naziva i zadatak vektorske optimizacije. Radi jednostavnosti ovde se razmatraju samo problemi maksimizacije. Poznato je da se zadatak minimizacije jednostavno prevodi u zadatak maksimizacije množenjem kriterijumske funkcije sa  $-1$ . Sve nadalje izložene definicije i metode moguće je prilagoditi da važe za rešavanje zadatka minimizacije.

Kažemo da je  $\mathbf{X} \subseteq \mathbb{R}^n$  skup dopustivih rešenja ako važi

$$\mathbf{X} = \{\mathbf{x} | f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m; \quad h_i(\mathbf{x}) = 0, \quad i = 1, \dots, k\}.$$

Svakom dopustivom rešenju  $\mathbf{x} \in \mathbf{X}$  odgovara skup vrednosti kriterijumskih funkcija, tj. vektor  $Q(\mathbf{x}) = (Q_1(\mathbf{x}), Q_2(\mathbf{x}), \dots, Q_l(\mathbf{x}))$ . Na taj način se skup dopustivih rešenja preslikava u *kriterijumski skup*, tj.  $S = \{Q(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$ .

U daljem tekstu biće korišćeni sledeći pojmovi:

- *Marginalna rešenja* zadatka VKO se određuju optimizacijom svake od funkcija cilja pojedinačno nad zadatim dopustivim skupom, tj. rešavanjem  $l$  jednokriterijumskih zadataka

$$\begin{aligned} \max & \quad Q_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \\ \text{p.o.} & \quad f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & \quad h_i(\mathbf{x}) = 0, \quad i = 1, \dots, k. \end{aligned}$$

Marginalna rešenja ćemo obeležavati sa  $\mathbf{x}^{(j)*} = (x_1^{(j)*}, x_2^{(j)*}, \dots, x_n^{(j)*})$ , gde je  $x^{(j)*}$  optimalno rešenje dobijeno optimizacijom  $j$ -te funkcije cilja nad zadatom dopustivim skupom  $\mathbf{X}$ .

- *Idealne vrednosti funkcija cilja*, označene sa  $Q_j^*$  jesu vrednosti funkcija cilja za marginalna rešenja

$$Q_j^* = Q_j(\mathbf{x}^{(j)*}), \quad j = 1, \dots, l.$$

- Idealne vrednosti funkcija cilja određuju *idealnu tačku* u kriterijumskom prostoru, tj. idealnu vrednost vektorske funkcije

$$Q^* = (Q_1^*, Q_2^*, \dots, Q_l^*).$$

- Ako postoji rešenje  $\mathbf{x}^*$  koje istovremeno maksimizira sve funkcije cilja, tj.

$$\mathbf{x}^* = \{\mathbf{x} | Q_j(\mathbf{x}) = Q_j^*, j = 1, \dots, l\},$$

onda se takvo rešenje naziva *savršeno rešenje*.

U najvećem broju slučajeva marginalna rešenja se razlikuju i savršeno rešenje ne postoji. Kada savršeno rešenje postoji, tada se u suštini ne radi o problemu VKO.

Veoma je važno imati u vidu da su u realnim problemima ciljevi gotovo uvek u koliziji, što znači da ne mogu svi biti dostignuti u potpunosti. Zbog toga najčešće nije moguće strogo definisati optimum niti za svaka dva rešenja formalno odrediti koje je bolje od drugog. Iz tog razloga proces dobijanja rešenja zahteva učešće donosioca odluke (u daljem tekstu DO). To je najčešće neko ko ima dublji uvid u problem i po čijem se zahtevu pristupa rešavanju. Donosilaca odluke može biti i više i tada se problem može dodatno iskomplikovati zbog njihovih različitih ciljeva, uvela u odlučivanju i stepena odgovornosti koji su spremni da preuzmu.

Činjenica da zadaci VKO po pravilu nemaju savršeno rešenje upućuje na preispitivanje koncepta optimalnosti i definicije optimalnog rešenja. Ključnu ulogu u tome ima koncept *Pareto optimalnosti*. To je proširenje poznatog koncepta optimalnosti koj se koristi u klasičnoj jednokriterijumskoj optimizaciji.

*Pareto optimum* se definiše na sledeći način.

**Definicija 3.1.1.** *Dopustivo rešenje  $\mathbf{x}^*$  predstavlja Pareto optimum zadatka VKO ako ne postoji neko drugo dopustivo rešenje  $\mathbf{x}$  takvo da važi*

$$Q_j(\mathbf{x}) \geq Q_j(\mathbf{x}^*) \quad \forall j = 1, \dots, l$$

*pri čemu bar jedna od nejednakosti prelazi u strogu nejednakost >.*

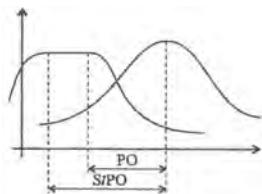
Drugim rečima,  $\mathbf{x}$  je Pareto optimum ako bi poboljšanje vrednosti bilo koje funkcije cilja prouzrokovalo pogoršanje vrednosti neke druge funkcije cilja. Za Pareto optimum postoje sledeći sinonimi: *efikasno, dominantno i nedominirano rešenje*.

Pored Pareto optimuma definišu se slabi i strogi (jaki) Pareto optimumi.

**Definicija 3.1.2.** *Dopustivo rešenje  $\mathbf{x}^*$  je slabi Pareto optimum ako ne postoji neko drugo dopustivo rešenje  $\mathbf{x}$  takvo da važi*

$$Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*) \quad \forall j = 1, \dots, l.$$

Drugim rečima  $\mathbf{x}^*$  je slabi Pareto optimum ako nije moguće istovremeno poboljšati sve funkcije cilja.



Slika 3.1.1. Geometrijska interpretacija Pareto optimalnih i Slabo Pareto optimalnih rešenja

**Definicija 3.1.3.** *Pareto optimalno rešenje  $\mathbf{x}^*$  je strogi Pareto optimum ako postoji broj  $\beta > 0$  takav da za svaki indeks  $j \in \{1, \dots, l\}$  i za svako  $\mathbf{x}$  koje zadovoljava uslov*

$$Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$$

*postoji bar jedno  $i \in \{1, \dots, l\} \setminus \{j\}$  takvo da je*

$$Q_i(\mathbf{x}) > Q_i(\mathbf{x}^*)$$

*i da važi*

$$\frac{Q_j(\mathbf{x}) - Q_j(\mathbf{x}^*)}{Q_i(\mathbf{x}^*) - Q_i(\mathbf{x})} \geq \beta.$$

Strogi Pareto optimum izdvaja ona Pareto rešenja čije promene ne prouzrokuju prevelike relativne promene u funkcijama cilja.

Odnos između opisanih optimuma je takav da svaki skup strožijih Pareto optimuma predstavlja podskup slabijih optimuma, tj. svaki Pareto optimum je istovremeno i slabi Pareto optimum, a svaki strogi Pareto optimum je i Pareto optimum. Odnos svih skupova dat je na slici.



Slika 3.1.2. Odnos između skupova Pareto optimalnih rešenja

## 3.2 Simboličke transformacije u višekriterijumskoj optimizaciji

Jedan od ciljeva našeg istraživanja bila je implementacija glavnih metoda višekriterijumske optimizacije u kompjuterskom algebarskom sistemu MATHEMATICA. Programski paket MATHEMATICA je jedan od raznovrsnih programskih jezika dostupnih danas, primenljiv i kod simboličkih i kod numeričkih izračunavanja [48, 113]. Nekoliko funkcija za ograničenu numeričku optimizaciju je dostupno u programskom paketu MATHEMATICA (videti [48], [113]).

Funkcije *Maximize* i *Minimize* dozvoljavaju specifikaciju funkcije cilja za maksimizovanje i minimizovanje, zajedno sa skupom ograničenja. U svim slučajevima pretpostavljeno je da su sve promenljive ograničene da nemaju negativne vrednosti.

*Minimize*[ $f$ , { $\text{cons}$ }, { $x$ ,  $y$ , ...}] ili *Minimize*[{ $f$ ,  $\text{cons}$ }, { $x$ ,  $y$ , ...}], minimizovati  $f$  u oblasti specifikovane ograničenjima  $\text{cons}$ ;

*Maximize*[ $f$ , { $\text{cons}$ }, { $x$ ,  $y$ , ...}] or *Maximize*[{ $f$ ,  $\text{cons}$ }, { $x$ ,  $y$ , ...}], naći maksimum od  $f$ , u oblasti specifikovanoj sa  $\text{cons}$ .

*Minimize* i *Maximize* se može u pravilnom rešavanju bilo kog problema polinomnog programiranja functionu kojem funkcije cilja  $f$  i ograničenja  $\text{cons}$  obuhvataju proizvoljne polinomske funkcije promenljivih [113]. Vazna karakteristika *Minimize* i *Maximize* je da oni uvek pronalaze globalne minimume i maksimume[113].

Funkcije *NMinimize* i *NMaximize* implementiraju nekoliko algoritama za pronalazenje ograničenja globalnog optimuma. Izrazi

*NMaximize*[{ $f$ ,  $\text{cons}$ },  $\text{vars}$ , Method -> {method, mopts}],

*NMinimize*[{ $f$ ,  $\text{cons}$ },  $\text{vars}$ , Method -> {method, mopts}],

pronalaze globalni maksimum i minimum , respektivno, za funkcije cilja  $f$  po ograničenjima  $\text{cons}$ , korišćenjem metoda optimizacije *method* sa opcijama metoda definisanim u *mopts*.

Glavni detalji višekriterijumske optimizacije koji su specifični za simbolička izračunavanja opisani su u radu [83]. Implementacija je izvedena u programskom paketu MATHEMATICA. Diskutovani su metod težinskih koeficijenata, glavni metodi prioriteta i metodi ciljnog programiranja. Posebno je obrađena simbolička konverzija datih funkcija cilja i ograničenja u odgovarajući problem jende funkcije cilja. Transformacije iz višekriterijumskog u jednokriterijumski problem u proceduralnim programskim jezicima su zapravo kombinacije realnih vrednosti, i podrazumevaju procedure koje zavise od funkcija cilja. U našoj implementaciji ove transformacije su izvršene u simboličkoj formi, uzimanjem kombinacija ciljnih funkcija, koje uključuju nedefinisane sbole i neoznačene promenljive.

Predlažemo sledeće jasne prednosti koje proističu iz implementacije problema višekriterijumske optimizacije u simboličkom programskom jeziku MATHEMATICA, poštujući traditionalnu implementaciju u proceduralnim programskim jezicima.

1. Mogućnost da se koriste proizvoljne ciljne funkcije i ograničenja (koji nisu definisani podprogramima) u toku izvršenja implementacione funkcije. Glavni aspekti ovih prednosti su:

(i) Problem višekriterijske optimizacije (3.1.1) je predstavljen pogodnom *unutrašnjom formom*, čiji elementi mogu biti korišćeni kao formalni parametri u optimizacijskom softveru. Unutrašnja forma problema (3.1.1) je uređena trojka

$$\begin{aligned} &\{Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})\}, \\ &\{f_1(\mathbf{x}) \leq 0, \dots, f_m(\mathbf{x}) \leq 0, h_1(\mathbf{x}) = 0, \dots, h_k(\mathbf{x}) = 0\}, \\ &\{x_1, \dots, x_n\} \end{aligned} \tag{3.2.1}$$

gde  $\mathbf{x}$  su argumenti za ciljne funkcije i ograničenja. Prvi element unutrašnje forme, označen sa  $q$ , je lista  $\{Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})\}$  čiji su elementi neodređeni izrazi koji predstavljaju funkcije cilja. Drugi element u (3.2.1) je lista ograničenja  $f_i(\mathbf{x}) \leq 0$ ,  $i = 1, \dots, m$ ,

$h_i(\mathbf{x}) = 0$ ,  $i = 1, \dots, k$ . Ovaj argument ćemo označavati kao *constr*. Treći element, u oznaci *var*, predstavlja generičku listu promenljivih  $\{x_1, \dots, x_n\}$ , određenu na osnovu  $\mathbf{x}$ . U tom smislu, dozvoljeno je da neki argumenti u  $\mathbf{x}$  mogu biti definisani u globalnom okruženju MATHEMATICA kernela.

(ii) Ako je  $f$  funkcija cilja jednokriterijumskog problema optimizacije dobijenog iz (3.2.1), možemo izračunati njen maksimum koristeći standardnu funkciju *Maximize*:  
**Maximize[f, constr, var]**.

Mogućnost softvera da procesира proizvoljnu funkciju cilja pri proizvoljnim ograničenjima omogućuje primenu naših optimizacionih modela.

2. Mogućnost korišćenja nizova funkcija, čiji elementi se mogu selektovati i kasnije primeniti na date argumente. Ove strukture nisu svojstvene proceduralnim programskim jezicima.

Svaki izraz sadržan u listi  $q = \{Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x})\}$  je odmah moguće primeniti na date argumente. Moguće je izračunati vrednost funkcije  $q[[i]]$  u tački  $x_0$  koristeći transformaciona pravila  $q[[i]]/.x \rightarrow x_0$  ili  $q[[i]]/.Thread[Rule[x, x_0]]$ . MATHEMATICA pretražuje delove funkcije  $q[[i]]$  koje je moguće zameniti koristeći navedeno pravilo, a onda izvodi zamenu.

Takođe, moguće je definisati funkciju koja uzima drugu funkciju kao argument, na primer:  $pf[f_, x_] := f[x] + f[1-x]$ . Kasnije je moguće koristiti izraze kao  $pf[q[[i]], x_0]$ . U izrazu  $f[x]$ , ime funkcije  $f$  je takođe izraz, i može se tretirati kao bilo kakav numerički izraz [113]. Specijalno, može se uzeti bilo koji element iz  $q$  kao argument u funkcijama *Minimize* i *Maximize*.

3. Efektivna i prirodna simbolička transformacija od višekriterijumskog modela u odgovarajući jedno-kriterijumska model.

Mnogo optimizacionih metoda je bazirano na podalgoritmima koji zahtevaju konstrukciju funkcija cilja opšte forme

$$G(q(\mathbf{x}), \Phi(f(\mathbf{x}), h(\mathbf{x}), \lambda)),$$

gde je  $\lambda$  mogući skup veštačkih promenljivih a  $\Phi$  je proizvoljna funkcija.

Transformacija višekriterijumskog u jednokriterijumske problem u proceduralnim jezicima je u osnovi kombinacija realnih vrednosti. Ove transformacije ćemo izvoditi u simboličkoj formi, koristeći kombinacije neizračunatih izraza i simbola.

### 3.3 Ispitivanje Pareto optimalnosti

Kao po pravilu, nemoguće je pronaći potpun beskonačan skup Pareto optimalnih rešenja za specijalne probleme iz realnog života. Iz ovog razloga inženjerski višekriterijumski problem naredbe nastoji da odredi konačan podskup kriterijumske različitih Pareto

optimalnih rešenja. Takođe, ovde postoje brojni metodi za dokazivanje Pareto optimalnosti. Ovi metodi takođe mogu biti korišćeni da se pronađe prvobitno Pareto optimalno rešenje [4], [111].

U radu [83] uveden je algoritam za određivanje Pareto optimalnosti rešenja višekriterijumskog problema, koristeći direktan dokaz u saglasnosti sa definicijom Pareto optimalne tačke.

---

**Algoritam 3.1** Ispitivanje Pareto optimalnosti fiksirane tačke  $\mathbf{x}^*$ .

---

**Ulaz:** Optimizacioni problem (3.1.1). Proizvoljna fiksirana tačka  $\mathbf{x}^*$ .

1: Odrediti skup  $X = \text{Reduce}[\text{constr}/.\{\text{List} \rightarrow \text{And}, \text{var}\}]$  i postaviti  $\text{Optimal}=\text{true}$ .

2: Za svaki indeks  $j = 1, \dots, l$  ponoviti Korake 2.1 i 2.2:

2.1: Generisati sledeću konjunkciju ograničenja

$$\text{Par} = X \ \&\& \ u_1(\mathbf{x}) \ \&\& \dots \ \&\& \ u_l(\mathbf{x}) \quad (3.3.1)$$

pri čemu je

$$u_i(\mathbf{x}) = \begin{cases} Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*), & j \neq i, \\ Q_i(\mathbf{x}) > Q_i(\mathbf{x}^*), & j = i. \end{cases} \quad (3.3.2)$$

2.2: Ako je  $\text{Par} = \emptyset$ , postaviti  $\text{Optimal} := \text{false}$  i ići na Korak 3.

3: **Rezultat:** Vratiti vrednost promenljive  $\text{Optimal}$  kao rezultat.

---

Odgovarajuća funkcija u našoj implementaciji je

`IsPareto[q_List, constr_List, var_List, sol_List]`,

pri čemu su formalni parametri uzeti u sledećem smislu:

`q, constr, var`: unutrašnja reprezentacija problema višekriterijumske optimizacije (3.1.1);

`sol`: rešenje odgovarajućeg problema jedno-ciljne optimizacije.

Pozivanje ove funkcije je oblika

`IsPareto[q, constr, var, First[Rest[Maximize[fun, constr, var]]]] ];`

gde je `fun, constr, var` reprezentacija odgovarajućeg problema jedno-ciljne optimizacije.

U ovoj funkciji koristimo sledeću verziju standardnih MATHEMATICA funkcija `Reduce` i `FindInstance` [113]:

`Reduce[expr, var]` pronalazi sve realne vrednosti promenljivih sadržanih u listi `var` koja zadovoljava skup brojeva koji sadrži logičke veze i invariјantne polinomske jednakosti i nejednakosti.

`FindInstance[expr, var]` određuje vrednosti promenljivih iz `var` u kojima je tvrdjenje `expr` tačno. Ako nije nađena ni jedna vrednost za `var`, rezultat je prazna lista. Parametar `expr` može sadržati jednakosti, nejednakosti, oblast specifikacije i kvantifikatore [113].

```
<<Algebra`InequalitySolve`
IsPareto[q_List, constr_List, var_List, res_List] :=
Module[{X = {}, l = Length[q]},
X = Reduce[constr/.{List->And}, var]; Ok = 1; (* Korak 1. *)
For[j = 1, j <= l, j++, (* Korak 2. *)
```

```

Par=X;
For[i = 1, i <= l, i++, (* Korak 2.1. *)
    If [j != i, Par = Par && q[[i]] >= (q[[i]]/. res),
        Par = Par && q[[i]] > (q[[i]]/. res)
    ];
]; (* Par is of the form (4.1), (4.2) *)
If[FindInstance[Par, var] != {}, Ok = 0; Break[] ]; (* Korak 2.2. *)
(*Ako je nadjena instanca za var, prekinuti ciklus*)
];
If[Ok == 1,
    Print["Resenje ", {q/.res, res}, " je Pareto optimalno"],
    Print["Resenje ", {q/.res, res}, " nije Pareto optimalno"];
];
Return[Ok]; (* Korak 3. *)
]

```

## 3.4 Metod težinskih koeficijenata

Metod težinskih koeficijenata je najstariji metod za VKO-u koji je korišćen. Po ovom metodu uvode se težinski koeficijenti  $w_i$  za sve kriterijumske funkcije  $Q_i(\mathbf{x})$ ,  $i = 1, \dots, l$ , pa se problem optimizacije svodi na sledeću skalarnu optimizaciju

$$\begin{aligned} \max \quad & Q(\mathbf{x}) = \sum_{i=1}^l w_i Q_i(\mathbf{x}) \\ \text{p.o.} \quad & \mathbf{x} \in \mathbf{X}, \end{aligned} \tag{3.4.1}$$

gde težine  $w_i$ ,  $i = 1, \dots, l$  ispunjavaju sledeće uslove

$$\sum_{i=1}^l w_i = 1, \quad w_i \geq 0, \quad i = 1, \dots, l.$$

Često se koristi metod težinskih koeficijenata tako što se zadaju vrednosti ovih koeficijenata. Međutim, to uvek izaziva određene teškoće i primedbe na ovakav postupak, jer se unosi subjektivan uticaj na konačno rešenje preko zadatih vrednosti težinskih koeficijenata.

Glavna ideja u metodu težinskih koeficijenata je da se odaberu težinski koeficijenti  $w_i$  koji odgovaraju ciljnim funkcijama  $Q_i(\mathbf{x})$ ,  $i = 1, \dots, l$ . Mnogi autori su razvili sistematske pristupe u selektovanju težina, čiji se pregled može naći u [31], [36] i [104]. Jedna od poteškoća ovog metoda je da variranje težina konzistentno i neprekidno ne mora uvek da rezultuje u tačnoj i kompletnoj reprezentaciji Pareto optimalnog skupa. Ovaj nedostatak je diskutovan u [22].

