

Оптимизациони алгоритми са инжењерским проблемима

- Циљ: сагледати особине оптимизационих алгоритама када су примењени на инжењерске оптимизационе проблеме
- Примери
 - Јаги антена
 - Трансверзални антенски низ
 - Одређивање поља око процесора за различите инструкције које се извршавају
- Кључна тачка: оптимизациона функција
- NLP и SAT проблеми

Пример #1: Јаги антена

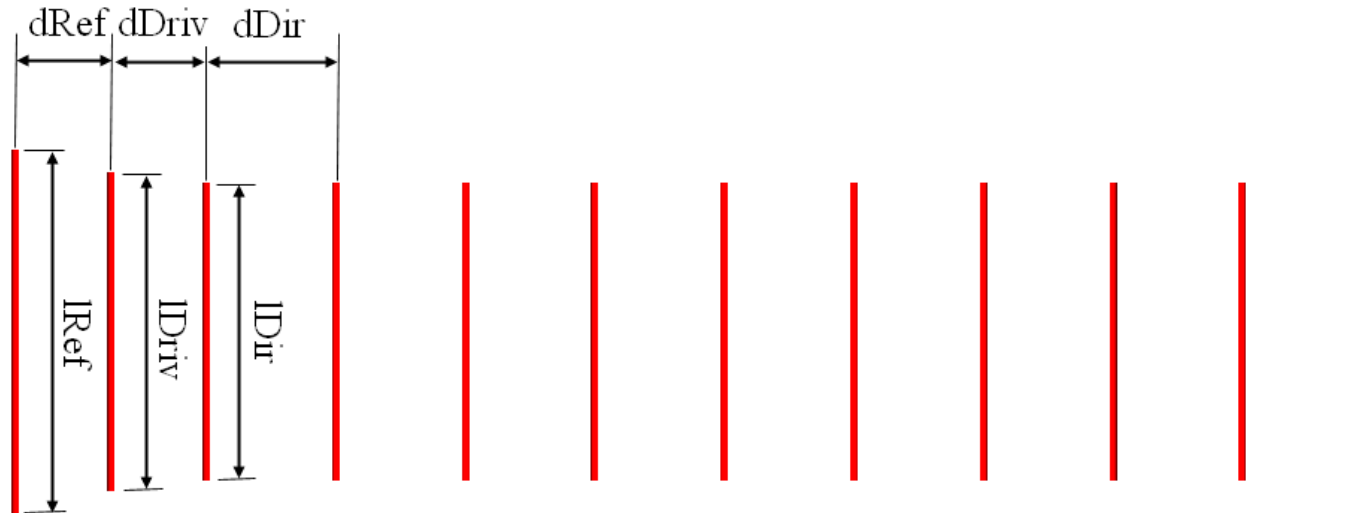
- Јаги антена се састоји из неколико линеарних дипола
- Само један дипол се директно напаја
- Остали диполи раде као паразитни радијатори чије струје су индуковане због међусобне спреге
- Антене овога типа се користе у опсегу 3 MHz - 3000 MHz

Модел јаги антене коришћен за оптимизацију

- Шест оптимизационих параметара:
 - l_{ref} – дужина рефлектора,
 - l_{drv} – дужина напајаног елемента,
 - l_{dir} – дужина сваког од директора (сви директори су једнаке дужине),
 - d_{ref} – растојање између рефлектора и напајаног елемента,
 - d_{drv} – растојање између рефлектора и првог директора,
 - d_{dir} – растојање између суседних директора

Оптимизациони простор и модел

- Оптимизациони простор:
параметри су мењани у
границама од 0,2 m до 0,8 m



Описна функција

- Описна функција
$$f = \sqrt{\frac{1}{n} \sum_{k=1}^n (gain - G_0)^2}$$
- n је број фреквенција у којима је рачунат добитак ($gain$) антене у децибелима
- У свим приказаним резултатима $G_0 = 20$ dB
- Добитак је рачунат на $n = 5$ еквидистантних учестаности
- Опсег учестаности 295-305 MHz
- За $G_0 = 0$ f је средњи добитак антене у посматраном опсегу учестаности

Поставка за поређење алгоритама

- За оптимизацију је коришћено шест различитих алгоритама
- Сваки метод је изнова покретан до укупно 10 000 итерација
- Додатним нумеричким експериментима је показано да је усвојени максималан број итерација довољан за поуздану статистику

Алгоритми који су упоређени

- Случајно претраживање
- Понављање градијентног алгоритма са случајно изабраним почетним решењем
- Понављање симплекс алгоритма са случајно изабраним почетним решењем
- Најбоље решење из 300 случајно изабраних као почетно решење за симплекс
- Хибридни ГА+симплекс
- Процена локалних минимума

Случајно претраживање

- Униформно шестодимензионално случајно претраживање
- Метод пружа увид у расподелу добитака јаги антене у оптимизационом простору
- Средње растојање међу решењима би требало да буде једнако теоријској вредности за униформну расподелу тачака (теоријска вредност 0,6 m)

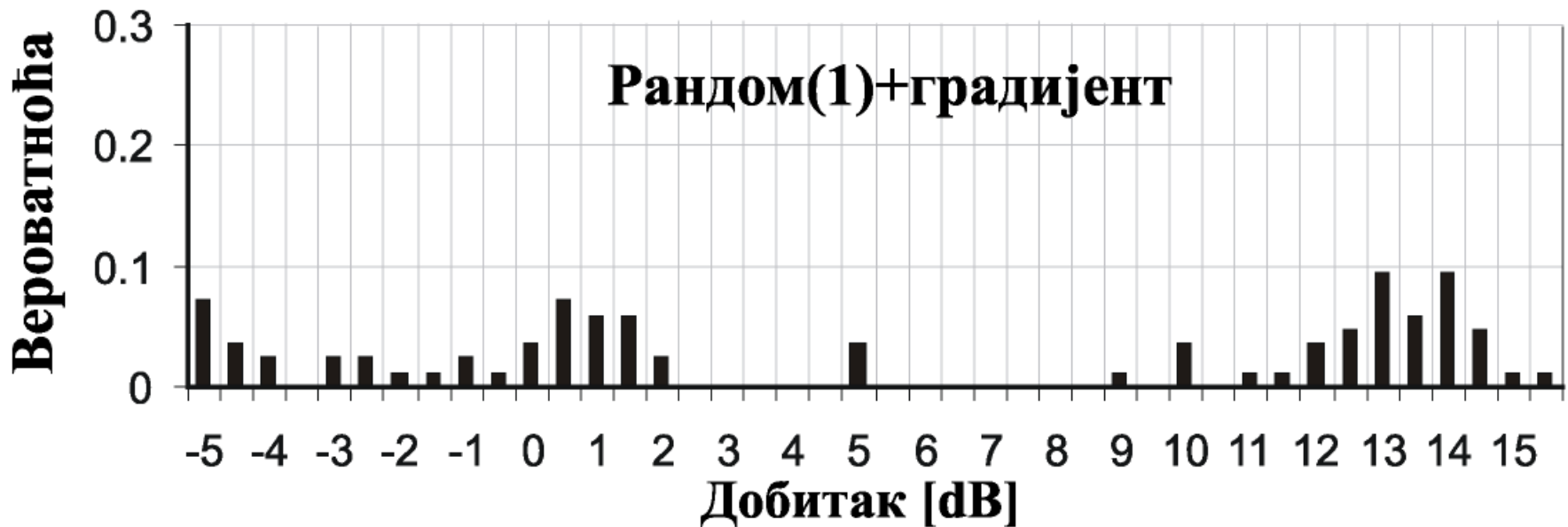
Случајно претраживање: резултати

- 25% решења: средњи добитак мањи од -5 dB
- Ниједно решење са добитком већим од 14 dB



