

Večkriterijska rekonstrukcija numerično kodiranih proceduralnih modelov dreves z diferencialno evolucijo

Aleš Zamuda, Janez Brest

Inštitut za računalništvo
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
Smetanova ul. 17, 2000 Maribor, Slovenija
E-pošta: ales.zamuda@uni-mb.si

Multi-objective Numerically Coded Procedural Tree Models Reconstruction by Differential Evolution

This paper presents an approach for multi-objective reconstruction of procedural models for woody plants (trees). The tree model works by building the 3D structure of a tree by applying a fixed procedure on a given set of numerically coded input parameters to recursively compute building parts of a tree. The parametrized procedural model can later be used for computer animation.

Reconstruction of a parametrized procedural model from the photographic images is done by differential evolution algorithm, which evolves the parametrized procedural model by fitting a set of its rendered images to a set of given photographic images. The two-objective comparisons are made on a pixel level of the images by integrating distances to nearest similar pixels. The obtained results show that the presented approach is viable for modeling of natural trees in computer animation by multi-objective evolution of the numerically coded procedural model. The use of multi-objective approach gives the decision maker a chance to select the final result.

1 Uvod

V članku predstavljamo nov večkriterijski pristop pri načrtovanju 3D modelov za naravna drevesa, ki jih uporabljamo v računalniški grafiki in animaciji. Rekonstrukcija združuje proceduralno modeliranje naravnih dreves [16] in večkriterijske evoliucijske algoritme [7]. Beaumont in Stepney sta na CEC 2009 [2] predstavila metodo, ki rekonstruira 2D proceduralne modele. Slednjo metodo za rekonstrukcijo smo že razširili na domeno 3D proceduralnih modelov [15]. Sedaj metodo razširjamo še z vpeljavo večkriterijske optimizacije in dodatnih metrik. Za izboljšanje kakovosti ujemanja dobljenih parametriziranih proceduralnih modelov z referenčnimi drevesi smo namreč uvedli več metrik in v evoliucijskem procesu uporabili večkriterijsko optimizacijo [14].

V naslednjem poglavju podrobneje predstavimo sorodna dela. V tretjem poglavju opišemo predlagan rekonstrukcijski postopek proceduralnih modelov naravnih dreves, ki za preiskovanje uporablja večkriterijsko dife-

rencialno evolucijo. V četrtem poglavju izvedemo eksperimente in opišemo rezultate ter diskutiramo, da predstavljen evoliucijski pristop omogoča modeliranje naravnih dreves. V petem poglavju podamo zaključek in predloge za nadaljnje delo.

2 Sorodna dela

Za predstavitev 3D drevesa uporabljamo zgrajen proceduralni model [16], katerega skušamo ustrezno parametrizirati. Rekonstrukcijski postopek temelji na analizi 2D slik naravnih dreves. Rekonstrukcija je toliko bolj uspešna, kolikor so si slike referenčnih dreves in slike upodobljenih parametriziranih proceduralnih modelov podobne. Parametrizacija proceduralnega modela poteka postopno, z evolucijo po vzoru iz narave, za kar uporabimo enega od trenutno svetovno najboljših evoliucijskih algoritmov, algoritem jDE [4, 6], ki smo ga nadgradili z večkriterijskim optimiranjem [7, 14].

2.1 Večkriterijska optimizacija

Večkriterijska optimizacija predstavlja reševanje večkriterijskega optimizacijskega problema. Slednjega definiramo kot iskanje dopustnega vektorja spremenljivk oziroma iskalnih parametrov $\mathbf{x} = (x_1, x_2, \dots, x_D)$, ki optimizira (v članku bomo na optimizacijo gledali kot na minimizacijo) vektorsko funkcijo $\mathbf{f}(\mathbf{x})$:

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})).$$

Z ovrednotenjem vektorja \mathbf{x} iz prostora spremenljivk \mathbb{R}^D dobimo kriterijski vektor $\mathbf{f}(\mathbf{x})$ v prostoru kriterijev \mathbb{R}^M . Vektor \mathbf{x} mora zadoščati m omejitvam:

$$g_i(\mathbf{x}) \leq 0; \quad i = 1, 2, \dots, m,$$

s katerimi podamo enostavne omejitve, kot npr. defini-cijsko območje in zaloga vrednosti optimizirane funkcije. Rešitev večkriterijskega optimizacijskega problema \mathbf{x} , dominira rešitev \mathbf{y} ($\mathbf{x} \preceq \mathbf{y}$), če velja:

