

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Juš Lozej

**Generiranje svetlobi prilagonjenih
dreves z uporabo genetskih
algoritmov**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Marko Bajec

Ljubljana 2015

Fakulteta za računalništvo in informatiko podpira javno dostopnost znanstvenih, strokovnih in razvojnih rezultatov. Zato priporoča objavo dela pod katero od licenc, ki omogočajo prosto razširjanje diplomskega dela in/ali možnost nadaljne proste uporabe dela. Ena izmed možnosti je izdaja diplomskega dela pod katero od Creative Commons licenc <http://creativecommons.si>

Morebitno pripadajočo programsko kodo praviloma objavite pod, denimo, licenco *GNU General Public License*, različica 3. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Besedilo teme diplomskega dela študent prepíše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Juš Lozej sem avtor diplomskega dela z naslovom:

Generiranje svetlobi prilagonjenih dreves z uporabo genetskih algoritmov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Bajca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 5. februarja 2014

Podpis avtorja:

Na tem mestu zapišite, komu se zahvaljujete za izdelavo diplomske naloge. Pazite, da ne boste koga pozabili. Utegnil vam bo zameriti. Temu se da izogniti tako, da pozabite na celo zahvalo.

Svoji dragi Alenčici.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Teorija	3
2.1	Generiranje dreves	3
2.2	Genetski algoritmi	3
3	Implementacija	13
3.1	Generiranje dreves	13
3.2	Genetski algoritmi	13
4	Rezultati	15
	Literatura	17

Seznam uporabljenih kratic

kratica	angleško	slovensko
GA	genetic algorithm	genetski algoritem
FF	fitness function	funkcija uspešnosti
SUS	stochastic universal sampling	univerzalno stohastično vzorčenje
RWS	roulette-wheel selection	izbor z ruletnim kolesmo

Povzetek

V vzorcu je predstavljen postopek priprave diplomskega dela z uporabo okolja L^AT_EX. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek.

Ključne besede: računalnik, računalnik, računalnik.

Abstract

This sample document presents an approach to typesetting your BSc thesis using L^AT_EX. A proper abstract should contain around 100 words which makes this one way too short.

Keywords: computer, computer, computer.

Poglavje 1

Uvod

Poglavje 2

Teorija

V tem poglavju bodo opisno teoretsko ozadje uporabljenih metod. Opisane bodo tudi nekatere značilnosti, ki niso pristal v končni verziji, vendar so bile v fazi načrtovanja diplomskega dela, mišljene kot možne alternative končni implementaciji. Opisno bo nekaj metod generiranja dreves in splošna teorija Genetskih algoritmov.

2.1 Generiranje dreves

2.1.1 L-sistemi

2.1.2 Space Colonization

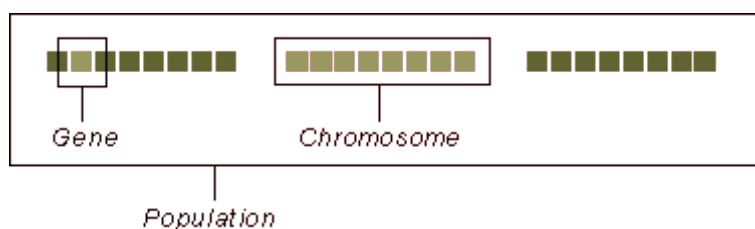
2.1.3 Parametrični modeli

2.2 Genetski algoritmi

Genetski algoritmi, v nadaljevanju označeni z kratico GA, so prilagodljivi heuristični iskalni algoritmi. Spadajo v skupino po naravi navdihnjenih algoritmov imenovanih evolucijski algoritmi. Delujejo na osnovi evolucijskih idejah naravnega izbora ter na osnovi genetike. Uporablja se jih za reševanje optimizacijskih problem z pomočjo pseudonaključnega priskovanja prostora.

Kljub svoji naključni osnovi algoritem ne preiskuje prostor naključno, saj uporablja mero uspešnosti prejšnjih generacij za vodenje nadaljnega preiskovanja prostora. GA so bili zasnovani tako, da simulirajo naravno okolje v katerem bi nato potekala evolucija glede na osnove, ki jih je prvič predstavil Charles Darwin v delu O nastanku vrst z delovanjem naravnega izbora ali ohranjanje prednostnih ras v boju za preživetje. Algoritmi se predvsem nanašajo na principe tekmovanja vrst med seboj in iz tega sledeče nadvlade ene nad drugo.