Рандом(1) + градијент

- Релативни корак за процену градијента 0,05%
координате оптимизационог простора



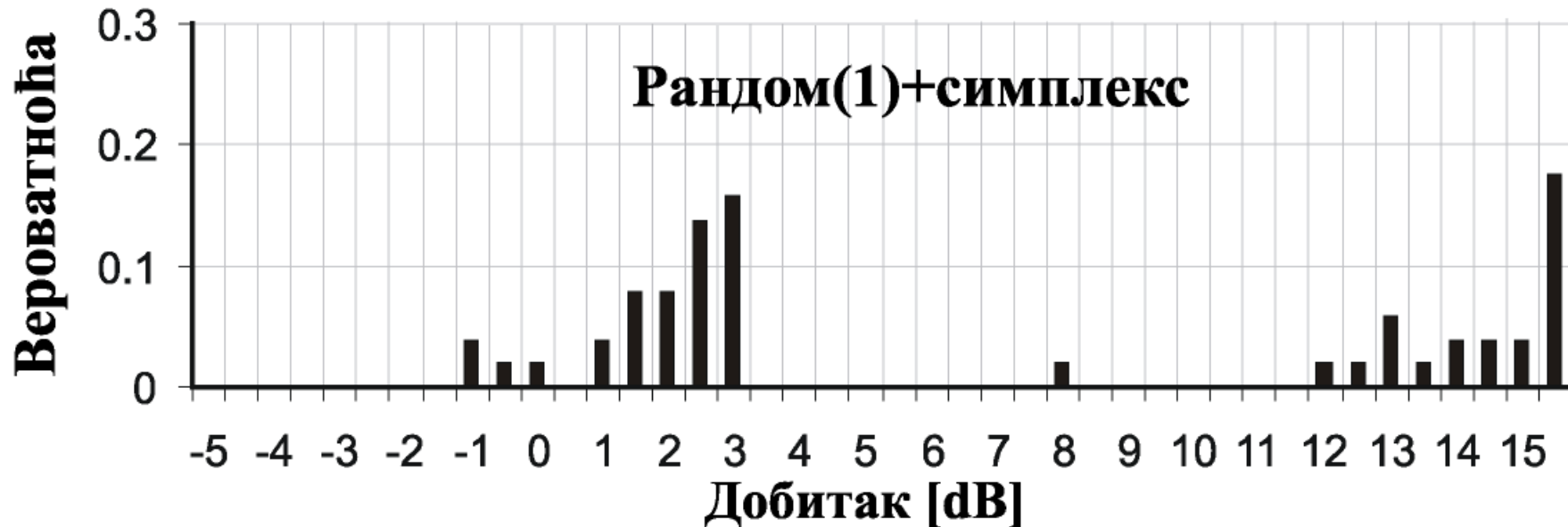
Рандом(1) + градијент

- Хистограм показује да постоји много минимума различитих дубина
- Градијентна метода конвергира ка једном од најближих локалних минимума у околини полазног решења
- Пронађени су минимуми у свим деловима оптимизационог простора у којем они постоје, без обзира на дубину минимума
- Решења добијена овом методом дају увид у расподелу положаја локалних минимума и њихових дубина у оптимизационом простору

Параметри симплекс алгоритма

- Коефицијенти износе:
рефлексија $\alpha = 1$, експанзија $\gamma = 2$,
контракција $\beta = 0,5$ и сажимање $\sigma = 0,5$
- Критеријуми за завршетак алгоритма су:
 - (а) релативна разлика, по свакој од координата, свих тачака симплекса мања од 0,05%
 - (б) апсолутна разлика функције грешке у свим тачкама симплекса мања од 10^{-3}
 - (в) укупно 300 итерација

Рандом(1) + симплекс

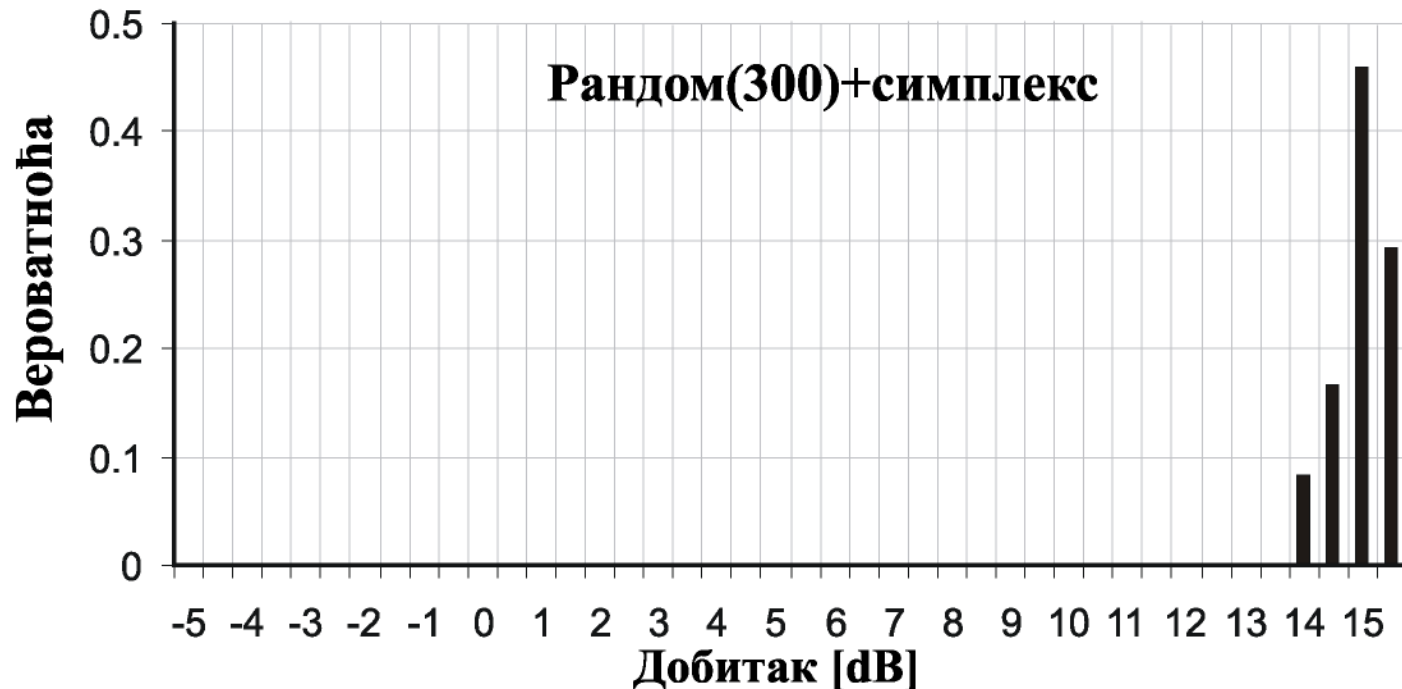


Рандом(1) + симплекс

- Понављањем симплекса проналазе се боља решења него понављањем градијентног алгоритма
- Овакви резултати су последица особине симплекса да успешно пронађе најбољи минимум у околини полазног решења