**Teorema 3.4.1.** *Ako su svi težinski koeficijenti  $w_i$  pozitivni, onda je rešenje problema (3.4.1) Pareto optimalno rešenje polaznog problema VKO.*

**Dokaz.** Neka je  $\mathbf{x}^*$  rešenje problema (3.4.1), i neka su svi težinski koeficijenti strogo pozitivni. Prepostavimo da ono nije Pareto optimalno, tj. da postoji  $\mathbf{x} \in S$  tako da za  $i = 1, \dots, l$  važi  $Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*)$ , pri čemu važi bar jedna stroga nejednakost (recimo za indeks  $j$ ). Kako je  $w_i > 0$  za svako  $i$ , važi

$$\sum_{i=1}^l w_i Q_i(\mathbf{x}) > \sum_{i=1}^l w_i Q_i(\mathbf{x}^*)$$

pa dobijamo kontradikciju sa pretpostavkom da je  $\mathbf{x}^*$  rešenje težinskog problema (3.4.1). Sledi da je  $\mathbf{x}^*$  Pareto optimalno.  $\square$

**Teorema 3.4.2.** *Ako je za svako  $i \in \{1, \dots, l\}$  ispunjen uslov  $w_i \geq 0$ , onda je rešenje problema (3.4.1) slabi Pareto optimum polaznog problema VKO.*

**Dokaz.** Neka je  $\mathbf{x}^*$  rešenje problema (3.4.1) i da je ispunjen uslov  $w_i \geq 0$ . Prepostavimo da ono nije slabo Pareto optimalno, tj. da postoji  $\mathbf{x} \in S$  tako da za  $i = 1, \dots, l$  važi  $Q_i(\mathbf{x}) > Q_i(\mathbf{x}^*)$ . Svi koeficijenti  $w_i$  su nenegativni i bar jedan je strogo veći od nule (zbog  $\sum_{i=1}^l w_i = 1$ ), pa važi

$$\sum_{i=1}^l w_i Q_i(\mathbf{x}) > \sum_{i=1}^l w_i Q_i(\mathbf{x}^*)$$

pa dobijamo kontradikciju sa pretpostavkom da je  $\mathbf{x}^*$  rešenje težinskog problema. Dakle,  $\mathbf{x}^*$  je slabi Pareto optimum.  $\square$

**Teorema 3.4.3.** *Ako je rešenje problema (3.4.1) jedinstveno, ono je onda i Pareto optimalno.*

**Dokaz.** Neka je  $\mathbf{x}^*$  jedinstveno rešenje problema (3.4.1). Prepostavimo da ono nije Pareto optimalno rešenje problema VKO, tj. da postoji  $\mathbf{x} \in S$  tako da za  $i = 1, \dots, l$  važi  $Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*)$ , pri čemu važi bar jedna stroga nejednakost (recimo za indeks  $j$ ). Primetimo da to znači  $\mathbf{x} \neq \mathbf{x}^*$ .

Kako je  $w_i \geq 0$  za svako  $i$ , važi

$$\sum_{i=1}^l w_i Q_i(\mathbf{x}) \geq \sum_{i=1}^l w_i Q_i(\mathbf{x}^*)$$

Ako bi važila stroga nejednakost, onda  $\mathbf{x}^*$  ne bi bilo rešenje problema (3.4.1). Dakle, važi jednakost. To znači da postoje dva različita rešenja  $\mathbf{x}$  i  $\mathbf{x}^*$  problema (3.4.1), što je kontradikcija.  $\square$

Pokazaćemo sada jedno jače tvrđenje.

**Teorema 3.4.4.** Ako su svi  $w_i > 0$ ,  $i \in \{1, \dots, l\}$ , tada je rešenje problema (3.4.1) strogi Pareto optimum problema VKO.

**Dokaz.** Neka je  $\mathbf{x}^*$  rešenje težinskog problema. Pokazali smo da je ono Pareto optimalno. Dokažimo da je to rešenje takođe i strogi Pareto optimum sa konstantom

$$M = (k - 1) \max_{i,j} \frac{w_j}{w_i}.$$

Pretpostavimo suprotno, da postoji  $\mathbf{x} \in S$  i indeks  $i$  takvi da je  $Q_i(\mathbf{x}) > Q_i(\mathbf{x}^*)$  pri čemu za svako  $j$  za koje je  $Q_j(\mathbf{x}^*) > Q_j(\mathbf{x})$  važi  $Q_i(\mathbf{x}^*) - Q_i(\mathbf{x}) < M(Q_j(\mathbf{x}) - Q_j(\mathbf{x}^*))$ . Zamenom

$$M = \frac{(k - 1)w_j}{w_i}$$

dobijamo

$$w_i \frac{Q_i(\mathbf{x}^*) - Q_i(\mathbf{x})}{k - 1} < w_j (Q_j(\mathbf{x}) - Q_j(\mathbf{x}^*)) > 0 .$$

Dakle, za svako  $j \neq i$  za koje je  $Q_j(\mathbf{x}^*) > Q_j(\mathbf{x})$  važi prethodna nejednakost. Za one indekse  $j \neq i$  za koje je  $Q_j(\mathbf{x}^*) \leq Q_j(\mathbf{x})$  gornja nejednakost svakako važi. Dakle, za svako  $j \neq i$  važi

$$w_i \frac{Q_i(\mathbf{x}^*) - Q_i(\mathbf{x})}{k - 1} < w_j (Q_j(\mathbf{x}) - Q_j(\mathbf{x}^*)),$$

pa sabiranjem ovih nejednakosti za  $j = 1, \dots, i - 1, i + 1, \dots, l$  dobijamo

$$w_i (Q_i(\mathbf{x}^*) - Q_i(\mathbf{x})) < \sum_{j=1, j \neq i}^l w_j (Q_j(\mathbf{x}) - Q_j(\mathbf{x}^*))$$

odnosno

$$\sum_{j=1}^l w_j Q_j(\mathbf{x}^*) < \sum_{j=1}^l w_j Q_j(\mathbf{x}) .$$

Dobijamo da  $\mathbf{x}^*$  nije rešenje težinskog problema tj. dolazimo do kontradikcije. Zaključujemo da je  $\mathbf{x}^*$  zaista strogi Pareto optimum polaznog problema.  $\square$

Težinski koeficijent na neki način treba da predstavi značaj kriterijumske funkcije kojoj je dodeljen. Da bismo to postigli najpre moramo da normalizujemo kriterijumske funkcije tj. da ih promenimo tako da imaju približno jednake vrednosti, a pri tome da zadrže sve bitne osobine. Na primer, ako je kriterijumska funkcija linearna  $Q_j(\mathbf{x}) = \sum_{i=1}^n a_i x_i$ , tada će normalizovan oblik ove funkcije biti

$$\frac{Q_j(\mathbf{x})}{\sum_{i=1}^n a_i} .$$

Ako je kriterijumska funkcija ograničena tada se za njen normalizovan oblik može uzeti sledeća funkcija čiji je kodomen  $[0, 1]$ :

$$\overline{Q}_i(\mathbf{x}) = \frac{Q_i(\mathbf{x}) - Q_i^*}{\max_{\mathbf{x} \in S} Q_i(\mathbf{x}) - Q_i^*} ,$$

Ukoliko donosilac odluke sam definiše težinske koeficijente, onda ovaj metod spada u grupu a priori metoda. Međutim, najčešće se malo zna o tome kako koeficijente treba izabrati. Zato se obično težinski problem rešava za razne vrednosti vektora  $(w_1, \dots, w_l)$  i na taj način dobijaju različita rešenja među kojima DO bira ono koje mu najviše odgovara. Ako se koristi ovakav pristup, onda ovaj metod postaje a posteriori metod.

Glavna mana ove metode je teškoća određivanja težinskih koeficijenata kada nema dovoljno podataka o problemu. Zato predlažemo algoritam za automatsko generisanje težinskih koeficijenata radi dobijanja Pareto optimalnih tačaka. Za pozitivne težine i konveksni problem, optimalna rešenja jednokriterijumskog problem jesu Pareto optimalna [114], tj. minimiziranje odgovarajućeg jednokriterijumskog problema je dovoljno za Pareto optimalnost. Međutim, formulacija ne obezbeđuje neophodan uslov za Pareto optimalnost [116]. Kada je višekriterijumska problem konveksan, primena funkcije **Compositions**[k,1] produkuje  $b$  Pareto optimalnih rešenja, gde integer  $b$  ispunjava  $1 \leq b \leq \binom{k+l-1}{l-1}$ .

---

**Algoritam 3.2** Izračunavanje Pareto optimuma generisanjem težinskih koeficijenata

---

**Ulaz:** Problem višekriterijumske optimizacije (3.1.1), zadat sa  $l$  funkcija cilja  $Q_i$ ,  $i = \overline{1, l}$ , sa skupom ograničenja  $X$  (reprezentovan skupom nejednačina).

- 1: Izabrati proizvoljno fiksiran prirodan broj  $k$ .
- 2: Generisati listu od  $p$  elemenata koja sadrži kompozicije broja 1 u  $l$  delova, na sledeći način:

$$W = \left\{ \left( \frac{\omega_1}{k}, \dots, \frac{\omega_l}{k} \right) \mid \omega_1, \dots, \omega_l \in \mathbb{Z} \wedge \sum_{i=1}^l \omega_i = k \right\}. \quad (3.4.2)$$

- 3: Za svako  $i = \overline{1, p}$  izvršiti Korake 2.1, 2.2 i 2.3:
  - 2.1: Upotrebiti listu  $W_i = \{W_{i,1}, \dots, W_{i,l}\}$  kao skup težinskih koeficijenata.
  - 2.2: Izračunati funkciju cilja:  $f_i(\mathbf{x}) := \sum_{j=1}^l W_{i,j} \cdot Q_j(\mathbf{x})$ .
  - 2.3: Rešiti sledeći problem jedno-kriterijumske optimizacije:

$$\begin{array}{ll} \text{Min.} & f_i(\mathbf{x}) \\ \text{p.o.} & \mathbf{x} \in X \end{array} \quad (3.4.3)$$


---

Sledi opis implementacije metoda težinskih koeficijenata. Pozivom standardne funkcije **Compositions**[n,1] možemo odrediti ” $l$ -dimenzionalne tačke”, čiji zbir koordinata daje  $n$ . Kada tu listu podelimo sa brojem  $n$ , možemo dobiti tačke u intervalu  $[0, 1]$  čije koordinate mogu predstavljati koeficijente  $w_i$ . Na taj način smo obezbedili automatsko generisanje koeficijenata  $w_i$  potrebnih za realizaciju metoda. Ostavljena je i mogućnost da korisnik sam izabere koeficijente  $w_i$ . To se postiže tako što pri pozivu funkcije **MultiW**, kojom se implementira metod težinskih koeficijenata, poslednji parametar zadamo kao nepraznu listu, tj. zadamo koeficijente  $w_i$  u obliku liste.

Sledi program kojim je implementiran metod težinskih koeficijenata [83].

**Ulazne veličine:**

*q\_,var\_List* - ciljna funkcija i lista njenih parametara;

*constr\_List* - lista ograničenja;

*var\_* - korak podele intervala  $[0,1]$ ;

*w1\_List* - lista težinskih koeficijenata u intervalu [0,1]. Prazna lista kao vrednost parametra *w1* označava da će težinski koeficijentu biti generisani pomoću funkcije *Compositions*. Inače, pretpostavlja se da je svaki element *w1[[i]]*,  $1 \leq i \leq \text{Length}[w1]$  liste *w1* jedan mogući skup koeficijenata  $w_j$ ,  $j = 1, \dots, l$ :  $w[[j]] = w1[[i, j]], j = 1, \dots, l$ .

### Lokalne promenljive:

*fun* - formirana funkcija za jednodimenzionalnu optimizaciju.

```
MultiW[q_, constr_List, var_List, w1_List] :=
Module[{i=0,k,l=Length[q],res={},W={},fun,sk={},qres={},mxs={},m,ls={}},
If[w1=={},k=Input["Initial sum of weighting coefficients?"]; W=Compositions[k,1]/k,
W=w1];
Print["Weighting Coefficients: "]; Print[W];
Print["Single-objective problems: "];
k=Length[W];
For[i=1, i<=k, i++,
fun=Simplify[Sum[W[[i,j]]*q[[j]], {j,1}]]; (* 3 *)
AppendTo[res, Maximize[fun, constr, var]]; (* 1 *)
];
Print["Solutions of single-objective problems: "]; Print[res];
For[i=1, i<=k, i++,
AppendTo[qres, q/.res[[i, 2]]]; AppendTo[mxs, res[[i, 1]]];
];
m=Max[mxs];
For[i=1, i<=Length[mxs], i++,
If[m==mxs[[i]], AppendTo[ls, {qres[[i]], res[[i,2]]}]];
];
Return[ls];
]
```

**Primer 3.4.1.** Maksimizovati funkcije koje se nalaze u listi:

$$Q = \{Q_1(x, y) = x + y, Q_2(x, y) = 2x - y\}$$

prema ograničenjima

$$-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0.$$

Prvo se koristi prazna lista za parametar *w1*, i na taj način se koeficijenti  $w_i$  generišu pomoću funkcije *Compositions*:

```
In[1]:=MultiW[\{x+y,2x-y\},\{-3x+5y<=9,3x+2y<=12,5x-4y<=9,x>=0,y>=0\},\{x,y\},\{\}]
```

U slučaju  $k = 5$ , izraz *w=Compositions[k,1]/k* produkuje

$$w=\{\{0,1\},\{\frac{1}{5},\frac{4}{5}\},\{\frac{2}{5},\frac{3}{5}\},\{\frac{3}{5},\frac{2}{5}\},\{\frac{4}{5},\frac{1}{5}\},\{1,0\}\}$$

Takođe, jednokriterijumski problemi optimizacije dati su sledećim unutrašnjim reprezentacijama.

Za  $\{w_1, w_2\} = \{0, 1\}$ :

$$\{2x-y, \{-3x+5y \leq 9, 3x+2y \leq 12, 5x-4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{\frac{1}{5}, \frac{4}{5}\}$ :

$$\{\frac{3}{5}(3x-y), \{-3x+5y \leq 9, 3x+2y \leq 12, 5x-4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{\frac{2}{5}, \frac{3}{5}\}$ :

$$\{\frac{1}{5}(8x - y), \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{\frac{3}{5}, \frac{2}{5}\}$ :

$$\{\frac{1}{5}(7x + y), \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{\frac{4}{5}, \frac{1}{5}\}$ :

$$\{\frac{3}{5}(2x + y), \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{1, 0\}$ :

$$\{x + y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Elementi liste  $res$ , koji odgovaraju rešenjima jednokriterijumske problema, jednaki su

$$\{\{\frac{9}{2}, \{x \rightarrow 3, y \rightarrow \frac{3}{2}\}\}, \{\{\frac{9}{2}, \{x \rightarrow 3, y \rightarrow \frac{3}{2}\}\}, \{\{\frac{9}{2}, \{x \rightarrow 3, y \rightarrow \frac{3}{2}\}\}, \{\{\frac{9}{2}, \{x \rightarrow 3, y \rightarrow \frac{3}{2}\}\}, \{\{5, \{x \rightarrow 2, y \rightarrow 3\}\}\}$$

a rezultat je lista koja sadrži vrednosti ciljnih funkcija u tački  $\{x \rightarrow 2, y \rightarrow 3\}$ , koja odgovara maksimalnoj vrednosti težinske funkcije (koja je jednaka 5):

**Out[1]=**  $\{\{5, 1\}, \{x \rightarrow 2, y \rightarrow 3\}\}$

Sada se vrednosti koeficijenata  $w_i$  definišu od strane korisnika. Vrednosti težinskih koeficijenata sadržane su u svakom elementu liste

$\{\{1, 0\}, \{0.9, 0.1\}, \{0.875, 0.125\}, \{0.8, 0.2\}, \{0, 1\}\}$ .

**In[2]:=MultiW[**  $\{x+y, 2x-y\}, \{-3x+5y \leq 12, 5x-4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\},$   
 $\{\{1, 0\}, \{0.9, 0.1\}, \{0.875, 0.125\}, \{0.8, 0.2\}, \{0, 1\}\}$  **]**

Sledeći problemi se rešavaju u ciklusu:

Za  $\{w_1, w_2\} = \{1, 0\}$ :

$$\{x + y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{0.9, 0.1\}$ :

$$\{1.1x + 0.8y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{0.875, 0.125\}$ :

$$\{1.125x + 0.75y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{0.8, 0.2\}$ :

$$\{1.2x + 0.6y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Za  $\{w_1, w_2\} = \{0, 1\}$ :

$$\{2x - y, \{-3x + 5y \leq 9, 3x + 2y \leq 12, 5x - 4y \leq 9, x \geq 0, y \geq 0\}, \{x, y\}\}$$

Dobija se sledeća vrednost za listu  $res$  koja odgovara rešenjima ovih problema:

$\{\{5, \{x \rightarrow 2, y \rightarrow 3\}\}, \{4.6, \{x \rightarrow 2., y \rightarrow 3.\}\}, \{4.5, \{x \rightarrow 3., y \rightarrow 1.5\}\},$   
 $\{4.5, \{x \rightarrow 3., y \rightarrow 1.5\}\}, \{\frac{9}{2}, \{x \rightarrow 3, y \rightarrow \frac{3}{2}\}\}\}$

a rezultat je

**Out[2]=**  $\{\{5, 1\}, \{x \rightarrow 2, y \rightarrow 3\}\}$

### 3.4.1 Linearni metod težinskih koeficijenata

Uočimo sledeći problem normalizovane jednokriterijumske optimizacije:

$$\text{Max.: } f(x) = \sum_{k=1}^l \omega_k Q_k^o(x), \quad (3.4.1)$$

$$\text{p.o.: } x \in X, \quad (3.4.2)$$

gde težine  $w_i, i=1, \dots, l$  odgovaraju funkcijama cilja zadovoljavajući sledeće uslove:

$$\sum_{i=1}^l w_i = 1, \quad w_i \geq 0, \quad i = 1, \dots, l, \quad (3.4.3)$$

i  $Q_k^o(x)$  je normalizovana  $k$ -ta funkcija cilja  $Q_k(x)$ ,  $k = \overline{1, l}$ .

U slučaju linearnih težinskih koeficijenata, posmatramo problem višekriterijumske optimizacije (3.1.1) sa linearnim funkcijama cilja sledećeg oblika:

$$Q_i(x) = \sum_{k=1}^l a_{ki} x_i, \quad a_{ki} \in \mathbf{R}.$$

Dakle, normalizovane funkcije cilja imaju sledeću formu:

$$Q_k^o(x) = \frac{Q_k(x)}{S_k} = \frac{a_{k1}}{S_k} x_1 + \frac{a_{k2}}{S_k} x_2 + \dots + \frac{a_{kn}}{S_k} x_n,$$

u kojem slučaju su normalizacione vrednosti  $S_k$  određene na sledeći način:

$$S_k = \sum_{j=1}^n |a_{kj}| \neq 0.$$

Očigledno, u mnogim praktičnim problemima, funkcije cilja su predstavljene promenljivim mernim jedinicama (na primer, ako je  $Q_1$  mereno u kilogramima,  $Q_2$  u sekundama, itd.). Iz tih je razloga neophodna normalizacija funkcija cilja. Jasno je da sada koeficijenti imaju vrednosti iz segmenta  $[0, 1]$ . Sada, posmatrajmo dobijeni normalizovani problem linearног programiranja

$$\begin{aligned} \text{Max.: } f(x) &= \sum_{k=1}^l \omega_k \frac{Q_k(x)}{S_k} = \omega_1 \frac{a_{k1}}{S_k} x_1 + \frac{a_{k2}}{S_k} x_2 + \dots + \omega_n \frac{a_{kn}}{S_k} x_n, \\ \text{p.o.: } x &\in X, \end{aligned} \quad (3.4.4)$$

Sledeća teorema daje praktičan kriterijum za otkrivanje nekih Pareto optimalnih rešenja problema (3.4.4).

**Teorema 3.4.5.** *Rešenje problema višekriterijumske optimizacije (3.1.1) u slučaju linearnih funkcija cilja, generisano metodom težinskih koeficijenata (3.4.4) je Pareto optimalno rešenje ako su sledeći uslovi zadovoljeni:  $\frac{\omega_k}{S_k} > 0$  za svako  $k \in \{1, \dots, l\}$ .*

**Dokaz.** Označimo sa  $x^*$  rešenje problema višekriterijumske optimizacije (3.4.4), dobijeno maksimizovanjem funkcije  $f(x) = \sum_{k=1}^l \omega_k Q_k^o(x)$ . Očigledno, zadovoljeno je da  $f(x^*) \geq f(x), \forall x \in X$ . dalje, dobijamo sledeća tvrđenja

$$\begin{aligned} & \sum_{k=1}^l \omega_k Q_k^o(x^*) \geq \sum_{k=1}^l \omega_k Q_k^o(x), \forall x \in X \\ \Leftrightarrow & \sum_{k=1}^l \omega_k (Q_k^o(x^*) - Q_k^o(x)) \geq 0, \forall x \in X \\ \Leftrightarrow & \sum_{k=1}^l \frac{\omega_k}{S_k} (Q_k(x^*) - Q_k(x)) \geq 0, \forall x \in X \end{aligned} \quad (3.4.5)$$

prepostavimo suprotno, da rešenje  $x^*$  problema (3.1.1) nije Pareto optimalno. onda tu postoje neki mogući slučajevi  $x'$  problema (3.1.1) za koje je zadovoljeno:  $Q_k(x') \geq Q_k(x^*)$ , koje podrazumeva da

$$Q_k(x^*) - Q_k(x') \leq 0 \quad \forall k \in \{1, \dots, l\}.$$

Pritom tu postoji na kraju jedan indeks  $k_i$  za koji je nejednakost stroga. Sumiranjem ovih nejednakosti i sa imajući u vidu prepostavku teoreme da su sve vrednosti  $\frac{\omega_k}{S_k}$  pozitivne

$$\sum_{k=1}^l \frac{\omega_k}{S_k} [Q_k(x^*) - Q_k(x')] < 0.$$

Naravno, ova nejednakost je suprotna tvrđenju (3.4.5). U ovom slučaju, posmatrano rešenje  $x^*$  mora biti Pareto optimalno.  $\square$

Ova teorema predstavlja način konstruisanja težinskih koeficijnata  $\omega_i$ ,  $i = \overline{1, l}$  u nameri da generiše samo Pareto optimalna rešenja primenom metoda težinskih koeficijenata. to je, ako donosilac odluke izabere pozitivan realan broj  $c$ , težinski koeficijenti su automatski generisani kao  $\omega_i = c \cdot S_i$ ,  $i = \overline{1, l}$ .