- rešitev \mathbf{x} ni slabša od rešitve \mathbf{y} po vseh kriterijih $f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall i = 1, \dots, M$ in
- rešitev \mathbf{x} je boljša od rešitve \mathbf{y} po vsaj enem kriteriju $\exists j \in \{1, \dots, M\}: f_j(\mathbf{x}) < f_j(\mathbf{y})$.

Pareto optimalna rešitev je takšna rešitev \mathbf{x} , ki je ne dominira nobena druga dopustna rešitev \mathbf{z} in bi veljalo $\mathbf{f}(\mathbf{z}) \preceq \mathbf{f}(\mathbf{x})$. Množica nedominiranih rešitev v množici rešitev P je množica vseh tistih rešitev, ki jih ne dominira nobena rešitev iz množice P . Kriterijski vektorji rešitev iz množice nedominiranih rešitev tvorijo nedominirano fronto. Nalogo večkriterijskega optimiranja lahko tako pretvorimo v nalogo iskanja aproksimacije množice nedominiranih rešitev.

2.2 Diferencialna evolucija

Diferencialna evolucija¹ (DE) [13] je algoritem, ki se uspešno uporablja za globalno optimizacijo realno kodiranih numeričnih funkcij. Algoritem zaradi svoje narave prilagajanja problemu in stabilnosti iskanja z elitističnim selekcijskim mehanizmom daje boljše rezultate od ostalih evolucijskih algoritmov [4, 10]. Algoritem DE [13] je sestavljen iz glavne evolucijske zanke, v kateri z evolucijskimi operatorji mutacije, križanja in selekcije postopno in vzporedno izboljšuje približek iskane rešitve. Evolucijski operatorji vplivajo na vsak primerek \mathbf{x}_i , $\forall i \in [0, NP]$ v populaciji rešitev, iz katerih se zgradi nova populacija za naslednjo generacijo. Eno kreiranje novega osebka imenujemo iteracija, skupno število ovrednotenih posameznikov pa označimo s FEs. V vsaki iteraciji operator mutacije izračuna mutiran vektor $\mathbf{v}_{i,G+1}$:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}),$$

kjer so $r_1, r_2, r_3 \in 1, 2, \dots, NP$ paroma in od i različni indeksi primerkov iz populacije v generaciji G , $i \in 1, 2, \dots, NP$ in $F \in [0, 2]$. Vektor r_1 imenujemo osnovni vektor. Izraz $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$ imenujemo diferenčni vektor in po množenju s faktorjem ojačanja F , utežen diferenčni vektor.

Po mutaciji dobljeni mutiran vektor $\mathbf{v}_{i,G+1}$ križamo s ciljnim vektorjem $\mathbf{x}_{i,G}$ in tako dobimo poskusni vektor $\mathbf{u}_{i,G+1}$. Binarni operator križanja v algoritmu DE zapišemo kot:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{rand}(0, 1) \leq CR \text{ ali } j = j_{rand} \\ x_{i,j,G} & \text{sicer} \end{cases},$$

kjer $j \in [1, D]$ označuje j -ti iskalni parameter v prostoru z D dimenzijami, funkcija $\text{rand}(0, 1) \in [0, 1]$ označuje vzorčenje uniformno (psevdo) naključno porazdeljenega naključnega števila in j_{rand} izbira uniformno naključen indeks iskalnega parametra, ki ga vedno izmenjamo (da bi s tem preprečili izdelavo enakih posameznikov). CR označuje že omenjen krmilni parameter stopnje križanja.

Poznamo več različic večkriterijskega algoritma DE, izmed katerih smo izbrali algoritem MOjDE [7, 14]. Algoritem MOjDE vključuje mehanizem samoprilagajanja krmilnih parametrov F in CR , ki so ga uvedli Brest s sod. [4]. V začetni populaciji smo parametra inicializirali uniformno naključno. Algoritem jDE ima številne razširitve [8, 5, 6], ki v tej študiji niso vključene.

¹Morda bi bil primeren prevod tudi diferenčna evolucija.

2.3 Numerično kodirani proceduralni model drevesa

Za predstavitev modela drevesa smo uporabili numerično kodirani proceduralni model iz programskega sistema EcoMod [16]. Parametriziran proceduralni model zgradi 3D strukturo drevesa z izvajanjem fiksnega postopka oz. procedure nad danim naborom numerično kodiranih vhodnih parametrov, kot so npr. širina debla, relativna dolžina vej in vejitvena struktura. Ta procedura rekurzivno izračunava sestavne dele drevesa. Parametriziran proceduralni model lahko kasneje uporabimo za računalniško animacijo, saj s spremembo glavnih ali pomožnih parametrov modela dobimo nekoliko spremenjene modele, ki so primerni za izračunavanje ključnih animacijskih okvirjev [17].

Izbran model se od nekaterih drugih modelov [1, 3, 9, 12, 11] razlikuje v tem, da je popolnoma numerično kodiran in je nespremenljive dimenzije. Slednje nam omogoča enostavno uporabo diferencialne evolucije [15].

3 Večkriterijska rekonstrukcija numerično kodiranih proceduralnih modelov dreves z diferencialno evolucijo

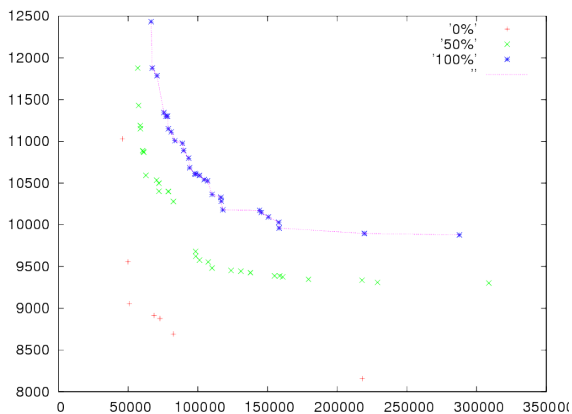
Rekonstrukcijski algoritem kodira parametre proceduralnega modela drevesa v genotip posameznika v populaciji DE. Na podlagi teh parametrov večkrat upodobi sliko drevesa, jo oceni z dvema metrikama in skuša izboljšati. V nadaljevanju opisujemo nekatere pomembnejše dele optimizacijskega postopka, tj. kodiranje genotipa, transformacijo v fenotip in njegovo ovrednotenje s primerjavo z referenčno sliko.

3.1 Kodiranje genotipa

Posameznik iz populacije predstavlja nabor parametrov za parametrizacijo proceduralnega modela. Posameznikov genotip \mathbf{x} kodira nabor parametrov za slednji model. Celoten proceduralni model drevesa, ki z izračunavanjem iz nabora parametrov upodobi naravno drevo, je dobro opisan v literaturi. Genotip $x_{i,j} \in [0, 1]$ je dimenzije $D = 4509$, z mutacijo in križanjem ga spreminjamo z DE, ter za $i \in 1..NP$ ter $j \in 1..D$ kodira seznam parametrov, kot opisano v literaturi [15].

3.2 Transformacija iz genotipa v fenotip

Kot smo omenili v uvodu, rekonstrukcijski postopek temelji na rekonstrukciji iz 2D slik naravnih dreves. Da bi lahko ujemanje genotipa za parametriziran proceduralni model primerjali s sliko referenčnega drevesa \mathbf{z}^* , ga moramo transformirati v fenotip. Fenotip v našem primeru predstavlja upodobljeno sliko \mathbf{z} . Sliki \mathbf{z}^* in \mathbf{z} sta dimenzij $X \times Y$. Obe sliki pred dejansko primerjavo še pretvorimo v črno beli, tako da vsak piksel, ki je različen od ozadja, postavimo na 1 (črno), ostale piksele na 0 (belo). Sliko referenčnega drevesa še skaliramo na dimenzijo slike za



Slika 1: Ploskve 0%, 50% in 100% dosega za $NP=100$.

upodabljanje proceduralnega modela. Upodobitev proceduralnega modela vzorčimo dvakrat, iz različnih kotov, z razliko 90° , da dobimo različne projekcije in se izognemo ploskim drevesom.

3.3 Primerjava fenotipa in referenčne slike

Rekonstrukcija je toliko bolj uspešna, kolikor so si referenčne slike dreves in slike upodobljenih parametriziranih proceduralnih modelov podobne. V ta namen smo izbrali dva kriterija. Metrika za prvi kriterij je vsota manhattanskih razdalj od vsakega črnega piksla slike do najbližjega črnega piksla referenčne slike in obratno [2]. Metrika za drugi kriterij je število različnih pikslov med slikama. Posamezni fenotip smo tako ovrednotili s kriterijema f_1 in f_2 :

$$f_1(\mathbf{x}) = f(\mathbf{g}(\mathbf{x}, \beta_1), \mathbf{g}(\mathbf{x}, \beta_2)) = h_1(\mathbf{z}_1) + h_1(\mathbf{z}_2)$$

$$h_1(\mathbf{z}_i) = \sum_{x,y} m_1(z_{x,y}^i, z_{x,y}^*) + \sum_{x,y} m_1(z_{x,y}^*, z_{x,y}^i)$$

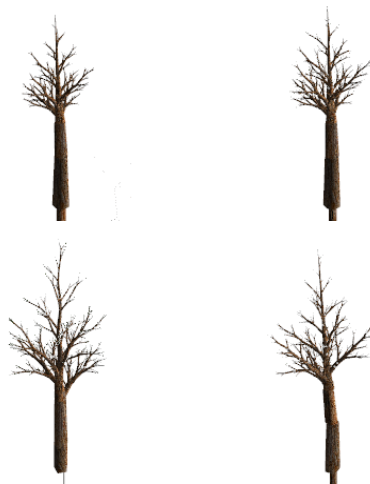
$$f_2(\mathbf{x}) = f(\mathbf{g}(\mathbf{x}, \beta_1), \mathbf{g}(\mathbf{x}, \beta_2)) = h_2(\mathbf{z}_1) + h_2(\mathbf{z}_2)$$

$$h_2(\mathbf{z}_i) = \sum_{x,y} w(z_{x,y}^i, z_{x,y}^*) + \sum_{x,y} w(z_{x,y}^*, z_{x,y}^i),$$

kjer z m_1 označujemo funkcijo za izračun manhattanske razdalje do najbližjega piksla v sliki \mathbf{z}^* z vrednostjo 1, w funkcijo za razlikovanje istoležečih pikslov $i \in \{1, 2\}$.



Slika 2: Drevo, ki ga rekonstruiramo.



Slika 4: Upodobitev končne aproksimacijske množice parametriziranih proceduralnih modelov za prvi zagon.

4 Rezultati

Algoritem smo preizkusili na testnem primeru iz slike 2. Dimenzijo vzorčenja upodabljanja parametriziranega proceduralnega modela smo nastavili na 250×250 , največje število žil v drevesu $S = 410$, ter velikost populacije jDE na $NP = 100$, ostale parametre smo pustili privzete, kot so v originalnih algoritmih iz literature. Dobljene ploskve dosega evucijskega procesa za 30 zagonov vidimo na sliki 1. Na sliki [?] vidimo slike posameznih izboljšav v evucijskem procesu za prvi zagon. Na sliki 4 vidimo končno aproksimacijsko množico velikosti 10 za prvi zagon.

5 Zaključek

Predstavili smo algoritem za večkriterijsko rekonstrukcijo proceduralnih modelov dreves. Rekonstrukcija poteka iterativno, s pomočjo diferencialne evolucije, ki naključno vzorči parametre proceduralnega modela drevesa. Dobljene modele upodobimo v programskem sistemu EcoMod in dobljene slike primerjamo z referenčno sliko, da bi dobili čim boljše ujemanje. Ustreznost je merjena z več kriteriji, ki niso posebej uteženi, zato uporabljamo večkriterijsko optimizacijo. Prikazali smo še primer rekonstrukcije drevesa in podali konvergenčni graf za izbran evucijski zagon.

V nadaljevanju raziskovalnega dela bi lahko uporabili segmentacijske postopke za poravnavo in samodejno razpoznavo lege osnovnega debla drevesa na fotografijah iz narave, prikazali razpoznavo več dreves in pri izračunu podobnosti slik namesto črno-belih slik uporabljali vsote volumetričnih podatkov [11]. Za vodenje optimizacijskega procesa bi lahko uporabili tudi interaktivne metode, ki bi jih kombinirali s preferencami v evucijskem algoritmu. Ker uporabljen programski sistem EcoMod omogoča tudi upodabljanje ekosistemov, bi lahko kasneje



Slika 3: Upodobitev v trenutni populaciji Pareto optimalnih parametriziranih proceduralnih modelov skozi evolucijo, za prvi zagon.

razpoznavali pokrajinsko geometrijo z drevesnimi ekosistemi, tako razporeditev dreves kot tudi njihove parametrizirane proceduralne modele.

Literatura

- [1] M. Aono and T. Kunii. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34, 1984.
- [2] Darren Beaumont and Susan Stepney. Grammatical Evolution of L-systems. In *The 2009 IEEE Congress on Evolutionary Computation CEC 2009*, pages 2446–2453. IEEE Press, 2009.
- [3] J. Bloomenthal. Modeling the mighty maple. In B. A. Barsky, editor, *SIGGRAPH '85 Conference Proceedings (San Francisco, CA, 22–26 July 1985)*, pages 305–311, 1985.
- [4] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.
- [5] J. Brest and M. Sepesy Maučec. Population Size Reduction for the Differential Evolution Algorithm. *Applied Intelligence*, 29(3):228–247, 2008.
- [6] J. Brest, A. Zamuda, B. Bošković, M. S. Maučec, and V. Žumer. Dynamic Optimization using Self-Adaptive Differential Evolution. In *IEEE Congress on Evolutionary Computation 2009*, pages 415–422. IEEE Press, 2009.
- [7] J. Brest, A. Zamuda, B. Bošković, and V. Žumer. Večkriterijska optimizacija: primerjava algoritmov MOjDE in DEMO. In *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, volume B*, pages 85–88, 2007.
- [8] J. Brest, A. Zamuda, B. Bošković, and V. Žumer. An Analysis of the Control Parameters' Adaptation in DE. In Uday K Chakraborty, editor, *Advances in Differential Evolution, Studies in Computational Intelligence*, volume 143, pages 89–110. Springer, 2008.
- [9] M. Holton. Strands, gravity, and botanical tree imagery. *Comput. Graph. Forum*, 13(1):57–67, 1994.
- [10] Efrén Mezura-Montes and Blanca Cecilia Lopez-Ramirez. Comparing bio-inspired algorithms in constrained optimization problems. *The 2007 IEEE Congress on Evolutionary Computation*, pages 662–669, 25–28 Sept. 2007.
- [11] Boris Neubert, Thomas Franken, and Oliver Deussen. Approximate image-based tree-modeling using particle flows. *ACM Trans. Graph.*, 26(3):88, 2007.
- [12] W. Reeves. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of SIGGRAPH'85*, pages 313–322, 1985.
- [13] R. Storn and K. Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.
- [14] A. Zamuda, J. Brest, B. Bošković, and V. Žumer. Večkriterijska optimizacija: eksperimentalni rezultati algoritmov MOjDE in DEMO. In *Zbornik šestnajste mednarodne Elektrotehniške in računalniške konference ERK 2007, volume B*, pages 89–92, 2007.
- [15] A. Zamuda, J. Brest, B. Bošković, and V. Žumer. Woody Plants Model Recognition by Differential Evolution. In *The Fourth International Conference on Bioinspired Optimization Methods and their Applications, May 20 - 21 2010, Ljubljana, Slovenia*, pages 205–215, 2010.
- [16] A. Zamuda, J. Brest, N. Guid, and V. Žumer. Construction of virtual trees within ecosystems with ecomod tool. *Proceedings of IPSI-2006 Slovenia, International Conference on Advances in the Internet, Processing, Systems, and Interdisciplinary Research*, 2006.
- [17] A. Zamuda, J. Brest, N. Guid, and V. Žumer. Modelling, Simulation, and Visualization of Forest Ecosystems. In *The IEEE Region 8 EUROCON 2007: International conference on Computer as a tool*, September 9–12, 2007, Warsaw, Poland, pages 2600–2606. IEEE Press, 2007.