Рандом(300) + симплекс

- Прескачу се делови простора у којима постоје локални минимуми мањих дубина

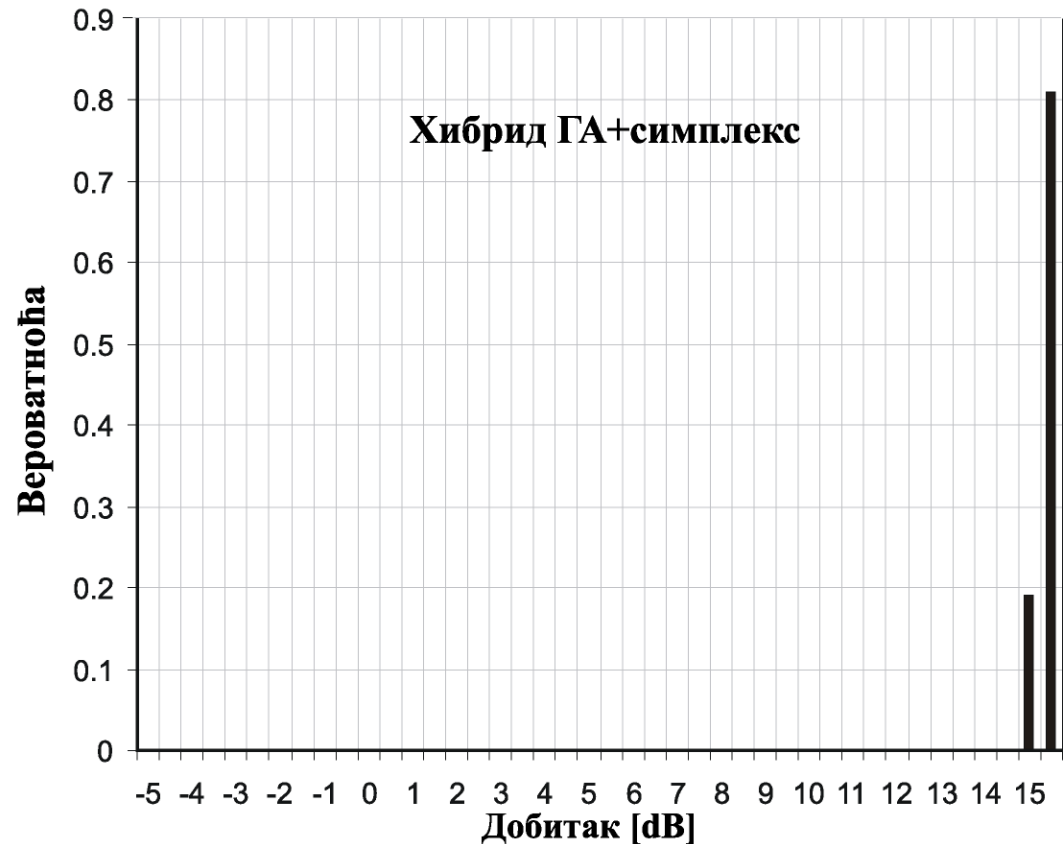


Хибридни ГА+симплекс

- Боље од обичног ГА у смислу много мањег броја потребних итерација
- Параметри ГА су: шеснаестобитно кодовање оптимизационих параметара, величина популације 60, у свакој генерацији 20 најбољих решења учествује у стварању наредне генерације, укупан број генерација 5, вероватноћа укрштања је 0,8 и вероватноћа мутације 0,1
- ГА има 300 итерација и најбоље решење после тога је полазна тачка за симплекс

ГА + симплекс

- Вероватноћа проналажења најбољег решења 80% (глобални минимум)



Процена локалних минимума

- За сваку тачку у скупу од 100 случајно изабраних, суседних 10 тачака је кориштено за проверу да ли је посматрана тачка локални минимум
- Шест независних покретања је било потребно да би се добио укупан потребан број итерација
- У оквиру сваког покретања процењено је у просеку 9 минимума

Процена локалних минимума



Рачунање растојања између решења

- У циљу стицања увида у распоред решења у оптимизационом простору, средње растојање између решења је рачунато као

$$R = \sqrt{\frac{2 \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=1}^6 (x_k^i - x_k^j)^2}{N(N-1)}}$$

- N је укупан број пронађених решења
- x_k^i је k -ти параметар i -тог решења

Средње растојање између пронађених решења

Алгоритам	Средње растојање [m]
Рандом	0,5676
Рандом(1)+градијент	0,5694
Рандом(1)+симплекс	0,5654
Рандом(300)+симплекс	0,4022
Хибрид ГА+симплекс	0,3189
Процена локалних минимума ПЛМ	0,5412

Број итерација за антену са средњим добитком већим од 15 dB

Алгоритам	Просечан број итерација
Случајно претраживање	>10.000
Рандом(1)+градијент	~10.000
Рандом(1)+симплекс	2.400
Рандом(300)+симплекс	1.200
Хибридни ГА+симплекс	690
Процена локалних минимума – ПЛМ	650

Пример #2: Потискивање лобова трансверзалног антенског низа

- Трансверзални антенски низ сачињен од 42 електрички кратка дипола (тачкасти извори зрачења)
- Растојање између суседних дипола је једна половина таласне дужине
- Амплитуде простопериодичних побуда дипола су оптимизоване тако да се добију минимални бочни лобови у дијаграму зрачења

Поставка оптимизације

- Симетрија коефицијената: само једна половина коефицијената низа је оптимизована
- Да би се избегао бесконачан број решења, који је последица могућег скалирања коефицијената, једном коефицијенту је дата апсолутна вредност
- На овај начин је добијен двадесетодимензиони оптимизациони проблем

Дијаграм зрачења низа

- Карактеристична функција зрачења

$$F(\theta) = 2 \cdot \sum_{k=1}^N a_k \cos\left(\left(k - N + \frac{1}{2}\right) \cdot \beta d \cos(\theta)\right)$$

- θ је угао између осе низа и правца у коме се рачуна функција
- $2N$ је укупан број елемената низа
- a_k је амплитуда побуде k -тог елемента
- β је коефицијент зрачења и
- d је растојање између суседних елемената

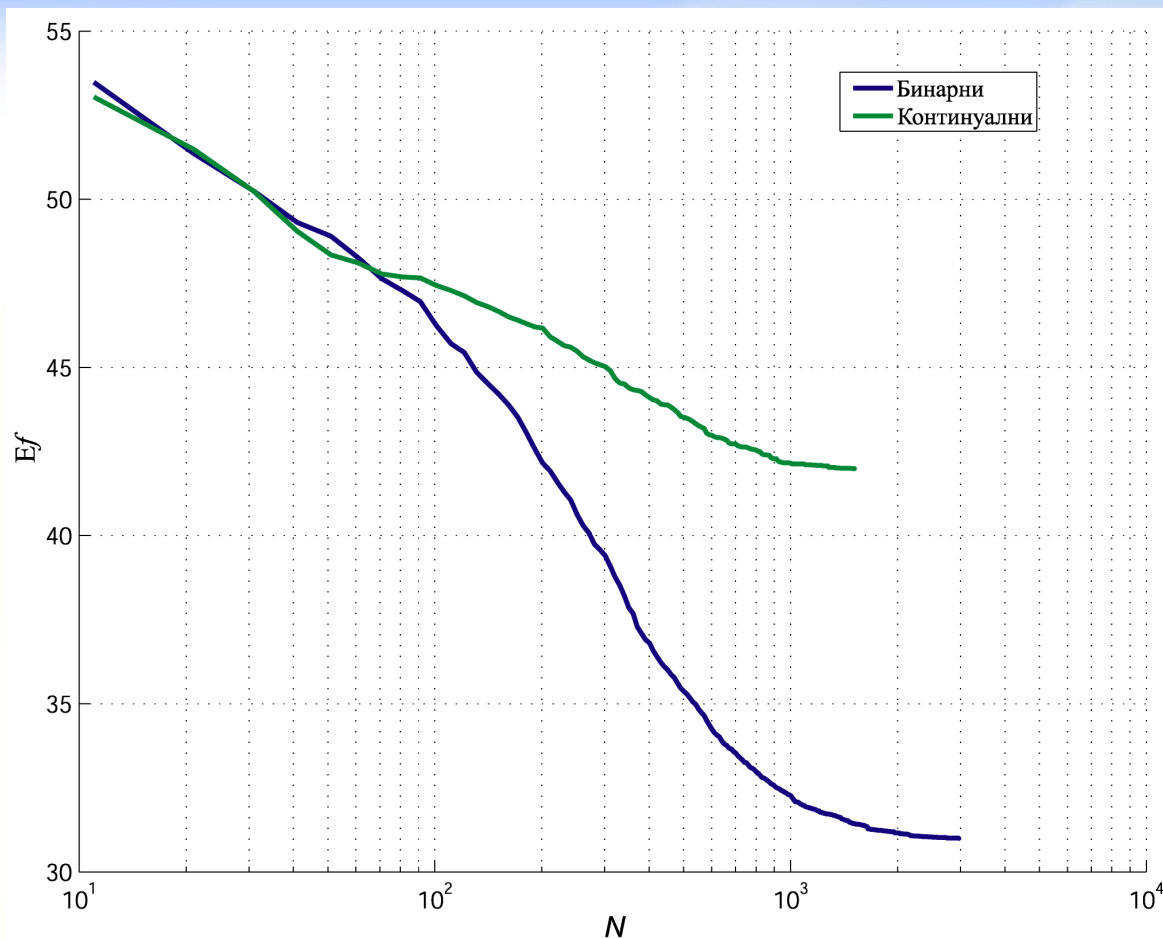
Описна функција

- Критеријум за оптимизацију је да нивои бочних листова буду нижи од -80 dB за $\theta > 25^\circ$ (мерено од осе низа)
- Оптимизациона функција

