Večkriterijska optimizacija z evolucijskimi algoritmi

Tea Robič, Bogdan Filipič
Odsek za inteligentne sisteme
Institut "Jožef Stefan"
Jamova 39, SI-1000 Ljubljana
tea.robic@ijs.si, bogdan.filipic@ijs.si

Multi-Objective Optimization with Evolutionary Algorithms

Most practical optimization problems require several objectives to be optimized simultaneously. As a result, not only a single optimal solution, but a set of optimal solutions to a problem exists. Recently, evolutionary algorithms have been proposed as an efficient multi-objective optimization method as they are capable of providing multiple solutions in a single run. This paper introduces the domain of multi-objective optimization, the underlying concepts of solution dominance and Pareto optimality, and two ways of solving multi-objective optimization problems, the preference-based and ideal approach. It further reviews the traditional weighted sum technique and the newly-proposed multi-objective evolutionary algorithms. The paper concludes with a comment on open questions and directions for future work in this area.

1. Uvod

V praksi se pogosto srečujemo z zahtevo po sočasnem optimiranju po različnih kriterijih. Večkrat so si kriteriji tudi konfliktni, kar pomeni, da izboljšanje rešitve po enem kriteriju povzroči njeno poslabšanje po drugih kriterijih. Takrat nimamo opravka samo z eno optimalno rešitvijo, temveč z množico optimalnih rešitev. Rešitev je *Pareto optimalna*, če ne obstaja neka druga rešitev, ki bi bila boljša od te rešitve po vseh kriterijih. Če ne poznamo dodatne informacije, ki bi podala pomembnosti kriterijev, si želimo, da bi poznali vse Pareto optimalne rešitve in bi se šele nato odločili za eno izmed njih. Evolucijski algoritmi kot populacijska metoda omogočajo sočasno optimizacijo več rešitev v enem samem zagonu algoritma in so zato zelo primerni za večkriterijsko optimizacijo.

V nadaljevanju tega prispevka opisujemo večkriterijsko optimizacijo in jo primerjamo z enokriterijsko. Predstavljamo dva pristopa k reševanju problema večkriterijske optimizacije – prednostnega in idealnega. Na kratko opisujemo tudi najpogosteje uporabljeno neevolucijsko metodo za reševanje problemov večkriterijske optimizacije. V tretjem razdelku naštevamo lastnosti evolucijskih algoritmov, ki omogočajo učinkovito večkriterijsko optimizacijo. Na kratko opisujemo tri evolucijske algoritme, ki so se izkazali kot najboljši. To so NSGA-II, SPEA2 in PESA-II. Zaključimo z razmislekom o možnostih nadaljnjega raziskovanja na tem področju.

2. Večkriterijska optimizacija

Kot pove že ime, se večkriterijska optimizacija ukvarja z reševanjem problemov, v katerih nastopa več kriterijskih funkcij. Zaradi pomanjkanja metod, primernih za večkriterijsko optimizacijo, se je v preteklosti večkriterijske probleme reševalo kot enokriterijske. Vendar pa obstaja mnogo pomembnih razlik med enokriterijsko in večkriterijsko optimizacijo, ki jih bomo navedli v nadaljevanju.

2.1. Definicija problema

Problem večkriterijske optimizacije je definiran kot problem iskanja dopustnega vektorja spremenljivk, ki optimira vektorsko funkcijo, katere elementi so kriterijske funkcije. Drugače povedano, iščemo vektor $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ iz definicijskega območja, ki zadošča m neenakostim

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

in optimira vektorsko funkcijo

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})).$$

Ker lahko vsak problem maksimizacije kriterijev enostavno prevedemo na problem minimizacije, bomo v nadaljevanju vedno predpostavljali, da želimo vektorsko funkcijo $\mathbf{f}(\mathbf{x})$ minimizirati po vseh kriterijih.

