



## INTERAKTIV GENETIK ALGORITMLAR ASOSIDA CSS ORQALI VEB-INTERFEYS DIZAYNLARINI AVTOMATIK GENERASIYA QILISH METODIKASI

**Yakubjanova Dilduza Qodirovna<sup>1</sup>,**

**Safarov Sardor Nuriddin o'gli<sup>2</sup>**

Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari  
universiteti Samarqand filiali

*Annotatsiya. Maqolada veb-sahifa dizayn-maketlarini avtomatik ravishda yaratish va takomillashtirishda evolyutsion (genetik) yondashuvdan foydalanish metodikasi tavsiflanadi. CSS qoidalari «xromosoma» sifatida kodlanib, mutatsiya va skreujivaniye operatorlari orqali yangi dizayn variantlari hosil qilinishi, foydalanuvchi bahosi esa fitnes-funksiya vazifasini bajarishi asoslanadi. Tadqiqotda taklif etilgan axborot tizimi konsepsiyasiga tayanilgan holda interaktiv evolyutsion qidiruv sikli, uchta genetik algoritmi ssenariysi (butun maket, alohida element, adaptiv moslash) va IDEF3 orqali jarayonlar dekompozitsiyasi yoritiladi. Shuningdek, loyiha bosqichlari muddatlari hamda dissertatsiya kontenti bo'yicha statistik tahlil natijalari keltiriladi.*

***Kalit so'zlar:** evolyutsion metodlar, genetik algoritmi, interaktiv genetik algoritmi, CSS, veb-interfeys, dizayn-generatsiya, prototiplash, IDEF3, UX.*

**Kirish.** Veb-interfeys (UI) va veb-sahifa dizaynini ishlab chiqish jarayonida tayyor «shablon»lardan foydalanish tezlikni oshirsa-da, ko'p holatlarda mahsulotning o'ziga xosligi va brend talablarini to'liq qondirmaydi. Dizayn vazifasi bir vaqtda texnik cheklovlarga (responsive maket, HTML/CSS semantikasi, brauzerlar o'rtasidagi moslik) hamda sub'ektiv estetik mezonlarga (vizual muvozanat, kontrast, o'qilishlilik, navigatsiya qulayligi) tayanadi. Shu sabab



dizaynni optimallashtirish ko'p o'lchamli, nolney va qiyin formallashtiradigan muammolar toifasiga kiradi.

Maqolada dizaynni «optimallashtirish muammosi» sifatida ko'rib, evolyutsion qidiruv orqali «populyatsiya»dagi maketlarni bosqichma-bosqich yaxshilash g'oyasi ilgari surilgan. Bu yerda inson (foydalanuvchi) algoritm ishida faol ishtirok etib, hosil bo'lgan variantlarga baho beradi va qidiruv yo'nalishini belgilab boradi. Maqolaning maqsadi — ushbu yondashuvni metodik jihatdan tizimlashtirish, uning amaliy qo'llanish chegaralarini ko'rsatish va loyihani joriy etish rejasi hamda kontent ma'lumotlari asosida statistik xulosalar berish.

Evolyutsion hisoblashlar algoritmlari biologik evolyutsiyadan ilhomlangan populyatsion qidiruv mexanizmlariga tayanadi [1]. Genetik algoritmlar nazariyasi J. H. Holland [3] ishlarida shakllangan: individuumlar fitnes qiymatiga ko'ra tanlanadi, keyin skreçivaniye va mutatsiya orqali yangi avlod paydo bo'ladi. Genetik dasturlash (J. R. Koza) [5] esa muammolar yechimini «dastur» ko'rinishida sintez qilishga imkon berib, inson tomonidan qo'l bilan yoziladigan qoidalar o'rniga avtomatik evolyutsiyani taklif qiladi. Kompyuter grafikasida sun'iy evolyutsiya (K. Sims) [4] kabi ishlar evolyutsion yondashuvning vizual ob'ektlar uchun ham samarali ekanini ko'rsatgan.

Veb-dizayn sohasida CSS ga asoslangan evolyutsion optimallashtirish tadqiqotlari interfeys parametrlarini «gen» sifatida tasvirlab, dizayn variantlarini avtomatik izlash g'oyasiga asoslanadi. Biroq dizayn sifatini avtomatik baholash (UX va estetika) ko'p omilli va sub'ektiv bo'lgani uchun fitnes funksiyani formallashtirish qiyin. Shu nuqtada interaktiv genetik algoritm (IGA) muhim: fitnes inson bahosi orqali belgilanadi va algoritm «qayta aloqa» (feedback) bilan qidiruvni maqbul zonaga yo'naltiradi. Tadqiqotda IGA yordamida butun maketni generatsiya qilish, alohida elementlarni qayta yaratish hamda turli ekranlarga moslashtirish ssenariylari taklif etilgan.