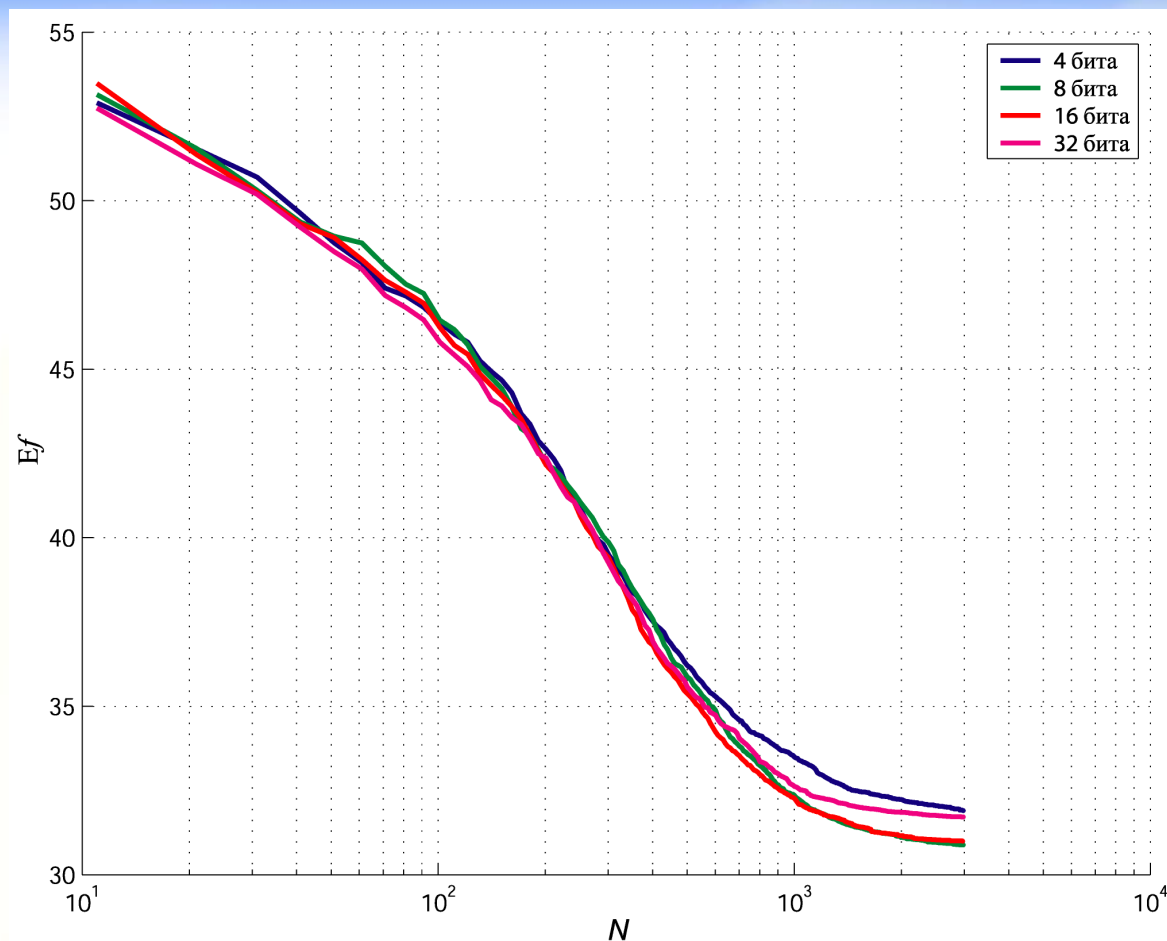
$$Cost Func. = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} \{\max[0, 80 - F_{\max} - F(\theta_i)]\}^2}$$

