UNIVERZA V LJUBLJANI

BIOTEHNIŠKA FAKULTETA

Jana Obšteter

**SELEKCIJA Z OPTIMIZIRANIMI PRISPEVKI V SLOVENSKIH REJSKIH PROGRAMIH ZA GOVEDO**

Seminarska naloga pri predmetu METODOLOGIJA EKONOMSKIH IN DRUŽBOSLOVNIH ZNANOSTI V BIOZNANOSTI

Doktorski študij Bioznanosti

Ljubljana, 2018

Kazalo vsebine

[1 predstavitev raziskovalnega problema in ciljev 1](#_Toc512241277)

[1.1 SELEKCIJA PRI GOVEDU 1](#_Toc512241278)

[1.1.1 Napoved plemenskih vrednosti 1](#_Toc512241279)

[1.1.2 Cilji selekcije 3](#_Toc512241280)

[1.2 algoritmi za optimizacijo prispevkov v selekciji 5](#_Toc512241281)

[1.2.1 Optimizacijski algoritmi 6](#_Toc512241282)

[1.2.2 Razvoj rešitve za selekcijo z optimalnimi prispevki 8](#_Toc512241283)

[1.3 RAZISKOVALNI PROBLEM 10](#_Toc512241284)

[1.4 RAZISKOVALNI CILJI 11](#_Toc512241285)

[2 METODE IN MATERIALI DELA 11](#_Toc512241286)

[2. 1 PRIPRAVA ORODJA IN PARAMETROV ZA SIMULACIJO 11](#_Toc512241287)

[2.2 TESTIRANI SCENARIJI 12](#_Toc512241288)

[2.3 OPTIMIZACIJA PRISPEVKOV 12](#_Toc512241289)

[2. 4 PRIMERJALNI PARAMETRI 14](#_Toc512241290)

[3 REZULTATI IN RAZPRAVA 14](#_Toc512241291)

[3.1 Genetski napredek 14](#_Toc512241292)

[3.2 GENETSKA VARIABILNOST 15](#_Toc512241293)

[3.3 UČINKOVITOST SELEKCIJE 16](#_Toc512241294)

[4 zaključki 17](#_Toc512241295)

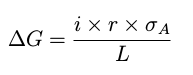
[5 Viri in literatura 18](#_Toc512241296)

# 1 predstavitev raziskovalnega problema in ciljev

## 1.1 SELEKCIJA PRI GOVEDU

Selekcija stremi h genetskemu izboljšanju populacije. V živinoreji izvajamo umetno selekcijo na gospodarsko pomembne lastnosti. To so predvsem proizvodne lastnosti, npr. prireja mleka ali prirast mesa, lastnosti zunanjosti živali in lastnosti zdravja. Ključen korak selekcijE je odbira najboljših živali za starše naslednje generacije. Zaradi dolgih generacijskih intervalov, velikega vpliva nekaj izbranih moških živalih ter visokih stroškov testiranja in vzreje živali je odbira genetsko najboljših živali izrednega pomena. Hkrati pa je identifikacija teh živali težavna zaradi več virov variabilnosti. Med temi so genetska variabilnost za lastnost, velik okoljski vpliv, segregacija in rekombinacija pri potomcih že identificiranih elitnih živali ter vključenost različnih organizacij v selekcijsko delo. Živali so rangirane in odbrane na podlagi plemenskih vrednosti (PV), ki so dvakratno odstopanje potomcev od povprečja populacije (Falconer in MacKay, 1996).

Uspešnost selekcije merimo z doseženim genetskim napredkom (ΔG), za izračun katerega velja:



*(1)*

; kjer je ∆G = letni genetski napredek, merjen v enotah standardnega odklona na leto, i = intenzivnost selekcija, r = točnost selekcije, σ = genetski standardni odklon, L = generacijski interval (Lush, 1945). Le ta je odvisen od točnosti selekcije (r), intenzivnosti odbire (i), genetske variabilnosti za lastnost (σA) in generacijskega intervala (L). Ti parametri se bistveno razlikujejo med moško in žensko potjo selekcije pri govedi, zato sta Rendel in Robertson, 1950, definirala štiri poti selekcije pri govedu: matere mater, matere očetov, očetje mater in očetje očetov. Najstrožja selekcija se vrši pri odbiri očetov, saj je vpliv moških živali zaradi velikega števila potomcev, ki jih dosežemo preko umetnega osemenjevanja, znatno večji od vpliva ženskih živali, ki imajo v povprečju le enega potomca letno.

### 1.1.1 Napoved plemenskih vrednosti

V selekciji živali lahko PV napovemo na dva načina: klasičnega ali genomskega. Oboje PV napovemo z metodo mešanega modela, pri čemer je vpliv živali naključen vpliv (enačba 2). Za napoved potrebujemo vektor fenotipskih vrednosti (rešitev), matriko sistematskih vplivov in matriko sorodstva. Model za napoved je sledeč:

*y =* ***X****b +* ***Z****a + e (2)*

; kjer je **X** matrika dogodkov, *b* je vektor ocen za sistematske vplive, **Z** je matrika sorodstva, zgrajena na podlagi rodovniških podatkov, *a* je vektor iskanih plemenskih vrednosti, *e* pa je vektor ostankov.

V **klasični** napovedi je matrika sorodstva zgrajena na podlagi rodovniških podatkov, ki jo imenujemo tudi matrika sorodstva C (angl. *coancestry matrix*). V napovedi uporabimo ½ C, ki jo imenujemo matrika A. Slednja odraža pričakovano sorodstvo oz. kovarianco med živalmi (slika 1). Matriki C in A sta pozitivni definitni matriki. Elementi matrike C so koeficienti sorodstva *c*(angl. c*oeancestry coefficient)*, matrike **A** pa koeficienti *a,* ki so ½ koeficientov sorodstva *c .* Diagonalni elementi matrike **A** izražajo sorodnost posameznika samega s sabo, in jih zapišemo kot *aii= (1 + Fi)*, kjer je *Fi*koeficient inbridinga za žival *i*. Nediagonalni elementi pa predstavljajo sorodstvo med dvema različnima posameznikoma in so oblike *aij  = ½ (aij’ + aij’’)*, kjer sta *j’* in *j’’* oče in mati živali *j*. Pri tem je *Fi =* ½ *(asd),* kjer je *s* oče, *d* pa mati živali *i*.



Slika 1: Primer matrike A, ki je ½ matrika sorodstva C, za posameznike 1 - 8.

Točnost klasičnih PV je tako močno odvisna od lastnih izmerjenih vrednosti oz. izmerjenih vrednosti potomcev in posledično zaznamovana z dolgimi generacijskimi intervali (povprečna starost starša ob rojstvu izbranih potomcev). Ker mlade živali še nimajo lastnih fenotipskih vrednosti, je njihova PV preprosto povprečje PV njihovih staršev (angl. parent average, PA), točnost katere je zaradi rekombinacije in segregacije majhna.

V zadnjih letih se je povečala dostopnost genomskih informacij. Znižanje cene genotipizacije (postopek pridobivanja genomskih podatkov za posameznika) je vodilo do razvoja t.i. genomske selekcije (Meuwissen in sod., 2001). Genomska selekcija omogoča napoved PV, imenovanih genomske plemenske vrednosti (gPV), za vse genotipizirane živali, tudi mlade brez lastnih fenotipskih vrednosti. Točnost gPV je večja od točnosti PA, kar omogoča zgodnejšo odbiro staršev (krajši generacijski interval) in bolj točno odbiro mladih živali (Pryce in Daetwyler, 2012).

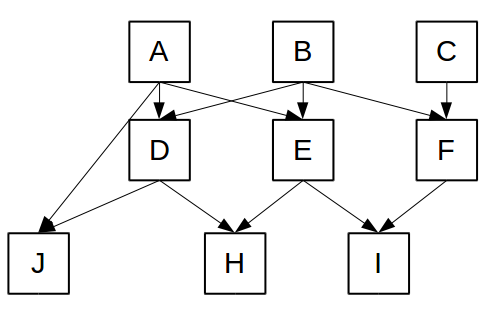
V **genomski** napovedi je matrika sorodstva zgrajena na podlagi genomskih podatkov . To zahteva genotipizacijo živali na izbranih številih mest (označevalci). Pri napovedi genomskih PV matriko sorodstva namesto na podlagi rodovniških podatkov zgradimo na podlagi genotipov živali na pregledanih označevalcih. Takšno matriko imenujemo genomska matrika sorodstva ali matrika **G**, ki je pozitivna semidefinitna matrika (slika 2). Genomsko sorodstvo imenujemo tudi realizirano sorodstvo. V genomski napovedi ocenjujemo vplive posameznih označevalcev, na podlagi katerih napovedmo gPV. Za napoved gPV zaradi večjega števila spremenljivk kot podatkov uporabimo metodo L2 regularizacije (Ridge regresija) ali pa uporabimo Bayesovski pristop, kjer za efekte posameznih označevalcev uporabimo aprirorij distribucijo (Meuwissen in sod., 2001).



Slika 2: Primer genomske matrike sorodstva za posameznike 1 - 8.

### 1.1.2 Cilji selekcije

Čeprav si selekcija prizadeva za genetski napredek populacije, je treba ob tem paziti tudi na ohranjanje genetske variabilnosti. Iz enačbe *(1)* je razvidno, da manjšanje genetske variabilnost manjša tudi genetski napredek. Povečanje intenzivnosti selekcije oz. uporaba majhnega števila elitnih posameznikov kratkoročno vodi do večjega genetskega napredka, vendar pa ne omogoča dolgoročnega napredka zaradi zmanjšanja genetske variabilnosti v populaciji. Slednje je posledica večjega parjenja v sorodstvu – inbridinga - preko večje možnosti za odbiro sorodnikov (Woolliams in sod., 2015). Koeficient inbridinga *F* je lastnost diploidnih organizmov – ki imajo dva seta kromosomov - in je definiran kot verjetnost, da sta v posamezniku oba alela na nevtralnem mestu na genomu (brez pritiska selekcije) identična po izvoru (Wright, 1921). Večji inbriding poleg onemogočanja dolgoročnega genetskega napredka vodi tudi do depresije zaradi inbridinga, ki pomeni zmanjšano preživetje in plodnost potomcev sorodnih posameznikov. Čeprav je inbriding neizbežen v končno velikih populacijah, je pomembno, da uravnavamo stopnjo, s katero koeficient inbridinga narašča (*ΔF*). Samo spremljanje koeficienta inbridinga *F* v starših in njihovih potomcih ni zadosten kriterij za trajnostno upravljanje s populacijo, saj se vpliv paritvenih odločitev na koeficient inbridinga pozna z zamikom dveh generaciji, ko se ustvarijo zanke v rodovniku (slika 3) (Woolliams in sod., 2015). Poleg tega je *F* v populaciji relativen na *F* v prvi generaciji rodovnika, za katero predpostavimo, da je neinbridirana in nesorodna. Posledično sama globina rodovnika vpliva na ocenjen *F* v populaciji (Woolliams in sod, 1998).