**Posledica 3.4.6.** *Rešenje problema višekriterijumske optimizacije (3.1.1) u slučaju nenormalizovanih linearnih funkcija cilja, generisanih metodom težinskih koeficijenata (3.4.4) je Pareto optimalno ako su sledeći uslovi zadovoljeni:  $\omega_k > 0$  za svako  $k \in \{1, \dots, l\}$ .*

Mehanizovanje procesa konstruisanja skupa Pareto optimuma može biti izvšeno kompjuterski potpomognutom metodom težinskih koeficijenata  $w_i$ , zadovoljavajući (3.4.3). Ovaj metod je baziran na standardnoj MATHEMATICA funkciji Compositions[].

Na osnovu Teoreme 3.4.5, za pozitivne težine i konveksan problem, optimalno rešenje smenjenog problema (3.4.1) je Pareto optimalno (sličan rezultat je dođen u [114]). Minimizovanje problema (3.4.1) sa strogo pozitivnim težinama je dovoljan uslov za Pareto optimalnost. Ipak, formulacija ne obezbeđuje neophodan uslov za Pareto optimalnost [116]. Kada je višekriterijumski problem konveksan, primena funkcije W=Compositions[k,1] daje  $b$  Pareto optimalnih rešenja, gde ceo broj  $b$  zadovoljava  $1 \leq b \leq \frac{(k+l-1)!}{k!(l-1)!}$ .

**Primer 3.4.2.** u slučaju  $k = 5, l = 2$ , izraz `W=Compositions[k,1]/k` daje sledeću listu  $W$ :

$$\{\{0, 1\}, \{\frac{1}{5}, \frac{4}{5}\}, \{\frac{2}{5}, \frac{3}{5}\}, \{\frac{3}{5}, \frac{2}{5}\}, \{\frac{4}{5}, \frac{1}{5}\}, \{1, 0\}\}.$$

broj Pareto optimalnih tačaka je ceo između 1 i 6.

Takođe je dopušteno da donosilac odluka eksplicitno definiše koeficijente  $w_i$ ,  $i = 1, \dots, l$ .

### 3.4.2 Implementacioni detalji za dvodimenzionalni slučaj

Posmatrajući opštu formu problema višekriterijske optimizacije u  $\mathbb{R}^2$ :

$$\begin{aligned} \text{Max.: } & Q(\mathbf{x}) = Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2 \\ \text{p.o.: } & a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i \leq 0, \quad i \in I_1 \\ & a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i \geq 0, \quad i \in I_2 \\ & x, y \geq 0. \end{aligned} \quad (3.4.1)$$

gde  $I_1 \cup I_2 = \{1, \dots, m\}$ ,  $I_1 \cap I_2 = \emptyset$  and  $a_{ij}, b_i, c_j$  su zadati realni brojevi  $m = |I_1| + |I_2|$ , sto je objašneno u [79]. Svaka nejednakost je ograničena u (3.4.1) određuje podskup  $D_i \subset \mathbb{R}^2$ ,  $i = 1, \dots, m$ , predstavljajući skup tačaka na jednoj strani od pdgovarajućih realnih algebarskih krivih

$$a_{11}^i x^2 + a_{22}^i y^2 + 2a_{12}^i xy + 2a_1^i x + 2a_2^i y + a_0^i = 0.$$

Dakle, skup mogućih rešenja (označen kao  $\Omega_P$  u  $\mathbb{R}^2$ ) je određen kao presek

$$\Omega_p = D_1 \cap D_2 \cap \dots \cap D_m \cap D_{m+1} \cap D_{m+2},$$

gde su podskupovi  $D_{m+1}, D_{m+2}$  of  $\mathbb{R}^2$  dobijeni iz uslova  $x \geq 0, y \geq 0$ .

Formalni parametri funkcije `MultiW` su korišeni u sledećem smislu:

$q_-, constr\_List, var\_List$ : Lista neizračunatih izraza (predstavljena funkcijama cilja), lista zadatih uslova i lista neoznačenih promenljivih, respektivno (u internoj formi problema).

$w1\_List$ : Prazna lista je korišćena kao vrednost parametra  $w1$  znači da će težinski koeficijenti biti generisani sa značenjem funkcije `Compositions`. Inače, to je simulirano da svaki element  $w1[[i]], 1 \leq i \leq Length[w1]$  liste  $w1$  je mogući skup koeficijenata  $w_j$ ,  $j = 1, \dots, l$ :  $w[[j]] = w1[[i, j]], j = 1, \dots, l$ .

Lokalna promenljivars u funkciji `MultiW` sabira međurezultate. Takođe, lokalna promenljiva  $fun$  predstavlja izraz  $Q(\mathbf{x}) = \sum w_i Q_i(\mathbf{x})$  u (3.2.1).

```

MultiW[q_, constr_List, var_List, w1_List] :=
Module[{i = 0, k, l = Length[q], res = {}, w = {}, fun, sk = {},
qres = {}, mxs = {}, Paretos = {}, m, ls = {}},
If[w1 == {},
k = Input["Initial sum of weighting coefficients? "];
w = Compositions[k, l] / k,
w = w1;
];
Print["Weighting Coefficients: "]; Print[w];
Print["Single-objective problems: "];
k = Length[w];
For[i = 1, i <= k, i++,
fun = Simplify[Sum[w[[i, j]] * q[[j]], {j, l}]];
Print[{fun, constr, var}];
Temp = Maximize[fun, constr, var];
AppendTo[res, Temp];
If[IsPareto[q, constr, First[Rest[Temp]], var] == 1,
AppendTo[Paretos, List[Temp[[2, 1, 2]], Temp[[2, 2, 2]]]]];
];
Print["Solutions of single-objective problems: "];
Print[res];
For[i = 1, i <= k, i++,
AppendTo[qres, q /. res[[i, 2]]]; AppendTo[mxs, res[[i, 1]]];
];
Print["Values of objectives corresponding to solutions
of single-objective problems: "];
Print[qres];
m = Max[mxs];
For[i = 1, i <= Length[mxs], i++,
If[m == mxs[[i]], AppendTo[ls, {qres[[i]], res[[i, 2]]}]];
];
(* Print["Paretos = ", Paretos]; *)
DrawParetos[Paretos, constr, var];
Return[ls];
]
]

```

Primer 3.4.3. Rešimo sledeći problem višekrijumske optimizacije:

$$\begin{aligned}
&\max \quad [40x + 10y; x + y] \\
&p.o. \quad 2x + y \leq 6 \\
&\quad x + y \leq 5 \\
&\quad x \leq 2 \\
&\quad x, y \geq 0
\end{aligned}$$

formiravši koeficijente  $w_i$ :

```

pts = MultiW[{40x + 10y, x + y}, {2x + y <= 6, x + y <= 5, x <= 2,
x >= 0, y >= 0}, {x, y}, {}]

```

U slučaju  $k = 5$  imamo  $w = \{\{0, 5\}, \{\frac{1}{5}, \frac{4}{5}\}, \{\frac{2}{5}, \frac{3}{5}\}, \{\frac{3}{5}, \frac{2}{5}\}, \{\frac{4}{5}, \frac{1}{5}\}, \{5, 0\}\}$ . Neophodno je rešiti sledećih šest problema linearog programiranja, za svaku podlistu  $w$ . Naš modul najpre iskorišćava probleme linearog programiranja, i onda odgovarajuća rešenja.

$$\begin{aligned} \max \quad & \frac{5(x+y)}{2}, & \{\{80, 5\}, \{x \rightarrow 1, y \rightarrow 4\}\} \\ \max \quad & \frac{14x+11y}{5}, & \{\{80, 5\}, \{x \rightarrow 1, y \rightarrow 4\}\} \\ \max \quad & \frac{31x+19y}{10}, & \{\{80, 5\}, \{x \rightarrow 1, y \rightarrow 4\}\} \\ \max \quad & \frac{17x+8}{5}, & \{\{100, 4\}, \{x \rightarrow 2, y \rightarrow 2\}\} \\ \max \quad & \frac{37x+13y}{10}, & \{\{100, 4\}, \{x \rightarrow 2, y \rightarrow 2\}\} \\ \max \quad & \frac{4x+y}{1}, & \{\{100, 4\}, \{x \rightarrow 2, y \rightarrow 2\}\} \end{aligned}$$

Uzimajući težinske koeficijente  $w_1 = \frac{1}{5}$  i  $w_2 = \frac{4}{5}$ , rešavamo problem maksimizacije funkcije  $\frac{14x+11y}{5}$  na datom skupu uslova. Rešenje  $x = 1, y = 4$  je dobijeno. U toj tački prva funkcija cilja ima vrednost 80, i druga 5. Rešenje  $\{x = 1, y = 4\}$  je Pareto optimalno po definiciji. U slučaju težinskih koeficijenata  $w_1 = \frac{3}{5}$  and  $w_2 = \frac{2}{5}$ , sa maksimizovanjem funkcije  $\frac{14x+11y}{5}$  dobijamo rešenje  $\{x = 2, y = 2\}$ , koje je takođe Pareto optimalno.

Funkcija za dokazivanje uslova Pareto optimalnosti i funkcije za ilustraciju Pareto optimalnih tačaka su opisane u radu [83]. Sva generisana rešenja zadovoljavajući **IsPareto** implementirana u [83] su korišćena za generisanje Pareto optimalnih tačaka, označenih sa **Paretos**. Standardna MATHEMATICA funkcija **ListPlot** iscrtava tače koje su sadržane u listi **Paretos**. Funkcija **RegionPlot[ineqs, vars]** daje grafičko prikazivanje skupa nejednakosti **ineqs**, sa promenljivim **vars**. Sa značenjem grafičkih funkcija **Show** kombinujemo grafike Pareto optimalnih tačaka i grafike skupa ograničenja.

Svaki od odgovarajućih realnih algebarskih krivih deli oblast u opsegu koji je moguć za ove uslove i opseg koji je nemoguć za ove uslove. Dopušteni uslovi su smešteni u opseg  $\Omega_P$  to je dopušteno za sve uslove (oblast mogućih uslova). Funkcija **DrawParetos** koristi sledeće parametre:

1. **Paretos\_List**: Pareto tačke;
2. **constr\_List**: lista ograničenja;
3. **var\_List**: lista promenljivih.

```
DrawParetos[Paretos_List, constr_List, var_List]:= Block[{p1, p2},
  p1=ListPlot[Paretos, PlotStyle->\{PointSize[0.033], Hue[1]\},
    DisplayFunction->Identity];
  p2=RegionPlot[constr,{var[[1]]},{var[[2]]}, AspectRatio->1,
    DisplayFunction->Identity];
  Show[p2,p1 ,DisplayFunction->$DisplayFunction ];
]
```

Najpre iscrtavamo skup Pareto optimalnih tačaka **Paretos** primenom standardne MATHEMATICA funkcije **ListPlot** (videti [113]). U sledećem koraku, crtamo skup ograničenja (grafike **constr**) preimenjući standardnu MATHEMATICA funkciju **RegionPlot**.

**Primer 3.4.4.** Rešimo sledeći problem:

$$\begin{aligned} \text{Max.: } & Q_1(x_1, x_2) = 8x_1 + 12x_2, \quad Q_2(x_1, x_2) = 14x_1 + 10x_2, \quad Q_3(x_1, x_2) = x_1 + x_2 \\ \text{p.o.: } & 8x_1 + 4x_2 \leq 600, \quad 2x_1 + 3x_2 \leq 300, \\ & 4x_1 + 3x_2 \leq 360, \quad 5x_1 + 10x_2 \geq 600, \quad x \geq 0, y \geq 0. \end{aligned}$$

Koristimo, na primer, metod težinskih koeficijenata, i biramo sledeće težinske koeficijente:

$$\{\{0, 0, 1\}, \{0, \frac{2}{5}, \frac{3}{5}\}, \{1, 0, 0\}\}$$

Ovaj problem može biti rešen izrazom

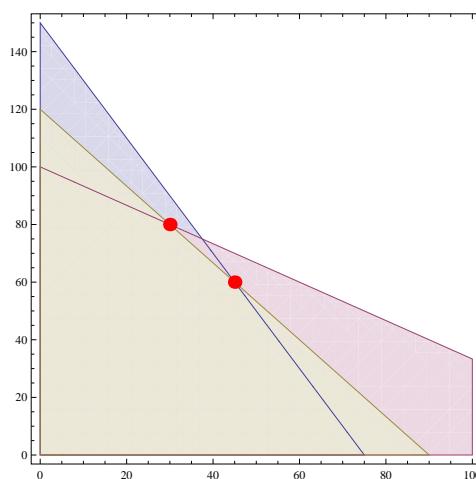
```
MultiW[\{8x1+12x2, 14x1+10x2, x1+x2\},
\{8x1+4x2<=600, 2x1+3x2<=300, 4x1+3x2<=360, 5x1+10x2>=600, x1>=0, x2>=0\},
{x1,x2}, {\{0,0,1\}, \{0,2/5,3/5\}, \{1,0,0\}}]
```

Program generiše tri rešenja, odgovarači svakoj selekciji težinskih koeficijenata. Za prvi odabir ovih koeficijenata rešenje problema je  $\{\{1200, 1220, 110\}, \{x_1 \rightarrow 30, x_2 \rightarrow 80\}\}$ , i Pareto optimalno je.

Rešenje  $\{\{1080, 1230, 105\}, \{x_1 \rightarrow 45, x_2 \rightarrow 60\}\}$ , dobijeno korišćenjem težinskih koeficijenata  $\{0, \frac{2}{5}, \frac{3}{5}\}$  je takođe Pareto optimalno.

Optimalno rešenje koje odgovara težinskim koeficijentima  $\{1, 0, 0\}$  dato je u obliku  $\{\{1200, 1000, 100\}, \{x_1 \rightarrow 0, x_2 \rightarrow 100\}\}$ . ovo rešenje nije Pareto optimalno, zato što rešenje odgovara prvom izboru težinskih koeficijenata , gde  $\{x_1 \rightarrow 30, x_2 \rightarrow 80\}$ , daje veće vrednosti funkcija  $Q_2$  i  $Q_3$ , i vrednosti za funkciju  $Q_1$  u tačkama  $\{x_1 \rightarrow 30, x_2 \rightarrow 80\}$  i  $\{x_1 \rightarrow 0, x_2 \rightarrow 100\}$  su jednake.

Ovaj rezultat je ilustrovan Slikom 3.4.1, gde je prostor ograničenja predstavljen kao oblast na kojoj Pareto optimalne tačke leže.



Slika 3.4.1. Grafička reprezentacija rešenja problema.

### 3.4.3 Implementacioni detalji za trodimenzionalni slučaj

Posmatrajmo opštu formu problema višekriterijumske optimizacije u  $\mathbb{R}^3$  ( $3D$  problem):

$$\begin{aligned} \text{Max.: } \quad & Q(\mathbf{x}) = Q_1(\mathbf{x}), \dots, Q_l(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^3 \\ \text{p.o.: } \quad & a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + \\ & 2a_{23}yz + 2a_1x + 2a_2y + 2a_3z + a_0 \leq 0, \quad i \in I_1 \\ & a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + \\ & 2a_{23}yz + 2a_1x + 2a_2y + 2a_3z + a_0 \geq 0, \quad i \in I_2 \\ & x, y, z \geq 0. \end{aligned}$$

Slično slučaju  $2D$ , skup dopustivih rešenja (u  $\mathbb{R}^2$  označen sa  $\Omega_P$ ) je određen kao presek

$$\Omega_p = D_1 \cap D_2 \cap \dots \cap D_m \cap D_{m+1} \cap D_{m+2} \cap D_{m+3},$$

gde  $D_i \subset \mathbb{R}^3$  je skup rešenja nejednakosti  $i$ -th i  $D_{m+1}, D_{m+2}, D_{m+3} \subset \mathbb{R}^3$  je dobij iz uslova  $x \geq 0, y \geq 0, z \geq 0$ .

```
MultiW3D[q_, constr_List, var_List, wl_List] :=
Module[{i = 0, k, l = Length[q], res = {}, w = {}, fun,
sk = {}, cfs, qres = {}, mxs = {}, m, ls = {}},
If[wl == {},
k = Input["How many times? "];
w = Compositions[k, l] / k,
k = Length[wl] - 1; w = wl;
];
Print["Weighting Coefficients: "]; Print[w];
(* Normalization *)
For[i = 1, i ≤ l, i++,
cfs = Coefficient[q[[i]], var];
AppendTo[sk, Sum[cfs[[j]], {j, Length[cfs]}]]];
];
Print["Single-objective problems: "];
For[i = 1, i <= k + 1, i++,
fun = Simplify[Sum[w[[i, j]] * q[[j]] / sk[[j]], {j, l}]];
Print[{fun, constr, var}];
AppendTo[res, Penalty[fun, constr, var, {5, 5},
10^-4, 10^-4, 30, Interior]];
];
Print["Solutions of single-objective problems: "];
Print[res];
For[i = 1, i <= k + 1, i++,
AppendTo[qres, q /. res[[i, 2]]]; AppendTo[mxs, res[[i, 1]]];
];
Print["Values of objectives corresponding to
solutions of single-objective problems: "];
Print[qres];
m = Max[mxs];
For[i = 1, i ≤ Length[mxs], i++,
If[m == mxs[[i]], AppendTo[ls, {qres[[i]], res[[i, 2]]}]];
];
Return[ls];
]
```

Odgovarajući algoritam je implementiran MATHEMATICA funkcijom `DrawParetos3D` koja rešava dati  $3D$  problem i daje interaktivnu vizuelizaciju Pareto optimalnog rešenja. Ova funkcija ima sledeću formu:

```
DrawParetos3D[Paretos_List, constr_List, var_List, {}] :=
Module[{p1,p2},
p1 =Graphics3D[Table[{Blue, PointSize[0.08], Point[Paretos[[i]]]}, {i, Length[Paretos]}], DisplayFunction -> Identity],
p2 = RegionPlot[constr,(*Display the constraint set*)
{var[[1]]}, {var[[2]]}, {var[[3]]}, AspectRatio -> 1,
DisplayFunction -> Identity],
Show[p2, p1, DisplayFunction -> dip[EdgeForm[]],
ViewPoint -> {3, 4, 0}, AspectRatio -> 1]]]
```

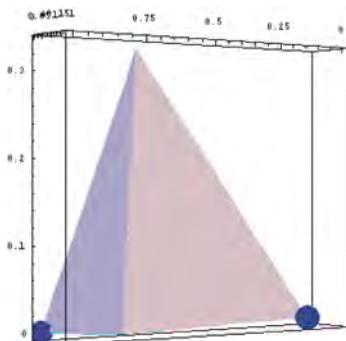
**Primer 3.4.5.** Posmatrajmo sledeći problem višekriterijske optimizacije, za slučaj tri promenljive

$$\begin{aligned} \text{Max.: } Q(\mathbf{x}) &= [x + y + z, -x - 3y - 6z, x + y + z] \\ \text{p.o.: } x + y + z &\leq 1 \\ x + 3y + z &\geq 0 \\ -x + 5y + 2z &\leq 0 \\ x \geq 0, y \geq 0, z \geq 0 \end{aligned}$$

```
pts = MultiW3D[{x + y + z, -x - 3y - 6z, x + y + z},
{ x + y + z <= 1, x + 3y + z >= 0, -x + 5y + 2z <= 0 ,
z >= 0, y >= 0, x >= 0}, {x,y,z}, {}]
```

Za slučaj  $k = 6$  imamo više mogućnosti za bolju grafičku ilustraciju (videti Sliku 3.4.2).

```
DrawParetos3D[pts, { x + y + z <= 1, x + 3y >= -z, -x + 5y <= -2z,
z>= 0, y >= 0, x >= 0}, {x, y, z}, {}]
```



Slika 3.4.2. Reprezentacija skupa dopustivih rešenja i skupa Pareto optimuma

**Primer 3.4.6.** Razmatrajmo sledeći 3D problem višekriterijumske optimizacije

$$\begin{aligned} \text{Max.: } Q(\mathbf{x}) &= [x + y + z, -x - 3y - 6z, x + y + z] \\ \text{p.o.: } &x^2 + y^2 + z^2 \leq 1 \\ &x \geq 0, y \geq 0, z \geq 0 \end{aligned}$$

Korišćenjem sledeće funkcije

```
pts =MultiW3D[{x+y+z, -x-3y-6z, x+y+z},
{x^2+y^2+z^2<=1, z>=0, x>=0}, {x,y,z}, {}]
```

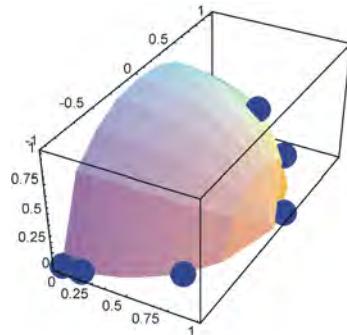
kada  $k = 9$ , dobijamo sledeće Pareto optimalne tačke:

$$\begin{aligned} &\{0, -1, 0\}, \left\{-\frac{1}{\sqrt{2}} + \sqrt{2}, -\frac{1}{\sqrt{2}}, 0\right\}, \left\{-\frac{49}{5\sqrt{2}} + 5\sqrt{2}, -\frac{7}{\sqrt{2}}, 0\right\}, \\ &\left\{-\frac{2}{\sqrt{3}} + \sqrt{3}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right\}, \left\{\frac{1}{5}\left(-\frac{1}{\sqrt{26}} + \sqrt{26}\right), \frac{1}{\sqrt{26}}, 0\right\}, \left\{\frac{1}{7}\left(-2\sqrt{\frac{2}{\sqrt{39}}} - \frac{25}{\sqrt{78}} + \sqrt{78}\right), \frac{5}{\sqrt{78}}, \sqrt{\frac{2}{39}}\right\}. \end{aligned}$$

Korišćenjem sledeće funkcije

```
DrawParetos3D[pts, {x^2 + y^2 + z^2 <= 1, z >= 0, x >= 0}, {x, y, z}, {}]
```

dobijamo grafičko predstavljanje Pareto optimalnih tačaka



Slika 3.4.3. Pareto optimalne tačke leže na granici skupa dopustivih rešenja

Primenom prvobitnog koda dobijamo lepu interaktivnu demonstraciju. Demonstracija takođe pruža mogućnost korisniku da rotira sliku i vidi iz različitih uglova i tačaka viđenja, i mogućnost da vide sve Pareto optimalne tačke.