- 100 статистички независних покретања алгоритма

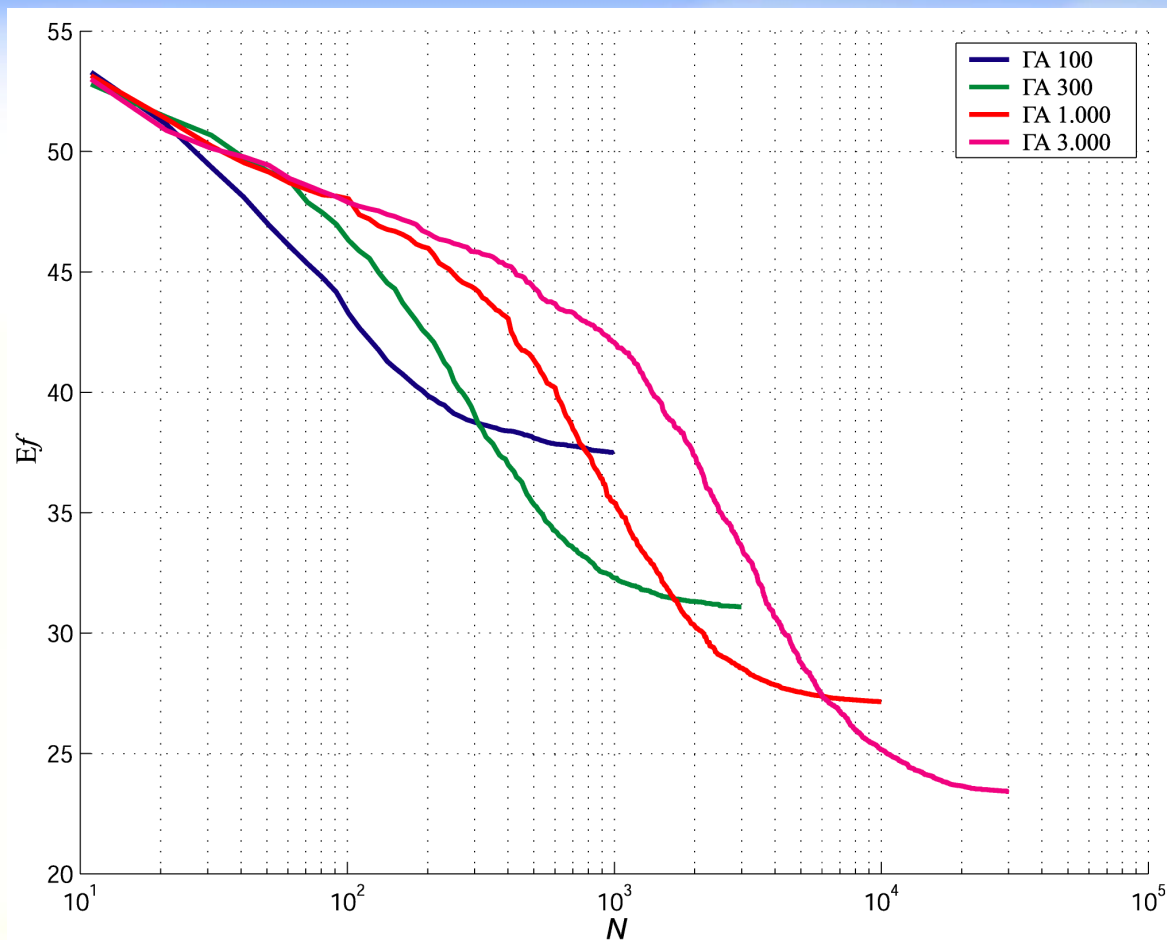
ГА: бинарно / континуално представљање



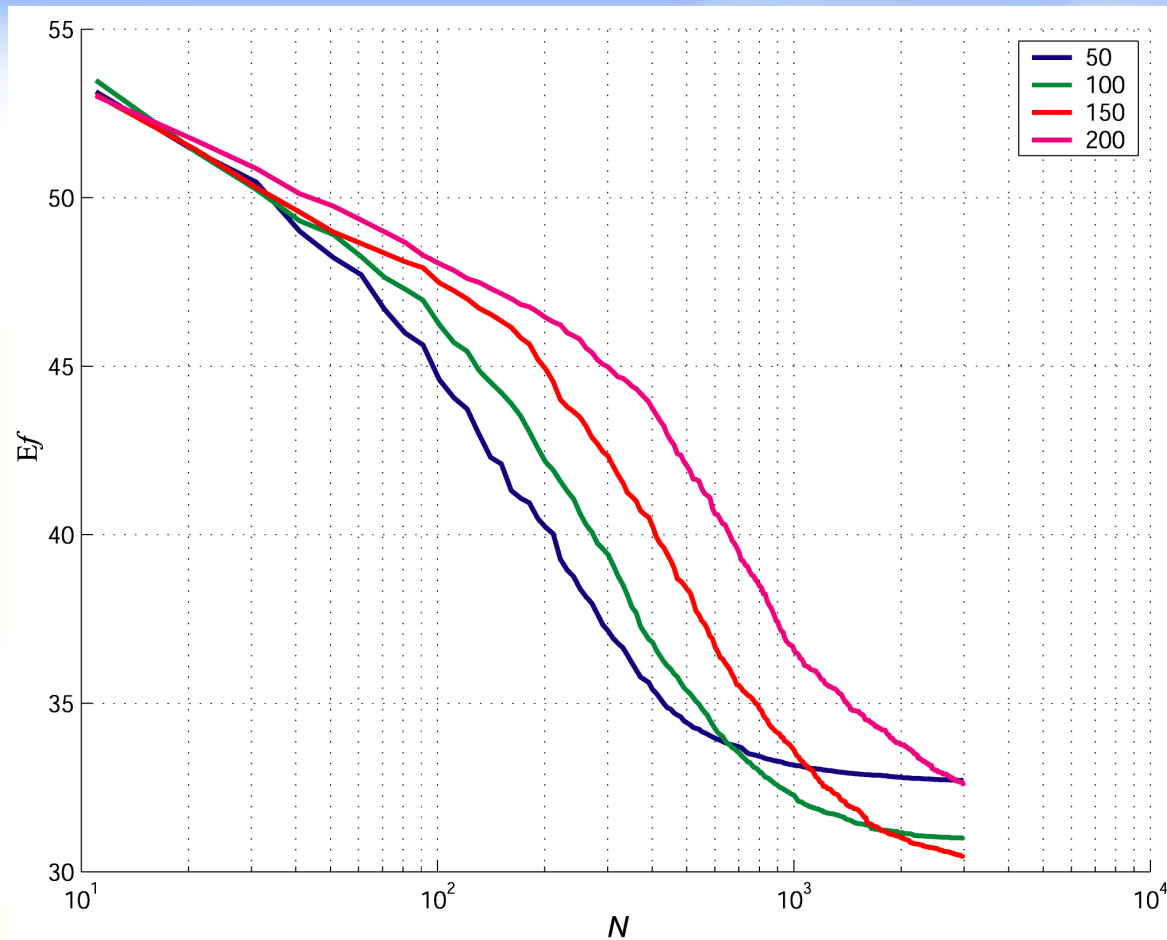
Утицај броја бита за представљање променљивих