Pri enokriterijski optimizaciji je prostor kriterijev množica realnih števil $\mathbb{R}$, ki je z relacijo $\leq$ popolno urejena. Tako za poljubni rešitvi $\mathbf{x}$ in $\mathbf{y}$ enokriterijskega optimizacijskega problema velja natanko ena od trditev "$\mathbf{x}$ je boljša od $\mathbf{y}$", "$\mathbf{x}$ je slabša od $\mathbf{y}$" ali "$\mathbf{x}$ in $\mathbf{y}$ sta enakovredni". Pri večkriterijskem optimizacijskem problemu pa je prostor kriterijev večdimenzionalen ($\mathbb{R}^k$). Zato za relacijo $\leq$ ne velja več popolna urejenost, temveč le delna urejenost. Dve rešitvi sta tako pogosto neprimerljivi. Večina večkriterijskih optimizacijskih algoritmov si pri primerjavi rešitev pomaga s konceptom dominantnosti.

2.2. Dominantnost in Pareto optimalnost

Rešitev večkriterijskega optimizacijskega problema $\mathbf{x}$ *dominira* rešitev $\mathbf{y}$ ($\mathbf{x} \preceq \mathbf{y}$), če sta izpolnjeni naslednji zahtevi:

1. Rešitev $\mathbf{x}$ ni slabša od rešitve $\mathbf{y}$ po vseh kriterijih ($\mathbf{f}_i(\mathbf{x}) \leq \mathbf{f}_i(\mathbf{y})$ za vse $i = 1, \ldots, k$).
2. Rešitev $\mathbf{x}$ je boljša od rešitve $\mathbf{y}$ po vsaj enem kriteriju ($\mathbf{f}_j(\mathbf{x}) < \mathbf{f}_j(\mathbf{y})$ za vsaj en $j \in \{1, \ldots, k\}$).

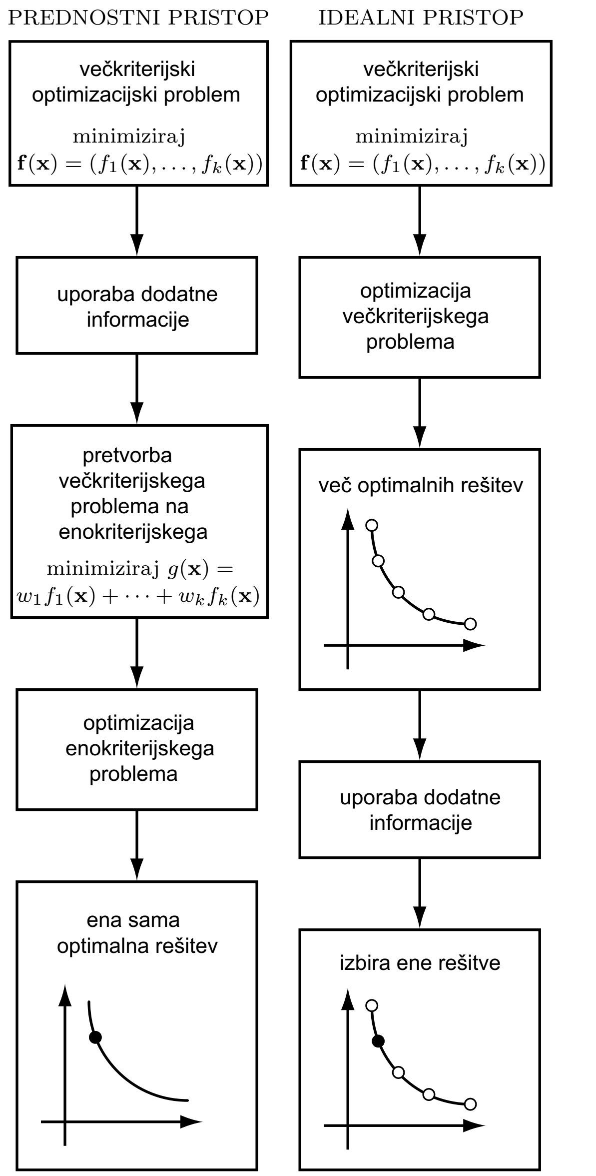
Množica nedominiranih rešitev v množici rešitev P je množica vseh tistih rešitev, ki niso dominirane s strani rešitev iz množice P. Množica nedominiranih rešitev iz celotnega prostora dopustnih rešitev se imenuje *Pareto fronta*, njeni elementi pa *Pareto optimalne rešitve* (po Vilfredu Paretu, italijanskem ekonomistu in sociologu ter pionirju na področju večkriterijske optimizacije).

2.3. Pristopi k reševanju večkriterijskih problemov

Če imamo pri večkriterijski optimizaciji opraviti s konfliktnimi kriteriji, obstaja več Pareto optimalnih rešitev. Brez dodatne informacije o pomembnosti posameznih kriterijev ne moremo povedati, katera od teh je boljša. Zato je večkriterijska optimizacija zahtevnejša od enokriterijske. Podobno kot pri enokriterijski optimizaciji, tudi pri večkriterijski navadno želimo dobiti samo eno optimalno rešitev. To lahko dosežemo z dvema pristopoma, ki sta prikazana na sliki 1.

Rešitev, ki jo dobimo s prednostnim pristopom, je seveda odvisna od funkcije, s katero smo večkriterijski problem pretvorili v enokriterijskega. Z drugačno funkcijo bi lahko dobili drugačno rešitev. Poleg tega ta pristop potrebuje dodatno informacijo o pomembnosti kriterijev, ki navadno ni vnaprej poznana.

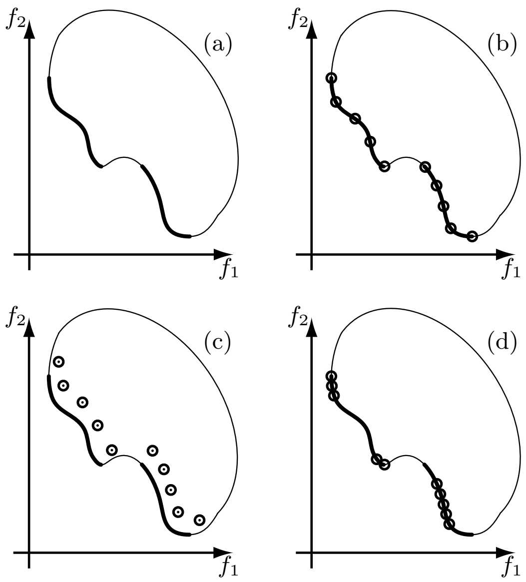
Pri idealnem pristopu pa najprej dobimo množico optimalnih rešitev, iz katere izberemo rešitev, ki nam najbolj ustreza. Tudi pri tem pristopu potrebujemo dodatno informacijo o problemu, vendar jo potrebujemo le za izbiro ene rešitve iz množice optimalnih in ne za konstrukcijo novih rešitev. Zato je idealni pristop bolj praktičen in manj subjektiven od prednostnega pristopa. Seveda, če poznamo informacijo o kriterijih, ki nam omogoča ciljno usmerjeno uporabiti prednostni pristop, ni nobenega razloga, da bi namesto tega uporabljali idealni pristop.



Slika 1: Prednostni in idealni pristop reševanja nalog večkriterijske optimizacije [4]

Metode za večkriterijsko optimizacijo lahko razdelimo na t.i. *klasične* in *evolucijske*. Klasične metode uporabljajo prednostni pristop. Najbolj razširjena je metoda utežene vsote (angl. weighted sum method), kjer večkriterijski problem pretvorimo v enokriterijskega tako, da izberemo uteži, ki določajo pomembnost kriterijev (ta pretvorba je bila uporabljena na sliki 1 levo). Na ta način za vsak nabor uteži dobimo eno optimalno rešitev problema. To je široko uporabljena in enostavna metoda, ki pa ima pomembno slabost. Ne glede na nastavitev uteži namreč ne more najti rešitev na konkavnem delu Pareto fronte.

Med klasične metode štejemo še metodo ϵ-omejitev (angl. ϵ-constraint method), metodo uteženih metrik (angl. weighted metric method), Bensonovo metodo



Slika 2: Primer kriterijskega prostora, kjer minimiziramo f_1 in f_2: Pareto fronta (a) ter množice nedominiranih rešitev (b), (c) in (d)

in druge (za opis teh metod glej [4]). Klasične metode večinoma dajejo eno samo rešitev in so občutljive na obliko in zveznost Pareto fronte. Evolucijske algoritme kot predstavnike idealnega pristopa predstavljamo v naslednjem razdelku.

Pri reševanju z idealnim pristopom želimo, da večkriterijska optimizacijska metoda najde čim več Pareto optimalnih rešitev. Ker bomo izmed teh rešitev (s pomočjo informacije o pomembnosti kriterijev) izbirali najboljšo, si želimo, da so dobljene rešitve kar se da enakomerno razporejene po prostoru kriterijev. Slika 2(a) prikazuje prostor kriterijev in Pareto fronto na njem (narisana poudarjeno), slika 2(b) pa "idealni" nabor Pareto optimalnih rešitev. Nalogo večkriterijske optimizacijske metode tako lahko prepišemo v nalogo iskanja množice nedominiranih rešitev, za katere velja:

– da so čim bliže Pareto fronti in
– da so enakomerno razporejene po prostoru kriterijev.

To pa sta pogosto dva konfliktna cilja. Sliki 2(c) in 2(d) prikazujeta dve množici nedominiranih rešitev, za kateri ne moremo reči, katera je boljša. Množica rešitev na sliki 2(c) je dobro razporejena, a je daleč od Pareto fronte, medtem ko je množica rešitev na sliki 2(d) Pareto optimalna, a neenakomerno razporejena.

2.4. Kakovost množic nedominiranih rešitev

Tudi pri manj ekstremnih primerih množic nedominiranih rešitev od tistih, ki sta prikazani na slikah 2(c) in 2(d), se pogosto postavlja vprašanje, kako oceniti kakovost množice. Če upoštevamo, da je mnogokrat težko primerjati dve rešitvi večkriterijskega optimizacijskega problema, si lahko predstavljamo, da je primerjanje dveh množic rešitev še zahtevnejši problem. V zadnjem času so se pojavili predlogi različnih mer za ocenjevanje kakovosti množic rešitev [6, 8]. Večina mer ocenjuje množice rešitev glede na izpolnjenost obeh ciljev večkriterijske optimizacije, tj. bližino Pareto fronti in enakomerno razporejenost rešitev. Vendar je oba cilja težko natančno opredeliti, kaj šele optimirati.

3. Evolucijski algoritmi

Evolucijski algoritmi temeljijo na Darwinovi teoriji o evoluciji: s selekcijo in kombinacijo genetskega zapisa osebkov skozi generacije ustvarjajo vedno boljše in boljše rešitve. Ti algoritmi se uspešno uporabljajo v enokriterijski optimizaciji. Ker so populacijski, so primerni tudi za naloge večkriterijske optimizacije, kjer (po idealnem pristopu) želimo v enem zagonu algoritma dobiti več nedominiranih rešitev. Poleg tega so robustni – nimajo težav pri optimizacijskih nalogah z nekonveksno ali nezvezno Pareto fronto.

V zadnjih dveh desetletjih je sunkovito naraslo število objav o uporabi evolucijskih algoritmov v večkriterijski optimizaciji [1]. Med njimi sta tudi dve knjigi [4, 2]. Predstavljeni so bili številni evolucijski pristopi k reševanju večkriterijskih problemov. Za najuspešnejše so se izkazale metode, ki so uporabljale koncept Pareto dominantnosti in elitizem – ohranjanje najboljših dobljenih rešitev. Metode se med seboj v glavnem razlikujejo po tem, na kakšen način opravljajo selekcijo in s kakšnimi prijemi dosegajo enakomerno razporejenost rešitev.

Zaradi prostorskih omejitev si bomo ogledali samo tri izbrane evolucijske metode za večkriterijsko optimizacijo. To so metode, ki so zaradi uporabe elitizma in domiselnih pristopov za doseganje enakomerne porazdelitve rešitev imenovane tudi metode druge generacije. Predstavljamo jih v naslednjih podrazdelkih.

3.1. NSGA-II

Deb in sodelavci [5] so predlagali učinkovit algoritem za večkriterijsko optimizacijo, ki uporablja elitizem tako, da v naslednjo generacijo vedno prenese najboljše rešitve iz predhodne in trenutne generacije. Algoritem rešitve najprej sortira glede na nedominiranost (rešitve rangira po frontah), nato pa še glede na gostoto rešitev v prostoru kriterijev. Na ta način hkrati optimira oba cilja večkriterijske optimizacije. Gostoto rešitev definira za vsako rešitev posebej s pomočjo metrike, s katero meri razdalje do sosednjih rešitev.

NSGA-II je priljubljen algoritem, ki se je v številnih primerjavah z drugimi algoritmi dobro izkazal. Njegova največja prednost je, da ne potrebuje nobenih dodatnih parametrov (poleg velikosti populacije), od katerih bi bila odvisna učinkovitost algoritma.

3.2. SPEA2

Algoritem SPEA2, ki so ga razvili Zitzler in sodelavci [7], implementira elitizem s pomočjo zunanjega arhiva najboljših dobljenih rešitev, ki je lahko drugačne velikosti kot populacija v evolucijskem algoritmu. Arhiv se osveži v vsaki generaciji. Vrednotenje posamezne rešitve poteka s pomočjo informacije o tem, koliko drugih rešitev iz populacije in arhiva ta rešitev dominira, in koliko drugih rešitev dominira njo. Poleg tega algoritem upošteva še informacijo o gostoti rešitev, ki se izračuna na podoben način, kot v metodi k najbližjih sosedov.

Algoritem SPEA2 se je izkazal kot zelo učinkovit, predvsem pri zagotavljanju enakomerne razporejenosti rešitev. Tudi ta algoritem odlikuje lastnost, da (poleg velikosti populacije in arhiva) ne potrebuje dodatnih parametrov.

3.3. PESA-II

PESA-II, Corna in sodelavcev [3], se od algoritmov NSGA-II in SPEA2 razlikuje predvsem po načinu selekcije. Uporablja namreč selekcijo s pomočjo hiperkock. Prostor kriterijev je razdeljen na hiperkocke, ki jih ocenimo glede na to, koliko rešitev vsebujejo. Najprej izvedemo selekcijo v hiperkockah, nato pa iz vsake hiperkocke izberemo naključno rešitev. Elitizem je izveden podobno kot pri metodi SPEA2 – z uporabo zunanje populacije najboljših rešitev.

Tudi PESA-II je dober algoritem, ki pa potrebuje nastavitev dodatnega parametra, ki pove velikost uporabljenih hiperkock.

4. Zaključek

Evolucijski algoritmi so zaradi svoje populacijske narave in robustnosti zelo uporabni za reševanje problemov večkriterijske optimizacije. Omogočajo namreč izračun več nedominiranih rešitev v enem samem zagonu algoritma. Kljub njihovi popularnosti in uspešnosti pa je na tem področju še veliko nerešenih vprašanj.

Ker imamo opravka z ocenjevanjem množic rešitev v večkriterijskem prostoru, je še veliko dela potrebnega pri analizi metrik. Zaenkrat še ni jasno, kakšna metrika je najbolj primerna za ocenjevanje množic rešitev in ali takšna metrika sploh obstaja [8]. Zato je veliko primerjav uspešnosti različnih algoritmov narejenih kar na podlagi grafičnih predstavitev nedominiranih rešitev (kot v našem primeru na sliki 2), ki pa pridejo v poštev samo v primeru dvokriterijske optimizacije.

Kot pogosto pri evolucijskih algoritmih, tudi tu pogrešamo bolj poglobljene analize o delovanju algoritmov, dokazih konvergence in podobno. Ker pa je to področje v zadnjih letih zelo aktivno, si lahko nadejamo še veliko predlogov novih metod in tudi teoretičnih študij.

Literatura

[1] C. A. Coello Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.

[2] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, New York, 2002.

[3] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 283–290, San Francisco, CA, 2001. Morgan Kaufmann.

[4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Chichester, UK, 2001.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the 6-th International Conference Parallel Problem Solving from Nature (PPSN-VI)*, pages 849–858, 2000.

[6] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Proceedings of the Congress on Evolutionary Computation (CEC2002)*, pages 711–716. IEEE Press, Piscataway, NJ, 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Tik report no. 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, 2001.

[8] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. D. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.