МАТФ
Универзитет у Београду
Математички факултет

Metode za rešavanje problema simboličke regresije

master rad

Jana Jovičić 1097/2019

mentor: dr Aleksandar Kartelj

Matematički fakultet
Univerzitet u Beogradu

10. septembar 2022.

Sadržaj

Motivacija

- Lakša analiza različitih fizičkih sistema.
- Modelovanje osobina različitih fizičkih sistema pomoću promenljivih i razumevanje odnosa između tih promenljivih.
- Pronalazak modela kojim se najbolje opisuje dostupni skup podataka.
- Ne zahteva prethodnu specifikaciju strukture modela, već istovremeno uči i o strukturi modela i njegove parametre.
- Pod manjim uticajem ljudske greške ili nedovoljnog domenskog znanja u odnosu na regresione metode koje unapred pretpostavljaju formu modela.
- Prednost u odnosu na metode dubokog učenja je u lakšoj interpretabilnosti.

Tip problema

- Problem kombinatorne optimizacije.
- Instance velikih dimenzija je nemoguće rešiti usled ograničenja vremenskih i memorijskih resursa.
- Najčešće se traže aproksimativna rešenja problema, uglavnom pomoću razlčitih metaheurističkih metoda.
- Smatra se da je NP-težak problem, ali još nije formalno dokazano.

Regresija

- Tehniku za modelovanje veze izmedu zavisne (ciljne) promenljive i jedne ili više nezavisnih promenljivih (atributa).
- Cilj - formiranje modela koji će na osnovu dostupnih uzoraka za trening i odgovarajućih izlaznih vrednosti predvidati vrednost kontinualne izlazne promenljive za novi uzorak.
- Različiti tipovi regresione analize: linearna regresija, polinomijalna regresija, simbolička regresija.

Linearna regresija

- Pretpostavlja se linearna forma modela, tj. važi pretpostavka da se vrednost ciljne promenljive može dobiti kao linearna kombinacija vrednosti ulaznih obeležja.
- Kako se pretpostavlja linearnu zavisnost po parametrima β_0, β_1, ..., β_m između atributa x_1, x_2, ..., x_m i ciljne promenljive y, onda se y predstavlja u obliku

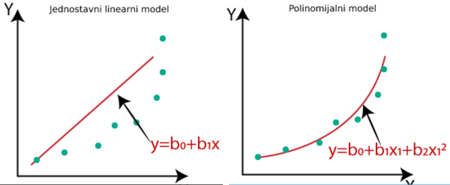
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m.$$

Polinomijalna regresija

- Veza izmedu atributa i ciljne promenljiv se modeluje pomoću polinoma n-tog stepena.
- U slučaju jedne nezavisne promenljive, ciljna promenljiva je oblika

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \ldots + \beta_m x_1^m$$

- Ovakav model se i dalje smatra linearnim, jer su težine pridružene atributima linearne. Samo je kriva koju modelujemo polinomijalnog obilka.



Simbolička regresija

- Generalizacija linearne ili polinomijalne regresije, ne pretpostavlja unapred formu modela.
- Cilj simboličke regresije je pronalazak matematičkog izraza u simboličkoj formi, koji dobro modeluje vezu izmedu ciljne promenljive i nezavisnih promenljivih.
- Formalno, ako je dat skup podataka (X_i, y_i), $i = 1, ..., n$, gde $X_i \in \mathbb{R}^n$ predstavlja i-ti skup atributa, a $y_i \in \mathbb{R}$ i-tu ciljnu promenljivu, cilj simboličke regresije je pronalazak funkcije $f : \mathbb{R}^n \to \mathbb{R}$ koja najbolje odgovara skupu podataka, odnosno za koju važi $y_i \approx f(X_i), i = 1, ..., n$.

Evaluacija modela simboličke regresije

- Ranije - u terminima metaheuristike kojom je problem rešavan, na primer:
  - na osnovu srednje vrednosti najboljih vrednosti funkcija prilagođenosti dobijenih pri većem broju nezavisnih pokretanja programa.
  - na osnovu broja uspešnih pokretanja od ukupnog broja pokretanja programa, gde se pod uspešnim pokretanjem smatralo ono u kom postoji barem jedna jedinka koja za svaku instancu iz skupa podataka daje grešku manju od nekog definisanog praga.
- Poslednjih godina - poput ostalih tipova regresije: pomoću metrika kao što su MSE, RMSE i R^2, uz podelu skupa podataka na trening i test deo.

Evaluacija modela simboličke regresije

- Srednje kvadratna greška (MSE, eng. *Mean Squared Error*)

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

  gde je n broj uzoraka, y_i stvarna vrednost ciljne promenljive za i-ti uzorak, a $\hat{y}_i$ predviđena vrednost.
- Koeficijent određenosti R^2 (koeficijent determinacije, eng. *coefficient of determination*)

  $$R^2 = 1 - \frac{MSE}{\mathrm{Var}(y)}$$

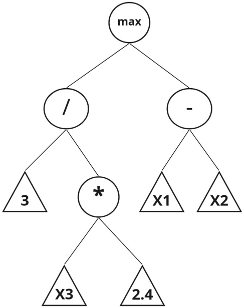
- Srednje kvadratna greška se izražava u terminima veličine ciljne promenljive, dok je vrednost koeficijenta određenosti normirana.

Reprezentacija izraza kod simboličke regresije

- Izraz se može predstaviti pomoću sintaksnog stabla.
- Lisovi mogu da sadrže samo terminale izraza (konstante i nezavisne promenljive iz datog skupa podataka).
- Unutrašnjim čvorovima su predstavljeneunarne i binarne funkcije.
- U okviru jednog stabla ista funkcija se može pojaviti veći broj puta. Isto važi i za konstante i promenljive.
- Skupovi funcija i terminala su fiksirani u skladu sa trenutnim problemom koji se rešava.

Reprezentacija izraza kod simboličke regresije

$$\max\left\{\frac{3}{x_3 * 2.4}; x_1 - x_2\right\}$$



Slika: Primer sintaksnog stabla

Reprezentacija izraza kod simboličke regresije

- Prilikom ocenjivanja tačnosti regresionog modela treba uzeti u obzir da se jedna ista funkcija može izraziti pomoću više simbolički različitih zapisa.
- Npr. ako je ciljna funkcija f jednaka $(u+v)/(1+uv/c^2)$ onda se i simbolički različit zapis $(v+u)/(1+uv/c^2)$ treba smatrati tačnim rešenjem.
- Smatra se da je ciljna funkcija f ispravno određena kandidatskom funkcijom f' ako algebarska simplifikacija izraza $f'-f$ daje simbol "0".