## 3.5 Leksikografska višekriterijumska optimizacija

Često se do optimalnog rešenja dolazi posle uzastopnog donošenja odluka. Prvo se nađe optimalno rešenje za najvažniju funkciju cilja. Ako je optimalno rešenje jedinstveno, tada je problem rešen. Međutim, ako optimalno rešenje nije jedinstveno, tada se na skupu svih optimalnih rešenja optimizuje funkcija cilja koja je druga po važnosti. Ako je optimalno

rešenje sada jedinstveno, problem je rešen; ako nije, optimizuje se funkcija cilja treća po važnosti na skupu optimalnih rešenja prve i druge funkcije cilja itd.

Dakle posmatrajmo problem minimizacije date uređene sekvene ciljnih funkcija

$$Q_1(x), \dots, Q_l(x)$$

i skup ograničenja

$$\mathbf{x} \in \mathbf{X}.$$

Trebalo bi da se reši sledeći skup konveksnih uslovnih nelinearnih programa

$$\begin{aligned} \max \quad & Q_i(\mathbf{x}) \\ \text{p.o.} \quad & \mathbf{x} \in \mathbf{X} \\ & Q_j(\mathbf{x}) \geq a_j, \quad j = 1, \dots, i-1, \quad i \geq 2, \end{aligned}$$

gde su  $a_j = Q_j(\mathbf{x}_j^*)$ ,  $j = 1, \dots, i-1$  optimalne vrednosti prethodno postavljenih jednokriterijumskih problema na nivoima prioriteta  $j = 1, \dots, i-1$ ,  $i \geq 2$ .

Nejednakosti u poslednjem problemu mogu biti zamenjene jednakostima [72]. leksikografski metod spada u grupu apriornih metoda.

Prepostavimo da su kriterijumske funkcije  $Q_1, \dots, Q_l$  poređane od najvažnije do najmanje važne. Polazni problem višekriterijumske optimizacije se rešava sledećim algoritmom.

---

### Algoritam 3.3 Leksikografski metod višekriterijumske optimizacije

---

**Ulaz:** Problem VKO (3.1.1) sa listom uređenih funkcija cilja  $Q_1, \dots, Q_l$ .

1: Postaviti vrednost  $S_0 := S$ ,  $i := 1$

2: Rešava se problem:

Minimizovati  $Q_i(\mathbf{x})$  pod uslovom  $\mathbf{x} \in S_{i-1}$

Neka je rešenje ovog problema  $\mathbf{x}^{(i)*}$ .

3: Ako je  $\mathbf{x}^{(i)*}$ , dobijeno u koraku 2, jedinstveno rešenje, ono se proglašava za rešenje višekriterijumskog problema i algoritam se završava.

4: Formira se skup  $S_i := \{\mathbf{x} \in S_{i-1} \mid Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^{(i)*})\}$ .

5: Ako je  $i = l$  skup  $S_l$  se proglašava za skup rešenje problema višekriterijumske optimizacije, i algoritam se završava.

6: Stavlja se  $i := i + 1$  i vraćamo se na korak 2.

---

Jedan od načina za implementaciju višekriterijumske optimizacije dat je funkcijom *MultiLex* [83].

```
MultiLex[q_List, constr_List, var_List] :=
Module[{res={}, s, f=constr, l=Length[q], i},
For[i=1, i<=l, i++,
  s=Maximize[q[[i]], f, var]; (* 2 *)
  If[i<l, AppendTo[f, q[[i]]]>=First[s]]; (* 3,2 *)
  AppendTo[res, {q/.Last[s], Last[s]}];
];
Print[res]; Return[{q/.Last[s], Last[s]}];
]
```

**Primer 3.5.1.** Problem maksimizacije funkcija koje su sadržane u listi

$$\{8x_1 + 12x_2, 14x_1 + 10x_2, x_1 + x_2\}$$

prema ograničenjima

$$\begin{aligned} \{8x_1 + 4x_2 &\leq 600, 2x_1 + 3x_2 \leq 300, 4x_1 + 3x_2 \leq 360, 5x_1 + 10x_2 \geq 600, \\ &x_1 \geq 0, x_2 \geq 0\} \end{aligned}$$

može biti rešen koristeći izraz

```
MultiLex[\{8x1+12x2, 14x1+10x2, x1+x2\},
{\8x1+4x2<=600, 2x1+3x2<=300, 4x1+3x2<=360, 5x1+10x2>=600, x1>=0, x2>=0\},
{x1, x2}]
```

Unutrašnje reprezentacije jednokriterijumskih problema su:

za  $i = 1$ :

$$\begin{aligned} &8x_1 + 12x_2, \\ \{8x_1 + 4x_2 &\leq 600, 2x_1 + 3x_2 \leq 300, 4x_1 + 3x_2 \leq 360, 5x_1 + 10x_2 \geq 600, \\ &x_1 \geq 0, x_2 \geq 0\}, \\ &\{x_1, x_2\} \end{aligned}$$

za  $i = 2$ :

$$\begin{aligned} &14x_1 + 10x_2, \\ \{8x_1 + 4x_2 &\leq 600, 2x_1 + 3x_2 \leq 300, 4x_1 + 3x_2 \leq 360, 5x_1 + 10x_2 \geq 600, \\ &x_1 \geq 0, x_2 \geq 0, 8x_1 + 12x_2 \geq 1200\}, \\ &\{x_1, x_2\} \end{aligned}$$

za  $i = 3$ :

$$\begin{aligned} &x_1 + x_2, \\ \{8x_1 + 4x_2 &\leq 600, 2x_1 + 3x_2 \leq 300, 4x_1 + 3x_2 \leq 360, 5x_1 + 10x_2 \geq 600, \\ &x_1 \geq 0, x_2 \geq 0, 8x_1 + 12x_2 \geq 1200, 14x_1 + 10x_2 \geq 1220\}, \\ &\{x_1, x_2\} \end{aligned}$$

Rešenja uzastopnih optimizacionih problema su zapamćena u listi  $res$ , koja je jednaka

$$\{\{1200, \{x_1 \rightarrow 0, x_2 \rightarrow 100\}\}, \{1220, \{x_1 \rightarrow 30, x_2 \rightarrow 80\}\}, \{110, \{x_1 \rightarrow 30, x_2 \rightarrow 80\}\}\}$$

**Teorema 3.5.1.** Rešenje dobijeno leksikografskom metodom je Pareto optimalno rešenje problema višekriterijumske optimizacije.

**Dokaz.** Označimo sa  $\mathbf{x}^*$  rešenje dobijeno leksikografskim metodom. Prepostavimo da ono nije Pareto optimalno tj. da postoji neko  $x \in S$  tako da je  $Q_i(x) \geq Q_i(\mathbf{x}^*)$  za svako  $i = 1, \dots, l$ , pri čemu za neko  $k$  važi  $Q_k(\mathbf{x}) > Q_k(\mathbf{x}^*)$ . Mogu nastati dva slučaja:

1. Prva mogućnost je da je  $\mathbf{x}^*$  jedinstveno rešenje i da su iskorišćeni svi kriterijumi do  $j$ -tog (to znači da je na kraju prethodno opisanog algoritma  $i = j$ ). Kako je  $\mathbf{x} \in S = S_0$  i  $Q_1(\mathbf{x}) \geq Q_1(\mathbf{x}^*)$  sledi da je  $Q_1(\mathbf{x}) = Q_1(\mathbf{x}^*)$ , kao i  $x \in S_1$ . Kako je sada  $x \in S_1$  i  $Q_2(x) \geq Q_2(\mathbf{x}^*)$  sledi  $Q_2(x) = Q_2(\mathbf{x}^*)$  i  $\mathbf{x} \in S_2$ . Nastavljujući ovakvo rezonovanje zaključujemo  $Q_{j-1}(\mathbf{x}) = Q_{j-1}(\mathbf{x}^*)$  i  $\mathbf{x} \in S_{j-1}$ . Kako je  $\mathbf{x}^*$  jedinstveno rešenje problema

Minimizovati  $Q_j(\mathbf{x})$  pod uslovom  $\mathbf{x} \in S_{j-1}$

a pri tome važi  $Q_j(\mathbf{x}) \geq Q_j(\mathbf{x}^*)$  i  $\mathbf{x} \in S_{j-1}$  zaključujemo da mora biti  $\mathbf{x} = \mathbf{x}^*$  pa je  $Q_k(\mathbf{x}) = Q_k(\mathbf{x}^*)$ , odakle dobijamo kontradikciju sa pretpostavkom  $Q_k(\mathbf{x}) > Q_k(\mathbf{x}^*)$ .

2. Druga mogućnost je da su iskorišćeni svi kriterijumi (tj. u prethodnom algoritmu je  $i = l$ ). Potpuno analogno kao u prvom slučaju se pokazuje da  $\mathbf{x} \in S_k$  i da za svako  $i = 1, \dots, k$  važi  $Q_i(\mathbf{x}) = Q_i(\mathbf{x}^*)$ , što je kontradikcija sa  $Q_k(\mathbf{x}) > Q_k(\mathbf{x}^*)$ .  $\square$

Iz ovog dokaza se može zaključiti da se ograničenje oblika  $Q_j(x) \leq Q_j(\mathbf{x}^{(i)*})$  može zameniti ograničenjem  $Q_j(\mathbf{x}) = Q_j(\mathbf{x}^{(i)*})$ .

Mane leksikografske metode su očigledne:

- U praksi je često teško odrediti koji je kriterijum važniji od drugog.
- Najčešće se dešava da se jedinstveno rešenje dobije pre nego što se iskoriste svi kriterijumi, ili čak nakon korišćenja samo jednog kriterijuma. Na taj način, neki kriterijumi uopšte ne učestvuju u doноšenju odluke, što je krajnje nepoželjno.

Zbog toga klasični leksikografski metod ima jako ograničenu primenu.

### 3.5.1 Primena geometrijskog metoda u leksikografskom metodu

Modifikacija leksikografskog metoda za slučaj dve ili tri promenljive predložena je u radu [73]. Pritom su korišćeni poznati rezultati linearног programiranja, koje ovde navodimo. Relacije između linearног programiranja i višekriterijumske optimizacije su dobro istražene. Neki od poznatih rezultata navedeni su u radu [39], od kojih ćemo ovde navesti neke od njih.

Posmatrajmo najpre bazični problem linearног programiranja u prostoru  $\mathbb{R}^n$ , koristeći ograničenja u obliku nejednakosti. Bez gubitka opštosti, možemo posmatrati samo problem maksimizovanja:

$$\begin{aligned} \text{Max.: } & f(\mathbf{x}) = f(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n \\ \text{p.o.: } & \sum_{j=1}^n a_{ij}x_j \geq b_i, \quad i = 1, \dots, m, \end{aligned} \tag{3.5.1}$$

gde su konstante  $a_{ij}, b_i, c_j \in \mathbb{R}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$  unapred zadate. Dalje ćemo navesti nekoliko poznatih rezultata korišćenih u geometrijskom (grafičkom) metodu.

Proizvoljno rešenje  $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$  gore navedenog sistema nejednačina se naziva *dopustivo rešenje*. U geometrijskoj interpretaciji, ono je tačka  $n$ -dimenzionalnog prostora  $\mathbb{R}^n$ . Sa  $\Omega \subset \mathbb{R}^n$  označavamo skup svih dopustivih rešenja problema (3.5.1). Pretpostavimo da je  $\Omega \neq \emptyset$ . Tada važi sledeće pomoćno tvrđenje.

**Lema 3.5.2.** *Granični skup  $\Omega_P = \{\mathbf{x} | \sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, \dots, m\}$  skupa dopustivih rešenja  $\Omega$  je konveksan skup.*

*Optimalna tačka  $x^* = (x_1^*, \dots, x_n^*) \in \Omega$  problema linearног programiranja (3.5.1) je dopustivo rešenje u kojoj je vrednost funkcije cilja  $f(x_1, \dots, x_n)$  maksimalna.*

**Lema 3.5.3.** *Optimalna vrednost funkcije cilja problema (3.5.1) dobija se u nekoj od ekstremnih tačaka skupa dopustivih rešenja  $\Omega$ .*

Značaj prethodne teoreme je u tome što se broj potencijalnih ekstremuma funkcije  $f(x)$  redukuje sa (u opštem slučaju) beskonačnog skupa  $\Omega_P$  na konačan skup temena koji ima maksimalno  $\binom{n}{m}$  elemenata. Ciljna funkcija  $f(x)$  dostiže maksimum ili minimum u temenima konveksnog skupa  $\Omega_p$ . Formalno, dovoljno je izračunati vrednosti ciljne funkcije u svim ekstremnim tačkama skupa  $\Omega_P$  i odrediti onu tačku, ili tačke, u kojima je vrednost funkcije  $f(x)$  ekstremna. U sledećoj teoremi su generalizovana neka poznata tvrđenja.

**Stav 3.5.4.** *Ako su dobijena dva optimalna rešenja problema (3.5.1), tada svaka tačka na duži određenoj tim tačkama predstavlja optimalno rešenje problema (3.5.1). Odnosno, ako su  $x^{(1)} \in \Omega_P$  i  $x^{(2)} \in \Omega_P$  optimalna rešenja, onda je svaki vektor oblika*

$$x_\lambda = \lambda x^{(1)} + (1 - \lambda)x^{(2)}, \quad 0 \leq \lambda \leq 1$$

*optimalno rešenje.*

**Dokaz.** Pošto su  $x^{(1)}$  i  $x^{(2)}$  optimalna rešenja, važi

$$f(x^{(1)}) \geq f(x), \forall x \in \Omega_P \wedge f(x^{(2)}) \geq f(x), \quad \forall x \in \Omega_P$$

Sledi da za svaki vektor  $x_\lambda$  važi

$$\begin{aligned} f(x_\lambda) &= f(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) = \lambda f(x^{(1)}) + (1 - \lambda)f(x^{(2)}) \\ &\geq \lambda f(x) + (1 - \lambda)f(x) = f(x), \quad \forall x \in \Omega_P. \end{aligned}$$

Dakle, svaki tačka  $x_\lambda$ ,  $0 \leq \lambda \leq 1$  je optimalno rešenje.  $\square$

Skup dopustivih rešenja geometrijski predstavlja poliedar u prostoru, a ciljna funkcija, koja se za  $f(x) = \text{const}$  interpretira kao hiperravan, dostiže maksimum (minimum) u jednom temenu poliedra (slučaj jedinstvenog optimalnog rešenja) ili po jednoj strani poliedra ako je hiperravan  $f(x) = \text{const}$  njoj paralelana (slučaj beskonačno mnogo optimalnih rešenja).

Kod slučaja  $n = 2$ , ako se ograničenja grafički predstave u koordinatnom sistemu  $x_1x_2$  dobija se konveksan poligon, na čijim se temenima (ekstremnim tačkama) nalaze moguća rešenja datog linearног modela. Teme najudaljenije od prave određene funkcijom cilja predstavlja optimalno rešenje datog modela.

Na osnovu prethodnih diskusija, implementacija geometrijskog metoda za slučaj dve promenljive može biti data funkcijom čiji su formalni parameteri funkcija cilja  $f$  i skup ograničenja  $g$ . Rešavanjem sistema nejednačina iz  $g$  dobijamo region dopustivih rešenja, zapamćen u promenljivoj  $h$ . Pritom koristimo funkciju `Reduce` (sličan način za rešavanje sistema nejednačina je dat u [5]). Ekstremne tačke su sačuvane u listi  $R$ . Ove tačke se određuju u preseku svake dve prave određene prevodenjem nejednakosti iz ograničenja u jednakosti. Nakon toga, lista ekstremnih tačaka  $R$  se sortira u neopadajući poredak na osnovu rastojanja od prave  $f(x_1, x_2) = 0$ . Optimalno rešenje je maksimalni element ove liste.

```

LinProgGraphicalM[f_, g_List] :=
Module[{res = {}, var = Variables[f], i, j, h, gE, t, r},
For[i = 1, i < Length[g], i++, var = Union[Variables[First[g[[i]]]], var]];
h = g /. {List -> And};
h = Reduce[h, var];
If[h == False, Print["The problem is inadmissible"]; Break[]];
gE = g /. {LessEqual -> Equal, GreaterEqual -> Equal, Less -> Equal, Greater -> Equal};
For[i = 1, i < Length[gE] - 1, i++,
For[j = i + 1, j < Length[gE], j++, t = FindInstance[gE[[i]] && gE[[j]] && h, var];
If[t != {}, AppendTo[res, {ReplaceAll[f, t[[1]]], t[[1]]}];];
];
For[i = 1, i < Length[res] - 1, i++,
For[j = i + 1, j < Length[res], j++, If[res[[i, 1]] < res[[j, 1]], r = res[[i]];
res[[i]] = res[[j]];
res[[j]] = r];
];
If[res[[1, 1]] == res[[2, 1]], r = {res[[1, 2]], res[[2, 2]]}, r = {res[[1, 2]]}];
Return[r];
]

```

### 3.5.2 Modifikovani algoritam leksikografskog metoda

Ovde navodimo modifikaciju algoritma leksikografskog metoda za rešavanje linearног problema višekriterijumske optimizacije, za slučaje dve ili tri nepoznate. Za razliku od klasičnog leksikografskog metoda, u svakom koraku se skup dopustivih rešenja smanjuje na skup optimalnih tačaka dobijenih u prethodnoj iteraciji, primenjujući geometrijski metod linearнog programiranja.

Ulagni parametri sledećeg algoritma su: lista funkcija cilja  $f$  (uređena po prioritetima) i lista ograničenja  $g$ .

---

**Algoritam 3.4** Rešavanje linearног problema višekriterijumske optimizacije (3.1.1), za slučaj dve nepoznate  $x_1, x_2$ .

#### Modifikovani leksikografski metod 2D

---

**Ulaz:** Lista uređenih funkcija cilja  $f$  i lista ograničenja  $g$ .

- 1: Postaviti vrednost indeksa  $i := 1$  i početnu vrednost skupa  $R := \emptyset$ . Uzeti prvu funkciju  $f_1$  kao funkciju cilja, a listu  $g$  kao skup ograničenja.
  - 2: Rešiti optimizacioni problem jedne funkcije cilja za datu funkciju cilja i ograničenja, koristeći geometrijski metod. Sačuvati rešenje u skupu  $R$ .
  - 3: Ako rešenje iz  $R$  predstavlja duž, reprezentovanu kao  $\lambda x^{(1)} + (1 - \lambda)x^{(2)}$ , gde je  $0 \leq \lambda \leq 1$ , ići na Korak 4. U suprotnom, ako je rešenje iz  $R$  tačka, ići na Korak 5.
  - 4: Ako je  $i < l$  tada: uvećati indeks  $i$  za 1, uzeti  $f_i$  kao funkciju cilja a skup  $R$  kao skup ograničenja, i ići na Korak 2. Ako je  $i = l$ , ići na Korak 5.
  - 5: **Rezultat:** Skup  $R$  kao rešenje polaznog problema.
- 

Zaključujemo da je rešenje problema VKO primenom prethodnog algoritma dato ili kao duž u ravni ili kao jedinstvena tačka (korak 5 algoritma). Nakon rešavanja svakog jedno-kriterijumskog problema u Koraku 2, uzima se najbolje rešenje (koje maksimizuje funkciju cilja). Dakle, očigledno sledi sledeća teorema.

**Teorema 3.5.5.** [73] Neka je dat linearни problem višekriterijumske optimizacije (3.1.1), gde je  $\mathbf{x} \in \mathbb{R}^2$ . Za datu listu uređenih funkcija cilja  $f$  i datu listu ograničenja  $g$ , rešenje dobijeno Algoritmom 3.4 je Pareto optimalna tačka prosmatranog problema VKO (3.1.1).

**Dokaz.** Najpre, primetimo da primenom Algoritma 3.4 imamo niz dobijenih rešenja maksimalno  $l$  jedno-kriterijumskih optimizacionih problema. Označimo ove skupove rešenja sa  $R_i$   $i \leq l$ . Za svaki indeks  $1 \leq i \leq l$ , postoje dve mogućnosti:

- 1) skup  $R_i$  se sastoji od jedinstvene ekstremne tačke.
- 2) skup  $R_i$  predstavlja duž određenu dvema ekstremnim tačkama.