Slika 3: Primer rodovnika z zankami, ki ustvarjajo inbriding.

**Selekcija z optimalnimi prispevki**

Upravljanje *ΔF* je pomembno v dveh situacijah: prva je ohranjanje populacij, kjer je cilj minimiziranje *ΔF*, druga pa selekcija, kjer je cilj čimbolj učinkovito uporabiti razpoložljivo genetsko variabilnost za ustvarjanje genetskega napredka – cilj je tako maksimiranje genetskega napredka ob omejitvi *ΔF* (Wooliams in sod., 2015).

Za trajnostno upravljanje s populacijo v selekciji in za dolgotrajni uspeh selekcije moramo tako stremeti k povečanju genetskega napredka in hkrati nadzirati *ΔF* v populaciji. James in McBride sta leta 1958 prvič postavila koncept dolgoročnih genetskih prispevkov posameznikov, *r,* ki so definirani kot natančen prispevek genov, ki jih posameznik prispeva v bodoči genetski »bazen«. Z drugimi besedami – *r* za posameznika *i* je delež genov v populaciji veliko generacij kasneje, ki izhajajo iz posameznika *i*. Dolgoročni genetski prispevki so preko preprostih razmerij povezani s ključnimi parametri populacijske dinamike:

* z genetskim napredkom preko

*ΔG = ∑ ri x ai*, (3)

kjer je *r* dolgoročen prispevek posameznika in *a* prispevek Mendelskega vzorčenja (angl. *Mendelian sampling term*), ki je odstopanje potomca od povprečja staršev (Wolliams in sod., 1999);

* s stopnjo inbridinga, kjer je

*ΔF = ¼ Σ ri2 (4)*

(Wray in Thompson, 1990).

Rejske organizacije imajo cilj maksimirati *ΔG* ob vzdrževanju predhodno določene vrednosti za stopnjo inbridinga *ΔF* (Gorjanc in Hickey, 2018). Ta ideja je implementirana v selekciji z optimiziranimi prispevki (angl. o*ptimal contribution selection,* OCS) (Meuwissen 1997; Grundy et al. 1998), ki teži k uravnoteženju dveh kontradiktornih ciljev - dosegi genetskega napredka in ohranjanju genetske variabilnosti. Nasprotje temu imenujemo selekcija s selektivno odbiro (angl. *truncation selection*), pri čemer je cilj samo maksimiranje genetskega napredka brez obzira na ohranjanje genetske variabilnosti. Genetska varianca skozi generacije inbridinga pada in po *t* generacijah znaša

*σ2A,t = (1 – Ft) x σ2A,0 (5)*

(Falconer in Mackay, 1996). Vidimo, da je padec genetske variance po generacijah sorazmeren z naraščanjem inbridinga. Posledično moramo za upravljanje z genetsko varianco nadzorovati *ΔF*, saj je *F* posameznika enak koeficientu sorodstva njegovih staršev (Wooliams in sod., 2015). Povprečni inbriding v generaciji t je tako enak povprečnemu sorodstvu izbranih očetov in mater v generaciji *t – 1*. Sorodstvo med dvema posameznikoma *i* in *j* je enako:

*Cij = 1/2(Coče(i), oće(j) + Coče(i), mati(j) + Cmati(i), oče(j) + Cmati(i), mati(j)) (6)*

Ker je *ΔF* povezan z *r2*  prednikov (enačba 4), je minimiziranje *ΔF* ekvivalentno minimiziranju *Σ ri2.* OCS tako nadzoruje povečanje inbridinga preko nadzora povečanja **skupnega sorodstva** izbranih staršev (Wooliams in sod., 2015). Skupno sorodstvo staršev je definirano kot povprečno sorodstvo te skupine uteženo glede na posameznikove bodoče prispevke k naslednji generaciji. Je tudi verjetnost, da sta dva naključno izbrana alela iz genskega »bazena« populacije identična po izvoru (Woolliams, 2015). Stopnjo inbridinga *ΔF* lahko tako nadzorujemo preko skupnega sorodstva izbranih staršev. Da bi se izognili ustvarjanju ozkih grl, mora biti *ΔF* konstanten, *F* pa mora slediti:

*Ft  =* 1 – *(δf)t-1* (7)

; kjer je *δf* tarčen *ΔF* . Ekvivalentno mora skupno sorodstvo isti poti slediti eno generacijo prej:

*C*t  = 1 – (*δf*)t. (8)

V OCS se določi omejitev za skupno sorodstvo, ki je:

*Ct =* 1 – *(1 – Co)(1 – δf)t* (9)

; če se optimizacija začne v generaciji ena ali:

*Ct* = *Ct-1* + *δf*(1 – *Ct-1*) (10)

; če je sorodstvo v prvi generaciji (*Co*) povišano (Woolliams in sod., 2015). Ker nastavimo fiksno pot za omejitev *Ct*, bo vsako odstopanje od optimalnih prispevkov v generaciji *t,* ki bi povzročilo preveliko ali premajhno sorodstvo populacije v generaciji *t+1*, v OCS avtomatsko popravljeno, saj *Ct-1* še naprej sledi isti poti. Posledično bo omejitev v generaciji *t+1* bolj stroga ali bolj sproščena glede na dejansko izmerjeno sorodstvo populacije (v primerjavi z načrtovanim).

Selekcija z optimalnimi prispevki skuša maksimirati genetski napredek v naslednji generaciji (*Gt+1*) in hkrati omejiti skupno sorodstvo na *Ct+1*. Pri tem so potrebne dodatne omejitve, ki izhajajo iz praktičnega vidika parjenja: zagotoviti moramo, da so prispevki vseh posameznikov ne-negativni, da je vsota moških kot tudi ženskih posameznikov ½ ter da je vsota prispevkov vseh posameznikov enaka 1 (Woolliams in sod., 2015).

Prispevke lahko optimiziramo na dva načina. Prvi način vključuje optimizacijo prispevkov z namenom: i) maksimiranja genetskega napredka z omejitvijo stopnje skupnega sorodstva med izbranimi posamezniki; ali ii) samo minimiziranja skupnega sorodstva. Tej optimizaciji prispevkov lahko sledi še ena optimizacija, in sicer za dodelitev partnerjev za parjenje za minimiziranje inbridinga posameznih parjenj, kar prepreči preveliko depresijo zaradi inbridinga v potomcih (Gorjanc in Hickey, 2018). Obe optimizaciji lahko rešimo deterministično z Lagrangeovimi multiplikatorji (Meuwissen, 1997), linearnim programiranjem (Toro in Perez-Enciso, 1990), kvadratnim programiranjem (Pong-Wong in Woolliams, 2007) ali hevrističnimi metodami optimizacije (Gorjanc in Hickey, 2018). Drugi način vključuje vzajemno optimizacijo prispevkov in določitve partnerjev za parjenje preko optimizacije načrta parjenja (Kinghorn in Shepherd, 1999). Ta skupna optimizacija ni rešljiva z determinističnimi metodami, pač pa mora biti rešena s stohastičnimi ali hevrističnimi metodami.

## 1.2 algoritmi za optimizacijo prispevkov v selekciji

Glavni cilj selekcije z optimalnimi prispevki je maksimirati genetski napredek, vendar ob tem ohraniti genetsko variabilnosti. Algoritmi za optimizacijo problema so se razvijali od preprostih linearnih modelov, preko linearnega celoštevilskega programiranja, linearnega programiranja s kvadratno omejitvijo, ki je bila sprva rešena z Lagrangeovimi multiplikatorji, kasneje pa s konveksnim programiranjem, in sicer semidefinitnim programiranjem in nazadnje stožčastem programiranjem drugega reda. Algoritmi se med seboj razlikujejo v optimizacijskem postopku, številu ciljev za optimizacijo, potrebnem računalniškem času, spominu … V raziskavi smo za optimizacijo odbire in prispevkov živali uporabili R-ov paket optiSel, ki implementira stožčasto programiranje drugega reda, in AlphaMate, ki uporabi hevrističen evolucijski algoritem. Zato bom podrobneje razdelala ta dva algoritma.

### 1.2.1 Optimizacijski algoritmi

**1.2.1.1 Deterministični algoritmi**

**Konveksna optimizacija**

Konveksni problemi so oblike:

min: *f0(x) (11a)*

omejitve: *fi(x) ≤ 0, i = 1 … , m (11b)*

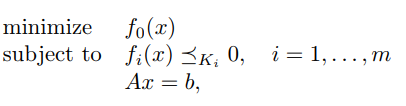
*aiTx = bi, i = 1 …, p; (11c)*

; kjer so *f0, …, fm* konveksne funkcije. Konveksni problemi imajo dodatne tri pogoje (Boyd, 2004):

* ciljna funkcija je konveksna,
* funkcija pogoja neenakosti mora biti konveksna,
* funkcija pogoja enakosti *hi(x) = aiTx – bi* mora biti afina.

Konveksni optimizacijski problemi minimizirajo konveksno funkcijo preko konveksne množice. Temeljna lastnost konveksne optimizacije je dejstvo, da je vsak lokalni optimum tudi globalni optimum. Tudi linearen program je konveksen optimizacijski problem z afino ciljno funkcijo in omejitvami. Afine funkcije so tudi konveksne funkcije, saj je vsaka linearna množica afina in vsaka afina množica je konveksna (Boyd, 2004).