Algoritam grube sile

- Zasniva se na sistematičnoj pretrazi prostora matematičkih izraza.
- Pretraga podrazumeva isprobavanje svih mogućih kombinacija podizraza sve dok se ne stigne do zadovoljavajućeg rešenja.
- Pretraga radi iterativno po visini sintaksnog stabla izraza [3].
- Prvo se proveravaju sva stabla visine 1 koja su generisana na osnovu svih datih funkcija i promenljivih. Ako među njima postoji izraz koji nad datim skupom podataka daje MSE grešku manju od definisanog parametra $\epsilon = 10^{-6}$, smatra se da je pronađeno tačno rešenje i pretraga se prekida.
- Ako takav izraz nije pronađen generišu se sva stabla visine 2. Postupak se ponavlja sve dok se ne pronađe tačno rešenje ili dok se ne dostigne definisano vremensko ograničenje.

Algoritam grube sile

Algoritam 1 Algoritam grube sile za simboličku regresiju

Ulaz: X - skup vrednosti atributa, y - skup vrednosti ciljne promenljive, F - skup funkcija koje je moguće koristiti za kreiranje stabla izraza, ϵ - dozvoljena epsilon okolina greške, $vreme$ - maksimalan broj sati izvršavanja programa;

Izlaz: $resenje$ - izraz koji predstavlja tačno rešenje problema, $najblizeResenje$ - izraz koji daje najmanju grešku, $uspeh$ - indikator koji govori da li je pronadjeno tačno resenje;

Inicijalizacija: t = lista promenljivih // formiranih na osnovu ulaznog skupa podataka;
while vremenski kriterijum zaustavljanja nije ispunjen **do**
    T = svi mogući parovi terminala iz t; // *generisani kao permutacije dužine 2*
    $uspeh, resenje, S, najblizeResenje = konstruisiStabla(T, F, X, y, \epsilon)$;
    if $uspeh$ **then**
        return $uspeh, resenje, najblizeResenje$;
    end if
    $t = t \cup S$;
end while
return $uspeh, resenje, najblizeResenje$;

Algoritam grube sile

Algoritam 2 konstruisiStabla()

Opis: Algoritam za konstrukciju novih stabala i njihovu evaluaciju pri potrazi za rešenjem.

Ulaz: T - skup parova terminala, F - skup funkcija koje je moguće koristiti za kreiranje stabla izraza, X - skup vrednosti atributa, y - skup vrednosti ciljne promenljive, ϵ - dozvoljena epsilon okolina greške;

Izlaz: $uspeh$ - indikator koji govori da li je pronadjeno tačno resenje, $resenje$ - tačno rešenje čija je greška manja od ϵ, S - skup generisanih stabala, tj. skup izraza čije stablo ima visinu jednaku trenutnoj iteraciji, $najblizeResenje$ - izraz koji daje najmanju grešku, ali ne nužno manju od ϵ;

Inicijalizacija: $minGreska = inf$; $S = []$; $resenje = ''$; $uspeh = False$;
for $f \in F$ **do**
  for $par \in T$ **do**
    if f je unarna funkcija **then**
      if prvi član $para$ je jednak prvom članu prethodnog para **then**
        continue;
      end if
    end if
    $f.levo\_podstablo = par[0]$;
    if f je binarna funkcija **then**
      $f.desno\_podstablo = par[1]$;
    end if
    $S = S \cup f$
    $y_{pred} = f(X)$;
    $greska = MSE(y_{pred}, y)$;
    if $greska < minGreska$ **then**
      $najblizeResenje = f$;
      $minGreska = greska$;
    end if
    if $greska < \epsilon$ **then**
      $uspeh = True$;
      $resenje = f$;
      break;
    end if
  end for
  if $uspeh$ **then**
    break;
  end if
end for
return $uspeh$, $resenje$, S, $najblizeResenje$;

Algoritam grube sile

- Može se desiti da algoritam grube sile ne uspe da stigne do tačnog rešenja ne samo zbog vremenskog ograničenja, već i zbog ograničenja memorijskih resursa.
- U svakoj iteraciji se povećava broj stabala koje je potrebno čuvati, jer ih je u narednoj iteraciji potrebno iskoristiti kao terminale.
- Ako je n broj funkcijskih simbola, a m broj parova terminala u trenutnoj iteraciji, broj stabala generisanih u toj iteraciji biće, u najgorem slučaju, jednak nm.

Genetsko programiranje

Algoritam 3 Osnovna struktura GP metode

Ulaz: GP parametri i kriterijum zaustavljanja;
Izlaz: najbolja jedinka tj. izraz koji predstavlja rešenje problema;

Generisati početnu populaciju jedinki;
Izračunati prilagođenost svake jedinke u populaciji;
while nije zadovoljen kriterijum zaustavljanja **do**
    Izabrati iz populacije skup jedinki za reprodukciju;
    Primenom operatora ukrštanja kreirati nove jedinke;
    if ispunjen uslov za mutaciju **then**
        Primeniti mutaciju nad novim jedinkama;
    end if
    Izračunati prilagođenost novih jedinki;
    Formirati novu generaciju na osnovu novih jedinki;
end while
return najbolja jedinka;

- Zasniva se na iterativnoj popravci inicijalne populacije rešenja.
- Kod GP jedinke su predstavljene stabloidinim strukturama.
- GP metoda je implementirana po ugledu na radove Džona Koze [1].