Očigledno je  $f_k(x^{(l)}) \geq f_k(x^{(k)})$  zadovoljeno za svako  $k = \overline{1, l-1}$ . Pretpostavimo suprotno, da rešenje  $x^{(l)}$  nije Pareto optimalna tečka problema (3.1.1). Tada postoji druga tačka  $x^*$ , takva da je

$$f_k(x^*) \geq f_k(x^{(l)}), \quad \forall k \in \{1, \dots, l\},$$

gde je najmanje jedna stroga nejednakost. Neka je na primer, za indeks  $t$  validna nejednakost  $f_t(x^*) > f_t(x^{(l)})$ . S obzirom da važi  $f_t(x^{(l)}) \geq f_t(x^{(t)})$ , nejednakost  $f_t(x^*) > f_t(x^{(t)})$  je zadovoljena. Ovo povlači da je  $x^* \notin \Omega'_P$ , pri čemu je

$$\Omega'_P = \Omega_P \cap \{x \mid f_1(x) \geq f_1(x^{(1)}), \dots, f_{t-1}(x) \geq f_{t-1}(x^{(t-1)})\}.$$

Dakle,  $x^* \in \Omega_P \setminus \Omega_{Pt}$ , što znači da za neko  $i \in \{1, \dots, k-1\}$  je zadovoljena nejednakost  $f_i(x^*) < f_i(x^{(i)})$ . Ovo je u kontradikciji sa polaznim tvrđenjem  $f_k(x^*) \geq f_k(x^{(k)})$ , za svako  $k = 1, \dots, l$ . Očigledno, rešenje  $x^{(l)}$  mora viti Pareto optimalno.  $\square$

S obzirom na varijante geometrijskog metoda za dve i tri promenljive, analogni algoritam sa Algoritmom 3.4 se može navesti za trodimenzionalni slučaj.

**Algoritam 3.5** Rešavanje linearogn problema višekriterijumske optimizacije (3.1.1), za slučaj tri nepoznate  $x_1, x_2, x_3$ .

### Modifikovani leksikografski metod 3D

**Ulaz:** Lista uređenih funkcija cilja  $f$  i lista ograničenja  $g$ .

- 1: Postaviti vrednost indeksa  $i := 1$  i početnu vrednost skupa  $R := \emptyset$ . Uzeti prvu funkciju  $f_1$  kao funkciju cilja, a listu  $g$  kao skup ograničenja.
- 2: Rešiti optimizacioni problem jedne funkcije cilja za datu funkciju cilja i ograničenja, koristeći geometrijski metod. Sačuvati rešenje u skupu  $R$ .
- 3: Ako se skup  $R$  sastoji od jedinstvene tačke, ići na Korak 5. U suprotnom, ako je skup rešenja  $R$  određen kao  $\lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)}$ , gde je  $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1$ ,  $\lambda_1 + \lambda_2 + \lambda_3 = 1$ , gde su najmanje dva koeficijenta različita od nule, ići na Korak 4.
- 4: Ako je  $i < l$  tada: uvećati indeks  $i$  za 1, uzeti  $f_i$  kao funkciju cilja a skup  $R$  kao skup ograničenja, ići na Korak 2. Ako je  $i = l$ , ići na Korak 5.
- 5: **Rezultat:** Skup  $R$  kao rešenje polaznog problema.

**Teorema 3.5.6.** [73] Neka je dat linearni problem višekriterijumske optimizacije (3.1.1), gde je  $\mathbf{x} \in \mathbb{R}^3$ . Za datu listu uređenih funkcija cilja  $f$  i datu listu ograničenja  $g$ , rešenje dobijeno Algoritmom 3.5 je Pareto optimalna tačka prosmatranog problema VKO (3.1.1).

**Dokaz.** Dokaz je sličan dokazu Teoreme 3.5.5, sa razlikom da se za svako  $1 \leq i \leq l$ , skup rešenja  $R_i$  može sastojati od jedinstvene tačke, duži ili dela ravnih. Skup rešenje je dat kao linearna kombinacija tri optimalne tačka:  $\lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)}$ , where  $0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1$ ,  $\lambda_1 + \lambda_2 + \lambda_3 = 1$ .  $\square$

Ovde navodimo i implementaciju modifikovanog leksikografskog metoda koristeći geometrijski metod linearog programiranja.

```

LexicModif[f_List, g_List] :=
Module[{l = Length[f], var = Variables[f], r, q, q1, q2, i, m1, m2},
r = LinProgGraphicalM[f[[1]], g];
Print[r];
q = RegionPlot[g, {var[[1]], 0, 10}, {var[[2]], 0, 10}, AspectRatio -> 1];
For[i = 2, (i <= l) && (Length[r] == 2), i++, m1 = ReplaceAll[f[[i]], r[[1]]];
m2 = ReplaceAll[f[[i]], r[[2]]];
If[m1 > m2, r = {r[[1]]}];
If[m2 > m1, r = {r[[2]]}];
If[Length[r] > 1, Print["Pareto optimal solutions have the form: [Lambda]*",
r[[1]], "+(1-[Lambda])*", r[[2]], ", 0<=[Lambda]<=1"];
q1 = Graphics[{Thick, Line[r]}];
Print["Pareto optimal solution is: ", r[[1]]];
q2 = ListPlot[r, PlotStyle -> {PointSize[0.03]}, DisplayFunction -> Identity];
Return[{ReplaceAll[f, r[[1]]], r[[1]]}];]

```

Ovde koristimo funkciju `RegionPlot` [113], radi reprezentacije skupa dopustivih rešenja. Ovaj se skup nakon svakog koraka smanjuje na skup optimalnih tačaka iz prethodne iteracije algoritma. Ova ideja ubrzava izvršenje programas.

### 3.5.3 Numerički eksperimenti

Ukoliko je ciljna funkcija od dve ili tri promenljive, procedura geometrijskog metoda se može primeniti [68]. Tada je optimalno rešenje dobijeno iscrtavanjem grafika modifikovane funkcije cilja  $f(x_1, x_2) = 0$  i njenim paralelnim pomeranjem u smeru vektora gradijenta.

Ako su nađena sva različita rešenja, tada su sve tačke na duži između njih takođe optimalna rešenja. Dakle, modifikacija leksikografskog metoda se ne ograničava samo na nalaženje jedne Pareto optimalne tačke.

**Primer 3.5.2.** Rešiti sledeći linearni problem VKO:

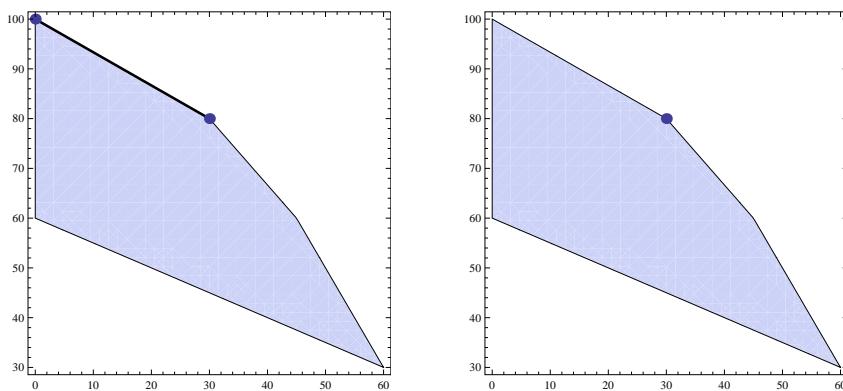
$$\begin{aligned}
\text{Max. : } & [8x + 12y, 14x + 10y, x + y] \\
\text{Such that : } & 2x + y \leq 150 \\
& 2x + 3y \leq 300 \\
& 4x + 3y \leq 360 \\
& x + 2y \geq 120 \\
& x, y \geq 0
\end{aligned}$$

koristeći navedeno leksikografsko uređenje kao u listi.

Najpre posmatramo prvu funkciju po važnosti:  $f_1 = 8x + 12y$ . Primenom grafičke metode imamo da svaka tačka na duži  $\lambda\{30, 80\} + (1-\lambda)\{0, 100\}$ ,  $0 \leq \lambda \leq 1$ , maksimizuje funkciju cilja  $f_1$ . Ovaj rezultat je ilustrovan na Slici 3.5.1, sa leve strane.

Nakon toga, ispitujemo samo tačke na dobijenoj duži, koje maksimizuju drugu funkciju cilja po leksikografskom uređenju:  $f_2 = 14x + 10y$ . Kako prava  $f_2 = 0$  nije paralelna sa dobijenom duži, to maksimalna tačka mora biti između krajnjih tačaka duži. Testiranjem, zaključujemo da je to tačka  $\{30, 80\}$ . Dalje izračunavanje ovde staje jer je rešenje problema jedno-kriterijumske optimizacije jedinstvena tačka.

Dakle, rešenje ovog problema je  $\{30, 80\}$  (videti desnu stranu na Slici 3.5.1), u kojem slučaju tri polazne funkcije cilja imaju respektivno sledeće vrednosti: 1200, 1220, 110.



Slika 3.5.1. Rezultati iz prve i druge iteracije Algoritma 3.4.

Sledeća instrukcija je korišćena za rešavanje ovog problema:

```
LexicModif[{8x + 12y, 14x + 10y, x + y},  
2x + y<= 150, 2x + 3y<= 300, 4x + 3y<= 360, x + 2y>= 120, x>= 0, y>= 0]
```

Pritom je, dakle, dobijen sledeći rezultat:

$$\{\{1200, 1220, 110\}, \{x \rightarrow 30, y \rightarrow 80\}\}.$$

**Primer 3.5.3.** Rešiti trodimenzionalni problem VKO:

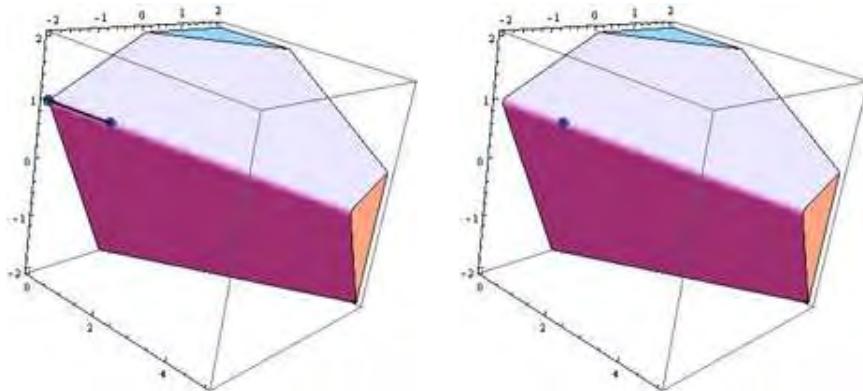
$$\begin{aligned} \text{Max. : } & [x - 2y - z, x + y + z] \\ \text{Such that : } & -x + 2y + z \geq -3, \\ & x - y + 2z \leq 4, \\ & x \geq 0. \end{aligned}$$

Geometrijskom metodom je dobijeno da svaka tačka duži  $\lambda\{1, -\frac{7}{5}, \frac{4}{5}\} + (1 - \lambda)\{0, -2, 1\}$ ,  $0 \leq \lambda \leq 1$  maksimizuje funkciju cilja  $f_1 = x - 2y - z$ , što je ilustrovano na Slici 3.5.2 sa leva. Dalje, pretražuju se tačke dobijene duži koje maksimizuju drugu po uređenju funkciju cilja:  $f_2 = x + y + z$ .

Sledeća komanda je primenjena na rešavanje polaznog problema:

```
LexicModif[{x-2y-z, x+y+z}, {-x+2y+z+3>=0, x-y+2z<=4, x>=0}]
```

Rešenje problema je tačka  $\{1, -\frac{7}{5}, \frac{4}{5}\}$  (desna strana Slike 3.5.2), u kojoj zadate tri funkcije cilja imaju sledeće vrednosti, respektivno:  $\{3, \frac{2}{5}\}$ .



Slika 3.5.2. Prve dve iteracije Algoritma 3.5 ilustrovane u MATHEMATICA.

## 3.6 Analiza drugih metoda višekriterijumske optimizacije

### 3.6.1 Relaksirani leksikografski metod

Hijerarhijski metod je modifikacija leksikografskog metoda, u kome se koriste ograničenja oblika [60]

$$Q_j(\mathbf{x}) \geq \left(1 + \frac{\delta_j}{100}\right) Q_j(\mathbf{x}_j^*).$$

Relaksacija se sastoji u povećanju desne strane ograničenja za procenat  $Q_j(\mathbf{x}_j^*) \cdot \delta_j$ . Varijanjem parametra  $\delta_j$  mogu se generisati različite Pareto optimalne tačke [50]. Još jedna varijacija leksikografskog metoda je uvedena u [107], gde su ograničenja jednaka

$$Q_j(\mathbf{x}) \geq Q_j(\mathbf{x}_j^*) - \delta_j.$$

U ovom slučaju,  $\delta_j \leq 100$  jesu pozitivne tolerancije definisane od strane DO.

Relaksirani leksikografski metod je iterativni postupak u kome se rešavaju jednokriterijumski zadaci optimizacije. Prepostavlja se da su od strane DO dati prioriteti kriterijuma i da su u skladu sa njima dodeljeni indeksi kriterijumima. U relaksiranoj leksikografskoj metodi se po svakom od  $p$  kriterijuma rešava jednokriterijumski zadatak optimizacije. Pri tome se u narednoj iteracijine postavlja kao ograničenje zahtev da rešenje bude optimalno po kriterijumu višeg prioriteta, već se ono relaksira tako da se zahteva da rešenje bude u okolini optimalnog rešenja dobijenog u prethodnoj iteraciji. Nataj način, svaki kriterijum utiče na konačno rešenje.

DO zadaje redosled kriterijuma po značajnosti. Pored toga, svakom kriterijumu, uzimajući poslednji, dodeljuje se vrednost  $\delta_k$ ,  $k = 1, \dots, l - 1$ , za koju kriterijum višeg prioriteta sme da odstupi od svoje optimalne vrednosti. Metoda obuhvata izvršavanje  $l$  koraka opisanih sledećim algoritmom.

**Algoritam 3.6** Relaksirani leksikografski metod

**Ulaz:** Problem VKO (3.1.1) sa uredjenom listom funkcija cilja  $Q_1, \dots, Q_l$ .

1: Rešiti problem

$$\begin{array}{ll} \max & Q_1(\mathbf{x}) \\ \text{p.o.} & \mathbf{x} \in \mathbf{X} \end{array}$$

Rešenje ovog problema označimo sa  $Q_1^*$ .

2: Za svako  $p = 2, l$  rešiti jednokriterijumske optimizacije.

$$\begin{array}{ll} \max & Q_p(\mathbf{x}) \\ \text{p.o.} & \mathbf{x} \in \mathbf{X} \\ & Q_j(\mathbf{x}) \geq Q_j^* - \delta_j, \quad j = 1, \dots, p-1, \end{array}$$

pri čemu je  $Q_j^*$  rešenje  $j$ -og po redu problema jednokriterijumske optimizacije.

3: **Rezultat:** Rešenje  $Q_l^*$  kao rešenje polaznog problema.

Za rešenje polaznog modela se usvaja rezultat dobijen u poslednjem koraku, a vrednosti funkcija cilja za dobijeno rešenje se moraju posebno računati.

Rešenje dobijeno ovim metodom obezbeđuje slabi Pareto optimum, a ako je rešenje jedinstveno, ono je i Pareto optimalno. Ako je dopustiva oblast konveksna, podešavanjem parametara  $\delta_k, k = 1, \dots, l-1$ , može se dobiti bilo koje Pareto optimalno rešenje.

**Teorema 3.6.1.** *Rešenje dobijeno relaksiranim leksikografskim metodom je slabi Pareto optimum problema višekriterijumske optimizacije.*

**Dokaz.** Prepostavimo suprotno, da rešenje  $\mathbf{x}^*$  dobijeno relaksiranim leksikografskim metodom nije slabo Pareto optimalno. To znači da postoji neko  $\mathbf{x} \in S$  tako da za svako  $i = 1, \dots, l$  važi  $Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*)$ . Kako je  $Q_i(\mathbf{x}^*) = Q_i^*$  tim pre važi  $Q_i(\mathbf{x}) \geq Q_i^* - \delta_i$ ,  $i = 1, \dots, l-1$ , pa kako je  $Q_k(\mathbf{x}) > Q_k(\mathbf{x}^*)$  dobijamo kontradikciju sa prepostavkom da je  $\mathbf{x}^*$  rešenje dobijeno relaksiranim leksikografskim metodom.  $\square$

**Teorema 3.6.2.** *Ako se relaksiranim leksikografskim metodom dobija jedinstveno rešenje, ono je Pareto optimalno.*

**Dokaz.** Prepostavimo suprotno, da jedinstveno rešenje  $\mathbf{x}^*$  dobijeno relaksiranim leksikografskim metodom nije Pareto optimalno. To znači da postoji neko  $\mathbf{x} \in S$  tako da za svako  $i = 1, \dots, l$  važi  $Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*)$  pri čemu za neko  $j \in \{1, \dots, l\}$  važi stroga nejednakost. Razlikujemo dva slučaja:

- 1)  $Q_k(\mathbf{x}) > Q_k(\mathbf{x}^*)$ . Kako iz prepostavljenog sledi da je  $Q_i(\mathbf{x}) \geq Q_i^* - \delta_i$ ,  $i = 1, \dots, k-1$ , zaključujemo da  $\mathbf{x}^*$  nije rešenje dobijeno u koraku  $k$  gore opisanog algoritma tj. da  $\mathbf{x}^*$  nije rešenje dobijeno relaksiranim leksikografskim metodom, što je kontradikcija.
- 2)  $Q_k(\mathbf{x}) = Q_k(\mathbf{x}^*)$ . Kao i pod 1), važi  $Q_i(\mathbf{x}) \geq Q_i^* - \delta_i$ ,  $i = 1, \dots, k-1$ , pa dobijamo da je  $\mathbf{x}$  rešenje dobijeno relaksiranim leksikografskim metodom. Međutim, kako je  $\mathbf{x}^*$  jedinstveno rešenje zaključujemo da mora biti  $\mathbf{x} = \mathbf{x}^*$ . Tada je i  $Q_j(\mathbf{x}) = Q_j(\mathbf{x}^*)$ , što je kontradikcija.

Dakle,  $\mathbf{x}^*$  je zaista Pareto optimalno rešenje.  $\square$

U praktičnoj primeni leksikografskog metoda vrednosti  $\delta_i$  moraju biti pažljivo odabrane. Ukoliko se ovim metodom dobije više rešenja bilo bi dobro smanjiti te vrednosti.

**Primer 3.6.1.** Primenom relaksiranog leksikografskog metoda rešimo sledeći zadatak VKO (funkcije cilja su relaksirane po prioritetu)

$$\begin{aligned} \max \quad & \{Q_1(\mathbf{x}), Q_2(\mathbf{x}), Q_3(\mathbf{x})\} \\ \text{p.o.} \quad & 2x_1 + x_2 \leq 6, \quad x_1 + 3x_2 \leq 6, \quad x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

gde je

$$\begin{aligned} Q_1(\mathbf{x}) &= x_1 + 4x_2, \quad \delta_1 = 1 \\ Q_2(\mathbf{x}) &= x_1, \quad \delta_2 = 1 \\ Q_3(\mathbf{x}) &= x_1 + x_2. \end{aligned}$$

**Korak 1.** Rešiti problem

$$\begin{aligned} \max \quad & Q_1(x) = x_1 + 4x_2 \\ \text{p.o.} \quad & 2x_1 + x_2 \leq 6, \quad x_1 + 3x_2 \leq 6, \quad x_1, x_2 \geq 0. \end{aligned}$$

Ovaj zadatak ima jedinstveno optimalno rešenje u tački  $x^{1*} = (0, 2)$ , pri čemu je  $Q_1^* = 8$ .

**Korak 2.** Rešiti problem

$$\begin{aligned} \max \quad & Q_2(\mathbf{x}) = x_1 \\ \text{p.o.} \quad & 2x_1 + x_2 \leq 6, \quad x_1 + 3x_2 \leq 6, \quad x_1, x_2 \geq 0, \quad x_1 + 4x_2 \geq 7. \end{aligned}$$

Dobija se rešenje  $x^{2*} = (2.43, 1.14)$ ,  $Q_2^* = 2.43$

**Korak 3.** Rešiti problem

$$\begin{aligned} \max \quad & Q_3(\mathbf{x}) = x_1 + 4x_2 \\ \text{p.o.} \quad & 2x_1 + x_2 \leq 6, \quad x_1 + 3x_2 \leq 6, \quad x_1, x_2 \geq 0, \quad x_1 + 4x_2 \geq 7, \quad x_1 \geq 1.43. \end{aligned}$$

Rešenje ovog zadatka se usvaja kao konačno. To je rešenje  $x_1^* = 3.6$ ,  $x_2^* = 1.2$ . Vrednosti funkcija cilja u ovoj tački su  $Q^* = (7.2, 2.4, 3.6)$ .

Relaksirani leksikografski metod je veoma osetljiv na izbor koeficijenata  $\delta_j$ , tako da se "pogrešnim" izborom njegove vrednosti mogu dobiti neprihvatljiva rešenja. Tako u prethodnom primeru imamo slučaj da se konačno rešenje poklapa sa marginalnim rešenjem funkcije cilja koja ima najniži prioritet, dok je vrednost najznačajnijeg kriterijuma smanjena. Kod primene ove metode se preporučuje da DO kritički preispita dobijena rešenja, uporedi ih sa marginalnim i da po potrebi koriguje zadate koeficijente.

Jedan od načina za implementaciju relaksiranog leksikografskog metoda dat je funkcijom **MultiRelax()** [83]:

**Ulagne veličine:**

$q$  - lista ciljnih funkcija;

$parcf$  - parametri ciljnih funkcija;

$constr$  - lista ograničenja.

```

MultiRelax[q_List, constr_List, var_List, delta_List] :=
Module[{res={}, s, f=constr, l=Length[q], i},
  For[i=1, i<=l, i++,
    s = Maximize[q[[i]], f, var];      (* 2 *)
    If[i<1, AppendTo[f, q[[i]]>=First[s]-delta[[i]] ] ] ; (* 3,2 *)
    AppendTo[res, s];
  ];
  Print[res]; Return[{q/.Last[s],Last[s]}];
]

```

### 3.6.2 Metod $\varepsilon$ -ograničenja

U ovom metodu DO izdvaja kriterijum  $Q_p(\mathbf{x})$ ,  $1 \leq p \leq l$  koji ima najviši prioritet i njega maksimizira, dok ostale funkcije cilja ne smeju imati vrednosti manje od unapred zadatih  $\varepsilon_k$ ,  $k = 1, \dots, l$ ,  $k \neq p$ . Sve druge ciljne funkcije se koriste da se kreiraju dodatna ograničenja oblika  $l_i \leq Q_i(\mathbf{x}) \leq \varepsilon_i$ ,  $i = 1, \dots, l$ ,  $i \neq p$  [37]. U ovoj relaciji,  $l_i$  i  $\varepsilon_i$  jesu donja i gornja granica ciljne funkcije  $Q_i(\mathbf{x})$ , respektivno.