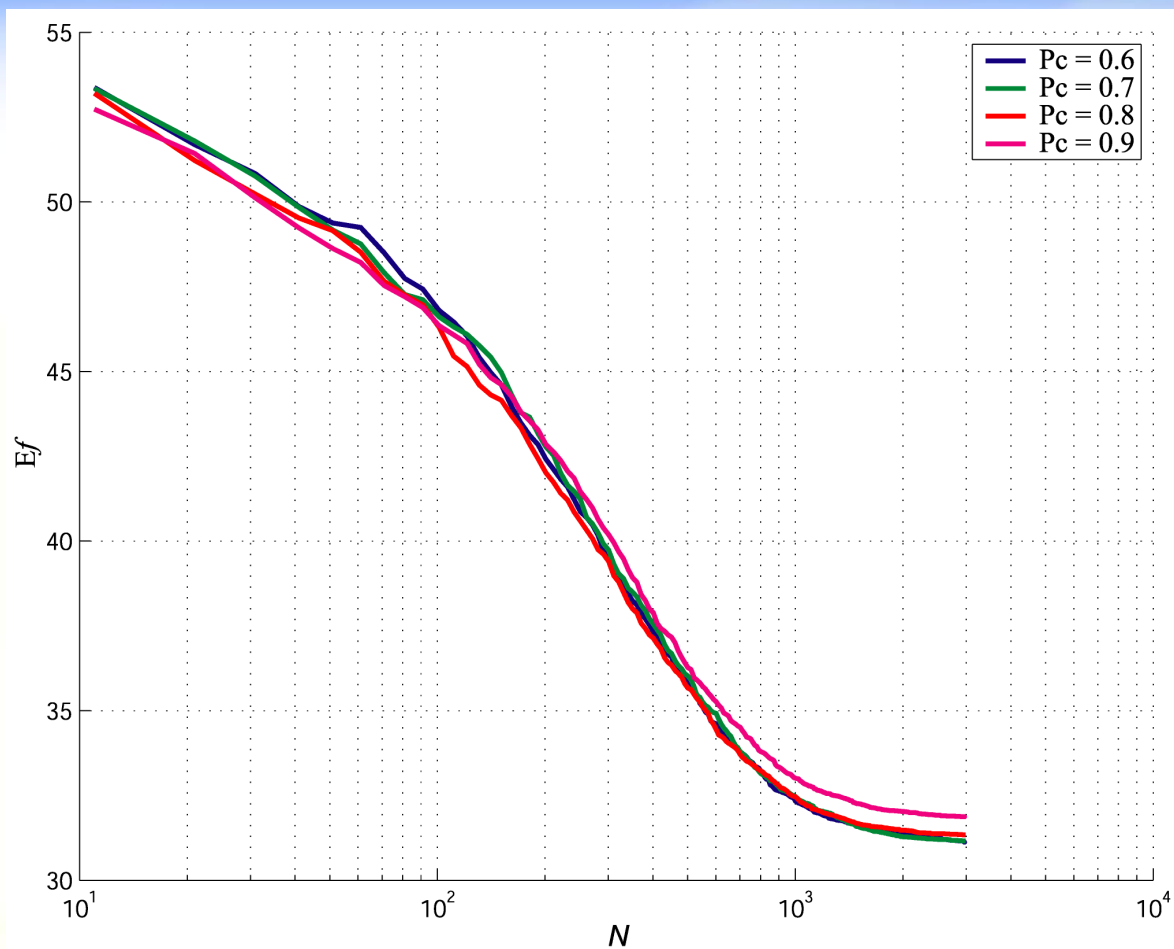
Утицај величине популације



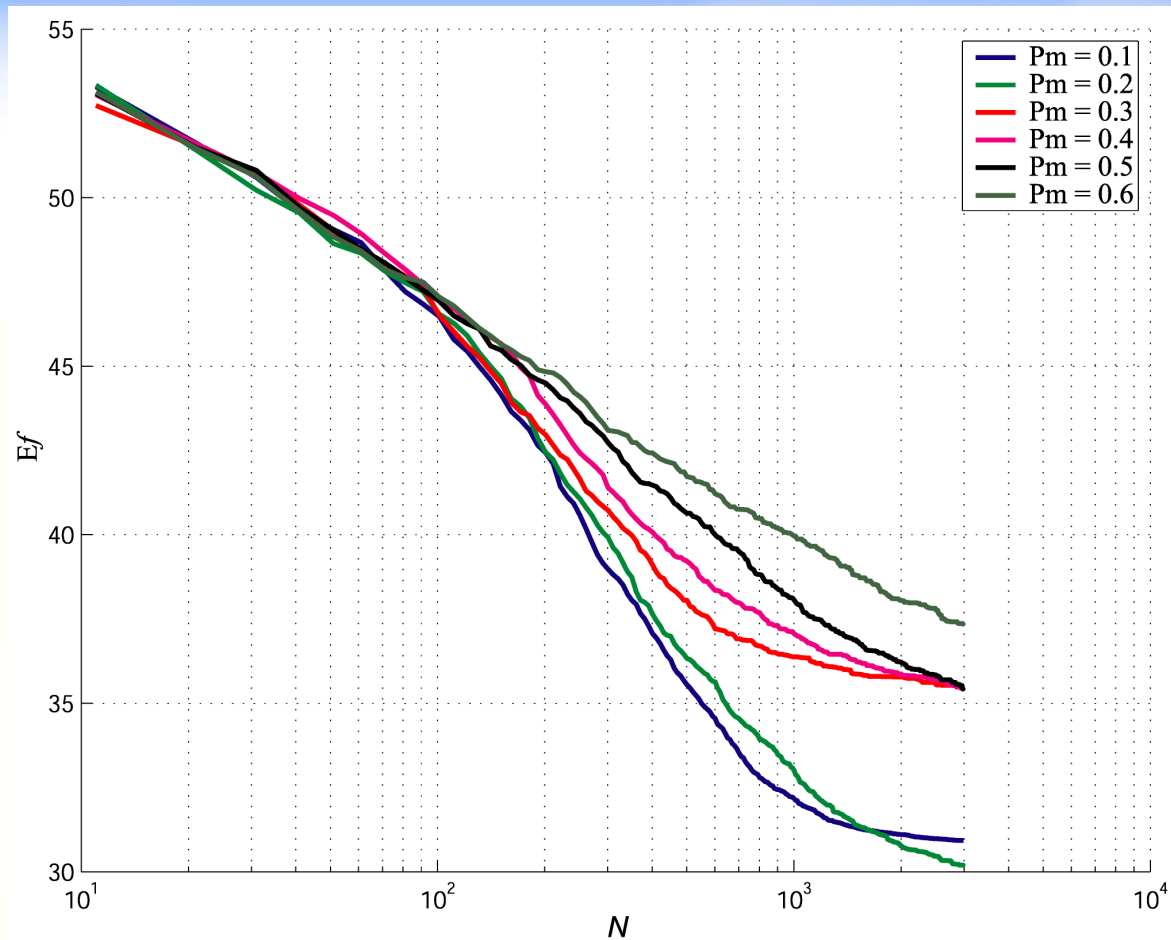
Број решења за формирање наредне генерације (30%)