Kot omenjeno prej, GA uporabljajo osnove naravnega izbora nad posamezniki zaporednih generacij populacije v namen doseči najboljše možne posameznike. Vsi posamezniki predstavljajo populacijo, katere velikost se skozi potek GA ne spreminja. Vsak posameznik je predstavljen z kromosomom, ki je lahko zaporedje števil ali bitov, niz znakov, permutacija, binarno drevo in podobno. Kromosomi posameznikov predstavlja točko v iskalnem prostoru našega problema in hkrati tudi njegovo rešitev. Kromosom je nato sestavljen iz manjših delov imenovanih geni 2.1, torej v primeru kromosoma, ki je zaporedje števil, bi bili geni posamezna števila znotraj zaporedja. Vsak posameznik, torej njegov kromosom, predstavlja točko v iskalnem prostoru in hkrati tudi rešitev našega optimizacijskega problema.



Slika 2.1: Grafična predstavitev populacije, kromosoma in gena

Uspešnost teh rešitev nato izmerimo z funkcijo uspešnosti, v nadaljevanju označene z kratico FF (Fitness Function), katera sprejme kot parameter po-

sameznika, oziroma njegov kromosom, in nam pove kako dobro ta posameznik rešuje naš problem. Funkcija v kontekstu evolucije predstavlja posameznikovo zmožnost tekmovati z ostalimi posamezniki populacije. Da bodo GA delali pravilno predpostavimo, da bodo posamezniki z visoko vrednostjo FF proizvedli posameznike, katerih vrednost FF bo večja od staršev. Pri tem upoštevamo, da posamezniki z visokimi vrednostmi FF se bodo tudi večkrat razmnoževali kot tisti z manjšimi vrednostmi FF in s tem izboljšali prihodnje generacije. Torej cilj GA je najti posameznika z največjo možno vrednostjo FF. To doseže z izmenjavanjem genov posameznikov glede na vrednost FF.

2.2.1 Algoritem

Kot je omenjeno zgoraj, GA skozi potek delovanja ne spreminja velikosti. Za vsakega posameznika hranimo njihov kromosom in vrednost FF. Prvo generacijo inicijaliziramo naključno, torej vsak posameznik dobi naključno generirani kromosom. Nato tem posameznikom izmerimo vrednost FF. Glede na vrednost FF nato posameznike razmnožimo. Posamezniki z visokimi vrednostmi FF dobijo več priložnosti za razmnoževanje kot posamezniki z nizkimi vrednostmi FF. Kromosome staršev križimo in s tem dobimo kromosome Potomcev. Potomci, ki nastanejo kot rezultat križanja, nosijo karakteristike staršev (mešanico genov staršev). Ker pri razmnoževanju dobimo nove posameznik moramo velikost populacijo prilagoditi prejšni velikosti. To najlažje naredimo tako, da se znebimo najslebsih posameznikov (najnižje vrednosti FF). S tem omogočamo, da se zaporedne generacije izboljšujejo. Da se populacij ne bi vjela v lokalni maksimum FF posameznike, z majhno vrjetnostjo, tudi mutiramo. Mutiramo jih tako, da jim naključno spremenimo neke gene. Ta postopek nato ponavljamo za neko fiksno število generacij ali pa dokler ne dobimo posameznika, ki je dovolj dober.

Delovanje algoritma lahko opišemo z naslednjo pseudo kodo:

1. Naključno inicializirajo populacijo(t)
2. Doloci vrednosti funkcije uspešnosti populacije(t)
3. Ponavljaaj
 1. Izberi starse , za krizanje , iz populacije(t)
 2. Z izbranimi starsi izvedi krizanje in s tem izdelaj populacijo
 3. Populacijo($t+1$) mutiramo
 4. Doloci vrednosti funkcije uspešnosti populacije($t+1$)
4. dokler najboljši posameznik ni dovolj dober

2.2.2 Inicializacija

Kot smo omenili prej, inicializiramo populacijo naključno, ponavadi omejeno na določeno območje. Pri tem prilagodimo velikost populacije kompleksnosti iskalne domene.