Reprezentacija jedinki

- Kako jedinka treba da odgovara rešenju problema, u slučaju simboličke regresije pomoću jedne jedinke je predstavljen jedan izraz koji je u obliku sintaksnog stabla.
- Čvorovi stabla odgovaraju unarnim i binarnim funkcijama i terminalima.
- Terminali mogu biti:
  - promenljive definisane na osnovu dostupnog skupa podataka
  - efemerne slučajne konstante (tj.slučajno generisani brojevi određenog tipa iz definisanog intervala, najčešće realni brojevi iz intervala [-1, 1])

Generisanje početne populacije

1. "full"metoda
   - Generisanje potpunog stabla.
   - Put između korena i svakog lista je jednak definisanoj maksimalnoj dubini.
   - Ako se trenutni čvor koji se generiše nalazi na dubini koja je manja od maksimalne, može se izabrati samo čvor koji predstavlja funkciju.
   - Ako se trenutni čvor koji se generiše nalazi na dubini koja je jednaka maksimalnoj, onda se može odabrati samo terminal.
   - Ako je trenutni čvor koji se generiše na dubini koja je manja od minimalne, moguće je odabrati samo funkciju.

Generisanje početne populacije

2. "grow"metoda
    - Generisanje stabala čiji oblici variraju.
    - Dužina puta od korena do bilo kog lista ne bude veća od definisane maksimalne dubine, a dozvoljeno je da bude kraća.
    - Ako se trenutni čvor koji se generiše nalazi na dubini koja je manja od maksimalne, može se izabrati ili čvor koji predstavlja funkciju ili čvor koji predstavlja terminal.
    - Ako se trenutni čvor koji se generiše nalazi na dubini koja je jednaka maksimalnoj, onda se može odabrati samo terminal.
    - Ako je trenutni čvor koji se generiše na dubini koja je manja od minimalne, moguće je odabrati samo funkciju.

Generisanje početne populacije

3. "ramped half-and-half"metoda
   - Generisanje stabala različitih visina i oblika.
   - Kombinacija "full"i "grow"metoda.
   - Kreiranje podjednakog broja stabala po svakom nivou dubine od nivoa 2 do nivoa definisanog parametrom maksimalne dubine.
   - Npr. ako je definisana maksimalna dubina jednaka 6, onda će 20% stabala imati dubinu 2, ..., 20% dubinu 6.
   - Za svaki nivo dubine 50% stabala se kreira pomoću "full", a 50% pomoću "grow"metode.

Funkcija prilagođenosti

- Daje ocenu kvaliteta jedinke i utiče na verovatnoću izbora te jedinke za proces formiranja nove generacije.
- Koza definiše 4 vrste funkcija prilagodenosti:
  1. "Raw"funkcija prilagođenosti
  2. Standardizovana funkcija prilagođenosti
  3. "Adjusted"funkcija prilagođenosti
  4. Normalizovana funkcija prilagođenosti

"Raw"funkcija prilagođenosti

- Izražava se u terminima problema koji se rešava.
- Kod simboličke regresije se može posmatrati kao funkcija greške.
- Njena vrednost za neku jedinku odgovara zbiru distanci između vrednosti koje daje izraz predstavljen tom jedinkom i pravih ciljnih vrednosti svih uzoraka iz trening skupa.
- Distanca se računa kao apsolutna vrednost razlike između predviđenih i pravih vrednosti.

$$r(i,t) = \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

- Bolje jedinke imaju manju vrednost funkcije.

Standardizovana funkcija prilagođenosti

- Redefiniše "raw"funkciju tako da njena vrednost za bolje jedinke uvek bude manja, bez obzira na vrstu problema koji se rešava.
- Kako je kod simboličke regresije "raw"funkcija već definisana tako da manje vrednosti reprezentuju bolje jedinke, standardizovana funkcija će biti jednaka "raw"funkciji.

$$s(i,t) = r(i,t)$$

- Kod problema kod kojih je "raw"prilagođenost definisana tako da se boljim jedinkama smatraju one sa većom vrednošću te funkcije, standardizovana funkcija bi bila jednaka razlici maksimalne moguće vrednosti "raw"funkcije i vrednosti "raw"funkcije za datu jedinku.

$$s(i,t) = r_{max} - r(i,t).$$

"Adjusted"funkcija prilagođenosti

- Računa se na osnovu standardizovane funkcije.

$$a(i,t) = \frac{1}{1+s(i,t)}.$$

- Vrednost $a(i,t)$ pripada intervalu [0,1] i veća je za bolje jedinke.
- Prednost: naglašava razliku između dobrih i vrlo dobrih jedinki.
- Npr. ako imamo dve loše jedinke čije su vrednosti standardizovane funkcije redom 64 i 65, njihove vrednosti prilagođene funkcije će biti 0.0154 i 0.0156. U oba slučaja, razlika između prilagođenosti ove dve loše jedinke nije velika.
- Ako imamo dve dobre jedinke čije su vrednosti standardizovane funkcije redom 4 i 3, njihove vrednosti prilagođene funkcije će biti 0.20 i 0.25. Ovde, iako su jedinke na osnovu vrednosti standardizovane funkcije bliske, na osnovu "adjusted"funkcije se dodatno ističe bolja od dve posmatrane dobre jedinke.

Normalizovana funkcija prilagođenosti

- Računa se na osnovu "adjusted"funkcije kao

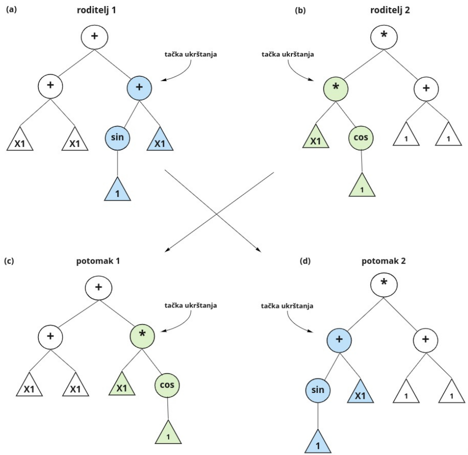
$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{M} a(k,t)},$$

  gde je M veličina populacije.
- Za jedniku i "adjusted"funkcija prilagođenosti se normalizuje u skladu sa "adjusted"funkcijama ostalih jedinki iz populacije.
- Njena vrednost pripada intervalu [0, 1].
- Veća je za bolje jedinke u populaciji.
- Zbir normalizovanih vrednosti prilagođenosti svih jedinki je jednak 1.
- Mana: Ako u populaciji postoji neka jedinka čiji izraz nije definisan nad svim tačkama iz skupa podataka, to će onemogućiti izračunavanje normalizovane funkcije prilagođenosti čak i jedinke koja je definisana u svim tačkama.