Haimes u [29] je uveo  $\varepsilon$ -constraint (ili trade-off) metod (metod  $\varepsilon$ -ograničenja), u kome su sve vrednosti  $l_i$  isključene. Sistematsko variranje vrednosti  $\varepsilon_i$  produkuje skup Pareto optimalnih rešenja [37]. Međutim, nepodesna selekcija vrednosti  $\varepsilon \in \mathbf{R}^k$  može da proizvede rešenje koje ne ispunjava ograničenje. Opšte pravilo za selektovanje  $\varepsilon_i$  je sledeće ([7])

$$Q_p(\mathbf{x}_i^*) \leq \varepsilon_i \leq Q_i(\mathbf{x}_i^*),$$

gde je  $\mathbf{x}_i^*$  tačka koja maksimizira ciljnu funkciju  $Q_i(\mathbf{x})$ .

Neophodno je da se reši sledeći problem

$$\begin{aligned} \max & \quad Q_p(\mathbf{x}) \\ \text{p.o.} & \quad x \in \mathbf{X}, \\ & \quad Q_i(\mathbf{x}) \geq \varepsilon_i, \quad i = 1, \dots, l, \quad i \neq p, \end{aligned}$$

gde je  $Q_p(\mathbf{x})$  odabrani kriterijum sa najvećim prioritetom i  $\varepsilon_i$  su odabrani realni brojevi. Ako optimalno rešenje ne postoji, neophodno je da se smanji vrednost za  $\varepsilon_i$ .

**Teorema 3.6.3.** *Rešenje problema  $\varepsilon$  ograničenja je slabo Pareto optimalno rešenje početnog problema VKO.*

**Dokaz.** Prepostavimo da je  $\mathbf{x}^*$  rešenje problema  $\varepsilon$  ograničenja, i da ono nije slabi Pareto optimum. Dakle, postoji  $\mathbf{x} \in S$  tako da je  $Q_j(\mathbf{x}) > Q_j(x^*) \wedge Q_i(\mathbf{x}) > Q_i(x^*) \geq \varepsilon_i$  za svako  $i = 1, \dots, l$ ,  $i \neq j$ , odakle sledi da  $\mathbf{x}^*$  nije rešenje problema  $\varepsilon$  ograničenja, što je kontradikcija. Zaključujemo da je rešenje problema  $\varepsilon$  ograničenja slabi Pareto optimum.  $\square$

**Teorema 3.6.4.** *Tačka  $\mathbf{x}^* \in S$  je Pareto optimalno rešenje polaznog problema VKO ako za svako  $j = 1, \dots, l$  tačka  $\mathbf{x}^*$  jeste rešenje problema  $\varepsilon$  ograničenja pri čemu je  $\varepsilon_i = Q_i(\mathbf{x}^*)$ ,  $i = 1, \dots, l$ ,  $i \neq j$ .*

**Dokaz.** Uslov je dovoljan jer iz njega sledi da ni za jedno  $j = 1, \dots, l$  ne postoji  $\mathbf{x} \in S$  tako da je  $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*) \wedge Q_i(\mathbf{x}) \geq \varepsilon_i = Q_i(\mathbf{x}^*)$ ,  $i \neq j$ , tj. sledi da je  $\mathbf{x}^*$  Pareto optimum.

Prepostavimo, sa druge strane, da je  $\mathbf{x}^*$  Pareto optimum, a da za neko  $j$  ono nije rešenje problema  $\varepsilon$  ograničenja sa  $\varepsilon_i = Q_i(\mathbf{x}^*)$ ,  $i \neq j$ . Dakle, postoji  $\mathbf{x} \in S$  tako da važi  $Q_j(\mathbf{x}) > Q_j(\mathbf{x}^*)$  i  $Q_i(\mathbf{x}) \geq Q_i(\mathbf{x}^*) = \varepsilon_i$ ,  $i \neq j$ . Dolazimo u kontradikciju sa pretpostavkom da je  $\mathbf{x}^*$  Pareto optimalno rešenje. Dakle, uslov je i potreban.  $\square$

Primetimo da iz ove teoreme sledi da se metodom  $\varepsilon$  ograničenja može dobiti proizvoljno Pareto optimalno rešenje.

**Teorema 3.6.5.** *Neka je vektor  $\varepsilon$  proizvoljan. Tada, ako problem  $\varepsilon$  ograničenja ima jedinstveno rešenje onda je ono Pareto optimalno rešenje polaznog problema VKO.*

**Dokaz.** Neka je  $\mathbf{x}^*$  jedinstveno rešenje problema  $\varepsilon$  ograničenja. Prepostavimo da  $\mathbf{x}^*$  nije Pareto optimalno tj. da postoji  $\mathbf{x}'$  iz  $S$  tako da važi  $Q_i(\mathbf{x}') \geq Q_i(\mathbf{x}^*)$   $i = 1, \dots, l$ , pri čemu važi stroga jednakost za  $i = l$ . Ako je  $l = j$ , tada dobijamo  $Q_j(\mathbf{x}') > Q_j(\mathbf{x}^*) \wedge Q_i(\mathbf{x}') \geq Q_i(\mathbf{x}^*) \geq \varepsilon_i$ ,  $i = 1, \dots, l$ ,  $i \neq j$ , tj. dobijamo kontradikciju sa pretpostavkom da je  $\mathbf{x}^*$  rešenje problema  $\varepsilon$  ograničenja. Ako je  $l \neq j$ , tada je  $Q_i(\mathbf{x}') \geq Q_i(\mathbf{x}^*) \geq \varepsilon_i$ ,  $i = 1, \dots, l$ ,  $i \neq j$  i  $Q_j(\mathbf{x}') \leq Q_j(\mathbf{x}^*)$ . Međutim kako je  $\mathbf{x}^*$  jedinstveno rešenje te iz  $Q_i(\mathbf{x}') \geq \varepsilon_i$ ,  $i = 1, \dots, l$ ,  $i \neq j$  sledi da mora biti  $Q_j(\mathbf{x}') < Q_j(\mathbf{x}^*)$ . Ponovo dolazimo do kontradikcije pa zaključujemo da je  $\mathbf{x}^*$  zaista Pareto optimalno rešenje polaznog problema VKO.  $\square$

U slučaju da problem  $\varepsilon$  ograničenja nema rešenja trebalo bi povećati vrednosti granica  $\varepsilon_i$ . Naravno, nijedno  $\varepsilon_i$  ne sme biti manje od idealne vrednosti funkcije cilja  $Q_i$ . U slučaju da DO sam izabere prioritetu funkciju  $Q_j$  i konstante  $\varepsilon_i$  na ovaj metod se može gledati kao na a priori metod. Međutim, on u tom slučaju ne može biti siguran da će biti zadovoljan dobijenim rešenjem. Zbog toga je pristup obično drugačiji. Postupak rešavanja problema  $\varepsilon$  ograničenja se ponavlja za razne vrednosti granica  $\varepsilon_i$  i za razne funkcije  $Q_j$  kao prioritetne, sve dok DO ne bude zadovoljan dobijenim rešenjem.

Da bismo dokazali da je rešenje dobijeno metodom ograničenja Pareto optimalno moramo dokazati da je ono jedinstveno ili da je rešenje  $l$  različitih problema za svaku od funkcija  $Q_j$ , što nije nimalo jednostavno. Ovo je glavna mana ove metode.

Sledi opis implementacije prema [83].

Ulagni parametri:

$q_-$ ,  $constr_-$ ,  $var_-$ : unutrašnja forma problema;  
 $p_-$ : redni broj ciljne funkcije sa najvećim prioritetom.

```
MultiEps[q_List, constr_List, var_List, p_Integer] :=
Module[{s, f = constr, i, j, k, l = Length[q], eps, lb = {}, ub = {}},
For[i = 1, i <= l, i++,
  xi = Maximize[q[[i]], constr, var];
  AppendTo[lb, q[[p]]/.First[Rest[xi]]];
  AppendTo[ub, q[[i]]/.First[Rest[xi]]];
];
For[i = 1, i <= l, i++,
  For[eps = lb[[i]], eps <= ub[[i]], eps = eps + 0.1,
```

```

f = constr;
For[k = 1, k <= l, k++,
  If[k != p, AppendTo[f, q[[k]]>=eps ] ] (* 3 *)
];
s = Maximize[q[[p]], f, var]; (* 2, 1 *)
]
]

```

**Primer 3.6.2.** Rešiti problem:

$$\begin{aligned} \text{max: } & Q_1(\mathbf{x}) = x_1, Q_2(\mathbf{x}) = x_2 \\ \text{p.o.: } & 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1. \end{aligned}$$

Prepostavimo da prvi kriterijum ima najveći prioritet. Ovaj problem se rešava sledećim pozivom funkcije *MultiEps*:

```
In[5]:= MultiEps[{x1,x2},{x1>=0,x2>=0,x1<=1,x2<=1},{x1,x2},{0.5},1]
```

U slučaju kada drugi kriterijum uzima vrednosti veće ili jednake od 0.5 ( $\text{eps} = 0.5$ ) neophodno je da se reši sledeći jednokriterijumski problem:

$$\begin{aligned} \text{max } & Q_1(\mathbf{x}) = x_1 \\ \text{p.o. } & 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, x_2 \geq 0.5. \end{aligned}$$

Njegova unutrašnja forma, korišćena u MATHEMATICA funkciji *Maximize* je

$$x1, \{x1>=0, x2>=0, x1<=1, x2<=1, x2>=0.5\}, \{x1, x2\}$$

Rešenje problema je

$$\{\{1., 0.5\}, \{x1 \rightarrow 1., x2 \rightarrow 0.5\}\}$$

i nije Pareto optimalno. Pareto optimalno rešenje se generiše u slučaju  $\text{eps} = 1$ , i jednako je

$$\{\{1., 1\}, \{x1 \rightarrow 1, x2 \rightarrow 1\}\}$$

### 3.6.3 Metodi rastojanja

Metodi rastojanja čine grupu za rešavanje zadataka VKO čija je osnovna ideja da se u kriterijumskom prostoru traži tačka koja je najbliža nekoj unapred određenoj tački koja se želi dostići ili ka kojoj treba težiti ako ona nije dopustiva. Drugim rečima, minimizira se rastojanje između željene tačke i dopustive oblasti. Razlike između metoda ove grupe potiču od toga kako se željena tačka određuje, na koji način se rastojanje od nje "meri", da li se uvode težinski koeficijenti, itd.

U opštem slučaju, DO za svaki od  $l$  kriterijuma zadaje željene vrednosti ili određuje način kako će se one izračunati. Na taj način se u  $l$ -dimenzionalnom kriterijumskom prostoru definiše tačka

$$f^z = (f_1^z, \dots, f_l^z)$$

koja po pravilu ne pripada dopustivoj oblasti  $S$ . U slučaju

$$f^z \in S$$

zadatak se rešava rešavanjem sistema jednačina koje su definisane skupom ograničenja.

U metodu rastojanja rešava se sledeći opšti zadatak

$$\begin{array}{ll} \min & d(f^z - f(x)) \\ \text{p.o.} & \mathbf{x} \in \mathbf{X} \end{array}$$

gde  $d(\cdot, \cdot)$  označava rastojanje definisano pogodnom metrikom.

U kontekstu merenja rastojanja u kriterijumskom prostoru ovde ćemo ponoviti definicije metrika koje se najčešće koriste u metodama rastojanja

$$\begin{aligned} l_1 \text{ (pravougaona) metrika: } d_{l_1}(f^z, f(\mathbf{x})) &= \sum_{k=1}^l |f_k^z - f_k(\mathbf{x})|, \\ l_2 \text{ (Euklidova) metrika: } d_{l_2}(f^z, f(\mathbf{x})) &= \sqrt{\sum_{k=1}^l (f_k^z - f_k(\mathbf{x}))^2}, \\ l_\infty \text{ (Čebiševljeva) metrika: } d_{l_\infty}(f^z, f(\mathbf{x})) &= \max_{1 \leq k \leq l} \{|f_k^z - f_k(\mathbf{x})|\}. \end{aligned}$$

Kriterijumima je moguće dodeliti težinske koeficijente, tako da prethodne formule za rastojanje između željenog i traženog rešenja dobijaju oblik

$$d_{l_1}(f^z, f(x)) = \sum_{k=1}^l w_k |f_k^z - f_k(\mathbf{x})|, \text{ za pravougaonu metriku,}$$

$$d_{l_2}(f^z, f(x)) = \sqrt{\sum_{k=1}^l w_k (f_k^z - f_k(\mathbf{x}))^2}, \text{ za Euklidovu metriku,}$$

$$d_{l_\infty}(f^z, f(\mathbf{x})) = \max_{1 \leq k \leq l} \{w_k |f_k^z - f_k(\mathbf{x})|\}, \text{ za Čebiševljevu metriku.}$$

Bez obzira na oblik polaznih funkcija cilja, zadaci minimizacije rastojanja su problemi nelinearnog programiranja, za koje, u opštem slučaju, nije jednostavno pronaći optimalno rešenje. Izuzetno se u slučaju  $l_1$  metrike i linearnih funkcija cilja i ograničenja zadatak minimizacije rastojanja može formulisati kao problem LP što ćemo objasniti na sledećem primeru.

**Napomena 3.6.1.** *Objašnjene transformacije su osnova za formulaciju zadatka višekriterijumskog linearog programiranja (VLP) i oblik koji se naziva ciljno programiranje. Prema tome, zadatak ciljnog programiranja je poseban oblik zadatka metoda rastojanja u VKO. Dodatno je moguće kriterijumima dodeliti prioritete dostizanja željenih vrednosti i težinske koeficijente.*

Rešenje zadatka VKO dobijeno metodom rastojanja, kada skupu vrednosti kriterijuma ne pripada željna vrednost, predstavlja slab Pareto optimum, a ako je jedinstveno, onda je i Pareto optimalno.

Metod rastojanja (ciljno programiranje ili goal programming method) je razvijen u [9], [10] [38]. Za svaku od  $l$  kriterijumskih funkcija odabiraju se ciljne vrednosti, ili se izračunavaju pomoću funkcije *Maximize*. Na taj način, generiše se  $l$ -dimenzionalna tačka  $b(\mathbf{x}) = (b_1, \dots, b_l)$ , koja sadrži ciljne vrednosti za ciljne funkcije. Potom se minimizira

rastojanje između tačke  $b(\mathbf{x})$  i  $Q(\mathbf{x}) = (q_1(\mathbf{x}), \dots, q_l(\mathbf{x}))$ . Drugim rečima, rešava se sledeći opšti zadatak

$$\begin{array}{ll} \min & d(b - Q(\mathbf{x})) \\ \text{p.o.} & \mathbf{x} \in \mathbf{X}, \end{array}$$

gde je  $d(\cdot, \cdot)$  metrika koja definiše rastojanje između tačaka  $b$  i  $Q(\mathbf{x})$ . Uzmimo, na primer metriku definisanu  $L_1$ -normom

$$d(b - Q(\mathbf{x})) = \sum_{i=1}^l |b_i - q_i(\mathbf{x})|.$$

U ovom slučaju, prirodno je da se koriste zamene  $y_i = b_i - q_i(\mathbf{x})$ ,  $i = 1, \dots, l$ . Koristeći dodatne smene  $y_i = \$[i]^+ - \$[i]^-$ ,  $i = 1, \dots, l$ , u kojima je  $\$[i]^+, \$[i]^- \geq 0$ , (i tada je  $|y_i| = \$[i]^+ + \$[i]^-$ ,  $i = 1, \dots, l$ ), dobija se sledeći jednokriterijumski optimizationi problem

$$\begin{array}{ll} \min & \sum_{i=1}^l (\$[i]^+ + \$[i]^-) \\ \text{p.o.} & \mathbf{x} \in \mathbf{X} \\ & Q_i(\mathbf{x}) - \$[i]^+ + \$[i]^- = b_i, \quad i = 1, \dots, l \\ & \$[1]^+ \geq 0, \dots, \$[l]^+ \geq 0, \$[1]^- \geq 0, \dots, \$[l]^- \geq 0. \end{array}$$

Lista ograničenja se transformiše ciklusom:

```
For[i=1,i<=l,i++,
  AppendTo[constr,q[[i]]-\$[2i-1]+\$[2i]==b[[i]]];
];
For[i=1,i<=2l,i++, AppendTo[constr,\$[i]>=0] ];
```

Odgovarajuća jednokriterijumska ciljna funkcija se generiše i minimizira izrazom

```
s=Minimize[Array[\$,2l,1,Plus],constr,Union[Variables[Array[\$,2l,1]],var]];
```

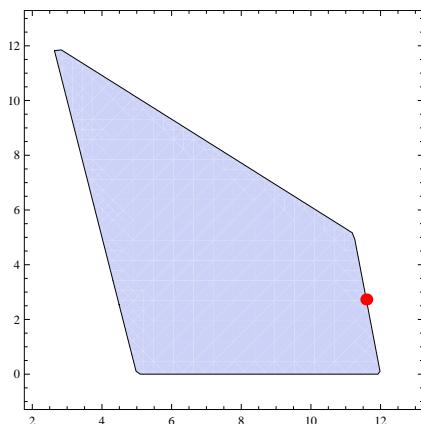
Standardna funkcija `Variables[expr]` daje listu nezavisnih promenljivih u izrazu `expr`. Izraz `Union[l1, l2, ...]` daje sortiranu listu različitih elemenata koji se pojavljuju u bilo kom od izraza  $l_i$  [113].

**Primer 3.6.3.** Rešiti sledeći zadatak VLP metodom rastojanja sa pravougaonom metrikom, ako se želi dostići idealna tačka.

$$\begin{array}{ll} \max & [Q_1(\mathbf{x}), Q_2(\mathbf{x}), Q_3(\mathbf{x})] \\ \text{p.o.} & 35x_1 + 7x_2 \geq 175 \\ & 136x_1 + 170x_2 \leq 2400 \\ & 20x_1 + 3x_2 \leq 240 \\ & x_1, x_2 \geq 0 \end{array}$$

gde su ciljne funkcije

$$Q_1(\mathbf{x}) = 4x_1 + 5x_2, \quad Q_2(\mathbf{x}) = x_1 + x_2, \quad Q_3(\mathbf{x}) = 40x_1 + 6x_2.$$



Slika 3.6.1. Ilustracija skupa dopustivih rešenja iz primera.

U ovom zadatku DO je odredio da su željene vrednosti kriterijumske funkcija idealne vrednosti tih funkcija. Zato je najpre potrebno odrediti marginalna rešenja i odgovarajuće idealne vrednosti funkcije

$f_1^* = 60$  za višestruko rešenje koje pripada duži **AB** sa krajnjim tačkama

$$x_1^{(1)*} = 3.09, \quad x_2^{(1)*} = 9.52$$

i

$$x_1^{(1)**} = 11.59, \quad x_2^{(1)**} = 2.73.$$

$f_2^* = 14.32$  za :  $x_1^{(2)*} = 11.59, x_2^{(2)*} = 2.73$  (tačka **B**),

$f_3^* = 480$  za višestruko rešenje koje pripada duži **BC** sa krajnjim tačkama

$$x_1^{(3)**} = 11.59, \quad x_2^{(3)**} = 2.73$$

i

$$x_1^{(3)*} = 12, \quad x_2^{(3)*} = 0.$$

Zadatak VKO prilagođen za rešavanje metodom rastojanja sada ima oblik

$$\min \Phi(\mathbf{x}) = |f_1^* - f_1(\mathbf{x})| + |f_2^* - f_2(\mathbf{x})| + |f_3^* - f_3(\mathbf{x})|$$

pri istim ograničenjima.

Pošto sve kriterijume trebamo maksimizovati, a po definiciji marginalnih rešenja je  $f_k^* \geq f_k(x)$ , može se staviti da je

$$|f_k^* - f_k(\mathbf{x})| = f_k^* - f_k(\mathbf{x}), \quad k = 1, 2, 3.$$

Tada se umesto polaznog zadatka može rešavati zadatak minimizacije funkcije

$$\Phi(\mathbf{x}) = 554.32 - 45x_1 - 12x_2,$$

pri istim ograničenjima. Pokazaćemo kako se ovaj zadatak rešava opštim postupkom koji koristi definiciju prebačaja i podbačaja. U tom slučaju matematički model ima oblik

$$\begin{aligned} \min \quad & F(x, y) = y_1^+ + y_1^- + y_2^+ y_2^- + y_3^+ + y_3^- \\ \text{p.o.} \quad & 4x_1 + 5x_2 - y_1^+ + y_1^- = 60 \\ & x_1 + x_2 - y_2^+ + y_2^- = 14,32 \\ & 40x_1 + 6x_2 - y_3^+ + y_3^- = 480 \\ & 35x_1 + 7x_2 \geq 175 \\ & 136x_1 + 170x_2 \leq 2400 \\ & 20x_1 + 3x_2 \leq 240 \\ & x_1, x_2, y_1^+, y_1^-, y_2^+, y_2^-, y_3^+, y_3^- \geq 0. \end{aligned}$$

Rešavanjem ovog zadatka dobija se jedinstveno rešenje  $F^* = 0$ ,  $x_1^* = 11.59$  i  $x_2^* = 2.73$ .