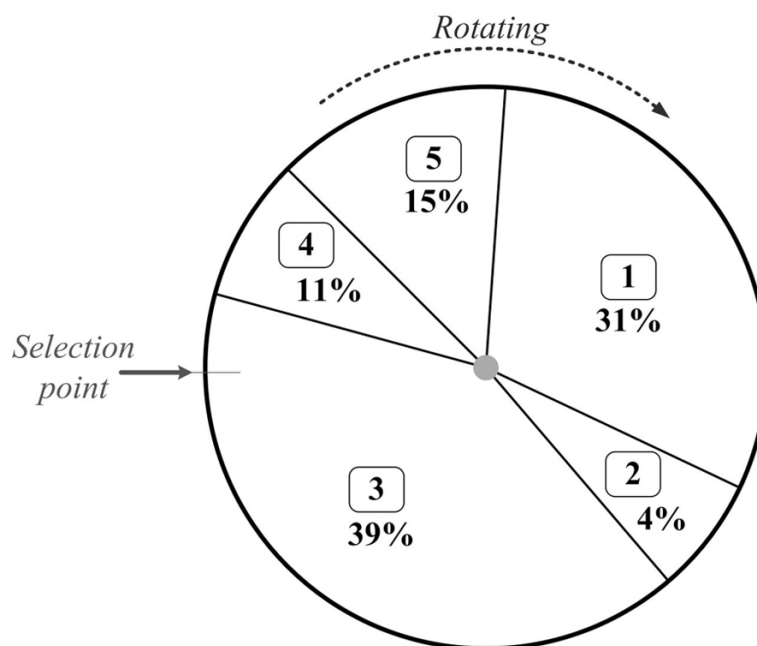
2.2.3 Izbor staršev

Starše za razmnoževanje se izbere na podlagi vrednosti FF. Torej posamezniki z visokimi vrednostmi bodo imeli večjo vrjetnost biti izbrani za starše. Najbolj enostaven način je, da posameznike uredimo glede na vrednost FF in izberemo N najboljših. N je število staršev. Težava pri tem je, da izbor staršev ni proporcionalen glede na vrednost FF ampak samo glede na zaporedno mesto v urejeni populaciji. Metode, ki upoštevata to sta izbor z ruletnim kolesmo in univerzalno stohastično vzorčenje.

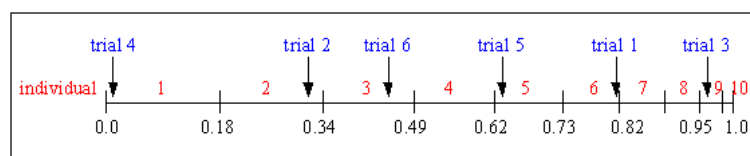
Izbor z ruletnim kolesmo

Izbor z ruletnim kolesmo, v nadaljevanju označen z kratico RWS(roulette-wheel selection), znan tudi kot vrednost upsešnosti proporcionalni izbor (fitness proportionate selection), je metoda izbiranja staršev, ki izbira posameznike proporcionalno glede na vrednost FF. Starše izbira z naključnim

vzorčenjem prostora, ki vsebuje normiran nabor vrednosti FF posameznikov populacije. Ta prostor dobimo tako, da vrednosti FF uredimo po velikosti in jih normiramo tako, da je seštevek vseh enak 1. Nato, na intervalu od 0 do 1, izberemo naključno število. Starša izberemo glede na to v čigav interval to število nato pade 2.2. To nato ponavljamo dokler ne dobim željeno število staršev 2.3.



Slika 2.2: Primer enega vzorca RWS

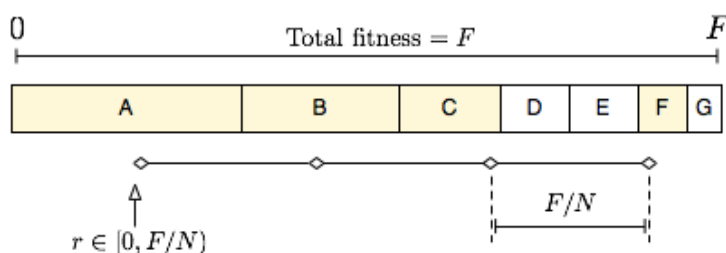


Slika 2.3: Primer celotnega izbora RWS

Algoritem izbira starše proporcionalno na vrednosti FF, vendar težava se pojavi, če en posamezni zavzema večji del celotnega intervala in se tako zgodi, da ostali posamezniki nikoli niso izbrani.