Selekcija

- Obezbeđuje čuvanje i prenošenje dobrih osobina populacije na sledeću generaciju.
- Turnirska selekcija - Iz tekuće populacije se na slučajan način bira k jedinki, gde k predstavlja veličinu turnira, a zatim se od njih bira najbolje prilagođena jedinka.

Operatori ukrštanja - Standardni operator ukrštanja



Operatori ukrštanja - Standardni operator ukrštanja

- Ukrštanje dve jedinke podrazumeva razmenu njihovih slučajno odabranih podstabala.
- Kod oba roditelja se odabere po jedan čvor, a zatim se razmene podstabla određena tim čvorom kao korenom.
- Postoji parametar kojim se ograničava veličina jedinki koje nastaju ukrštanjem.
- Ako se formira jedinka nedozvoljene veličine, umesto nje će u narednu generaciju ići jedan od roditelja.

Operatori ukrštanja - Operatori zasnovani na semantici [2]

- *Semantika uzorkovanja* (SS, eng. *Sampling Semantics*) nekog podstabla se aproksimira pomoću vrednosti dobijenih evaluacijom tog podstabla na predefinisanom skupu tačaka iz domena problema.
- Neka je F funkcija koja je izražena pomoću (pod)stabla T na domenu D i neka je P skup tačaka iz domena D, $P = \{p_1, p_2, ..., p_N\}$. Tada je *semantika uzorkovanja* stabla T na skupu P u domenu D, skup $S = \{s_1, s_2, ..., s_N\}$ takav da je $s_i = F(p_i), i = 1, 2, .., N$.
- Na osnovu SS definiše se *rastojanje semantike uzorkovanja* (SSD, *Sampling Semantics Distance*) između dva podstabla. Neka je $P = \{p_1, p_2, ..., p_N\}$ *semantika uzorkovanja* podstabla St_1, a $Q = \{q_1, q_2, ..., q_N\}$ *semantika uzorkovanja* podstabla St_2. Onda se SSD između St_1 i St_2 definiše kao

$$SSD(St_1, St_2) = \frac{1}{N}(|p_1 - q_1| + |p_2 - q_2| + ... + |p_N - q_N|).$$

Operatori ukrštanja - Operatori zasnovani na semantici

- Pomoću *SSD* se definišu *semantička ekvivalentnost* i *semantička sličnost* između dva podstabla.
- Dva podstabla su *semantički ekvivalentna* (SE, eng. *Semantically Equivalent*) na domenu ako je njihova *SSD* vrednost dovoljno mala

$$SE(St_1, St_2) = \begin{cases} true, & \text{ako je } SSD(St_1, St_2) < \epsilon \\ false, & \text{inače} \end{cases}$$

- Parametar ϵ predstavlja *semantičku osetljivost* (eng. *semantic sensitivity*). Za njega je najbolje uzeti neku vrednost iz skupa $\{0.01, 0.02, 0.04, 0.05, 0.06, 0.08, 0.1\}$.

Operatori ukrštanja - Operatori zasnovani na semantici

- Dva podstabla su *semantički slična* (SS_i, eng. *Semantically Similar*) na domenu ako njihova *SSD* vrednost leži na nekom pozitivnom intervalu.

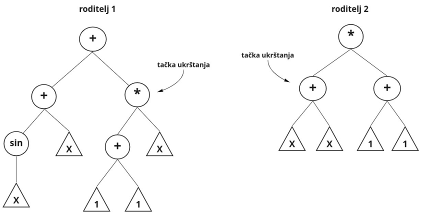
$$SS_i(St_1, St_2) = \begin{cases} true, & \text{ako je } \alpha < SSD(St_1, St_2) < \beta \\ false, & \text{inače} \end{cases}$$

- Paraametri α i β donja i gornja granica semantičke osetljivosti (LBSS, eng. *Lower Bound Semantic Sensitivity* i UBSS, eng. *Upper Bound Semantic Sensitivity*). Kod simboličke regresije najbolje rezultate daju vrednosti između 0.4 i 0.6 za *UBSS*, a vrednosti 10^{-2} ili manje za *LBSS*.
- Motivacija za korišćenje semantičke sličnosti - verovatnije je da će razmena podstabala biti korisnija ukoliko se odvija izmedu dve jedinke koje nisu semantički identične, ali nisu ni semantički suviše različite.

Operatori ukrštanja - Operatori zasnovani na semantici

Operator ukrštanja koji je svestan semantike, SAC (eng. Semantics Aware Crossover)

- Onemogućavanje razmene semantički ekvivalentnih podstabala koji dovode do kreiranja potomaka koji su identični svojim roditeljima.



- Ako su podstabla ekvivalentna, ponovo se na slučajan način biraju tačke ukrštanja.

Operatori ukrštanja - Operatori zasnovani na semantici

Operator ukrštanja koji je zasnovan na semantičkoj sličnosti, SSC (eng. Semantic Similarity-based Crossover)

- Proširenje SAC operatora.
- Proverava se semantička sličnost podstabala odabranih za ukrštanje.
- Semantičku sličnost mnogo teže zadovoljiti u odnosu na semantičku ne-ekvivalentnost, pa je verovatnije da će dolaziti do uzastopnih neuspešnih pokušaja tokom potrage za takvim podstablima.
- Koristi se veći broj pokušaja za pronalazak semantički sličnog para.
- Ako se pređe dozvoljeni broj pokušaja, podstabla se biraju na slučajan način.

Operatori ukrštanja - Operatori zasnovani na semantici

Algoritam 4 Algoritam ukrštanja zasnovanog na semantičkoj sličnosti

Ulaz: $skupJedinki$ - skup jedinki iz kog je potrebno izabrati dve jedinke za ukrštanje, α - donja granica semantičke osetljivosti, β - gornja granica semantičke osetljivosti, $maxBrojPokusaja$ - maksimalan broj pokušaja odabira semantički sličnog para jedinki

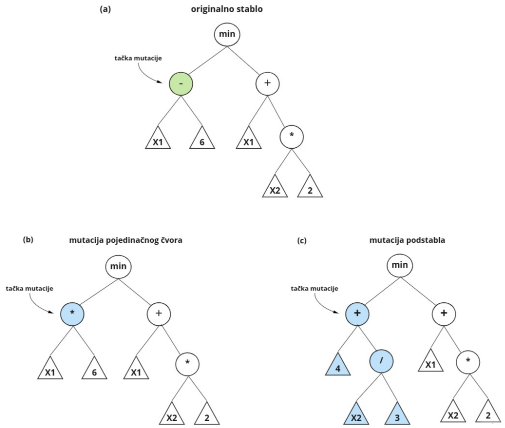
Izlaz: ;