Manbalar bazasida GOST standartlari ulushi yuqori ekani metodikaning yana bir xususiyatini ko'rsatadi: ishlanma faqat konsepsiya yoki demonstratsion prototip



emas, balki to'liq dasturiy mahsulot hayot sikli doirasida ishlab chiqilishi nazarda tutiladi. Demak, algoritm yadrosidan tashqari texnik topshiriq, arxitektura, ishchi hujjatlar, qabul-sinov va ekspluatatsiya jarayonlari ham muhim.

Metodika quyidagi komponentlardan tashkil topadi:

1) Kodlash (genotip) va dekodlash (fenotip). CSS selektorlari va xossalari ma'lum tuzilmada saqlanadi. Genotip — parametrlar to'plami (rang, o'lcham, otstup, shrift va h.k.), fenotip — brauzerda ko'rinadigan vizual maket. Kodlashda qiymatlar chegaralari va sintaktik qoidalar belgilanadi.

2) Boshlang'ich populyatsiya. Bazaviy dizayndan boshlab, parametrlar intervallari asosida tasodifiy o'zgarishlar qo'llanib, N ta variant yaratiladi. Populyatsiya xilma-xilligi konvergensiya tezligi va optimumga yaqin yechim topish ehtimoliga ta'sir qiladi.

3) Seleksiya. Foydalanuvchi bergan baholar (1–5 yoki 1–10) asosida tanlash amalga oshiriladi. Praktikada elitizm qo'llab, eng yaxshi 1–2 variantni keyingi avlodga o'zgarishsiz o'tkazish, qolganlarini esa skremivaniye/mutatsiya orqali hosil qilish tavsiya etiladi.

4) Chatishtirish. Blokli chatishtirish usuli qo'llanadi: masalan, header, navigation, content, footer kabi bloklarning CSS parametrlari ota-ona variantlardan meros qilinadi. Bu usul «yaxshi» lokal yechimlarni saqlab qolishga yordam beradi.

5) Mutatsiya. Mutatsiya operatori dizaynni tadqiq qilish (exploration)ni ta'minlaydi: rang palitrasidan yangi rang tanlash, shrift turini almashtirish, otstuplarni o'zgartirish, border-radius'ni rostlash va h.k. Mutatsiya ehtimoli  $p_{mut}$  juda katta bo'lsa dizayn «chaqnab ketadi», juda kichik bo'lsa lokal optimumda turib qolishi mumkin; shuning uchun  $p_{mut}$  ni iteratsiyalar bo'yicha adaptiv o'zgartirish mumkin.

6) Foydalanuvchi bilan interaksiya. IGAda iteratsiya sikli: (a) N ta variantni ko'rsatish; (b) foydalanuvchi baholashi; (v) tanlangan variantlardan yangi avlod hosil qilish; (g) to'xtash mezoni (iteratsiyalar soni, yaxshilanish to'xtashi,



foydalanuvchi qoniqishi) tekshirish. Tadqiqotda foydalanuvchi belgilangan elementni qayta generatsiya qilish imkoni ham ko'rsatilgan.

7) Uchta algoritm ssenariysi. (i) Global generatsiya — butun CSS ni evolyutsiyalash; (ii) lokal generatsiya — ma'lum element yoki blok uchun parametrlarni optimallashtirish; (iii) adaptatsiya — media-holatlar (media queries) va blok o'lchamlarini qayta hisoblab, turli ekranlar uchun bir xil UXni saqlash.

8) Konseptual modellashtirish. IDEF3 diagrammalari orqali modullar o'zaro bog'lanishi, parallel ishlarning ketma-ketligi va qaror nuqtalari tasvirlanadi. Bu metodika hujjatlashtirish va dasturiy realizatsiyada izchillikni ta'minlaydi.

Axborot tizimi arxitekturasi. Tadqiqotdagi tizim konsepsiyasidan kelib chiqib, arxitekturani quyidagi modullarga ajratish mumkin: (1) «Dizayn generatori» — genetik operatorlar va populyatsiyani boshqarish; (2) «CSS validator» — sintaktik tekshiruv va qiymat cheklovlari; (3) «Render moduli» — HTML shablon + CSS orqali brauzerda ko'rsatish; (4) «Baholash interfeysi» — foydalanuvchidan ball va izohlarni qabul qilish; (5) «Natijalar ombori» — variantlarni saqlash va tarixni yuritish; (6) «Adaptivlik moduli» — ekran o'lchamlari uchun alohida parametrlar va media-qoidalarni hisoblash.

Ma'lumot oqimi: foydalanuvchi talablari → bazaviy shablon tanlash → populyatsiya generatsiyasi → render va ko'rsatish → baholash → seleksiya/operatorlar → yangi avlod → natijani saqlash. Bu sxema «inson halqasi» (human-in-the-loop)ni algoritmnining markaziga qo'yadi, ya'ni dizayn sifatini inson tajribasi va qarorlari bilan boshqaradi.

IGAning asosiy afzalligi — estetik mezonlarni formallashtirmasdan turib, inson bahosi orqali qidiruvni maqbul zonaga yo'naltira olishidir. Bu adaptiv va personal dizayn talab qilinganda juda qimmat. Biroq sub'ektiv baholash tufayli baholarning barqarorligi va reproduksiyalanuvchanligi muammosi paydo bo'ladi: bir foydalanuvchi bir xil variantni turlicha baholashi yoki turli foydalanuvchilarning mezonlari keskin farq qilishi mumkin. Shuning uchun amaliyotda (a) baholash shkalasini standartlashtirish, (b) bir necha foydalanuvchi bahosini agregatlash, (v)



«ta'riflangan cheklovlar» (constraints) va avtomatik filtrlar yordamida nomaqbul variantlarni oldindan chiqarib tashlash tavsiya etiladi.

CSS kodlashda texnik xavflar ham mavjud: noto'g'ri kombinatsiyalar (masalan, past kontrast, juda kichik shrift, qo'llashga noqulay otstuplar) UXni yomonlashtiradi. Shu sabab genlar uchun «xavfsiz» intervallar, kontrast va o'qilishlilik qoidalari, hamda UI komponentlar uchun minimal standartlar kiritish maqsadga muvofiq. Tajribalar shuni ko'rsatdiki standartizatsiya va ishlab chiqish bosqichlariga katta e'tibor berilgani bunday cheklovlarni hujjatlashtirish va testlash orqali ham nazorat qilish mumkinligini ko'rsatadi.

Loyiha muddatlari tahlilidan ko'rinadiki, «algoritm yadrosi»ni ishlab chiqishdan tashqari arxitektura, ma'lumotlar modeli, interfeys prototiplari, ishchi hujjatlar va qabul ishlari katta ulushni egallaydi. Bu esa IGA asosidagi dizayn generatsiya tizimini joriy etishda to'liq injenerlik yondashuvi, jumladan talablar injeneriyasi va standartlarga muvofiq hujjatlashtirish muhim ekanini tasdiqlaydi.

Xulosa. Interaktiv genetik algoritmlar asosida CSS qoidalarini xromosoma sifatida modellashtirish veb-dizayn variantlarini avtomatik generatsiya qilish uchun amaliy metodika sifatida asoslandi. Taklif etilgan yondashuv butun maketni generatsiya qilish, alohida elementlarni lokal optimallashtirish va adaptiv moslashtirish ssenariylarini qamrab oladi. Kelgusi ishlarda avtomatik UX-filtrlar, ko'p foydalanuvchili baholash, hamda dizayn elementlari uchun ontologiya/qoidalar to'plamiga asoslangan cheklovlar bilan IGAni boyitish maqsadga muvofiq.

## FOYDALANILGAN ADABIYOTLAR

1. Сознание и язык. Проблема искусственного интеллекта. [Электронный ресурс]. URL: [https://studopedia.ru/10\\_192176\\_struktura-soznaniya-soznanie-i-samosoznanie.html](https://studopedia.ru/10_192176_struktura-soznaniya-soznanie-i-samosoznanie.html) (дата обращения: 25.05.2020).

2. Эволюционное программирование. [Электронный ресурс]. URL: [http://wi-ki.ru/wiki/Эволюционное\\_программирование](http://wi-ki.ru/wiki/Эволюционное_программирование) (дата обращения: 25.05.2020).



3. Holland, J. H. Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions / J. H. Holland // *Evolutionary Computations*. – 2000 – Vol. 8 (4). – P. 373–391.

4. Sims, K. Artificial Evolution for Computer Graphics / K. Sims, // *SIGGRAPH '91 Proceedings*. – 1991. – S. 319 – 328.

5. Koza, J.R. (1990), *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*, Stanford University Computer Science Department technical report STAN-CS-90-1314.

6. Панченко, Т. В. Генетические алгоритмы: учебно-методическое пособие / Т. В. Панченко, под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 6 с.

7. Кольчугина, Е. А. Биоинспирированные методы саморазработки программ для распределенных систем / Е. А. Кольчугина // *Известия высших учебных заведений. Поволжский регион. Технические науки*. – 2018 – № 4 (48). – С. 96–104. – DOI 10.21685/2072-3059-2018-4-9.

8. Саяпин, О. В., Тиханычев, О. В., Чискидов, С. В., Быстракова И. А. Разработка интерфейсов прикладных программ: макетирование или прототипирование / О. В. Саяпин // *Прикладная информатика*. — №1(85), 2020. — Москва, ООО «Синергия Пресс», 2020. — С. 49–56.

9. Тиханычев О.В. Пользовательские интерфейсы в автоматизированных системах: проблемы разработки // *Программные системы и вычислительные методы*. – 2019. – № 2. – С. 11 - 22. [Электронный ресурс] URL: [https://nbpublish.com/library\\_read\\_article.php?id=28443](https://nbpublish.com/library_read_article.php?id=28443) (дата обращения: 25.05.2020).

10. Sunghyun, R. K. Webpage design optimization using genetic algorithm driven CSS / R. K. Sunghyun, R. Baer, V. Honavar // *Iowa State University*. – 2007. – 104 с. University. – 2007. – 104 s.