Univerzalno stohastično vzorčenje

Univerzalno stohastično vzorčenje, v nadaljevanju označeno z kratico SUS (Stochastic universal sampling), želi popraviti težave, ki so se pojavile pri algoritmu RWS. Vzorčni prostor pripravi na enak način kot pri RWS. Težavo RWS reši z enakomernim vzorčenjem prostor od naključne točke naprej. To da možnost tudi šibkejšim posameznikom, da so izbrani za razmnoževanje. Prvo izbere naključno točko, na intervalu od 0 do $1/N$, kjer bo začel vzorčenje. Ta točka predstavlja prvi vzorec. Nato naredi še $N-1$ vzorcev, vsak zamaknjen za $1/N$ od prejšnjega. Starše nato izberemo glede na to v čigav interval te vzorci padejo 2.4.



Slika 2.4: Primer SUS

Elitizem

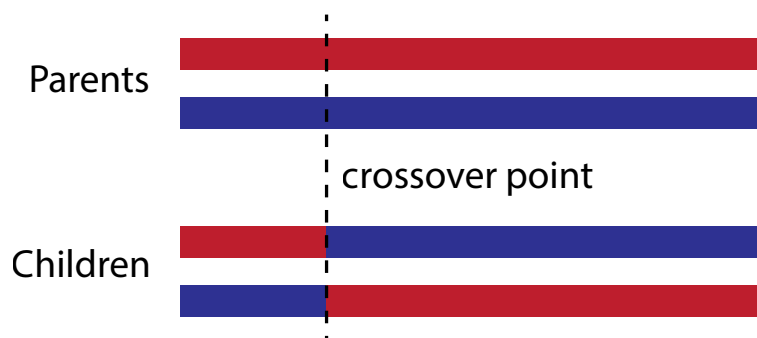
Elitizem pomeni, da vsaki zaporedni generaciji dodamo nekaj najboljših prejšnje generacije. Pri GA, pri katerih zaporedne generacije ne nujno pomenijo boljše posamznike, izboljša elitizem, iskanje saj so najboljši kromosomi vedno prisotni. Elitizem tako nudi pospešitev pri iskanju. Kritike elitizma so predvsem v tem, ker niso skladne z principi naravne selekcije, katere GA simulirajo.

2.2.4 Genetski operatorji

Naslednji korak je, da iz izbranih staršev naredimo novo generacijo. To naradimo z pomočjo genetskih operatorjev: Križanja in mutacija.

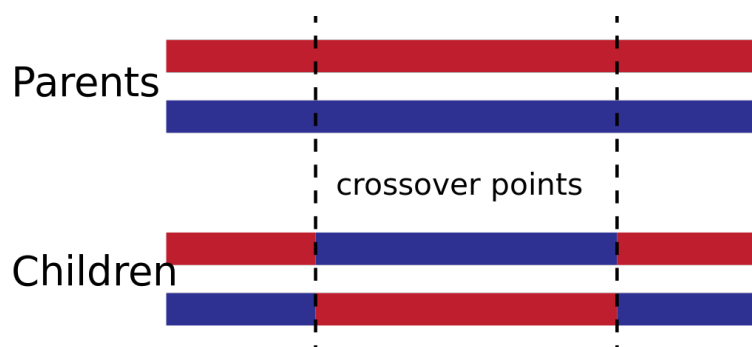
Križanje

Križanje je postopek, ki združi kromosome dveh ali več staršev v kromosom otroka. Križanje mora biti prilagojeno glede na tip kromosoma. Torej pri kromosom v obliki permutacije moramo paziti da križanje ne bodo dodali enakih števil, pri kromosomu v obliki drevesa naredimo križanje tako da izmenamo veje dreves in tako naprej. V nadaljevanju bo opisanih nekaj metod kako križati kromosom v obliki vektorja števil saj so najbolj pogosti tipi kromosomov. Najbolj enostaven način križanja je križanje po eni točki 2.5. Prvo določimo eno točko, ki razdeljuje kromosom na dva dela. Pri tem načinu križanja dobimo dva otroka. Prvi otrok dobi levi del kromosoma prvega starša in desni del kromosoma drugega starša.



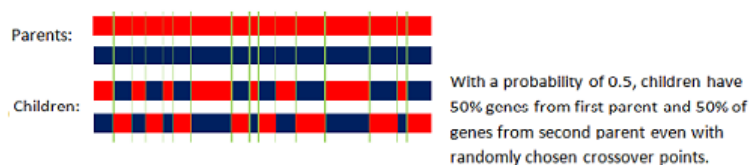
Slika 2.5: Primer križanja po eni točki

Najlažja izboljšava je tega je da dodamo več točk in za vsako točko spremenimo čigave gene prejme. Te algoritmi se imenijo gleda na število točk, ki jih uporabimo. Torej če imamo 2 točke se bo križanje imenovalo križanje po dveh točkah 2.6.



Slika 2.6: Primer križanja po dveh točkah

Težava pri takem načinu križanja je, da se geni izmenjujejo vedno po istem ključu to težavo rešuje enakomerno križanje 2.7. Pri enakomernem križanju uporabljamo fiksno mešalno vrjetnost, ki nam pove ali se bo gen izmenjav. Torej za vsak gen generiramo naključno število, na intervalu od 0 do 1, in če je ta večja od mešalne vrjetnosti se gena izmenjata, če je manjša se ne. Torej z mešalno vrjetnostjo 0.5 se bo okoli 50% genov izmenjalo.



Slika 2.7: Primer enakomernega križanja

Za križanje lahko uporabimo tudi več staršev in stem izboljšamo tudi kvaliteto dobljenega kromosoma. To najlažje naredimo tako, da za vsak gen še naključno izberemo starše katerih gene bomo križali.

Mutacije

GA se ujema v lokalne maksimum, to bojujemo z mutacijo. Po tem ko smo starše križali je dobro, da dobljene otroke še mutiramo. To pomeni, naključno spremenimo določene gene v kromosomu. To lahko naredimo fiksno, tako da

se vedno isti gen spremni, ali pa naključno, podobno kot pri univerzalnem križanju. Ker se gen naključno spremeni lahko dobimo posameznika, ki se ne nahaja v enaki okolici iskalne domene kot ostali posamezniki populacije. Z nadaljnim izvajanjem GA se lahko izkaže, da se te posamezniki nahajajo bližje globalnemu maksimumu kot kromosomi pred mutacijo.

2.2.5 Prekinitev

Kako vediti kdaj algoritem prekiniti? Najbolj osnovna metoda prekinitve je, da algoritem prekinemo po nekem fiksnem številu generaciji. Pri preprostejših problemih, algoritem hitro konvergira proti neki rešitvi, pri zahtevnejših ne. Zato je dobro slediti spreminjanju najboljšega kromosoma. Če se po nekaj generacijah ne spremeni potem lahko algoritem prekinemo. V nekaterih primerih želimo dobiti le dovolj dobro rešitev. V tem primeru postavimo prag vrednosti FF in če najboljši posameznik preseže ta prag potem lahko algoritem prekinemo.

2.2.6 Prednosti in slabost

Prednosti GA so, da so bolj robustni in manj prostorsko zahtevni kot drugi podobni iskalni algoritmi (naprimer linearno programiranje, iskanje v globino, iskanje v širino...). Nudijo tudi prednosti pri optimizaciji visoko prostorskih, več modelnih in visko dimenzijonalnih problemov, saj prostor ne preiskujejo izčrpno. Težava GA je, da se ujamejo v lokalne maksimume. Za reševanje visoko dimenzijonalnih večmodelnih problemov rabimo enako kompleksno FF, te funkcije so ponavadi zelo časovno zahtevne, zato za zaporedno računanje te funkcije porabimo veliko časa. Težava je tudi, da se z kompleksnostjo problema večja tudi iskalna domena in to zahteva tudi večje populacije, ki pa zahtevajo temu primerno večjo število izračunov FF. Uporabljati jih tudi ne morem na dinamičnih podatki, saj bi se z podatki spreminjala tudi iskalna domena in s tem tudi maksimumi FF.

Poglavje 3

Implementacija

3.1 Generiranje dreves

3.2 Genetski algoritimi

Poglavje 4

Rezultati

Literatura

- [1] L. Fortnow, “Viewpoint: Time for computer science to grow up”, *Communications of the ACM*, št. 52, zv. 8, str. 33–35, 2009.
- [2] D. E. Knuth, P. Bendix. “Simple word problems in universal algebras”, v zborniku: *Computational Problems in Abstract Algebra* (ur. J. Leech), 1970, str. 263–297.
- [3] L. Lamport. *LaTEX: A Document Preparation System*. Addison-Wesley, 1986.
- [4] O. Patashnik (1998) BIBTEXing. [Online]. Dosegljivo: <http://ftp.univie.ac.at/packages/tex/biblio/bibtex/contrib/doc/btxdoc.pdf>. [Dostopano 18. 9. 2014].
- [5] PDF/A. [Online]. Dosegljivo: <http://en.wikipedia.org/wiki/PDF/A>. [Dostopano 19. 9. 2014].
- [6] GNU General Public Licence. [Online]. Dosegljivo: <https://www.gnu.org/copyleft/gpl.html>. [Dostopano 20. 9. 2014].