To što je vrednost funkcije cilj jednako nuli ukazuje da je dobijeno rešenje savršeno jer nema ni prebačaja ni podbačaja vrednosti funkcija cilja od idealne tačke. Ovo je očigledno i na grafiku. Vidimo da u tački **B** sve funkcije cilja dostižu svoju najveću vrednost.

Jedino je tačka **B**(11.59, 2.73) Pareto optimalna. Sve tačke na duži **AB** na jednu stranu i **BC** na drugu stranu su slabi Pareto optimumi.

Sledi funkcija **MultiDist** kojom se daje implementacija metode rastojanja. Kao pomoćna funkcija koristi se funkcija **ideal** [83].

```
ideal[q_,constr_,var_]:=Module[{res={},i,l=Length[q]},For[i=1,i<=l,i++,AppendTo[res,First[Maximize[q[[i]],constr,var]]]];Return[res];]

MultiDist[q_,constr_,w_List,var_]:=Module[{con=constr,point={},l=Length[q],s},If[w=={}, point=ideal[q,con,var], point=w];For[i=1,i<=l,i++, AppendTo[con,q[[i]]-$[2i-1]+$[2i]==point[[i]]]; (* 3,2 *)];For[i=1,i<=2l,i++, AppendTo[con,$[i]>=0] ]; s=Minimize[Array[$,2l,1,Plus],con,Union[Variables[Array[$,2l,1]],var]];(* 3,2 *)Return[{q/.Last[s],Last[s]}];]
```

**Primer 3.6.4.** Rešiti problem

$$\begin{aligned} \max \quad & \{Q_1(\mathbf{x}) = x_1, Q_2(\mathbf{x}) = x_2\} \\ \text{p.o.} \quad & 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1. \end{aligned}$$

Ovaj problem se rešava izrazom

**In[5]:= MultiDist[{x1,x2},{x1>=0,x2>=0,x1<=1,x2<=1},{},{}]**

Pozivom funkcije *ideal*, dobija se idealna tačka {2, 3}. Posle toga, dobija se sledeći

$$\begin{aligned} \min \quad & \Phi(\mathbf{x}) = |2 - x_1| + |3 - x_2| \\ \text{p.o.} \quad & x_1 \leq 1, \quad x_2 \leq 1 \quad x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

Posle zamena  $y_1 = 2 - x_1 = \$[1]^- - \$[1]^+$ ,  $y_2 = 3 - x_2 = \$[2]^- - \$[2]^+$ , koristeći  $|y_1| = \$[1]^+ + \$[1]^-$ ,  $|y_2| = \$[2]^+ + \$[2]^-$ , s obzirom na (2.7), polazni problem se transformiše u sledeći

$$\begin{aligned} \min \quad & \$[1]^+ + \$[1]^- + \$[2]^+ + \$[2]^- \\ \text{p.o.} \quad & x_1 - \$[1]^+ + \$[1]^- = 2, \quad x_2 - \$[2]^+ + \$[2]^- = 3, \\ & x_1 \leq 1, \quad x_2 \leq 1, \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad \$[1]^+ \geq 0, \quad \$[1]^- \geq 0, \quad \$[2]^+ \geq 0, \quad \$[2]^- \geq 0. \end{aligned}$$

Unutrašnja reprezentacija ovog problema je

$$\begin{aligned} & \$[1] + \$[2] + \$[3] + \$[4], \\ & \{x1 \geq 0, x2 \geq 0, x1 \leq 1, x2 \leq 1, \\ & \quad x1 - \$[1]^+ + \$[1]^- = 2, x2 - \$[2]^+ + \$[2]^- = 3, \\ & \quad \$[1]^+ \geq 0, \$[2]^+ \geq 0, \$[3] \geq 0, \$[4] \geq 0\}, \\ & \{x1, x2, \$[1], \$[2], \$[3], \$[4]\} \end{aligned}$$

dok je njegovo optimalno rešenje

$$\{3, \{x1 \rightarrow 1, x2 \rightarrow 1, \$[1] \rightarrow 0, \$[2] \rightarrow 1, \$[3] \rightarrow 0, \$[4] \rightarrow 2\}\}$$

Prema tome, optimalno rešenje polaznog problema je

$$\text{Out[5]= } \{\{1, 1\}, \{x1 \rightarrow 1, x2 \rightarrow 1, \$[1] \rightarrow 0, \$[2] \rightarrow 1, \$[3] \rightarrow 0, \$[4] \rightarrow 2\}\}$$

## 4. Zaključak

U ovoj doktorskoj disertaciji analizirane su dve grupe metoda:

1. Simbolička izračunavanja dekompozicija i generalisanih inverza konstantnih, racionalnih i polinomijalnih matrica;
2. Linearni i nelinearni metodi višekriterijumske optimizacije i njihove modifikacije.

Glavni cilj ove disertacije je detaljna analiza algoritamskog pristupa kod simboličkog izračunavanja široke klase generalisanih inverza konstantnih, racionalnih i polinomijalnih matrica, kao i u višekriterijumskoj optimizaciji. Činjenica je da su ove oblasti još uvek neistražene i da ima prostora za dalja istraživanja u ovim poljima. Poznata literatura je najčešće posvećene ili teoriji optimizacije, ili teoriji uopštenih inverza. U ovoj disertaciji smo ukazali na povezanost ove dve discipline. Zapravo, moguće je posmatrati teoriju i metode bezuslovne optimizacije, kao korisno sredstvo za određivanje uopštenih inverza matrica, a u nekim slučajevima, za nalaženje uopštenih inverza linearno ograničenih operatora na Hilbertovim prostorima. Ovde smo izveli analizu i nove rezultate za višekriterijumsku optimizaciju i uopštene inverze, zasebno. Kasnije je moguće ovde prezentovane rezultate upotrebiti za konstrukciju novih algoritama za rešavanje specifičnih optimizacionih problema, radi određivanja različitih tipova generalisanih inverza.

Cilj istraživanja bio je doprinos kako u teorijskom (naučnom) tako i u praktičnom unapređenju postojećih metoda i algoritama za efikasna simbolička matrična izračunavanja i rešavanje problema višekriterijumske optimizacije. Radi ostvarenja tih ciljeva u ovoj doktorskoj disertaciji posebna pažnja je posvećena sledećim problemima:

1. pronalaženju novih i usavršavanju postojećih metoda za dekompoziciju polinomijalnih i racionalnih matrica;
2. konstrukciji efikasnih algoritama za izračunavanje uopštenih inverza matrica;
3. upoređivanju novih metoda i algoritama sa postojećim i prikazivanje poboljšanja i unapređenja u smislu složenosti, brzine i efikasnosti;
4. korišćenju gore pomenutih metoda za simbolička izračunavanja različitih vrsta uopštenih inverza polinomijalnih i racionalnih matrica;
5. konstrukciji novih algoritama i modifikaciju postojećih algoritama za rešavanje problema višekriterijumske optimizacije.

U ovoj, zaključnoj glavi dajemo kratak pregled svih izloženih rezultata, kao i neke ideje i motivaciju za dalja istraživanja.

Izveden je efikasan algoritam za izračunavanje  $\{1, 2, 3\}$ ,  $\{1, 2, 4\}$  inverza i Moore-Penrose-ovog inverza date racionalne matrice, zasnovan na  $LDL^*$  faktorizaciji. Ovaj algoritam je veoma pogodan za implementaciju u simboličkom programskom paketu MATHEMATICA. Međutim, polazni algoritam nije pogodan za korišćenje u proceduralnim programskim jezicima, zbog poteškoća sa simplifikacijom racionalnih izraza. Iz tog razloga su izvedena dva algoritma za izračunavanje Moore-Penrose-ovog inverza za polinomijalne i racionalne matrice. U njima je korišćena  $LDL^*$  faktorizacija potpunog ranga odgovarajuće polinomijalne, odnosno racionalne matrice. Pritom je izbegnuto korišćenje kvadratnih korena koji se javljaju u Cholesky dekompoziciji.

Poznato je da Gausova eliminacija nije optimalan algoritam. Čak i da je polinomijalna matrica invertibilna, standardna procedura eliminacije može imati probleme sa efikasnošću. Dugotrajne procedure normalizacije se moraju izvoditi (na primer za određivanje da li je pivot element jednak nuli, ili kod kancelacija). Stepen brojilaca i imenilaca među-rezultata može narasti eksponencijalno od  $n$ , pa dakle algoritam ima eksponencijalnu složenost. Ona se može svesti na kompleksnost u polinomijalnom vremenu primenom Gauss-Bareiss-ovih tehnika, i čak i na manji stepen primenom interpolacionih tehnika i rekonstrukcijom racionalne funkcije.

Gausov algoritam bez razlomaka uveden je u [2], koji predstavlja veliko poboljšanje u odnosu na algoritam Gausove eliminacije, i koji izvodi sve korake gde je egzaktno deljenje (tj. bez ikakvih ostataka) moguće. Nekoliko skorijih unapređenja smanjila su kompleksnosti većine problema sa polinomijalnim matricama na red veličine kompleksnosti množenja matrica.

Jedna od motivacija za buduće istraživanje je izrada sličnih algoritama, zasnovanih na  $LDL^*$  ili  $LDU$  dekompoziciji, za rad sa loše uslovjenim matricama. Takođe, potrebno je pronaći modifikacije uvedenih algoritama na nekim Hermitskim strukturalnim matricama.

Numerički algoritam za izračunavanje  $A_{T,S}^{(2)}$  inverza izведен je u [80], koji generalizuje poznate reprezentacije Moore-Penrose-ovog inverza  $A^\dagger$  iz [44]. Uvedeni algoritam je zasnovan na izračunavanju  $QR$  dekompozicije potpunog ranga matrice odgovarajuće matrice  $W$ .

Dokazali smo da dobijena reprezentacija spoljnih inverza i odgovarajuća opšta reprezentacija zasnovana na proizvoljnoj faktorizaciji potpunog ranga proizvode identične rezultate. Istraživana je i eksplicitna tranziciona formula između ovih reprezentacija.

Takođe, specijalne faktorizacije potpunog ranga koje generišu  $\{2, 4\}$  i  $\{2, 3\}$ -inverze su dobijene. Svi uvedeni algoritmi zasnovani na  $QR$  dekompoziciji su vrlo efikasni u poređenju sa ostalim metodama za izračunavanje Moore-Penrose-ovog i Drazinovog inverza.

Dobijena su i dva algoritma za simboličko izračunavanje  $A_{T,S}^{(2)}$  inverza racionalnih matrica jedne promenljive. Posmatrali smo simboličko izračunavanje uopštenih inverza primenom  $LDL^*$  i  $QDR$  dekompozicije potpunog ranga fiksirane matrice  $W$ . Upotreboom dodatne matrice  $D$  u oba slučaja, izbegnuto je korišćenje kvadratnog korena, što je od ključne važnosti kod simboličkog izračunavanja polinomijalnih i racionalnih izraza. Data

---

su i uporedna vremena izračunavanja naših algoritama sa Leverrier–Faddeev algoritmom, Grevilovim metodom pregrađivanja na skupovima racionalnih test matrica.

Opisan je metod za blok  $LDL^*$  dekompoziciju potpunog ranga konstantne blok matrice. Ovaj metod je primenjen na nalaženje inverza i Moore-Penrose-ovog inverza  $2 \times 2$  blok matrica. Na osnovu početnih rezultata, izrađen je metod za direktno direktno određivanje sub-matrica koje se javljaju u Moore-Penrose-ovom inverzu.

Proširivanje i uopštavanje ovih rezultata za slučaj racionalnih matrica i za slučaj dve promenljive predstavlja motivaciju za dalja istraživanja. Takođe, i drugi tipovi uopštenih inverza se mogu odrediti primenom blokovske  $LDL^*$  dekompozicije potpunog ranga.

Simbolička implementacija glavnih metoda višekriterijumske optimizacije je navedena, kao i ispitivanje uslova Pareto optimalnosti, koristeći mogućnosti simboličkih transformacija podržanih u paketu MATHEMATICA. Detaljno je opisan i implementiran linearni metod težinskih koeficijenata na dvodimenzionalnim i trodimenzionalnim problemima. Uvedena je i modifikacija leksikografskog metoda višekriterijumske optimizacije za slučajeve dve i tri nepoznate. Pokazalo se da novodobijeni metodi imaju vrlo efikasna vremena izračunavanja Pareto optimalnih tačaka.

Osim simboličkih mogućnosti, koristili smo i prednosti funkcionalnog programiranja, kao i grafičke reprezentacije dvodimenzionalnih i trodimenzionalnih skupova dopustivih rešenja. Dobijene implementacije su veoma korisne u smislu edukacije (videti [97]) i eksperimenata sa problemima VKO.

U cilju budućeg istraživanja iz oblasti višekriterijumske optimizacije, moguće je posmatrati postojeće višekriterijumske probleme iz sfera života i industrije, i pokušati primenu novodobijenih modifikovanih metoda. Neke od primena višekriterijumske optimizacije na praktičnim problemima, istraživane su u radovima [93, 76, 97].

# Literatura

- [1] A. G. Akritas, G. I. Malaschonok, *Computations in Modules over Commutative Domains*, Lecture Notes in Computer Science, **4770**, (2007), 11–23.
- [2] E. H. Bareiss, *Sylvester's Identity and Multistep Integer-Preserving Gaussian Elimination*, Mathematics of Computation, **22**, (1968), 565–578.
- [3] A. Ben-Israel, T.N.E. Greville, *Generalized inverses, Theory and applications, Second edition*, Canadian Mathematical Society, Springer, New York, 2003.
- [4] H.P. Benson, *Existence of efficient solution for vector maximization problems*, Journal of Optimization Theory and Applications, **28** (1978) 569–580.
- [5] M.A. Bhatti, *Practical optimization methods with Mathematica® applications*, Springer-Verlag, New York, 2000.
- [6] S.L. Campbell and C.D. Meyer Jr., *Generalized Inverse of Linear Transformation*, Pitman, London (1979), SIAM, 2009.
- [7] D.G. Carmichael, *Computation of Pareto optima in structural design*, Int. J. Numer. Methods Eng. 15, (1980) 925–952.
- [8] V. Chankong, Y. Haimes, *Multiobjective decision making: Theory and methodology Series, Volume 8*, North-Holland, New York, Amsterdam, Oxford, 1983.
- [9] A. Charnes, W.W. Cooper, *Management Models and Industrial Applications of Linear Programming*, New York: John Wiley and Sons, 1961.
- [10] A. Charnes, W.W. Cooper, *Goal programming and multiple objective optimization; part 1*, Eur. J. Oper. Res. 1, (1977) 39–54.
- [11] Y. Chen, *The generalized Bott–Duffin inverse and its application*, Linear Algebra Appl. **134** (1990), 71–91.
- [12] Y. Chen, X. Chen, *Representation and approximation of the outer inverse  $A_{T,S}^{(2)}$  of a matrix  $A$* , Linear Algebra Appl. **308** (2000), 85–107.
- [13] E. Chow, Y. Saad, *Approximate inverse techniques for block-partitioned matrices*, SIAM J. Sci. Comput. **18** (1997), 1657–1675.

- [14] R.E. Cline, *Inverses of rang invariant powers of a matrix*, SIAM J. Numer. Anal. **5** (1968), 182–197.
- [15] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*, The MIT Press, Cambridge, Massachusetts London, McGraw-Hill Book Company, Boston, New York, San Francisco, St. Louis, Montreal, Toronto, 2001.
- [16] P. Courrieu, *Fast computation of Moore-Penrose inverse matrices*, Neural Information Processing - Letters and Reviews **8** (2005), 25–29.
- [17] P. Courrieu, *Fast solving of weighted pairing least-squares systems*, Journal of Computational and Applied Mathematics **231** (2009), 39–48.
- [18] D.S. Cvetković-Ilić, *A note on the representation for the Drazin inverse of  $2 \times 2$  block matrices*, Linear Algebra and its Applications **429** (2008), 242–248.
- [19] D.S. Cvetković-Ilić, *New conditions for the reverse order laws for  $\{1,3\}$  and  $\{1,4\}$ -generalized inverses*, Electronic Journal of Linear Algebra, **23** (2012), 231–242.
- [20] D.S. Cvetković-Ilić, *New additive results on Drazin inverse and its applications*, Appl. Math. Comp. 218(7) (2011), 3019–3024.
- [21] D.S. Cvetković-Ilić, P. Stanimirović, M. Miladinović, *Comments on some recent results concerning  $\{2,3\}$  and  $\{2,4\}$ -generalized inverses*, Appl. Math. Comp. **217** (22) (2011), 9358–9367.
- [22] I. Das, J.E Dennis, *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in višekriterijumska optimizacija problems*, Struct. Optim. **14** (1997) 63–69.
- [23] G. Fragulis, B.G. Mertzios, A.I.G. Vardulakis, *Computation of the inverse of a polynomial matrix and evaluation of its Laurent expansion*, Int. J. Control., **53** (1991), 431–443.
- [24] P. Giorgi, C. P. Jeannerod, G. Villard, *On the Complexity of Polynomial Matrix Computations*, Proceedings of the 2003 international symposium on Symbolic and algebraic computation, (2003), 135–142.
- [25] G. H. Golub, C. F. Van Loan, *Matrix Computations, Third edition*, The Johns Hopkins University Press, Baltimore, 1996.
- [26] C.R. Goodall, *Computation Using the QR Decomposition*, C.R. Rao, ed., Handbook of Statistics, Vol. 9, 1993, 467–508.
- [27] T.N.E. Greville, *Some applications of the pseudo-inverse of matrix*, SIAM Rev. **3** (1960), 15–22.
- [28] A.J. Getson, F.C. Hsuan, *{2}-Inverses and their Statistical Applications*, Lecture Notes in Statistics **47**, Springer, Berlin, 1988.

- [29] Y.Y. Haimes, L.S. Lasdon, D.A. Wismer, *On a bicriterion formulation of the problems of integrated system identification and system optimization*, IEEE Trans. Syst. Man Cybern. SMC-1, (1971) 296–297.
- [30] R.E. Hartwig, *A method for calculating  $A^d$* , Math. Japonica **26** (1981), 37–43.
- [31] B.F. Hobbs, *A comparison of weighting methods in power plant siting.*, Decis. Sci., **11** (1980) 725–737.
- [32] N.J. Higham, *The Matrix Computation Toolbox*,  
<http://www.ma.man.ac.uk/~higham/mctoolbox>
- [33] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, Second Edition, SIAM, Philadelphia, 2002.
- [34] C. Hung, T.L. Markham, *The Moore-Penrose Inverse of a Partitioned Matrix  $M = \begin{pmatrix} A & D \\ B & C \end{pmatrix}$* , Linear Algebra and its Applications **11** (1975), 73–86.
- [35] F. Husen, P. Langenberg, A. Getson, *The  $\{\mathcal{D}\}$ -inverse with applications to statistics*, Linear Algebra Appl. **70** (1985), 241–248.
- [36] C.L. Hwang, K. Yoon, *Multiple attribute decision making methods and applications: a state-of-the-art survey.*, In: Beckmann, M.; Kunzi, H.P. (eds.) Lecture Notes in Economics and Mathematical Systems, No. 186. Berlin: Springer-Verlag, 1981.
- [37] C.L. Hwang, Md. Masud, A.S., in collaboration with Paidy, S.R., Yoon, K. *Multiple objective decision making, methods and applications: a state-of-the-art survey* In: Beckmann, M.; Kunzi, H.P. (eds.) Lecture Notes in Economics and Mathematical Systems, No. 164. Berlin: Springer-Verlag, 1979.
- [38] Y. Ijiri, *Management Goals and Accounting for Control*, Amsterdam: North-Holland, 1965.
- [39] J.P. Ignizio, *Linear programming in single-multiple-objective systems*, Englewood Cliffs: Prentice Hall, 1982.
- [40] C.P. Jeannerod, G. Villard *Essentially optimal computation of the inverse of generic polynomial matrices*, Journal of Complexity **21** (2005), 72–86.
- [41] J. Jones, N.P. Karampetakis, A.C. Pugh, *The computation and application of the generalized inverse via Maple*, J. Symbolic Computation **25** (1998), 99–124.
- [42] N.P. Karampetakis, *Computation of the generalized inverse of a polynomial matrix and applications*, Linear Algebra Appl. **252** (1997), 35–60.
- [43] N.P. Karampetakis, *Generalized inverses of two-variable polynomial matrices and applications*, Circuits Systems Signal Processing **16** (1997), 439–453.