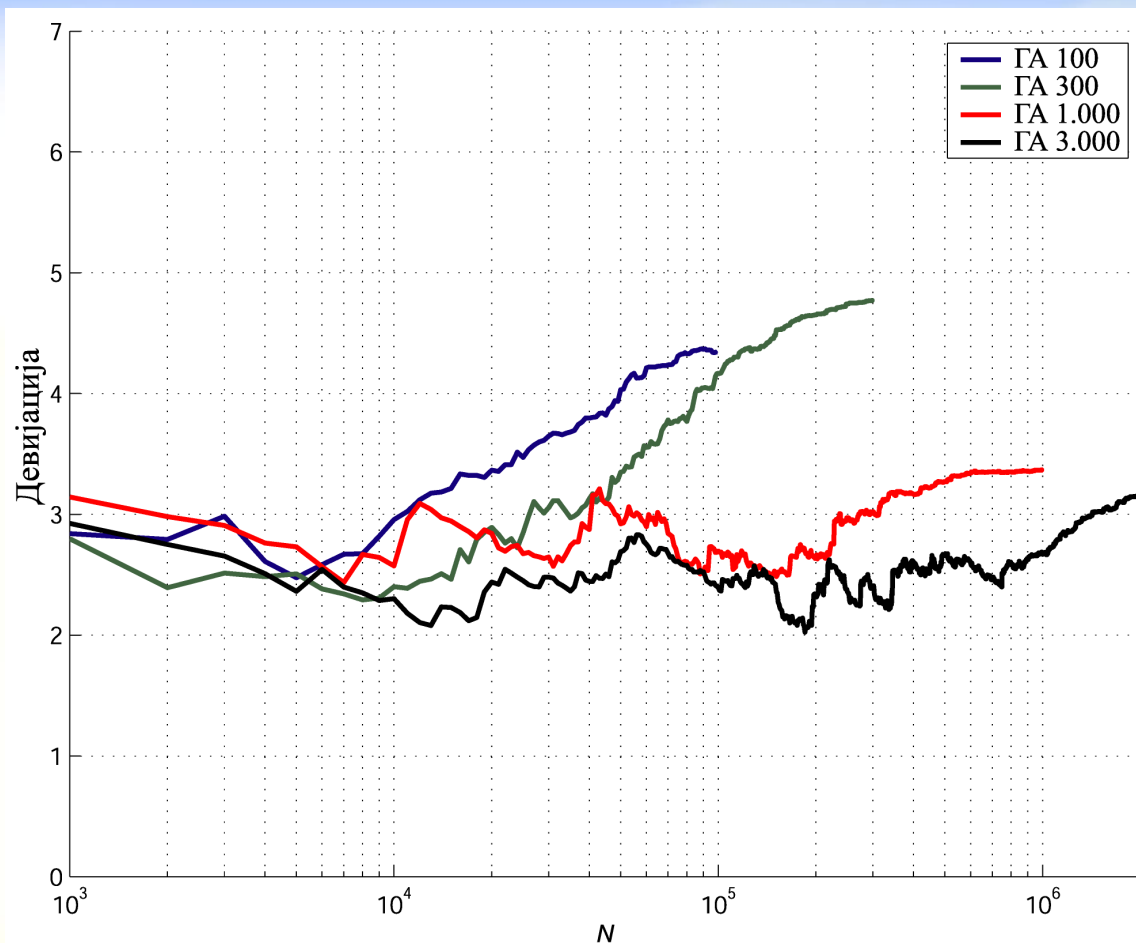
Утицај вероватноће укрштања



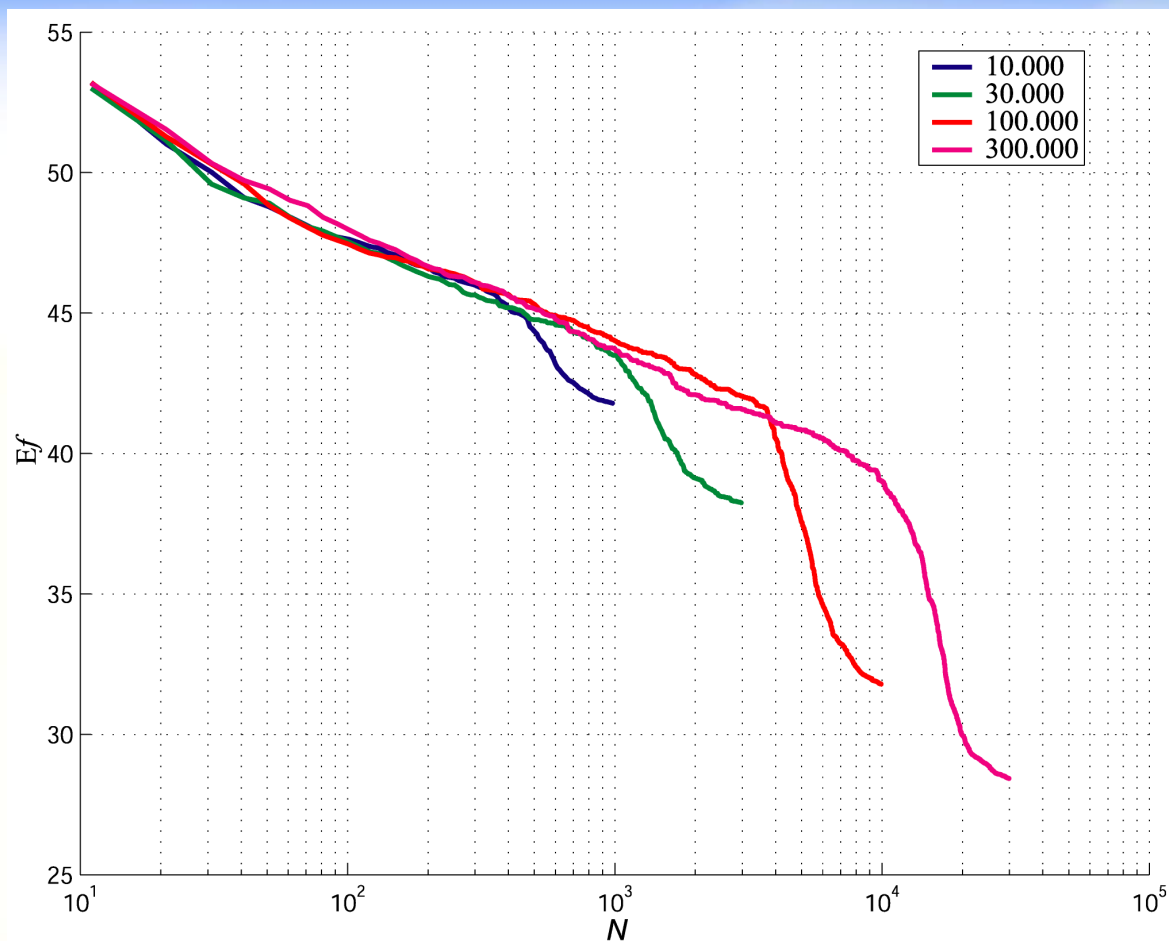
Утицај вероватноће мутације



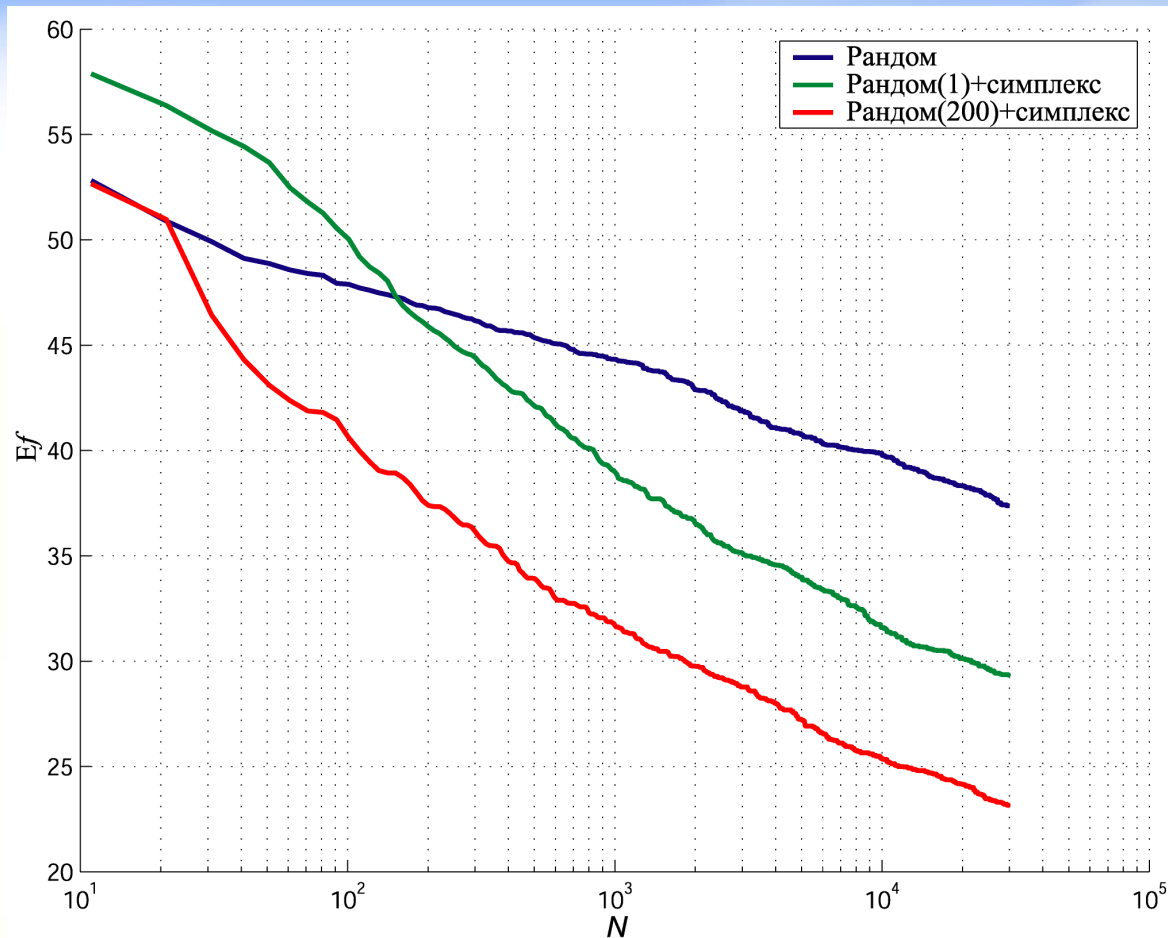
Девијација средњег најбољег решења (популације 100-3000)



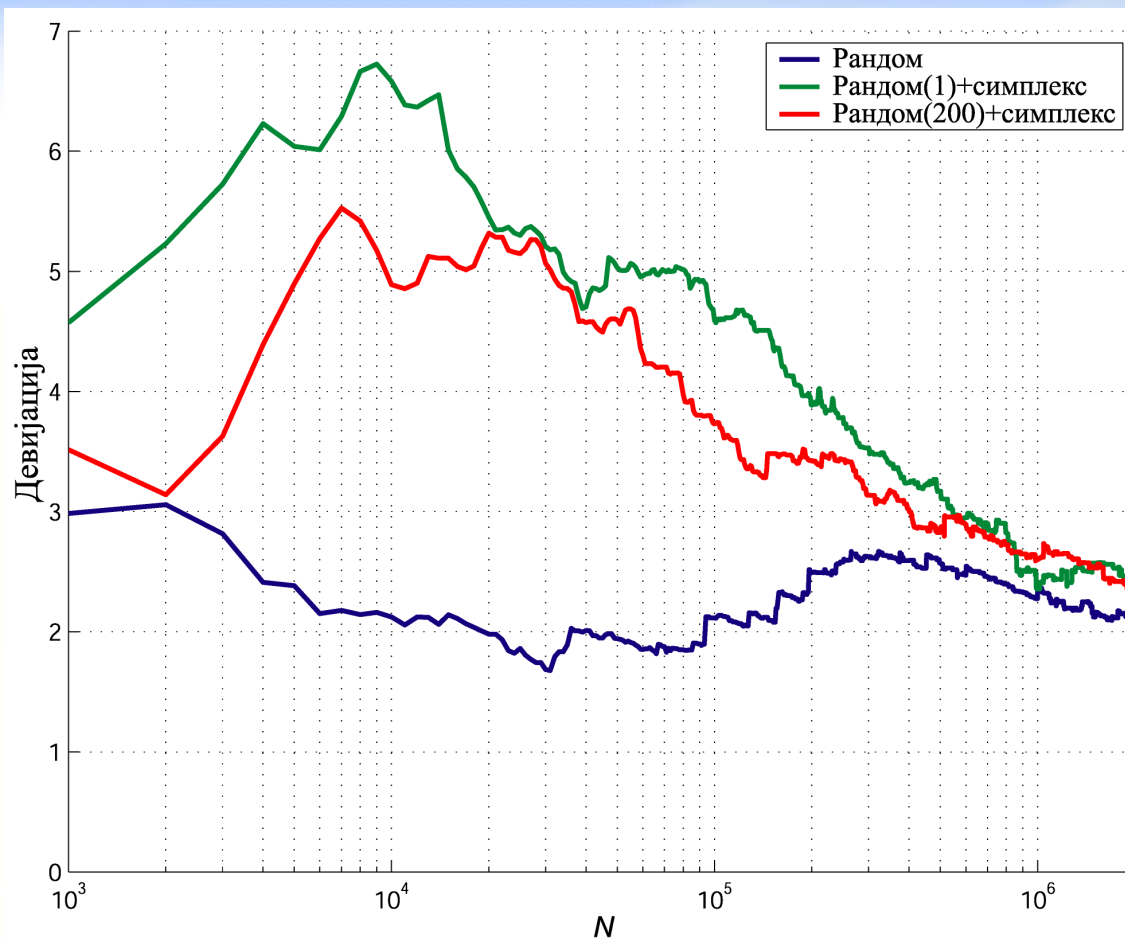
Симулирано каљење (ГА је ефикаснији)