Izabrati roditelja P_1 iz $skupaJedinki$;
Izabrati roditelja P_2 iz $skupaJedinki$;
$brojPokusaja = 0$;
while $brojPokusaja < maxBrojPokusaja$ **do**
    Na slučajan način izabrati podstablo St_1 u roditelju P_1;
    Na slučajan način izabrati podstablo St_2 u roditelju P_2;
    Na slučajan način generisati skup tačaka P iz domena D;
    Izračunati SSD između St_1 i St_2 na tačkama P;
    if $\alpha < SSD(\mathrm{St}_1, St_2) < \beta$ **then**
        // St_1, St_2 *su semantički slični*
        Izvršiti ukrštanje;
        Dodati decu u novu populaciju;
        return true;
    end if
    $brojPokusaja = brojPokusaja + 1$
end while
if $brojPokusaja == maxBrojPokusaja$ **then**
    Na slučajan način izabrati podstablo St_1 u roditelju P_1;
    Na slučajan način izabrati podstablo St_2 u roditelju P_2;
    Izvršiti ukrštanje;
    return true;
end if

Operatori mutacije

1. Mutacija pojedinačnog čvora (eng. *point mutation*)
    - Zamena odabranog čvora nekim drugim čvorom.
    - Terminal može biti zamenjen drugim slučajno izabranim terminalom.
    - Funkcija može biti zamenjena drugom funkcijom iste arnosti.
2. Mutacija celog podstabla (eng. *subtree mutation*)
    - Vrši se slučajni izbor čvora koji će predstavljati koren podstabla koje treba zameniti.
    - Na slučajan način se generiše novo stablo.
    - Odabrano podstablo se menja generisanim stablom.

Operatori mutacije



Zaustavljanje

Evolutivni proces stvaranja novih generacija se ponavlja sve dok nije zadovoljen neki uslov zaustavljanja:

1. Pronađeno je rešenje koje zadovoljava unapred zadati kriterijum.
2. Dostignut je zadati broj generacija.
3. Dostignut je definisani vremenski kriterijum zaustavljanja.
4. Funkcija prilagođenosti je izračunata zadati broj puta.

Ovde su korišćeni uslovi 1, 2 i 3, pri čemu se u uslovu 1. smatra da je pronađeno zadovoljavajuće rešenje ako mu je "adjusted"funkcija prilagođenosti veća od 0.9.

Metoda promenljivih okolina (VNS, eng. *Variable Neighbourhood Search*) [4]

- S-metaheuristika (Single-solution-based metaheuristika) - zasnovana na lokalnoj pretrazi i unapredivanju jednog rešenj.
- VNS metoda je zasnovana na 3 činjenice:
  1. Lokalni optimum u odnosu na jednu okolinu ne mora da predstavlja lokalni optimum u odnosu na drugu.
  2. Globalni optimum je lokalni optimum u odnosu na sve moguće okoline.
  3. Za mnoge probleme lokalni optimumi u odnosu na različite okoline su relativno blizu.
- Činjenica 3. ukazuje na to da lokalni optimum često daje neke informacije o globalnom. Zato se okoline lokalnog optimuma istražuju u potrazi za boljim rešenjem u nadi da će pretraga dovesti do globalnog optimuma.
- Osnovnu varijantu metode promenljivih okolina (Basic VNS, eng. *Basic Variable Neighbourhood Search*) čine lokalna pretraga i stohastička procedura razmrdavanja (eng. *shaking*).
- Cilj razmrdavanja je da spreči da se pretraga zaglavi u nekom lokalnom optimumu.

Metoda promenljivih okolina

1. Inicijalizacija: Izbor skupa okolina $N_k, k = 1, ..., k_{max}$; Konstruisanje početnog rešenja x; Izbor kriterijuma zaustavljanja.
2. Ponavljanje narednih koraka sve dok se ne ispuni kriterijum zaustavljanja:
    1. Postaviti k = 1
    2. Ponavljati naredne korake sve dok je $k \leq k_{max}$
        1. *Razmrdavanje* - Generisanje slučajnog rešenja x' iz okoline $N_k(x)$;
        2. *Lokalna pretraga* - Primeniti neku metodu lokalne pretrage sa početnim rešenjem x'. Rezultat pretrage označiti sa x'';
        3. *Prihvatanje rešenja i promena okoline* - Ako je tako dobijeno rešenje x'' bolje od x, postaviti $x = x''$ i $k = 1$; Inače, postaviti $k = k + 1$;

Tipovi okolina

1. $N(T)$ - struktura susedstva koje se koristi tokom lokalne pretrage. Članovi ove vrste susedstva se formiraju elementarnim transformacijama stabla.
2. $N_1(T)$ - struktura susedstva čiji se članovi dobijaju zamenom samo jednog čvora početnog stabla T. Odgovara operatoru mutacije pojedinačnog čvora kod GP.
3. $N_2(T)$ predstavlja operator zamene (eng. *Swap operator*). Odgovara operatoru mutacije celog podstabla kod GP.

Okoline $N_1(T)$ i $N_2(T)$ se koriste u proceduri razmrdavanja.

Razmrdavanje

- Dobija se k-ti sused stabla T, primenom istog poteza k puta.
- Prvo se nasumičnobira okolina $N_1(T)$ ili $N_2(T)$, a zatim se taj operator primenjuje k puta nad datim stablom.

Algoritam 5 Shake(T, k)

Ulaz: T - inicijalno stablo, k - broj ponavljanja primene operatora susedstva;
Izlaz: T' - stablo koje je sused stabla T;

$s \leftarrow$ slučajna vrednost iz $\{1, 2\}$;
for $i = 1, k$ **do**
    Na slučajan način izabrati $T' \in N_s(T)$;
    $T \leftarrow T'$;
end for
return T;

Elementarne transformacije stabla (ETT) i lokalna pretraga

- Susedi koji se razmatraju tokom lokalne pretrage se generišu elementarnim transformacijama trenutno najboljeg rešenja.
- Elementarne transformacije stabla (ETT, eng. *Elementary Tree Transformation*) se definišu na sledeći način. Neka je $G(V,E)$ neusmereni graf sa skupom čvorova V i skupom grana E i neka je $T(V,A)$ neko razapinjuće stablo grafa G. ETT transformiše stablo T u stablo T' (u oznaci $T' = ETT(T)$) sledećim koracima:
  1. U stablo T dodati granu a, takvu da $a \in E \setminus A$.
  2. Detektovati formirani ciklus i ukloniti bilo koju granu (osim one koja je dodata u prethodnom koraku) iz njega kako bi se dobio podgraf T', koji takođe predstavlja razapinjuće stablo grafa G.

Elementarne transformacije stabla (ETT) i lokalna pretraga

Algoritam 6 ETT(T(r, F, H, E, d))

```
Terminate ← false;
for each node i ∈ F do
  for each node j ∈ F ∪ H \ {i, parent(i), child(i)} do
    E' ← E ∪ {(i, j)}; // ciklus je formiran
    d_i ← d_i + 1; d_j ← d_j + 1;
    if j ∈ F then
      // slučaj 1
      E' ← E'\{(x, b)} / x ∈ {i, j}, b ∈ {parent(i), parent(j)};
      d_b ← d_b − 1;
      if x = i then
        d_i ← d_i − 1; E' ← E' ∪ {(b, child(j))};
        d_b ← d_b + 1; d_child(j) ← d_child(j) + 1;
        E' ← E'\{(j, child(j))};
        d_j ← d_j − 1; d_child(j) ← d_child(j) − 1;
      else
        d_j ← d_j − 1; E' ← E' ∪ {(b, child(i))};
        d_b ← d_b + 1; d_child(i) ← d_child(i) + 1;
        E' ← E'\{(i, child(i))};
        d_i ← d_i − 1; d_child(i) ← d_child(i) − 1;
      end if
    else
      // slučaj 2
      b ← parent(j); E ← E'\{(j, b)};
      d_j ← d_j − 1; d_b ← d_b − 1;
      E' ← E' ∪ {(b, child(i))};
      d_b ← d_b + 1; d_child(i) ← d_child(i) + 1;
      E' ← E'\{(i, child(i))};
      d_i ← d_i − 1; d_child(i) ← d_child(i) − 1;
    end if
    T' ← T(r, F, H, E', d);
    if f(T(r, F, H, E', d)) > f(T(r, F, H, E, d)) then
      E ← E'; Terminate ← true; Break;
    end if
  end for
  if Terminate = true then
    Break;
  end if
end for
```

Elementarne transformacije stabla (ETT) i lokalna pretraga

Naredbe algoritma se se mogu grupisati u četiri grupe (dve vrste dodavanja grane i dve vrste uklanjanja grane):

1. *Dodavanje grane (tip I)* (linije 4,5). Grana (i,j) se dodaje u trenutno stablo T, pri čemu ni i ni j ne smeju biti koren stabla. Dodatno, ili i ili j mora da bude funkcija. Nakon primene ovog koraka se formira ciklus u trenutnom stablu.
2. *Uklanjanje grane (tip I)* (linije 8,9,23,24). Ovde postoje dva slučaja. Ako su i i i j funkcije (slučaj 1 u algoritmu), onda se može ukloniti neka od grana $(i, parent(i))$, $(j, parent(j))$ kako bi se uklonio ciklus. Ako je jedan od čvorova terminal, npr. neka to bude j, onda se uklanja grana $(j, parent(j))$ (slučaj 2 u algoritmu).
3. *Dodavanje grane (tip II)* (linije 11,12,16,17,25,26). Ako je obrisana grana $(i, parent(i))$, onda se dodaje grana $(parent(i), child(j))$. Ako je obrisana grana $(j, parent(j))$, onda se dodaje grana $(parent(j), child(i))$.
4. *Uklanjanje grane (tip II)* (linije 13,14,18,19,27,28). Ove grane se uklanjaju kako bi stepen čvorova ostao zadovoljavajući.

Elementarne transformacije stabla (ETT) i lokalna pretraga

- Pomoću ETT je implementirana lokalna pretraga koja koristi strategiju *prvog poboljšanja*.
- Kada se naiđe na prvo stablo koje daje bolje rezultate na datom skupu podataka, ono se vraća kao trenutno najbolje rešenje iz čijeg susedstva će se dalje nastaviti pretraga.
- Kao funkcija na osnovu koje se poredi kvaliet stabala, uzet je koeficijent determinacije R^2.

Algoritam osnovne metode promenljivih okolina

Algoritam 7 Basic VNP(T, k$_{max}$)

repeat
  $k \leftarrow 1$;
  while $k < k_{max}$ **do**
    $T' \leftarrow Shake(T, k)$;
    $T'' \leftarrow ETT(T')$;
    $Neighborhood\_change(T, T'', k)$;
  end while
until nije ispunjen kriterijum zaustavljanja;

- Kriterijum zaustavljanja biti pronalazak tačnog rešenja, maksimalan dozvoljeni broj iteracija ili maksimalno vreme izvršavanja.

Podaci

Sve metode su testirane pomoću tri vrste skupova podataka:

1. Skup podataka generisan na osnovu funkcija koje se često razmatraju u literaturi. Za svaku funkciju je generisano 100 instanci na osnovu slučajno odabranih vrednosti nezavisnih promenljiih. Funkcije i intervali vrednosti iz kojeg su birane tačke su:
$F_1 = x^3 + x^2 + x, \quad x \in [-1, 1]$
$F_2 = x^4 + x^3 + x^2 + x, \quad x \in [-1, 1]$
$F_3 = x^5 + x^4 + x^3 + x^2 + x, \quad x \in [-1, 1]$
$F_4 = x^6 + x^5 + x^4 + x^3 + x^2 + x, \quad x \in [-1, 1]$
$F_5 = sin(x^2)cos(x) - 1, \quad x \in [-1, 1]$
$F_6 = sin(x) + sin(x + x^2), \quad x \in [-1, 1]$
$F_7 = log(x + 1) + log(x^2 + 1), \quad x \in [0, 2]$
$F_8 = sin(x_0) + sin(x_1^2), \quad x_0, x_1 \in [-1, 1]$
$F_9 = 2sin(x_0)cos(x_1), \quad x_0, x_1 \in [-1, 1]$

Podaci

2. Skup podataka generisan na osnovu jednostavnijih funkcija radi upoređivanja metaheurističkih metoda sa metodom grube sile.
$F_{01} = x_0 x_1 + x_1$
$F_{02} = x_1 + x_1^2 + x_0$
$F_{03} = x_0 x_1 + cos(x_0)$
$F_{04} = x_0 - x_1 x_1$
$F_{05} = x_0 - x_1 x_1 + x_1$

Podaci

3. Jedan od javno dostupnih skupova podataka za regresiju - "Yacht Hydrodynamics" skup. Kod njega je cilj predvideti vrednost rezidualnog otpora jahte na osnovu njenih karakteristika. Skup sadrži 308 instanci koje su određene pomoću 6 nezavisnih i jedne ciljne promenljive. Sve vrednosti su realnog tipa.

Evaluacija algoritma grube sile i poređenje sa metaheurističkim metodama

- Algoritam grube sile je uspešno mogao da dođe do optimalnog rešenja za primere $F_{01}, ..., F_{05}$ i F_1 i F_2.
- U ostalim primerima je, nakon više sati izvršavanja, dolazilo do nedostatka memorijskih resursa, bez pronalaska dovoljno dobrog rešenja.
- Svi metaheuristički pristupi su, uspešno pronašli optimalno rešenje za primere $F_{01}, ..., F_{05}$.

Evaluacija algoritma grube sile i poređenje sa metaheurističkim metodama

Tabela: Vreme izvršavanja (izraženo u sekundama) svih metoda na poblemima manjih dimenzija

	Algoritam grube sile	GP	GP sa SSC	VNP
F_{01}	1	12	19	4
F_{02}	<1	7	13	6
F_{03}	<1	6	12	6
F_{04}	<1	12	18	5
F_{05}	<1	7	18	6
F_{1}	<1	15	24	7
F_{2}	22	13	26	8

Evaluacija i poređenje metaheurističkih metoda

- Svaki skup podataka je podeljen na trening (70%) i test (30%) deo.
- Svaka metoda je evaluirana tako što je pokrenuta po 30 puta nad svim skupovima podataka.
- Prilikom svakog od tih nezavisnih pokretanja dobijen je izraz koji u tom izvršavanju najbolje odgovara datom skupu podataka.
- Za taj izraz je zatim izračunat koeficijent determinacije R^2 na trening i test skupu i provereno je da li i simbolički odgovara ciljnom izrazu.
- Pri svakom pokretanju mereno je i vreme koje je bilo potrebno za pronalazak najboljeg rešenja.

Evaluacija i poređenje metaheurističkih metoda

Tabela: Vrednosti parametara genetskog programiranja

Parametar	Vrednost
Veličina populacije	500
Maksimalan broj generacija	50
Broj jedinki za reprodukciju	300
Veličina turnira	3
Verovatnoća mutacie pojedinačnog čvora	0.2
Verovatnoća mutacije celog podstabla	0.2
Minimalna dubina stabla	2
Maksimalna dubina stabla u fazi inicijalizacije	6
Maksimalna veličina stabla	16
Vrsta funkcije prilagođenosti	adjusted
Skup funkcija	+, -, *, /, pow, sin, cos, log

Evaluacija i poređenje metaheurističkih metoda

Tabela: Vrednosti parametara genetskog programiranja za "Yacht Hydrodynamicsškup podataka

Parametar	Vrednost
Veličina populacije	2000
Maksimalan broj geneacija	100
Broj jedinki za reprodukciju	1200
Veličina turnira	10
Verovatnoća mutacie pojedinačnog čvora	0.2
Verovatnoća mutacije celog podstabla	0.2
Minimalna dubina stabla	2
Maksimalna dubina stabla u fazi inicijalizacije	12
Maksimalna veličina stabla	64
Vrsta funkcije prilagođenosti	adjusted
Skup funkcija	+, -, *, /, pow, sin, cos, log

Evaluacija i poređenje metaheurističkih metoda

Tabela: Vrednosti parametara metode promenljivih okolina

Parametar	Vrednost
k_{max}	4
Minimalna dubina stabla	2
Maksimalna dubina stabla u fazi inicijalizacije	4
Maksimalna dubina stabla kreiranog u fazi prerage	6
Maksimalan broj iteracija	1000
Skup funkcija	+, -, *, /, pow, sin, cos, log

Kod "Yachtškupa je za k_{max} uzeta vrednost 6, a broj iteracija je povećan na 2000.

Evaluacija i poređenje metaheurističkih metoda

Tabela: Prosečne vrednosti određenih karakteristika u 30 nezavisnih pokretanja

	Prosečna R^2 vrednost na trening skupu			Prosečna R^2 vrednost na test skupu		
	GP	GP sa SSC	VNP	GP	GP sa SSC	VNP
F_{01}	0.879	0.831	**0.949**	0.898	0.861	**0.945**
F_{02}	0.765	0.802	**0.848**	0.791	0.818	**0.897**
F_{03}	0.745	0.733	**0.863**	0.810	0.820	**0.926**
F_{04}	0.733	0.740	**0.914**	0.820	0.775	**0.922**
F_{05}	0.795	0.771	**0.846**	0.797	0.765	**0.813**

Evaluacija i poređenje metaheurističkih metoda

Tabela: Prosečne vrednosti određenih karakteristika u 30 nezavisnih pokretanja

	Broj pokretanja u kojima je pronađeno rešenje simbolički ekvivalentno ciljnom rešenju			Prosečno vreme izvršavanja (s)		
	GP	GP sa SSC	VNP	GP	GP sa SSC	VNP
F_{01}	7	3	**13**	12	19	**4**
F_{02}	1	3	**11**	7	13	**6**
F_{03}	5	5	**14**	6	12	**5**
F_{04}	2	1	**9**	12	18	**5**
F_{05}	3	1	**9**	**7**	18	**7**

Evaluacija i poređenje metaheurističkih metoda

Tabela: Prosečne vrednosti određenih karakteristika u 30 nezavisnih pokretanja

	Prosečna R^2 vrednost na trening skupu			**Prosečna R^2 vrednost na test skupu**		
	GP	**GP sa SSC**	**VNP**	**GP**	**GP sa SSC**	**VNP**
F_1	**0.914**	0.861	0.907	**0.907**	0.827	0.872
F_2	**0.827**	0.799	0.771	**0.824**	0.798	0.770
F_3	**0.851**	0.851	0.797	0.695	-1.428	**0.752**
F_4	0.746	0.691	**0.809**	**0.796**	0.743	0.778
F_5	0.643	0.606	**0.894**	0.607	0.589	**0.887**
F_6	0.928	0.917	**0.945**	0.883	0.881	**0.930**
F_7	0.960	0.968	**0.994**	0.950	0.959	**0.993**
F_8	0.857	0.837	**0.968**	0.716	0.657	**0.936**
F_9	0.950	0.940	**0.963**	0.955	0.938	**0.971**

Evaluacija i poređenje metaheurističkih metoda

Tabela: Prosečne vrednosti određenih karakteristika u 30 nezavisnih pokretanja

	Broj pokretanja u kojima je pronađeno rešenje simbolički ekvivalentno ciljnom rešenju			**Prosečno vreme izvršavanja (s)**		
	GP	**GP sa SSC**	**VNP**	**GP**	**GP sa SSC**	**VNP**
F_1	0	0	**13**	15	24	**7**
F_2	0	0	**9**	13	26	**9**
F_3	0	0	**2**	20	23	**10**
F_4	0	0	**2**	14	27	**12**
F_5	0	0	0	**14**	24	**14**
F_6	0	0	**1**	13	27	**12**
F_7	0	0	0	18	51	**13**
F_8	0	0	**3**	13	35	**7**
F_9	1	1	**2**	12	28	**8**

Evaluacija i poređenje metaheurističkih metoda

Tabela: Prosečne vrednosti određenih karakteristika u 30 nezavisnih pokretanja za "Yacht Hydrodynamicsškup podataka

Metoda	**Prosečna R^2 vrednost na trening skupu**	**Prosečna R^2 vrednost na test skupu**	**Prosečno vreme izvršavanja (s)**
1*GP	0.238	0.264	183
1*GP sa SSC	0.213	0.233	247
1*VNP	**0.477**	**0.457**	**30**

Evaluacija i poređenje metaheurističkih metoda

Tabela: Informacije o izrazu koji daje maksimalnu R^2 vrednost na test skupu od svih izraza dobijenih pri 30 nezavisnih pokretanja

	Maksimalna R^2 vrednost na test skupu			Izraz koji ima maksimalnu R^2 vrednost na test skupu		
	GP	GP sa SSC	VNP	GP	GP sa SSC	VNP
F_{01}	1.0	1.0	1.0	$x_1(x_0+1)$	$x_1(x_0+1)$	$x_1(x_0+1)$
F_{02}	1.0	1.0	1.0	$x_0+x_1^2+x_1$	$x_0+x_1^2+x_1$	$x_0+x_1^2+x_1$
F_{03}	1.0	1.0	1.0	$x_0x_1+cos(x_0)$	$x_0x_1+cos(x_0)$	$x_0x_1+cos(x_0)$
F_{04}	1.0	1.0	1.0	$x_0-x_1^2$	$x_0-x_1^2$	$x_0-x_1^2$
F_{05}	1.0	1.0	1.0	$x_0-x_1^2+x_1$	$x_0-x_1^2+x_1$	$x_0-x_1^2+x_1$

Evaluacija i poređenje metaheurističkih metoda

Tabela: Informacije o izrazu koji daje maksimalnu R^2 vrednost na test skupu od svih izraza dobijenih pri 30 nezavisnih pokretanja

	Simbolička ekvivalencija sa ciljnim izrazom		
	GP	**GP sa SSC**	**VNP**
$\boldsymbol{F}_{01}$	Da	Da	Da
$\boldsymbol{F}_{02}$	Da	Da	Da
$\boldsymbol{F}_{03}$	Da	Da	Da
$\boldsymbol{F}_{04}$	Da	Da	Da
$\boldsymbol{F}_{05}$	Da	Da	Da

Evaluacija i poređenje metaheurističkih metoda

Tabela: Informacije o izrazu koji daje maksimalnu R^2 vrednost na test skupu od svih izraza dobijenih pri 30 nezavisnih pokretanja

	Maksimalna R^2 vrednost na test skupu			Simbolička ekvivalencija sa ciljnim izrazom		
	GP	GP sa SSC	VNP	GP	GP sa SSC	VNP
F_1	0.992	0.985	1.0	Ne	Ne	Da
F_2	0.994	0.981	1.0	Ne	Ne	Da
F_3	0.981	0.995	1.0	Ne	Ne	Da
F_4	0.986	0.960	1.0	Ne	Ne	Da
F_5	0.943	0.964	0.999	Ne	Ne	Ne
F_6	0.971	0.987	1.0	Ne	Ne	Da
F_7	0.997	0.998	0.999	Ne	Ne	Ne
F_8	0.999	0.994	1.0	Ne	Ne	Da
F_9	1.0	1.0	1.0	Da	Da	Da

Evaluacija i poređenje metaheurističkih metoda

Tabela: Informacije o izrazu koji daje maksimalnu R^2 vrednost na test skupu od svih izraza dobijenih pri 30 nezavisnih pokretanja za "Yacht Hydrodynamicsškup podataka

Metoda	**Maksimalna R^2 vrednost na test skupu**	**Izraz koji ima maksimalnu R^2 vrednost na test skupu**
GP	0.929	$1.906^{(3.963x_5)}$
GP sa SSC	0.792	$\frac{0.822}{0.052^{x_5}}$
VNP	**0.956**	$0.000196^{\frac{x_1 - x_2 x_5}{x_2} * \frac{x_5}{x_1 - 2x_5^2}}$

Literatura I

J. R. Koza.
Genetic programming, on the programming of computers by means of natural selection.
MIT Press, 1998.

R. I. M. Quang Uy Nguyen, Nguyen Xuan Hoai.
Semantically-based crossover in genetic programming: Application to real-valued symbolic regression.
Genetic Programming and Evolvable Machines, 12:91–119, 2011.

Literatura II

M. T. Silviu-Marian Udrescu.
Ai feynman: a physics-inspired method for symbolic regression.
Science Advances, 6:eaay2631, 2020.

N. M. J. P. Souhir Elleuch, Bassem Jarboui.
Variable neighborhood programming for symbolic regression.
Optimization Letters, 16:191–210, 2020.