- [44] V.N. Katsikis, D. Pappas, A. Petralias, *An improved method for the computation of the Moore-Penrose inverse matrix*, Appl. Math. Comput. **217** (2011), 9828–9834.
- [45] T. Lu, S. Shiou, *Inverses of  $2 \times 2$  block matrices*, Computers and Mathematics with Applications **43** (2002), 119–129.
- [46] K. Miettinen, L. Kirilov, *Interactive reference direction approach using implicit parametrization for nonlinear multiobjective optimization*, MCDM 2004, Whistler, B. C. Canada August 6-11, 2004.
- [47] K.R. Mac-Crimmon, *An overview of multiple objective decision making*, in *J.L. Cochrane and M. Zeleny (Eds), Multiple criteria decision making*, University of South Carolina Press, Columbia, 1973.
- [48] R. Maeder, *Programming in Mathematica, Third Edition* Redwood City, California: Adisson-Wesley, 1996.
- [49] A.N. Malyshev, "Matrix equations: Factorization of matrix polynomials" *M. Hazewinkel (ed.), Handbook of Algebra I*, Elsevier (1995), 79-116.
- [50] R.T. Marler, *Survey of multi-objective optimization methods for engineering Struct. Multidisc. Optim.* **26** (2004) 369–395.
- [51] K. Miettien *Nonlinear Multiobjective Optimization*, Kluver Academic Publishers, Boston, London, Dordrecht, 1999.
- [52] T. Mulders, A. Storjohann, *On lattice reduction for polynomial matrices*, J. Symbolic Comput. **35** (4) (2003) 377-401.
- [53] Matrix Market, *National Institute of Standards and Technology, Gaithersburg, MD*. Available online from: <http://math.nist.gov/MatrixMarket>.
- [54] L. Nagy, V. Miller, D. Powers, *Research note: on the application of matrix generalized inverses to the construction of inverse systems*, Int. J. Control, **24** (1976), 733–739.
- [55] H.S. Najafi, M. S. Solary, *Computational algorithms for computing the inverse of a square matrix, quasi-inverse of a non-square matrix and block matrices*, Appl. Math. Comput. **183** (2006), 539–550.
- [56] M.Z. Nashed, *Generalized Inverse and Applications*, Academic Press, New York, 1976.
- [57] M.Z. Nashed, X. Chen, *Convergence of Newton-like methods for singular operator equations using outer inverses*, Numer. Math. **66** (1993), 235-257.
- [58] B. Noble, *A method for computing the generalized inverse of a matrix*, SIAM J. Numer. Anal. **3** (1966), 582–584.

- [59] S. Opricović, *Optimizacija sistema*, Građevinski fakultet Univerziteta u Beogradu, 1992.
- [60] A. Osyczka, *Multicriterion Optimization in Engineering with Fortran Programs*, New York: John Wiley and Sons, 1984.
- [61] M.D. Petković, P.S. Stanimirović, M.B. Tasić, *Effective partitioning method for computing weighted Moore–Penrose inverse*, Comput. Math. Appl. **55** (2008), 1720–1734.
- [62] M.D. Petković, P.S. Stanimirović, *Symbolic computation of the Moore-Penrose inverse using partitioning method*, Int. J. Comput. Math. **82** (2005), 355–367.
- [63] R. Piziak, P. L. Odell, *Full rang Factorization of Matrices*, Mathematics Magazine **72** (1999), 193–201.
- [64] C.R. Rao, S.K. Mitra, *Generalized Inverse of Matrices and its Applications*, John Wiley & Sons, Inc., New York, London, Sydney, Toronto, 1971.
- [65] M.A. Rakha, *On the Moore-Penrose generalized inverse matrix*, Appl. Math. Comput. **158** (2004), 185–200.
- [66] P. Robert, *On the Group inverse of a linear transformation*, J. Math. Anal. Appl. **22** (1968), 658–669.
- [67] L. Rodman, "Matrix functions" M. Hazewinkel (ed.) , *Handbook of Algebra* , I , Elsevier (1995), 117–154.
- [68] M. Sakaravitch, *Linear programming*, Springer-Verlag, New York (1983).
- [69] K. Schittkowski, *Multicriteria Optimization -User's Guide-*, <http://www.klaus-schittkowski.de>, November 2004.
- [70] X. Sheng, G. Chen, *Full-rang representation of generalized inverse  $A_{T,S}^{(2)}$  and its applications*, Comput. Math. Appl. **54** (2007), 1422–1430.
- [71] N. Shinozaki, M, Sibuya, K. Tanabe, *Numerical algorithms for the Moore-Penrose inverse of a matrix: direct methods*, Annals of the Institute of Statistical Mathematics **24** (1972), 193–203.
- [72] W. Stadler, *Fundamentals of multicriteria optimization*, In: Stadler,W. (ed.) *Multicriteria Optimization in Engineering and in the Sciences*, pp. 1–25. New York: Plenum Press, 1988.
- [73] I.P. Stanimirović, *Compendious lexicographic method for multi-objective optimization*, Facta Universitatis (Niš) Ser. Math. Inform. **27**, No. 1 (2012), 55–66.
- [74] I.P. Stanimirović, *Computing Moore-Penrose inverse using full-rank square root free factorizations of rational matrices*, submitted to Journal of Computational Mathematics.

- [75] I.P. Stanimirović, *Full-rank block LDL<sup>\*</sup> decomposition and the inverses of  $n \times n$  block matrices*, Journal of Applied Mathematics and Computing **40** (2012), 569–586.
- [76] I. Stanimirović, M. Petković, P. Stanimirović, M. Ćirić, *Heuristic algorithm for single resource constrained project scheduling problem based on the dynamic programming*, YUJOR **19** (2009), 281–298.
- [77] I.P. Stanimirović, M.B. Tasić, *Computation of generalized inverses by using the LDL<sup>\*</sup> decomposition*, Appl. Math. Lett., **25** (2012), 526–531.
- [78] I.P. Stanimirović, M.B. Tasić, A.M. Ilić, *Full-rank LDL<sup>\*</sup> Decomposition and Generalized Inverses of Polynomial Hermitian Matrices*, submitted to Journal of Symbolic Computations.
- [79] I. Stanimirović, M. Zlatanović, M. Petković, *On the linear weighted sum method for multi-objective optimization*, Facta Universitatis (Niš) Ser. Math. Inform. **26** (2011), 47–62.
- [80] P. Stanimirović, D. Pappas, V. Katsikis, I. Stanimirović, *Full-rank representations of outer inverses based on the QR decomposition*, Applied Mathematics and Computation **218** (2012), 10321–10333.
- [81] P. Stanimirović, D. Pappas, V. Katsikis, I. Stanimirović, *Symbolic computation of  $A_{T,S}^{(2)}$ -inverses using QDR factorization*, Linear Algebra Appl. **437** (2012), 1317–1331.
- [82] P.S. Stanimirović, *Block representations of  $\{2\}, \{1, 2\}$  inverses and the Drazin inverse*, Indian J. Pure Appl. Math. **29** (1998), 1159–1176.
- [83] P. Stanimirović, I. Stanimirović, *Implementation of polynomial multi-objective optimization in MATHEMATICA*, Structural and Multidisciplinary Optimization **36** (2008), 411–428.
- [84] P.S. Stanimirović, N.V. Stojković, M.D. Petković, *Run-time transformations in implementation of linear multi-objective optimization*, XVI Conference on Applied Mathematics, N. Krejic, Z. Luzanin, eds. Department of Mathematics and Informatics, Novi Sad, (2004), 157–165.
- [85] P.S. Stanimirović, M.B. Tasić, *Drazin Inverse of One-Variable Polynomial Matrices*, Filomat, **15** (2001), 71–78.
- [86] P.S. Stanimirović, M.B. Tasić, P. Krčolica, N.P. Karampetakis, *Generalized inversion by interpolation*, Filomat, **21:1** (2007), 67–86.
- [87] P.S. Stanimirović, M.B. Tasić, *Partitioning method for rational and polynomial matrices*, Appl. Math. Comput. **155** (2004), 137–163.
- [88] P.S. Stanimirović, M.B. Tasić, *On the Leverrier-Faddeev algorithm for computing the Moore-Penrose inverse*, J. Appl. Math. Comput. **35** (2011), 135–141.

- [89] P.S. Stanimirović, D.S. Djordjević, *Full-rank and determinantal representation of the Drazin inverse*, Linear Algebra Appl. **311** (2000), 31–51.
- [90] P.S. Stanimirović, D.S. Cvetković-Ilić, S. Miljković, M. Miladinović, *Full-rank representations of  $\{2, 4\}$ ,  $\{2, 3\}$ -inverses and successive matrix squaring algoritam*, Appl. Math. Comput. **217** (2011), 9358–9367.
- [91] P.S. Stanimirović, M.B. Tasić, *Computing generalized inverses using LU factorization of matrix product*, Int. J. Comput. Math. **85** (2008), 1865–1878.
- [92] P.S. Stanimirović, G.V. Milovanović, *Programski paket MATHEMATICA i primene*, Elektronski fakultet u Nišu, Edicija monografije, Niš, 2002.
- [93] J. Stefanović-Marinović, M. Petković, I. Stanimirović, M. Milovančević, *A model of planetary gear multicriteria optimization*, Transactions of FAMENA **35**, No. 4 (2011), 21–34.
- [94] A. Stojohann, G. Labahn, *A Fast Las Vegas algoritam for Computing the Smith Normal Form of a Polynomial Matrix*, Linear Algebra Appl. **253** (1997), 155–173.
- [95] M. Tasić, I. Stanimirović, *Implementation of partitioning method*, Facta Universitatis (Niš) Ser. Math. Inform. **25** (2010), 25–33.
- [96] M.B. Tasić, I.P. Stanimirović, *Symbolic computation of Moore-Penrose inverse using the LDL\* decomposition of the polynomial matrix*, Filomat, accepted for publication.
- [97] M. Tasić, P. Stanimirović, I. Stanimirović, M. Petković, N. Stojković, *Some useful MATHEMATICA teaching examples*, Facta Universitatis (Niš) Ser.: Elec. Energ. **18**, No. 2 (2005), 329–344.
- [98] M.B. Tasić, P.S. Stanimirović, M.D. Petković, *Symbolic computation of weighted Moore-Penrose inverse using partitioning method*, Appl. Math. Comput. **189** (2007), 615–640.
- [99] M.B. Tasić, P.S. Stanimirović, *Symbolic and recursive computation of different types of generalized inverses*, Appl. Math. Comput. **199** (2008), 349–367.
- [100] M.B. Tasić, P.S. Stanimirović, *Differentiation of generalized inverses for rational and polynomial matrices*, Appl. Math. Comput. **216** (2010), 2092–2106.
- [101] R.P. Tewarson, *A direct method for generalized matrix inversion*, SIAM J. Numer. Anal. **4** (1967), 499–507.
- [102] Y. Tian, *The Moore-Penrose inverses of  $m \times n$  block matrices and their applications*, Linear Algebra and its Applications, **283** (1998), 35–60.
- [103] F. Toutounian, A. Ataei, *A new method for computing Moore-Penrose inverse matrices*, Journal of Computational and Applied Mathematics **228** (2009), 412–417.

- [104] H. Voogd, *Multicriteria Evaluation for Urban and Regional Planning*, London: Pion, 1983.
- [105] G. Wang, Y. Wei, S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, Beijing/New York, 2004.
- [106] G.R. Wang, Y.L. Chen, *A recursive algorithm for computing the weighted Moore-Penrose inverse  $A_{MN}^\dagger$* , J. Comput. Math. **4** (1986), 74–85.
- [107] F.M. Waltz *An engineering approach: hierarchical optimization criteria*, IEEE Trans. Autom. Control AC-12, (1967) 179–180.
- [108] D. Watkins, *Fundamentals of Matrix Computations*, Wiley-Interscience New York 2002.
- [109] Y. Wei, *A characterization and representation of the generalized inverse  $A_{T,S}^{(2)}$  and its applications*, Linear Algebra Appl. **280** (1998), 87–96.
- [110] Y. Wei, H. Wu, *The representation and approximation for the generalized inverse  $A_{T,S}^{(2)}$* , Appl. Math. Comput. **135** (2003), 263–276.
- [111] R.E. Wendell, D.N. Lee, *Efficiency in multiple objective optimization problems*, Mathematical Programming, **12** (1977) 406–414.
- [112] F. Winkler, *Polynomial Algorithms in Computer Algebra*, Springer, 1996.
- [113] S. Wolfram, *The MATHEMATICA Book*, 5th ed., Wolfram Media/Cambridge University Press, Champaign, IL 61820, USA, 2003.
- [114] L.A. Zadeh, *Optimality and non-scalar-valued performance criteria*, IEEE Trans. Autom. Control AC-8, (1963) 59–60.
- [115] B. Zheng, R.B. Bapat, *Generalized inverse  $A_{T,S}^{(2)}$  and a rank equation*, Appl. Math. Comput. **155(2)** (2004), 407–415.
- [116] S. Zions, *Multiple criteria mathematical programming: an updated overview and several approaches.*, In: Mitra, G. (ed.) *Mathematical Models for Decision Support*, 1988, pp. 135–167, Berlin: Springer-Verlag
- [117] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36** (1986), 105–162.
- [118] G. Zielke, *A survey of generalized matrix inverses*, Computational Mathematics, Banach center Publications **13** (1984), 499–526.
- [119] K. Zotos, *Performance comparison of Maple and Mathematica*, Appl. Math. Comput. (2006), doi:10.1016/j.amc.2006.11.008.
- [120] S. Zlobec, *An explicit form of the Moore-Penrose inverse of an arbitrary complex matrix*, SIAM Rev. **12** (1970), 132–134.

# Spisak algoritama

2.1 Izračunavanje $\{i, j, k\}$ i Moore-Penrose-ovog inverza racionalne matrice - Algoritam LDLGInverse .....	16
2.2 Izračunavanje $LDL^*$ dekompozicije potpunog ranga zadate Hermitske polinomijalne matrice .....	23
2.3 Simboličko izračunavanje Moore-Penrose-ovog inverza polinomijalne matrice .....	31
2.4 Izračunavanje $LDL^*$ faktorizacije potpunog ranga racionalne Hermitske matrice ..	39
2.5 Simboličko izračunavanje Moore-Penrose-ovog inverza racionalne matrice .....	43
2.6 Izračunavanje $A_{T;S}^{(2)}$ inverza matrice na osnovu $LDL^*$ dekompozicije potpunog ranga matrice M - Algoritam LDLATS .....	47
2.7 Simboličko izračunavanje $A_{T;S}^{(2)}$ inverza polinomijalne matrice primenom $LDL^*$ faktorizacije potpunog ranga - Algoritam LDLATS2 .....	50
2.8 Izračunavanje $A_{T;S}^{(2)}$ inverza matrice koristeći $QR$ dekompoziciju matrice W - Algoritam QRATS2 .....	61
2.9 Izračunavanje $QDR$ dekompozicije potpunog ranga racionalne matrice A .....	69
2.10 Simboličko izračunavanje inverzne matrice $A_{T;S}^{(2)}$ korišćenjem $QDR$ faktorizacije - Algoritam QDRATS2 .....	73
2.11 Izračunavanje blokovske $LDL^*$ dekompozicije potpunog ranga date blok matrice	98
2.12 Grevilleov rekurzivni algoritam pregradjivanja .....	109
3.1 Ispitivanje Pareto optimalnosti fiksirane tačke $\mathbf{x}^*$ .....	118
3.2 Izračunavanje Pareto optimuma generisanjem težinskih koeficijenata .....	122
3.3 Leksikografski metod višekriterijumske optimizacije .....	134
3.4 Rešavanje linearног problema višekriterijumske optimizacije, za slučaj dve nepoznate - Modifikovani leksikografski metod 2D .....	138
3.5 Rešavanje linearног problema višekriterijumske optimizacije, za slučaj tri nepoznate - Modifikovani leksikografski metod 3D .....	139
3.6 Relaksirani leksikografski metod višekriterijumske optimizacije .....	143

# Indeks pojmove

- algoritam, 3
- dekompozicija matrice, 3
- blokovska  $LDL^*$  dekompozicija, 97
  - Cholesky dekompozicija, 10
  - dekompozicija potpunog ranga, 6
  - $LDL^*$  dekompozicija, 10, 13, 19, 23, 36, 39, 47, 96
  - LU dekompozicija, 9
  - QDR dekompozicija, 69, 73
  - QR dekompozicija, 55, 61
- donosilac odluke, 114
- Hermitska matrica, 10, 19, 35, 39, 97
- idealne vrednosti funkcija cilja, 113
- indeks matrice, 5
- MATHEMATICA, 3, 10, 17, 26, 34, 45, 54, 77, 88, 109, 115, 118, 123, 131, 140, 151
- MATLAB, 3, 62
- metod
- $\varepsilon$ -ograničenja, 145
  - geometrijski metod, 137
  - Grevilleov metod pregrađivanja, 109
  - Ispitivanje Pareto optimalnosti, 118
  - leksikografski metod, 134
  - linearni metod težinskih koeficijenata, 125
  - modifikovani leksikografski metod, 138
  - rastojanja, 147
  - relaksirani leksikografski metod, 142
  - težinskih koeficijenata, 119
- metrika
- Čebiševljeva, 148
  - Euklidova, 148
  - pravougaona, 148
- Pareto optimum, 114, 118
- slabi, 114
  - strog, 115
- polinomijalna matrica, 20, 29
- problem, 112
- linearog programiranja, 136
  - višekriterijumske optimizacije, 112, 113, 116
- racionalna matrica, 35, 39
- rešenja
- marginalna, 113
- rešenje
- savršeno, 114
- simboličko izračunavanje, 3, 31, 69, 116
- skalarna optimizacija, 119
- tačka
- idealna, 114, 151
- težinski koeficijenti, 119, 122
- uopšteni inverzi matrice, 4, 55
- $A_{T;S}^{(2)}$  inverz, 5, 47, 50, 55, 69, 73
  - Bott-Duffin-ov inverz, 6
  - definicija, 5
  - Drazinov inverz, 6, 8, 67
  - grupni inverz, 5, 6
  - Moore-Penrose-ov inverz, 5, 6, 15, 29, 31, 41, 64, 103
  - težinski Moore-Penrose-ov inverz, 6
  - uopšteni  $\{i, j, k\}$  inverz, 5, 14
  - uopšteni Bott-Duffin-ov inverz, 6
  - uopšteni inverzi blok matrice, 78, 81, 100



**ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ  
НИШ**

**КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА**

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	монографска
Тип записа, ТЗ:	текстуални / графички
Врста рада, ВР:	докторска дисертација
Аутор, АУ:	Иван Станимировић
Ментор, МН:	Милан Тасић
Наслов рада, НР:	Алгоритми за симболичка матрична израчунавања и оптимизацију
Језик публикације, ЈП:	српски
Језик извода, ЈИ:	енглески
Земља публиковања, ЗП:	Србија
Уже географско подручје, УГП:	Србија
Година, ГО:	2012.
Издавач, ИЗ:	ауторски репринт
Место и адреса, МА:	Ниш, Вишеградска 33.
Физички опис рада, ФО: (поплавња/страна/цита/табела/слика/графика/прилога)	165 стр., граф. прикази
Научна област, НО:	Рачунарске науке
Научна дисциплина, НД:	Симболичка израчунавања
Предметна одредница/Кључне речи, ПО:	Уопштени инверзи матрица, вишекритеријумска оптимизација
УДК	512.54.05 : 512.643 + 519.863
Чува се, ЧУ:	библиотека
Важна напомена, ВН:	
Извод, ИЗ:	Izведен je efikasan algoritam za izračunavanje spoljnih inverza i Moore-Penrose-ovog inverza date racionalne matrice, zasnovan na LDL* dekompoziciji. Dva algoritma za izračunavanje Moore-Penrose-ovog inverza polinomijalnih i racionalnih matrica su uvedena. Pritom je izbegnuto korišćenje kvadratnih korena koji se javljaju u Cholesky dekompoziciji. Opisana je simbolička implementacija glavnih metoda višekriterijumske optimizacije, kao i ispitivanje uslova Pareto optimalnosti. Detaljno je opisan linearni metod težinskih koeficijenata.
Датум прихватanja теме, ДП:	24. 12. 2012.
Датум одbrane, ДО:	
Чланови комисије, КО:	Председник: ..... Члан: ..... Члан, ментор: .....



**ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ  
НИШ**

**KEY WORDS DOCUMENTATION**

Accession number, <b>ANO:</b>	
Identification number, <b>INO:</b>	
Document type, <b>DT:</b>	<b>monograph</b>
Type of record, <b>TR:</b>	<b>textual / graphic</b>
Contents code, <b>CC:</b>	<b>doctoral dissertation</b>
Author, <b>AU:</b>	<b>Ivan P. Stanimirović</b>
Mentor, <b>MN:</b>	<b>Milan Tasić</b>
Title, <b>TI:</b>	Algorithms for symbolic matrix computations and optimization
Language of text, <b>LT:</b>	<b>Serbian</b>
Language of abstract, <b>LA:</b>	<b>English</b>
Country of publication, <b>CP:</b>	<b>Serbia</b>
Locality of publication, <b>LP:</b>	<b>Serbia</b>
Publication year, <b>PY:</b>	<b>2012</b>
Publisher, <b>PB:</b>	<b>author's reprint</b>
Publication place, <b>PP:</b>	<b>Niš, Višegradska 33.</b>
Physical description, <b>PD:</b> (chapters/pages/ref./tables/pictures/graphs/applications)	<b>165 p. ; graphic representations</b>
Scientific field, <b>SF:</b>	<b>Computer science</b>
Scientific discipline, <b>SD:</b>	<b>Symbolic computation</b>
Subject/Key words, <b>S/KW:</b>	<b>Generalized matrix inverses, multiobjective optimization</b>
<b>UC</b>	<b>512.54.05 : 512.643 + 519.863</b>
Holding data, <b>HD:</b>	<b>library</b>
Note, <b>N:</b>	
Abstract, <b>AB:</b>	An efficient algorithm for computing outer inverses and Moore-Penrose inverse of a rational matrix via LDL* decomposition, is developed. Two algorithms for evaluating the Moore-Penrose inverse of polynomial and rational matrices, are provided. Thereat, using square roots occurring in Cholesky decomposition, is avoided. Symbolic implementation of some main methods of multi-objective optimization are given, as well as verifying Pareto optimality. Linear weighted sum method is detailedly presented. Also, a modification of the lexicographic method is provided, for the cases of two and three unknowns.
Accepted by the Scientific Board on, <b>ASB:</b>	<b>24. 12. 2012.</b>
Defended on, <b>DE:</b>	
Defended Board, <b>DB:</b>	President: Member: Member, Mentor: