

# PSO GA Hybrid

Nikola Nesic

February 16, 2024

## 1 Opis problema

Ilustrovaćemo rešavanje optimizacionih problema sa nelinearnim ograničenjima na neprekidnom prostoru pretrage. Himmelblau-ov problem i pressure vessel problem.

### 1.1 Himmelblau-ov problem

Minimizovati funkciju:

$$tominimize(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Uz ograničenja:

$$78 < x_1 < 102, 33 < x_2 < 45, 27 < x_3, x_4, x_5 < 45$$

$$0 \leq g1(x) \leq 92$$

$$90 \leq g2(x) \leq 110$$

$$20 \leq g3(x) \leq 25$$

$$g1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5$$

$$g2(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_3^2$$

$$g3(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4$$

### 1.2 Pressure vessel

Rezervoar za skladištenje komprimovanog vazduha sa radnim pritiskom od 2000 psi i maksimalnom zapreminom od  $750ft^3$ .

Cilindrična posuda je pokrivena na oba kraja poluloptastim glavama.

Koristeći valjanu čeličnu ploču, školjka je napravljena u dve polovine koje su spojeni sa dva uzdužna zavara u cilindar.

Cilj je minimizirati ukupne troškove, uključujući troškove materijala, formiranje i zavarivanje.

Postoje četiri varijable povezane sa problemom, naime kao debljina posude pod pritiskom,  $T_s = k_1$ , debljina glave,  $T_h = k_2$ , unutrašnji poluprečnik posude,  $R = k_3$ , i dužina posude bez glave,  $L = k_4$ , tj. vektori su dati (u inčima) sa  $K_s = (T_s, T_h, R, L) = (k_1, k_2, k_3, k_4)$ .

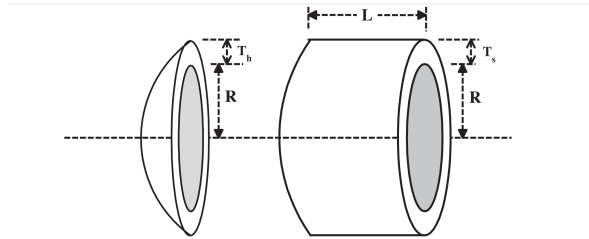


Figure 1: pressure vessel

Matematička sumarijacija, treba minimizovati:

$$tominimize(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Uz ograničenja:

$$1 * 0.625 \leq x_1, x_2 \leq 99 * 0.625, 10 \leq x_3, x_4 \leq 200$$

$$g1(x) = -x_1 + 0.0193x_3 \leq 0$$

$$\begin{aligned}
g2(x) &= -x_2 + 0.0095x_3 \leq 0 \\
g3(x) &= -\pi x_3^2 x_4 - 4/3\pi x_3^3 + 1296000 \leq 0 \\
g4(x) &= x_4 - 240 \leq 0
\end{aligned}$$

## 2 Moj pristup

Pozicija u prostoru rešenja je genetski kod jedinke.

Fitness funkcija je negativ funkcije koju minimalizujemo, s tim da cemo clip-ovati resenje za nedopustiva rešenja tipa  $\min \leq x_i \leq \max$ .

Za ostala ograničenja ćemo uvesti penalizaciju unutar fitness funkcije. Naime ako imamo ograničenje  $g(x) \leq \max$ , ako je  $g(x)$  veće od  $\max$  na izračunati fitness oduzimamo  $c^*(g(x)-\max)$  gde je  $c$  koeficijent kažnjavanja, analogno za ograničenja  $\min \leq g(x)$ .

Ovako se krecemo neprekidno kroz nedopustivi prostor.

Dodatno jedinka ima velocity (brzinu za PSO), age (broj iteracija koliko jedinka postoji) i life expectancy (broj iteracija koliko cemo dopustiti jedinki da vrši PSO dok je ne iskoristimo za iteraciju genetskog algoritma).

Dok se ne ispuni uslov zaustavljanja (while not stop).

Za svaku jedinku u populaciji (foreach ind).

Jedinka update-uje velocity, position i fitness, i inkrementira age.

Ako je napunila age do life expectancy iteracija, za nju se iz populacije bira jedinka sa kojom ce vršiti crossover i mutaciju.

Novonastala jedinka menja staru u populaciji, i age joj se postavlja na 0.

Što je veci life expectancy to se češće vrši PSO u odnosu na genetski algoritam.

## 3 Rezultati

### 3.1 Metodika

Uporediti Genetski algoritam, naš pristup i pristup iz [jedne studije](#) koji je takodje hibrid (doduše koristi se naša implementacija). Ne razmatramo PSO jer on sam po sebi retko nalazi dopustivo rešenje. Ni genetski algoritam ne nalazi uvek dopustivo rešenje. Num hits broj puta od pokušanih kada nam algoritam vraća dopustivo rešenje. Koristimo procesor 12th Gen Intel® Core™ i7-12700H × 20.

### 3.2 Parametrizacija

GA:

pop size=50

elitism size=6

mut prob=0.01 My hybrid:

Swarm size=50

$c_i=0.7$

$c_p = 1$

$c_g = 1$

life expectancy =1 0

Study hybrid:

Swarm size: 50

$c_i : 0.7$

$c_p : 1.0$

$c_g : 1.0$

ga min num : 10

ga max num : 50

ga min ps : 50

ga max ps : 100

ga min iter : 20

ga max iter : 100

y : 2  
b :3  
elitism size=6

### 3.3 Himmeblau

Method	NumHits	AverageMinimum	time
GA	730/1000	-28181.01238772747	0.01
MyHybrid	1000/1000	-29454.462609046153	0.01
studyHybrid	NA	NA	0.01

Method	NumHits	AverageMinimum	time
GA	717/1000	-28181.475586473614	0.1
MyHybrid	1000/1000	-29891.976516474388	0.1
studyHybrid	868/1000	-30003.642233559054	0.1

Method	NumHits	AverageMinimum	time
GA	736/1000	-28240.24716063069	1
MyHybrid	1000/1000	-30184.57028861193	1
studyHybrid	887/1000	-30229.959589093134	1

Malo vece vreme:

Method	NumHits	AverageMinimum	time
MyHybrid	50/50	-30 427.68485	50
studyHybrid	43/50	-30 453.01700	50

### 3.4 pressure vessel

Method	NumHits	AverageMinimum	time
GA	799/1000	48278.058944957746	0.01
MyHybrid	262/1000	15097.817554426572	0.01
studyHybrid	Na	NA	0.01

Method	NumHits	AverageMinimum	time
GA	805/1000	47913.48869576948	0.1
MyHybrid	182/1000	10048.394462206283	0.1
studyHybrid	824/1000	10090.475710738341	0.1

Method	NumHits	AverageMinimum	time
GA	790/1000	46872.178453021836	1
MyHybrid	197/1000	7832.455968444094	1
studyHybrid	823/1000	7911.2155552107215	1

Hajde sada da pogledamo rezultate ako smanjimo life expectancy sa 5 na 10:

Method	NumHits	AverageMinimum	time
MyHybrid	749/1000	17577.022950180828	0.01
MyHybrid	791/1000	11480.105373696335	0.1
MyHybrid	780/1000	8499.606242923526	1

Možemo da primetimo da su rešenja koja dobijamo lošija ali da učestalije uspevamo da izadjemo iz nedopustivog prostora pretrage. Naime genetski algoritam se izvršava duplo česce posto smo smanjili life expectancy duplo. Ovo nam govori da smo smanjili intenzifikaciju a povećali diverzifikaciju.

Malo vece vreme:

Method	NumHits	AverageMinimum	time
MyHybrid	38/50	6787.72352	50
studyHybrid	43/50	6560.20599	50

## 4 Zaključak

PSO većinu svoje diverzifikacije dobija samom inicijalizacijom, malo dobija iz pbest-a, ali ima jako dobru intenzifikaciju kroz iteracije.

Genetski algoritam ima jako dobru diverzifikaciju iz crossover-a i mutacije.

A intenzifikacija potiče iz selekcije i elitism-a pod heuristikom da će dobri roditelji ostaviti dobre potomke, što često nije dovoljno.

Naime mi želimo za naš konkretan problem da nadujemo optimalan odnos između diverzifikacije i intenzifikacije.

U našem slučaju odnos između iteracija genetskog algoritma i iteracija PSO .

Ovo zavisi od odnosa dopustivog prostora pretrage i prostora koji zapravo pretražujemo.

Ovo nije baš mera koju možemo da poredimo od problema do problema.

Različiti parametri različito rade za različite probleme.

Empirijski ako je num hits mali, treba povećati diverzifikaciju u odnosu na intenzifikaciju. Ako je num hits maksimalan, možemo povećavati intenzifikaciju dok num hits ne krene da opada i da ga držimo na graničnoj vrednosti tako da su parametri optimalni za dati algoritam.

Ovaj algoritam pretpostavlja da će biti više PSO iteracija nego GA iteracija. Neki uopšteni algoritam bi umesto life expectancy parametra mogao da ima pso to ga ratio i preko njega da regulise odnos između količine GA i PSO iteracija.

Ovde koristimo nasu implementaciju algoritma iz studije sa drugim parametrima kako bi imali vreme izvršavanja kao kriterijum poredjenja, u samoj studiji dobijaju boja rešenja, ali je vreme izvršavanja određeno brojem iteracija.

Algoritam iz studije ima veći akcenat na genetskom algoritmu i nasumičnoj pretrazi, dok naš algoritam ima veći akcenat na PSO. To što njihov algoritam malo bolje radi nam naslućuje da problem zahteva veću diverzifikaciju u odnosu na intenzifikaciju.

## References

[A hybrid PSO-GA for constraint optimization problems, Harish Garg](#)

[Vezbe iz predmeta Racunarska inteligencija na Matematičkom fakultetu u Beogradu](#)