Standardno formo konveksnega programiranja lahko posplošimo s tem, da omogočimo, da ima funkcija pogoja neenakosti vektorsko vrednost:



min:

omejjitve:

*(12a)*

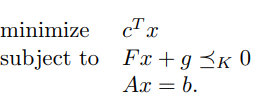
*(12b)*

*(12c)*

; kjer je *fo :* **R*n →*****R,** *Ki ⊆*  *Rki* pravi stožec in *fi:* **R*n*** *→***R***ki* *Ki*-konveksne. Takšnim problemom rečemo konveksni optimizacijski problemi s posplošeno omejitvijo neenakosti (angl. *convex optimization problems with generalized inequality constraints*). Med slednjimi je tudi stožčasto programiranje (Boyd, 2004).

**Stožčasto programiranje**

Problemi stožčaste optimizacije so eni izmed najenostavnejših primerov konveksne optimizacije s posplošenimi neenakostmi. Stožčasto programiranje optimizira linearno funkcijo *(13a*) z linearno omejitvijo *(13c)*, čemu je dodana ena ali več stožčastih omejitev (omejitev neenakosti v *(13b)*). Slednje pomeni, da je prostor rešitev (angl. *feasible space*) presek afinega prostora in konveksnega stožca (slika 4), ki je afin prostor (oz. K-konveksen) (Boyd, 2004).



min:

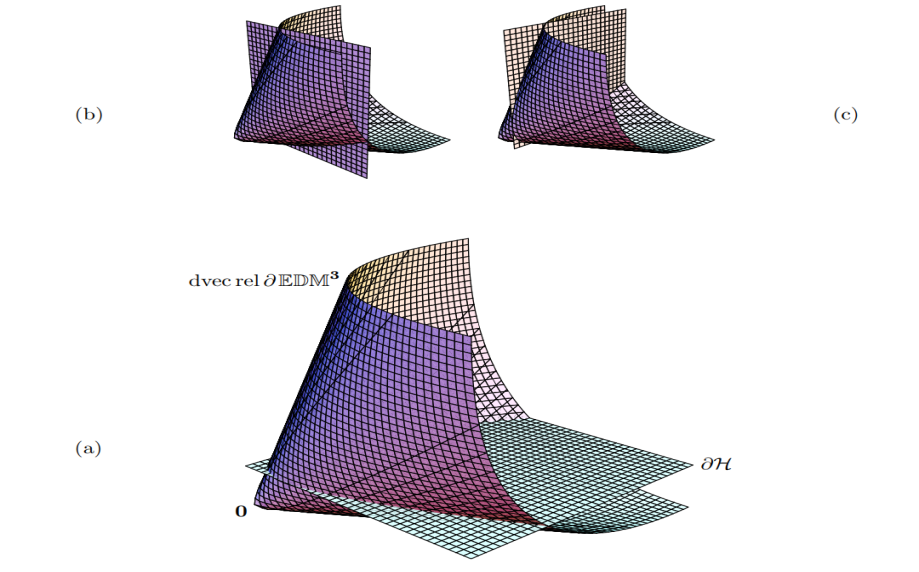
omejitve:

*(13a)*

*(13b)*

*(13c)*

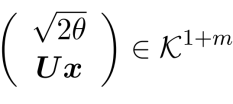
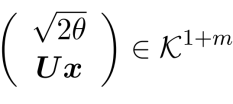
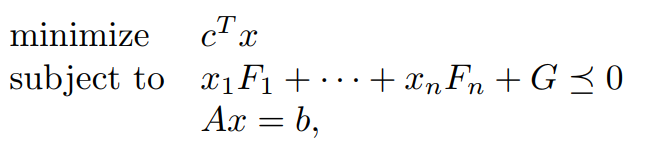
Če je *K* nenegativni orthant (ekvivalenten kvadrantu, vendar v višjih dimenzijah), se stožčast problem zreducira na linearen problem. Stožčasto programiranje je tako široka posplošitev linearnih programov (Boyd, 2004). Dva podrazreda sta tudi semidefinitno programiranje, ki ima kot omejitev semidefinitni stožec, in stožčasto programiranje drugega reda, ki ima kot omejitev Lorentzov stožec.



Slika 4: Presek konveksnega (semidefinitnega) stožca in afinega prostora (vir: Dattorro, 2005).

**Semidefinitno programiranje**

Semidefinitno programiranje (SDP) je stožčasto programiranje, ki minimizira linearno ciljno funkcijo z omejitvijo linearne neenakosti. Neenakost je linearna matrična neenakost (angl. *linear matrix inequality, LMI*), ki si lahko predstavljamo tudi kot presek stožca semidefinitne pozitivne matrike (slika 5) z afinim prostorom, kar imenujemo spektrahedron (Ramana, 1995). Tako je SDP optimiziranje linearne funkcije preko spektrahedrona. Standardna oblika SDP je tako:

; kjer so *G, F1,..., Fn* **S***k* in *A*  **R** *p x n.* Če so matrike diagonalne matrike, je problem zreduciran na linearen program (Boyd, 2004).

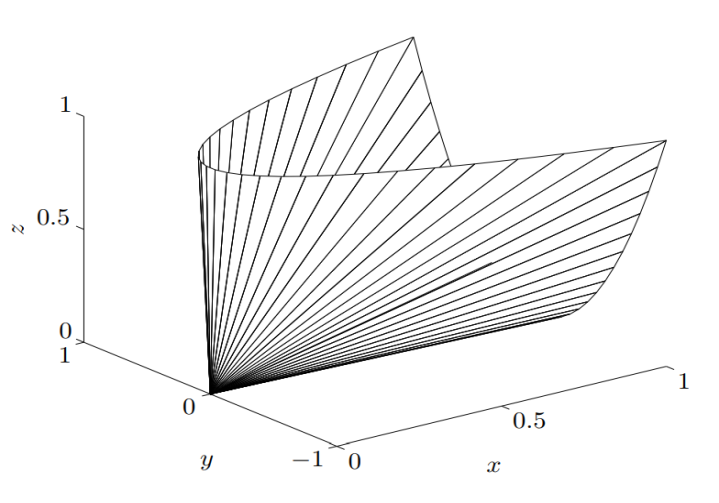
min:

omejitve: *F(x) ≥* 0 oz.

*(14a)*

*(14b)*

*(14c)*

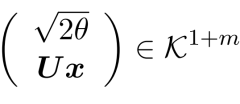
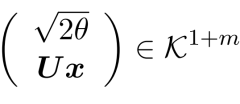
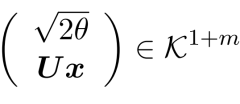
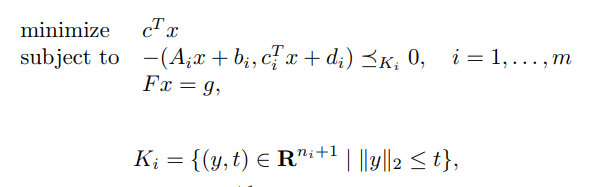


Slika 5: Omejitev s pozitivnim semidefinitnim stožcem v S2 (vir: Boyd, 2004).

Če je ena izmed omejitev kvadratna, jo moramo najprej prevesti na standardno formo oz. raven linearne matrične neenakosti. Da lahko kvadratne izraze vključimo v linearno matrično neenakost, jih moramo najprej transformirati v linearne izraze z uporabo Schurovega komplementa, omejitve pa združimo v bločno diagonalno matriko, kje je vsaka izmed omejitev zajeta v diagonalnih blokih. (Pong-Wong in Woolliams, 2007). Probleme semidefinintnega programiranja lahko učinkovitost rešimo z npr. metodami notranje točke. Slednja išče optimalno rešitev v strogo možnem prostoru parametrov. Glavni korak za izvedbo semidefinitnega programiranja je tako dokaz, da je problem konveksen in prevod problema na standardno obliko (Boyd, 2004).

**Stožčasto programiranje drugega reda**

Če je omejitev v obliki Lorentzovega stožca (tudi stožec drugega reda ali »kornet«) (slika 6), potem govorimo o stožčastem programiranje drugega reda. Stožčasto programiranje drugega reda tako optimizira linearno funkcijo *(15a)* preko preseka afinega prostora *(15c)* in stožca drugega reda *(15b).*

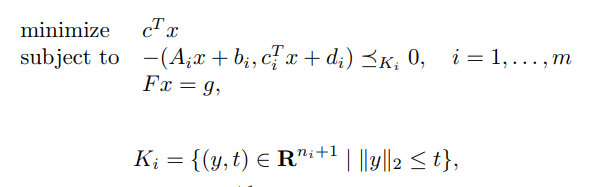
*; kjer je x* **R***n*spremenljivka, ki jo optimiziramo, *Ai* **R***ni x n* in *F* **R***p x n,*

*(15a)*

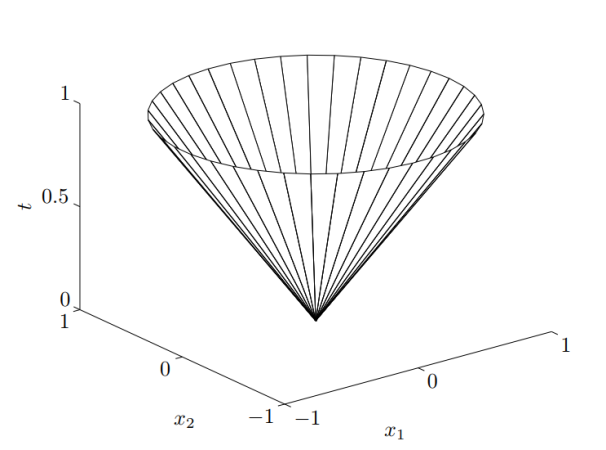
*(15b)*

*(15c)*

*(15d)*

****

pa je stožec drugega reda v **R***ni + 1* (Boyd, 2004)*.* Linearne programe, konveksne kvadratne programe in konveksne kvadratne programe s kvadratnimi omejitvami lahko vse formuliramo kot probleme stožčastega programiranja drugega reda.



Slika 6: Omejitev s stožcem drugega reda v R3 (vir: Boyd, 2004).

### 1.2.2 Razvoj rešitve za selekcijo z optimalnimi prispevki

Lindgren je prvi predlagal linearno razporeditev (angl. *linear deployment*), v kateri so prispevki kandidatov sorazmerni njihovim PV-jem (Lindgren in sod., 1989). Ta pristop zagotavlja, da imajo posamezniki z višjimi PV-ji večje prispevke, ne nadzira pa sorodstva med posamezniki. Metoda posledično zagotovi uspeh selekcije le v primeru, da so rodovniki neodvisni in posamezniki med seboj nesorodni, saj pristop ne omeji prispevkov posameznikov (Yamashita in sod., 2015). Napredek se tako skozi generacije občutno zmanjša zaradi depresije zaradi inbridinga in povečanja sorodnosti populacije.

Toro in Pérez-Enciso, 1990, sta uporabilo celoštevilsko linearno programiranje za optimizacijo odbire staršev in partnerjev za parjenje. Želena rešitev je v tem primeru matrika X ***=*** *[xij]*, kjer je *xij* binarna spremenljivka (0 ali 1) in predstavlja, ali sta oče *i* in mati *j* izbrana za parjenje. Matrika je izbrana z namenom maksimiranja genetskega napredka. Meuwissen je prvi uvedel kvadratno omejitev za nadzor skupnega sorodstva (Meuwissen, 1997). Optimizacijski problem za OCS tako predstavlja naslednji set enačb (Pong-Wong in Woolliams., 2007):

min: - *cT****g*** *(17a)*

omejitve: *c*T **A***c*/2 ≤ θ oz. *c*T **C***c* ≤ θ *(17b)*

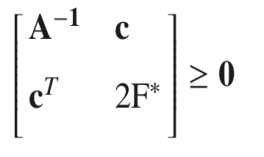
*cT*s = 0.5 *(17c)*

*cT*d = 0.5 *(17d)*

*c* ≥ 0 *(17e)*

*c* ≤ m *(17f)*

Ciljna funkcija je minimizacijska funkcija, ki minimizira negativni genetski napredek, kar je ekvivalentno maksimiranju genetskega napredka. S *c* označujemo prispevke posameznikov, *g* pa označuje vektor napovedanih PV. Z omejitvijo za skupno sorodstvo v enačbi *(17b)* zagotovimo, da skupno sorodstvo v naslednji generaciji ne preseže prednastavljene meje θ. Omejitev lahko izrazimo z matriko sorodstva **C** ali polovico le-te, ki je matrika A. Z omejitvami v *(17c)* in *(17d)* zagotovimo, da je prispevek vsakega izmed spolov ½ ter da je vsota prispekov 1. Omejitev v *(17e)* zagotavlja, da so prispevki nenegativni – prispevek 0 pomeni, da posameznik ni bil izbran kot starš naslednje generacije. Z omejitvijo v *(17f*) nastavimo zgornjo mejo za prispevek posameznika in s tem preprečimo preveliko uporabo ene živali. Meuwissen je problem rešil z metodo **Lagrangeovih multiplikatorjev** (Meuwissen, 1997). Čeprav se je ta metoda uporabljala v selekciji, pa ne zagotavlja vedno optimalne rešitve, saj se tekom iteracij določene vrednosti za prispevke fiksirajo na zgornjo ali spodnjo dovoljeno vrednost (Pong-Wong in Woolliams, 2007). Pong-Wong in Woolliams sta nato problem prevedla na problem **semidefinitnega programiranja** (Pong-Wong in Woolliams, 2007)**.** Izkoristila sta dejstvo, da je matrika sorodstva **C** **-** in posledično tudi matrika **A** **-** vedno pozitivno definitna, zaradi česar sta lahko omejitev za skupno sorodstvo prevedla na pozitiven semidefinitni pogoj s simetrično matriko. Za pretvorbo kvadratne omejitve sta uporabila Schurov komplement *(18a*), vse omejitve pa zapisala v obliki diagonalne bločne matrike. *(18b*). SDP najde optimalno rešitev - v nasprotju z Lagrangeovimi multiplikatorji - vendar pa zahteva veliko komputacijskega časa.

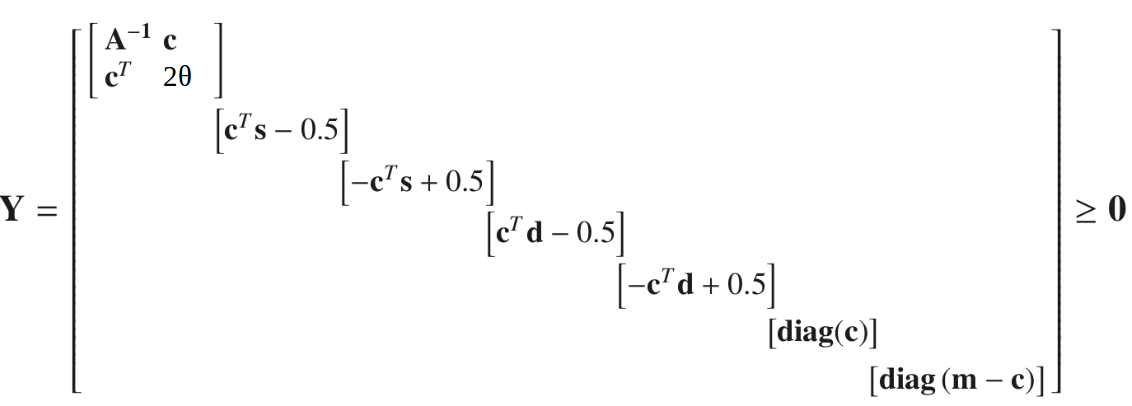


A-1 c

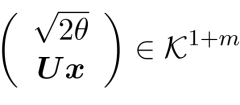
*cT 2θ*

cT Ac/2 ≤ θ →

*(18a)*



*(18b)*

Yamashita in sod., 2017, so predlagali problem prevesti na problem programiranja s stožci drugega reda. Ker je matrika A pozitivna definitna, lahko omejitev izrazimo tudi kot omejitev stožca drugega reda. Najprej nam pozitivno definitnost omogoči Cholesky faktorizacijo matrike A. S tem pridobimo zgornjo trikotno matriko, *A = UTU*, omejitev za skupno sorodstvo pa prevedemo na:

*cT Ac* / 2 ≤ θ ↔ cT *UTUc* ≤ 2*θ* ↔ ||*Uc|| ≤ √ 2θ* ↔

*c*

*(19)*

#### 1.2.1.2 Hevristični algoritmi

* 1. HEVRISTIČNo odločanje je odločanje v negotovosti
  2. V nasprotju z natančnimi metodami, ki vedno najdejo optimalno rešitev, hevrističnimi algoritmi ne more garantirati optimalne rešitve, vendar pa bo na podlagi izkušenj ali presoje zelo verjetno podala dobro rešitev..

Hevristične algoritme uporabimo, ko želimo zreducirati iskanje pri reševanju problema ali ko želimo znotraj omejenega komputacijskega časa pridobiti sprejemljivo rešitev. Pogosto jih imenujemo tudi pravilo palca, izobraženo ugibanj, intuitivna presoja ali preprosto kmečki razum. Hevristika je popularna tudi v operacijskih raziskavah, saj: a) velikokrat je malo verjetno, da bomo našli optimalno rešitev zaradi računalniške zahtevnosti problemov; b) če so problemi slabo definirani ali pa podatki netočni, najverjetneje najdena »optimalna« rešitev na bo optimalna za prave podatke. V takšnih primerih je bolje pridobiti robustno rešitev, ki je blizu optimalni. c) kadar uporabnik potrebuje več različnih rešitev za končno odločitev, še posebno v situacijah, kjer mora biti več kriterijev za uravnoteženih s človeškim faktorjem.

Uporaba hevristike je uporabna v naslednjih primerih (Zanakis in Evans, 1981):

a) netočni in pomanjkljivi podatki – ocena parametrov modela na podlagi le-teh lahko vsebuje večjo napako kot suboptimalna rešitev dobre hevristike.

b) uporabljamo poenostavljen model – le-ta je že tako ali tako netočna predstavitev problema, zaradi česar je »prava« rešitev zgolj akademska;

c) zanesljiva eksaktna metoda ni na voljo

d) eksaktna metoda je na voljo, vendar je računalniško neprivlačna – ali ima velike zahteve za računalniški čas ali pa spomin;

e) za izboljšanje performansa optimizacije, npr. za pridobitev začetne rešitve ali kot vodilo za zmanjšanje števila možnih rešitev

f) ponavljajoča se pogosta potreba po reševanju istega problema, pri čemer pomaga hitrost hevristike

g) hevristična rešitev je morda dovolj dobra – če je boljša od trenutnega

h) hevristika je enostavna, zato jih uporabnik razume in ji zaupa

i) kot naprava za učenje – pridobivanje vpogleda v kompleksne probleme

j) ostale omejitve sredstev – čas in denar, znanje ...

* 1. Glavna lastnosti hevrističnih optimizacijskih algoritmov je, da začnejo z neko poljubno rešitvijo, potem pa preko iteracij po nekem pravilu proizvajajo nove rešitve , ki jih ovrednotijo in sčasoma podajo najboljšo rešitev, ki so jo našli tekom iskanja. Iterativni proces je ponavadi ustavljen, ko: i) ne dano število iteracij uspemo izboljšati rešitve (algoritem konvergira); ii) ko je najdena rešitev dovolj dobra, iii) ko proces doseže dovoljen komputacijski čas; ali iv) ko nek interni parameter konča izvajanje procesa. file:///home/jana/Downloads/9780387258522-c2.pdf

Skupina hevrističnih algoritmov je raznolika v tipu uporabljenih metod. Prav tako obstajajo različne razvrstitve hevrističnih algoritmov, ki delijo algoritme glede na različne parametre, ena izmed bolj uporabnih pa je delitev glede na število sočasno uporabljenih rešitev. Tako jih delimo na trajektorne metode, ki uporabljajo eno rešitev, in populacijske rešitve, ki delujejo nad populacijo rešitev. Za trajektorne metode je značilno, da pri iskanju opišejo tirnico v iskalnem prostoru. Iskalni proces populacijskih algoritmov pa opisuje evolucijo množice točk v iskalnem prostoru. Ker bomo v nalogi uporabili algoritem, ki spada med populacijske metode, se bomo podrobneje osredotočili na to skupino (Korošec).

Med najbolj raziskanimi populacijskimi algoritmi sta optimizacija s kolonijami mravelj (angl. *Ant-Colony Optimisation*) in evolucijsko računanje (angl. *Evolutionary computation*). Oba opsnemata fenomena iz narave: medtem ko prvi posnema orientacijo mravelj na podlagi feromonov, pa drugi posnema proces naravne selekcije. Prvi algoritem uporabi metodo iz narave – mravlje so namreč izjemno učinkovite pri iskanju najkrajše poti, pri potovanju puščajo sledi feromonov, ki njih samim in drugim mravljam pomaga pri orientaciji. Mravlje sledijo feromonom, krajše poti dobijo več feromonov, saj mravlje osvežujejo feromonsko pot.

Evolucijsko računanje pa posnema koncept evolucije oz. naravne selekcije – imamo populacijo osebkov, v kateri okoljski pritiski povzročijo naravno selekcijo (preživetje najmočnejšega), kar povzroči naraščanje fitnesa populacije. Najprej naključno ustvarimo populacijo osebkov ter definiramo t.i. fitnes funkcijo, ki je ciljna funkcija, s katero bomo ocenili fitnes osebkov. Glede na to najboljše osebke odberemo kot starše prihodnje generacije, ki jo ustvarimo z mutacijo in / ali rekombinacijo dveh staršev. V naslednjem koraku tako starše kot tudi potomce ovrednotimo s fitnes funkcijo in le najboljše prenesemo v naslednjo generacijo. Vsaka iteracija algoritma ustvari novo generacijo osebkov, vsaka osebek pa predstavlja možno rešitev. Potek evolucijskega algoritma lahko predstavimo z naslednjo shemo:

**inicializiraj** populacije (naključna)

**osebki** (kandidatne rešitve)

**ovrednoti** vseh osebkov (s fitnes funkcijo)

DOKLER ne STOP delaj

**izberi** starše in populacije

ustvari potomce z mutacijo in rekombinacijo staršev

**ovrednoti** novorojene potomcev

**zamenjaj** nekatere izmed staršev s potomci

Uspeh evolucijskih algoritmov je tako močno odvisen od upravljanja populacije. V splošnem sta v evolucijskem algoritmu dve gonilni sili: selekcija, ki si prizadeva za kakovost in zmanjšuje genetsko variabilnosti populacije, in variacija – implementirana kot mutacija in rekombinacija - ki si prizadeva za novosti in povečuje genetsko variabilnosti. Zato je uravnoteženje teh dveh sil ključno za uspeh evolucijskega algoritma (Eiben).

**OCS in evolucijski algoritem**

Gorjanc in Hickey, 2018, sta za rešitev OCS uporabila evolucijski algoritem, in sicer algoritem diferencialne evolucije. Slednji vsebuje naslednje korake:

1. **inicializacija** začetne populacije, kjer naključno definiramo N število osebkov – **X***iG -* oz. rešitev:  
   **X***i, G =* {*x0, x1, …, xj*}; 0 ≤ *i < N, 0* ≤ *j < D, G = 0*

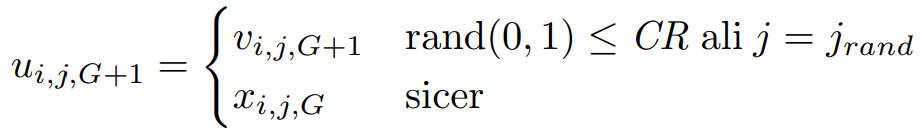
; kjer je *xj j-*ta neznanka v problemu, *N* je število osebkov znotraj populacije, *D* je dimenzija problema, *G* je generacija.

1. **mutacija:** za vsakega osebek x*i,G*, ki ga imenujemo tudi **ciljni vektor** (angl. target *vector* ali *target vector*), naključno izberemo dodatne tri osebke *r1, r2 in r2,* tako, da so indeksi *r1, r2* in *r3* različni od *i*. Iz njih ustvarimo t.i. **mutiran vektor** (angl. d*onor vector*):

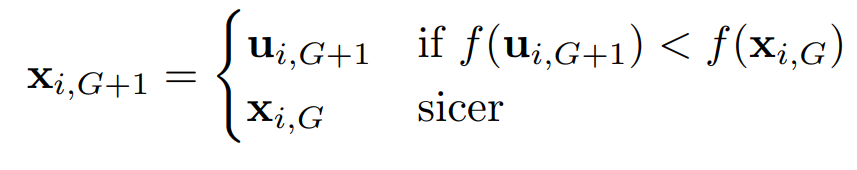
**v***i, G* = **x***r1, G  + F x (***x***r2,G –* **x***r3,G*);

kjer **x***r1, G* imenujemo osnovni vektor (angl. b*ase vector*), izraz *(***x***r2,G –* **x***r3,G*) pa diferenčni vektor (angl. *difference vector*). *F* je mutacijski faktor in zavzema vrednosti [0, 2]. Slednji nam uteži diferenči vektor, ki se zatem imenuje utežen diferenčni vektor (angl. *weighted difference vector*).

1. **rekombinacija:** z rekombinacijo ciljnega in mutiranega vektorja ustvarimo t.i. **poskusni vektor** (angl. t*rial vector*) *ui,j, G+1*

; kjer je rand*j,i* ∼ *U*[0, 1], *jrand* pa naključno število iz [1,2,..., *D*]. *jrand zagotovi, da vi,G+1 ≠xi,G.*

1. **selekcija:** na tej točki ciljni in poskusni vektor ocenimo z ocenitveno funkcijo *f.* Če je poskusni vektor boljši od ciljnega, potem preizkusni vektor v populaciji zamenja ciljnega.



## 1.3 RAZISKOVALNI PROBLEM

Genomska selekcija omogoča večji genetski napredek zaradi skrajšanega generacijskega intervala in večje točnosti zgodnjih selekcijskih odločitev (Schaeffer, 2006). Posledica krajšega generacijskega intervala je hitrejše obračanje generacij. Ohranjanje oz. optimalno upravljanje z genetsko variabilnostjo postane tako še pomembnejše za trajnostno upravljanje s populacijo in dolgoročno doseganje genetskega napredka. Pri tem še vedno ostaja vprašanje, na kakšno vrednost omejiti skupno sorodstvo oz. kako uravnotežiti ohranjanje genetske variance relativno na doseganje genetskega napredka, da bo scenarij učinkovitejši kot scenarij brez optimiziranja genetskih prispevkov.

Čeprav genomska selekcija omogoča večji genetski napredek, je njena uvedba predvsem v majhnih populacijah otežena zaradi majhnega števila živali za referenčno populacijo in omejenih sredstev. V Sloveniji se nekatere pasemske organizacije vključujejo v mednarodne organizacije za napoved genomskih plemenskih vrednosti, druge pa takšne možnosti nimajo. Zato je vprašanje tudi, ali bi z OCS lahko dosegli večjo učinkovitost in genetski napredek slovenske genomskega selekcijskega scenarija in s tem omogočili mednarodno konkurenčnost slovenske selekcije.

Kot predstavljeno obstaja za optimizacijo prispevkov v selekciji veliko različnih algoritmov, ki se razlikujejo v načinu optimizacije, računalniškem času, spominu in možnosti nastavitve ciljev optimizacije. Kot videno tudi za druge primere problemov v genomski selekciji, kot npr. napoved gPV, se lahko rezultati različnih algoritmov razlikujejo. Zato nas je zanimalo, kako primerljivi so algoritmi – specifično deterministično stožčasto programiranje in hevristični evolucijski algoritem – med seboj.

## 1.4 RAZISKOVALNI CILJI

Splošni cilj naloge je analiza vpliva uporabe optimizacije prispevkov na genetski napredek in učinkovitost selekcije v simulirani populaciji slovenske rjave pasme. To bo doseženo skozi naslednje cilje:

* razvoj orodja za stohastično simulacijo specifičnega rejskega programa s prekrivajočimi generacijami, ki omogoča tudi optimizacijo prispevkov;
* analiza slovenskih rejskih programov;
* priprava alternativnih scenarijev selekcije z optimiziranimi prispevki;
* primerjava slovenskega genomskega scenarija in OCS scenarijev: primerjava genetskega napredka, genetske variance in selekcijske učinkovitosti;
* primerjava OCS scenarijev, ki uporabijo determinističen ali hevrističen algoritem za optimizacijo: primerjava genetskega napredka, genetske variance in selekcijske učinkovitosti.

# 2 METODE IN MATERIALI DELA

Najprej smo razvili orodje za stohastično simulacijo, ki nam je omogočalo nastavitev vseh selekcijskih parametrov in s tem simulacijo slovenske populacije govedi. Testirali smo šest scenarijev z ali brez optimiziranja prispevkov živali, od tega je en scenarij posnemal genomsko selekcijo v slovenski populaciji s trenutnim režimom odbire živali. Preostalih pet scenarijev je implementiralo optimizacijo prispevkov očetov z različnimi relativnimi težami za maksimiranje genetskega napredka in minimiziranje skupnega sorodstva. Testirane scenarije smo primerjali na podlagi doseženega genetskega napredka, izgubljene genske variance in dosežene učinkovitosti selekcije.

## 2. 1 PRIPRAVA ORODJA IN PARAMETROV ZA SIMULACIJO

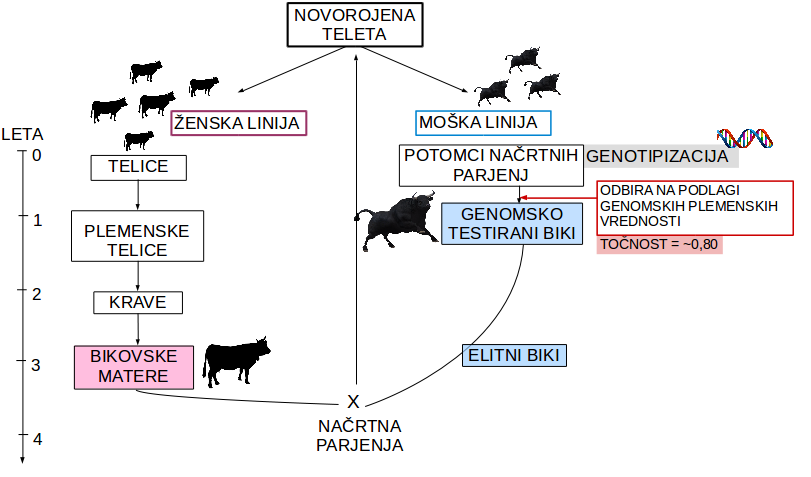
Najprej smo razvili simulator populacije govedi v selekciji. Uporabnik lahko nastavi vse selekcijske parametre, kar mu omogoča selekcijo specifične populacije govedi. Simulator je zgrajen kot Python ogrodje, ki povezuje programe za izvedbo vseh korakov enega kroga selekcije:

1. AlphaSim (Faux in sod., 2016) za stohastično simulacijo rodovnika in genoma živali,
2. blupf90 (Misztal in sod., 2002) za ocenitev genomskih plemenskih vrednosti,
3. R program, ki uporabi R-ov paket optiSel (Wellman, 2017) ali AlphaMate za optimizacijo prispevkov in
4. Python script za določitev staršev nove generacije potomcev.

V naši študiji so bili vsi parametri povzeti po slovenski populaciji rjave pasme govedi, ki šteje ~30.000 aktivnih živali, od tega 10.000 krav. Simulirani genomski podatki vključujejo 10 kromosomov, iz katerih je bilo 10.000 mest izbrranih za vzročna mesta za lastnost s heritabiliteto 0,25. Da smo dosegli primerno začetno strukturo populacije, smo najprej smo simulirali 20 generacij selekcijo z naključno odbiro staršev, nato pa še 20 generacij klasične selekcije (selektivna odbira). Temu je sledila simulacija 20 generacij za vsakega izmed testiranih scenarijev.

## 2.2 TESTIRANI SCENARIJI

V raziskavi smo analizirali šest scenarijev selekcije z ali brez optimizacije prispevkov. Vsi testirani scenariji so implementirali genomsko selekcijsko shemo, v kateri je bila genomska informacija uporabljena le za odbiro očetov, matere pa so bile odbrane po klasični selekcijski shemi (slika 7). Izmed vseh korakov v selekcijski shemi smo optimizirali le korak odbire in prispevkov očetov. V scenariju brez optimizacije prispevkov so bili parametri odbire očetov – število očetov, leta v uporabi in prispevki - povzeti po slovenskih rejskih programih (SLO\_selekcija). Število odbranih očetov in leta v uporabi so bili stalni in prednastavljeni, prispevki pa so bili le navzgor omejeni. V scenariju slovenske genomske selekcije so bili očetje odbrani le na podlagi njihovih PV, brez obzira na sorodstvo z ostalimi živalmi (selektivna odbira). Preostalih pet scenarijev je vključevalo korak optimizacije prispevkov očetov v skladu z OCS. Število izbranih očetov, leta v uporabi in prispevki tako niso bili stalna. V teh scenarijih so bili očetje odbrani na podlagi njihovih PV in sorodstva z ostalimi živalmi. V vsakem izmed petih optimiziranih scenarijev smo drugače uravnotežili maksimiranje genetskega napredka in nadzor skupnega sorodstva, kar smo izrazili s t.i. tarčnimi stopinjami. Testirali smo pet tarčnih stopinj, in sicer 15 (OCS15), 30 (OCS30), 45 (OCS45), 60 (OCS60), in 75 (OCS75).



Slika 7: Testirana genomska selekcija shema. Genomska informacija je bila uporabljena je za odbiro očetov.

## 2.3 OPTIMIZACIJA PRISPEVKOV

## 2.3.1 DETERMINISTIČNA OPTIMIZACIJA S STOŽČASTIM PROGRAMIRANJEM DRUGEGA REDA

V optimiziranih scenarijih smo optimizirali le korak odbire in prispevkov genomsko testiranih očetov (slika 7). Prispevke smo optimizirali v vsakem izmed 20 krogov selekcije. Prispevkov mater nismo optimizirali, prav tako nismo optimizirali določitve partnerjev za parjenje. Med potencialne očete smo vključili vse selekcijske kandidate ter že odbrane genomsko testirane bike iz obdobja zadnjih petih let. Da bi skrajšali optimizacijski čas, smo v optimizacijo vključili naključno izbran vzorec 20 % ženskih živali in nato izbrane očete sorazmerno uporabili na celotni populaciji. Za optimizacijo prispevkov očetov smo uporabili: i) deterministično stožčasto programiranje drugega reda, ki je implementirano v R-ovem paket optiSel (Wellman, 2017); ii) hevristično optimizacijo z evolucijskim algoritmom diferencialne evolucije, ki je implementiran v programu AlphaMate (Gorjanc in Hickey, 2018).

Pri obeh programih so bile želena tarča optimizacije t.i. trigonometrične stopinje, ki opisujejo razmerje med genetskim napredkom in izgubo genetske variabilnosti (slika 8). Podrobneje - merijo kot med rezultatom selekcije s ciljem maksimiranja genetskega napredka brez omejitve za sorodstvo (0 stopinj) in selekcije s ciljem minimiziranja sorodstva brez obzira na genetski napredek (90 stopinj). Manjše stopinje tako pomenijo, da dajemo relativno večji poudarek maksimiranju genetskega napredka, večje pa, da dajemo prednost minimiziranju skupnega sorodstva populacije (Gorjanc in Hickey, 2018). Trigonometrične stopinje lahko izračunamo iz maksimalnega odstotka napredka, ki je normalizirana vrednost za genetski napredek, ali minimalnega odstotka sorodstva, ki je normalizirana vrednost za skupno sorodstvo. Trigonometrične stopinje tako izračunamo kot:

*acos(maksimalni odstotek napredka / 100) x 180/π*  *(20a)*

ali:

*asin(minimalen odstotek sorodstva / 100) x 180/π. (20b)*

V vsakem krogu selekcije je optimizacija potekala v treh stopnjah:

1. optimizacija za maksimiranje genetskega napredka, iz česar smo pridobili maksimalen odstotek napredka;
2. optimizacija za minimiziranje sorodstva, iz česar smo pridobili minimalen odstotek sorodstva;
3. optimizacija za tarčne stopinje, pri čemer je šlo za maksimiranje genetskega napredka glede na omejitev sorodstva izračunanega na podlagi 1), 2) in tarčnih stopinj.

AlphaMate v vsaki iteraciji algoritma s ciljno funkcijo preveri vse rešitve, ali so njihove stopinje pod dovoljenimi. Rešitve, ki tega pogoja ne izpolnjujejo, so penalizirane in tako malo izločene iz boja za vstop v naslednjo generacijo rešitev. OptiSel pa ped začetkom optimizacije na podlagi izračuna maksimalnega dovoljenega sorodstva iz 3) nastavi omejitev za sorodstvo in nato maksimira genetski napredki ob omenjeni omejitvi. V raziskavi smo simulirali pet scenarijev uravnotežene selekcije za vsakega izmed programov optiSel in AlphaMate s tarčnimi stopinjami 15, 30, 45, 60 in 75.

## 

Slika 8: Prikaz trigonometričnih stopinj kot razmerje med maksimalnim genetskim napredkom in minimalnim sorodstvom (vir: Gorjanc in Hickey, 2018).

## 2. 4 PRIMERJALNI PARAMETRI

Po simulaciji smo testirane scenarije med seboj primerjali glede na dosežen letni genetski napredek, izgubo genetske variance in selekcijsko učinkovitost. Letni genetski napredek smo pridobili z regresijo povprečne PV na generacijo. Genetski napredek smo za vsak scenarij standardizirali tako, da je imel povprečje 0, izražen je pa bil v enotah standardnega odklona v prvi generaciji primerjave. Gensko varianco smo standardizirali tako, da je imela v prvi generaciji primerjave vrednost 1. Učinkovitost selekcije smo definirali kot naklon regresije genetskega napredka na izgubo genskega standardnega odklona. Tako genetski napredek kot tudi genski standardni odklon sta bila standardizirana z genskim standardnim odklonom v prvi generaciji primerjave. Učinkovitost selekcije tako izraža, kakšen genetski napredek bi dosegli (v enotah genskega standardnega odklona), če bi se vsa genetska variabilnosti pretvorila v genetski napredek.

# 3 REZULTATI IN RAZPRAVA

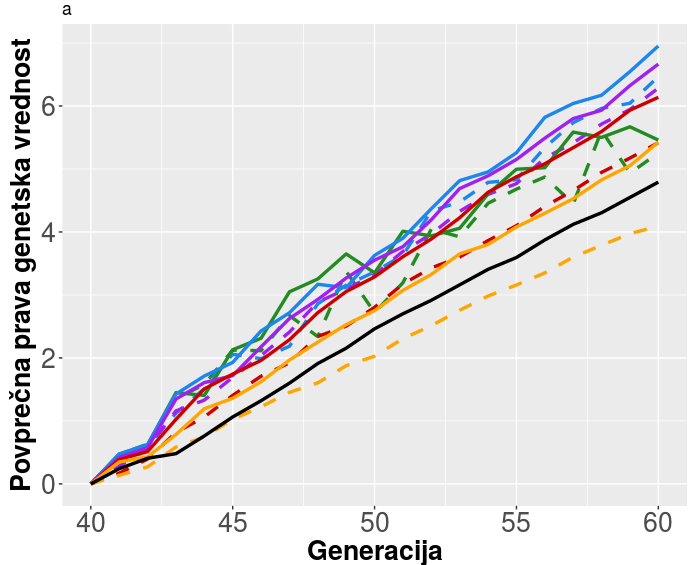
Pri testiranih scenarijih smo primerjali genetski napredek, gensko varianco in učinkovitost po 20 generacijah selekcije. Najprej smo med seboj primerjali scenarij slovenske genomske selekcije in scenarije z optimiziranimi prispevki (OCS scenariji). Ugotovili smo, da lahko z optimizacijo prispevkov v slovenski populaciji govedi dosežemo večji genetski napredek kot genomska selekcija s trenutnim režimom dobire, pri čemer se genetski napredek načeloma veča z manjšanjem tarčnih stopinj. Nasprotno temu smo ugotovili, da scenariji z večjimi stopinjami, t.i. večjo težo na minimiziranju skupnega sorodstva, izgubijo manj genetske variabilnosti, prav tako pa je pri ohranjanju genetske variabilnosti uspešen tudi slovenski genomski selekcijski scenarij. Pri analizi učinkovitosti smo ugotovili, da scenariji z največjim genetskim napredkom niso najbolj učinkoviti ter da je le OCS z največjimi tarčnimi stopinjami bolj učinkovita od slovenske genomske selekcije.

Nato smo med seboj primerjali še rezultate dveh optimizacijskih programov, in sicer optiSel (deterministična optimizacija) in AlphaMate (hevristična optimizacija). Ugotovili smo, da AlphaMate pri istih tarčnih stopinjah prinaša manj genetskega napredka, vendar pa ohrani več genetske varibiabilnosti kot optiSel. Vseeno pa je učinkovitost selekcije v AlphaMate scenarijih manjša od učinkovitosti v optiSel scenarijih.

## 3.1 Genetski napredek

Rezultati za genetski napredek so prikazani na sliki 9. Če najprej primerjamo rezultate slovenske genomske selekcije in selekcije z optimiziranimi prispevki s slovensko genomsko selekcijo, vidimo, da lahko z optimizacijo prispevkov povečamo genetski napredek. Genetski trend slovenske genomske selekcije je na sliki 9 prikazan s črno i prikazuje potencial ob obstoječem režimu odbire, t.j. 0,247 enot genetskega napredka letno (4,79 v generaciji 60). Z optimizacijo prispevkov očetov lahko genetski napredek povečamo za 8,6 do 37,5 %. Pri primerjavi OCS scenarijev smo ugotovili, da genetski napredek narašča z manjšanjem tarčnih stopinj. Slednje je bilo pričakovano, saj se z manjšanjem stopinj veča relativna teža za maksimiranje genetskega napredka pred minimiziranjem skupnega sorodstva. Posledično izmed OCS scenarijev najmanj genetskega napredka prinaša AlphaMate75, in sicer 0,213 (standardni odklon = 0,0122) enot letnega genetskega napredka, v generaciji 60 pa doseže 4,09 enot. Največ napredka prinaša optiSel30, in sicer 0,340 enot (6,95 enot v generaciji 60) kar je 60 % več kot AlphaMate75. Čeprav je teža za genetski napredek največja pri 15 stopinjah, pa ta scenarija – ne med optiSel in ne med AlphaMate scenariji - ne dosegata največjega genetskega napredka. Slednje je posledica močnega pritiska na odbiro najboljših posameznikov, zato je v tem scenariju število očetov zelo majhno. Medtem, ko je pri tarči 75 stopinj povprečno število očetov na generacijo 32 v optiSel in 25 v AlphaMate scenariju, to pri 15 stopinjah znaša le 4 v optiSel in 3 v AlphaMate scenariju. V določeni generaciji so v optiSel15 in AlphaMate15 tako vsi novorojeni osebki potomci parih očetov, kar povzroči velika nihanja v povprečni genetski vrednostmi med generacijami - standardni odklon je 0,0282 v optiSel15 in 0,0390 v AlphaMate15. Ker pa so očetje odbrani na podlagi plemenske vrednosti, točnost katere pa je ~0,6, lahko odbira ni dejansko optimalna, kar povzroči padec povprečne genetske vrednosti populacije v določeni generaciji. Rezultati so razkrili tudi splošen trend večanja standardnega odklona povprečne genetske vrednosti s padajočimi stopinjami, saj večji poudarek na maksimiranju genetskega napredka odbira manjše število očetov - standardni odklon narašča od 0,0163 in 0,0122 pri optiSel75 in AlphaMate75 do 0,0282 in 0,0390 pri optiSel15 in AlphaMate15. Vidimo tudi, da začetni superiornosti scenarijev s tarčnimi stopinjami 15 navkljub, začne letni genetski napredek po približno 10 generacijah padati (pri AlphaMate15 še prej), scenarij pa izgubi relativno prednost pred ostalimi. Razlog za to je velika izguba genetske variabilnosti, ki onemogoča dolgoročni genetski napredek.

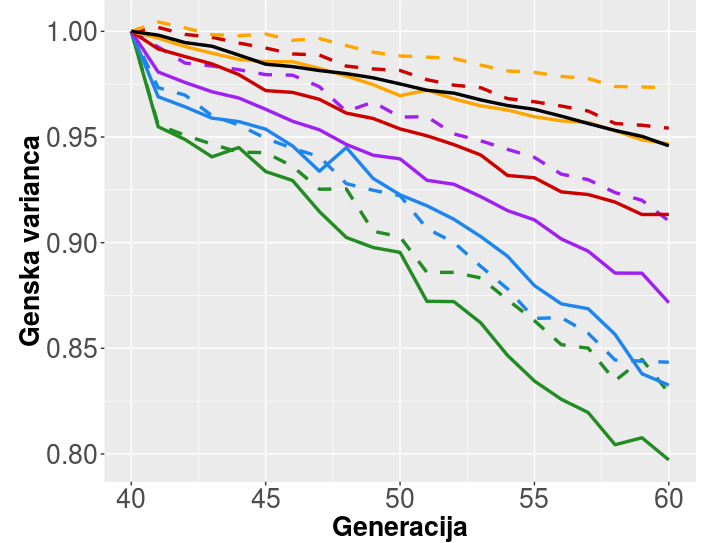
Če nato primerjamo še oba programa, opazimo, da prihaja do razlik med rezultati, pridobljenih s stožčastim programiranjem (optiSel) in hevrističnim evolucijskim algoritmom (AlphaMate). Opazimo, da pri istih tarčnih stopinjah AlphaMate prinaša manjši genetski napredek (merjen v generaciji 60). Relativna prednost optiSel rešitev pred AlphaMate se načeloma manjša z manjšanjem stopinj. Tako je največja pri največjih stopinjah, t.j. 75, in sicer 33% (5,43 optiSel in 4,09 AlphaMate); najmanjša pa je pri najmanjših stopinjah, t.e. 15, in sicer 1,1 % (5,67 optiSel in 5,61 AlphaMate).

Slika 9: Povprečen genetski napredek testiranih scenarijev po generacijah. optiSel15 – 75: selekcija z optiSel-optimiziranimi prispevki s tarčo 15 – 75 stopinj, AlphaMate15 – 75: selekcija z AlphaMate-optimiziranimi prispevki s tarčo 15 – 75 stopinj, SLO\_selekcija = slovenski genomski selekcijski scenarij s trenutnim režimom odbire.

## 3.2 GENETSKA VARIABILNOST

Rezultati za izgubo genetske variabilnosti, merjene kot genska varianca, so prikazani na sliki 10. Če zopet najprej primerjamo slovenski genomski scenarij s scenariji selekcije z optimiziranimi prispevki, vidimo, daje slovenski scenarij relativno uspešen pri ohranjanju genetske variabilnosti skoz generacije. Slednji izgubi 5,4 % prvotne genske variance in je primerljiv z najboljšim optiSel scenarijem (optiSel75), ki izgubi 5,3% genske variance in drugim najboljšim AlphaMate scenarijem (AlphaMate 60), ki izgubi 4,6 % prvotne genske variance. V skladu s pričakovanji imajo scenariji z večjimi tarčnimi stopinjami – ki pomenijo večjo težo na minimiziranju skupnega sorodstva – manjšo izgubo genske variance. Manjše stopinje pomenijo manjšo težo na ohranjanju genetske variabilnosti v optimizaciji, kar povzroči večjo izgubo genske variance. Tako znotraj optiSel in AlphaMate scenarijev tako najmanj genske variance izgubita scenarij s 75 stopinjami, in sicer izgubi optiSel75 izgubi 5,3 % prvotne genske variance, AlphaMate75 pa 2,7 %. Izguba variance nato monotono narašča s padanjam stopinja - največ genske variance tako izgubita scenarija s 15 stopinjami, in sicer optiSel15 v 20 generacijah izgubi 20,3 % prvotne genske variance , kar je 3,6-krat več kot optiSel75, AlphaMate15 pa izgubi 17,3 % genske variance, kar je 6,4-krat več kot AlphaMate75.

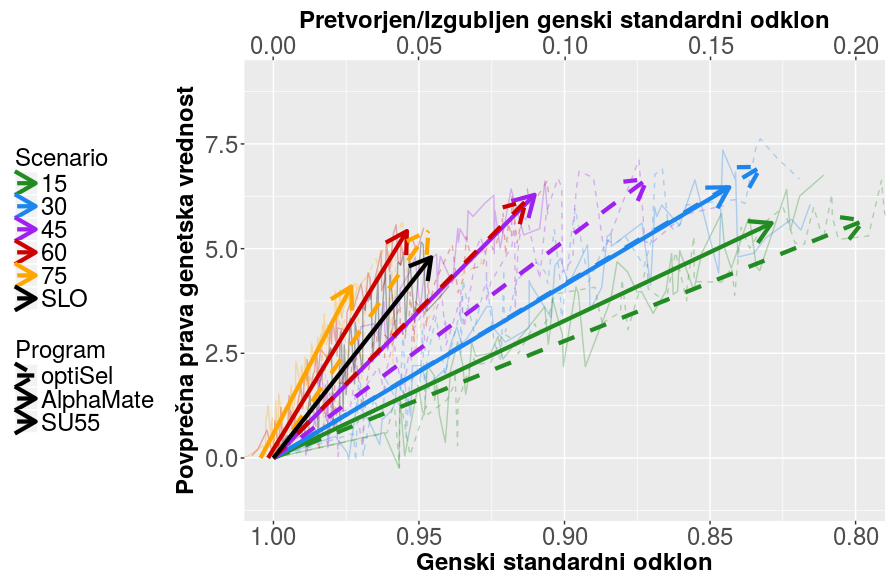
Pri primerjanju različnih algoritmov za optimizacijo, ki sta implementirana v optiSel in AlphaMate, opazimo tudi, da med njima prihaja do razlik. Tukaj vidimo obratno zgodbo kot pri doseganju genetskega napredka, saj je AlphaMate pri vseh tarčnih stopinjah bolj uspešen pri ohranjanju genetske variabilnosti kot optiSel - slednje pomeni, da je AlphaMate bolj konzervativen. Najmanj genske variance tako izgubita scenarija AlphaMate75 in AlphaMate60, ki izgubita 2,7 % in 4,6 % prvotne genske varianca. Slednja scenarija sta tudi edina bolj uspešna pri ohranjanju genske variance kot slovenski genomski scenarij.



Slika 10: Povprečna genska varianca testiranih scenarijev po generacijah. optiSel15 – 75: selekcija z optiSel-optimiziranimi prispevki s tarčo 15 – 75 stopinj, AlphaMate15 – 75: selekcija z AlphaMate-optimiziranimi prispevki s tarčo 15 – 75 stopinj, SLO\_selekcija = slovenski genomski selekcijski scenarij s trenutnim režimom odbire.

## 3.3 UČINKOVITOST SELEKCIJE

Na podlagi genetskega napredka in izgube genske variance – oz. genskega standardnega odklona - smo izračunali tudi učinkovitost selekcije za vse scenarije (slika 11). Učinkovitost scenarijev se giblje med 28 in 120 enot. Vidimo, da lahko z optimizacijo prispevkov očetov izboljšamo učinkovitost slovenske genomske selekcije. Učinkovitost slovenske genomske selekcije je 89 enot, boljšo učinkovitost pa dosežejo scenariji AlphaMate75, AlphaMate60 in optiSel75 z učinkovitostjo 120, 102 in 100 enot. Iz grafa je razvidno, da učinkovitost OCS scenarijev narašča z naraščanjem stopinj. Slednje je posledica manjše izgube genskega standardnega odklona, ki je relativno večja kot povečanje genetskega napredka pri scenarijih z manjšimi stopinjami. Najbolj učinkovit je tako scenarij AlphaMate75, ki doseže učinkovitost 120 enot. Najmanj učinkovit je scenarij optiSelS15, ki doseže učinkovitost 28 enot, kar je 73 % manj od najbolj učinkovitega scenarija. Majhna učinkovitost scenarija z večjimi tarčnimi stopinjami je posledica velike izgube genske variance oz. standardnega odklona. Opazimo tudi, da čeprav sta bila scenarija OCS75 in slovenski genomski selekcijski scenarij po izgubi genske variance zelo primerljiva, pa se po učinkovitostih razlikujeta. Slovenski selekcijski scenarij ima manjšo učinkovitost, in sicer 89 enot. Razlika je posledica dejstva, da navkljub isti izgubi genetske variabilnosti, scenarij OCS75 doseže večji genetski napredek. Prav tako vidimo, da čeprav je OCS30 imel največji genetski napredek, pa ni med najbolj učinkovitimi scenariji, saj porabi tudi relativno veliko genske variance oz. standardnega odklona.

Slika 11: Povprečna učinkovitost selekcije testiranih scenarijev. optiSel15 – 75: selekcija z optiSel-optimiziranimi prispevki s tarčo 15 – 75 stopinj, AlphaMate15 – 75: selekcija z AlphaMate-optimiziranimi prispevki s tarčo 15 – 75 stopinj, SLO\_selekcija = slovenski genomski selekcijski scenarij s trenutnim režimom odbire.

# 4 zaključki

Selekcija stremi h genetskemu izboljšanju populacije. Poleg maksimiranja genetskega napredka je v za trajnostni uspeh selekcije in dolgoročni genetski napredek ključno tudi ohranjanje genetske variabilnosti za lastnosti v populaciji. Selekcija z optimiziranimi prispevki združuje cilj maksimiranja genetskega napredka z omejitvijo skupnega sorodstva populacije. Razvoj rešitve za selekcijo z optimiziranimi prispevki se je razvijal od linearnega celoštevilskega programiranja, preko rešitve z Lagrangeovimi multiplikatorji, do semidefinitnega programiranje in nazadnje stožčastega programiranja drugega reda.

V naši raziskavi smo želeli analizirati uspeh selekcije z optimiziranimi prispevki v slovenski populaciji govedi in rezultate primerjati z uspehom slovenske genomske selekcije s trenutnim realnim režimom odbire. Razvoj orodja za stohastično simulacijo nam je omogočal nastavitev vseh selekcijskih parametrov in s tem simulacijo genomske selekcije v slovenski populaciji govedi. Poleg scenarija, ki posnema slovensko genomsko selekcijo, smo testirali pet uravnoteženih scenarijih z različnim razmerjem tež za maksimiranje genetskega napredka oz. minimiziranje skupnega sorodstva, kar je izraženo v t.i. tarčnih stopinjah.

Ugotovili smo, da OCS prinaša večji genetski napredek kot slovenska genomska selekcija. Kot pričakovano smo ugotovili, da večja teža na maksimiranju genetskega napredka, i.e. manjše tarčne stopinje, načeloma prinaša večji genetski napredek. Izjema je uravnotežena selekcija z najmanjšimi stopinjami (OCS15), ki začne po desetih generacijah izgubljati genetski napredek zaradi izgube genetske variabilnosti. To potrjuje, da za dolgoročni uspeh selekcije ni pomembna le odbira najboljših živali za večji genetski napredek, ampak je pomembna tudi sorodnost teh živali, da se v populaciji ohranja genetska variabilnost. Rezultati so pokazali tudi, da je izguba genetske variabilnosti, merjene kot genske variance, večja pri manjših tarčnih stopinjah. Slednje je pričakovano, saj manjše stopinja pomenijo večjo težo na ohranjanje genetske variabilnosti. Ugotovili smo tudi, da je slovenska genomska selekcija uspešna pri ohranjanju genetske variabilnosti, saj so bili rezultati primerljivi z OCS75, ki ohrani največ genske variance.

Na koncu smo rezultate za genetski napredek in gensko varianco povezali v merilo učinkovitosti selekcije. Ugotovili smo, da so OCS scenariji z večjimi stopinjami bolj učinkoviti. Slednji scenariji sicer manjši genetski napredek, vendar pa hkrati izgubijo manj genske variance oz. standardnega odklona – kar pomeni večjo učinkovitost. Čeprav sta imela OCS30 in OCS15 največja genetska napredka, sta po učinkovitosti na zadnjih dveh mestih, kar je posledica relativno velike izgube genskega standardnega odklona. Slovenska genomska selekcija je po učinkovitosti na drugem mestu, kar pomeni, da bi lahko z uvedbo OCS še izboljali učinkovitost trenutnega režima odbire živali.

V raziskavi smo pokazali možnost uporabe OCS v slovenski populaciji govedi. Slednja lahko prinese kar za 45 % večji genetski napredek, vendar pa bistveno ne izboljša učinkovitosti selekcije. Rezultati te raziskave bodo lahko v pomoč slovenskim rejskim organizacijam pri pripravah rejskih programov. V prihodnje bi lahko raziskavo ponovili še s katerim izmed preostalih optimizacijskih algoritmov, s čimer bi pridobili podatek o primerljivosti in zanesljivosti rezultatov. Prav tako bi lahko optimizirali korak parjenja oz. določitve partnerja za parjenje, za kar pa bi za tako veliko populacijo potrebovali večje komputacijske kapacitete.

# 5 Viri in literatura

Boyd S., Vandenberghe L. 2004. Convex Optimization. New York, ZDA, Cambridge University Press: 716 str.

Dattorro J. 2005. Convex Optimization & Euclidian Distance Geometry. Palo Alto, Kalifornija, ZDA, Meboo Publishing USA: 690 str.

Falconer D.S., Mackay, T.F.C. 1996. Introduction to Quantitative Genetics. 4. izdaja. Harlow, UK, Longman: 464 str.

Faux A.-M., Gorjanc G., Gaynor R.C., Battagin M., Edwards S.M., Wilson D.L., Hearne, S.J., Gonen S., Hickey J.M. 2016. AlphaSim: Software for Breeding Program Simulation. Plant Genome, 9, 3.

Gorjanc G., Hickey J.M. 2018. AlphaMate: a program for optimising selection, maintenance of diversity, and mate allocation in breeding programs. bioRxiv 250837.

Grundy B., Villanueva B., Woolliams J.A. 1998. Dynamic selection procedures for constrained inbreeding and their consequences for pedigree development. Genetics Research, 72, 2: 159–168.

James J.W., McBride G. 1958. The spread of genes by natural and artificial selection in closed poultry flock. Journal of Genetics, 56, 55–62.

Kinghorn B.P., Shepherd R.K. 1999. RK: Mate selection for the tactical implementation of breeding programs. Proceedings of the Association for the Advancement of Animal Breedingand Genetics, 13: 130 – 133.

Lindgren D., Libby W. S., Bondesson F. L. 1989. Deployment to plantations of numbers and proportions of clones with special emphasis on maximizing gain at a constant diversity. Theoretical and Applied Genetics, 77, 6: 825–831.

Lush J.L. 1945. Animal Breeding Plans. 3. Izdaja, Ames, Iowa, ZDA, Iowa State University Press: 442 str.

Meuwissen T. H. E. 1997. Maximizing the response of selection with a predefined rate of inbreeding. Journal of Animal Science, 75, 4: 934–940.

Meuwissen T.H.E., Hayes B.J., Goddard M.E. 2001. Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics, 157, 4: 1819–1829.

Misztal I., Tsuruta S., Strabel T., Auvray B., Druet T., Lee D.H. 2002. BLUPF90 and re-lated programs (BGF90). V: Proc. 7th World Congress on Genetics Applied to LivestockProduction. WCGALP, Montpellier, France, 19-23 avg. 2002. Montpellier, WCGALP: 1-2.

Pong-Wong R., Woolliams J.A. 2007. Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming. Genetics Selection Evolution, 39, 1: 3-25.

Pryce J.E., Daetwyler H.D. 2012. Designing dairy cattle breeding schemes under genomic selection: a review of international research. Animal Production Science, 52: 107–114

Ramana M. in Goldman A. J. 1995. Some Geometric Results in Semidefinite Programming. Journal of Global Optimization, 7, 1: 33–50.

Rendel J.M., Robertson A. 1950. Estimation of genetic gain in milk yield by selection in a closed herd of dairy cattle. Journal of Genetics, 50, 1: 1–8.

Schaeffer L. R., 2006. Strategy for applying genome-wide selection in dairy cattle. Journal of Animal Breeding and Genetics, 123, 4: 218-223.

Toro M., Pérez-Enciso M. 1990. Optimization of selection response under restricted inbreeding. Genetics Selection Evolution, 22, 1: 93 - 107.

Wellmann R. 2017. optiSel: optimum contribution selection and population genetics. R package version 0.7.1. https://cran.r-project.org/web/packages/optiSel/optiSel.pdf.

Woolliams J.A., Berg P., Dagnachew B.S., Meuwissen T.H.E. 2015. Genetic contributions and their optimization. Journal of Animal Breeding and Genetics, 132, 2: 89–99.

Woolliams J.A., Bijma P., Villanueva, B. 1999. Expected Genetic Contributions and Their Impact on Gene Flow and Genetic Gain. Genetics 153, 2: 1009–1020.

Wray N.R., Thompson R. 1990. Prediction of rates of inbreeding in selected populations. Genetics Research. 55, 1: 41–54.

Wright S. 1921. Systems of Mating. II. the Effects of Inbreeding on the Genetic Composition of a Population. Genetics 6, 2: 124–143.

Yamashita M., Mullin T. M., Safarina, S. 2015. An efficient second-order cone programming approach for optimal selection in tree breeding. arXiv:1506.04487v2.