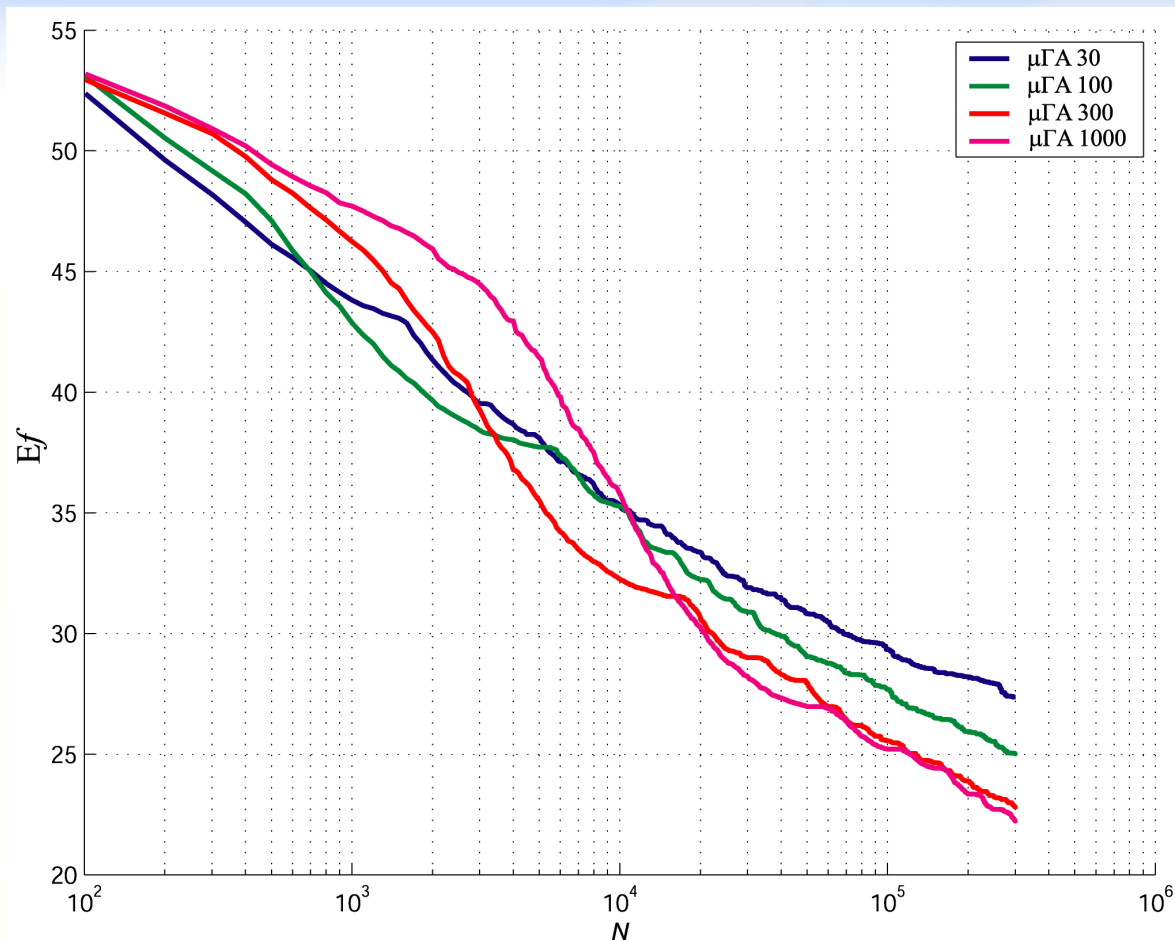
Методи понављања (ГА није много бољи)



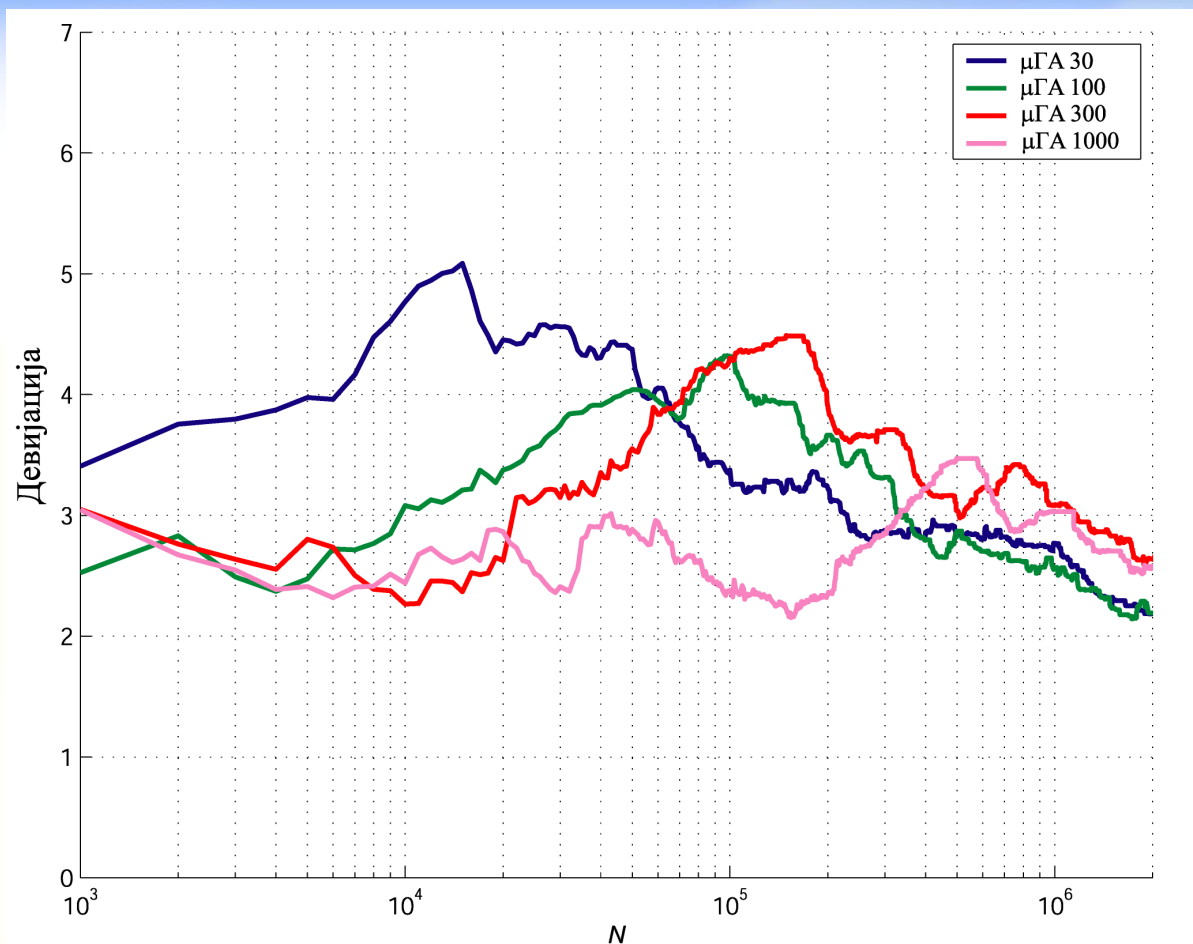
Методи понављања: девијација



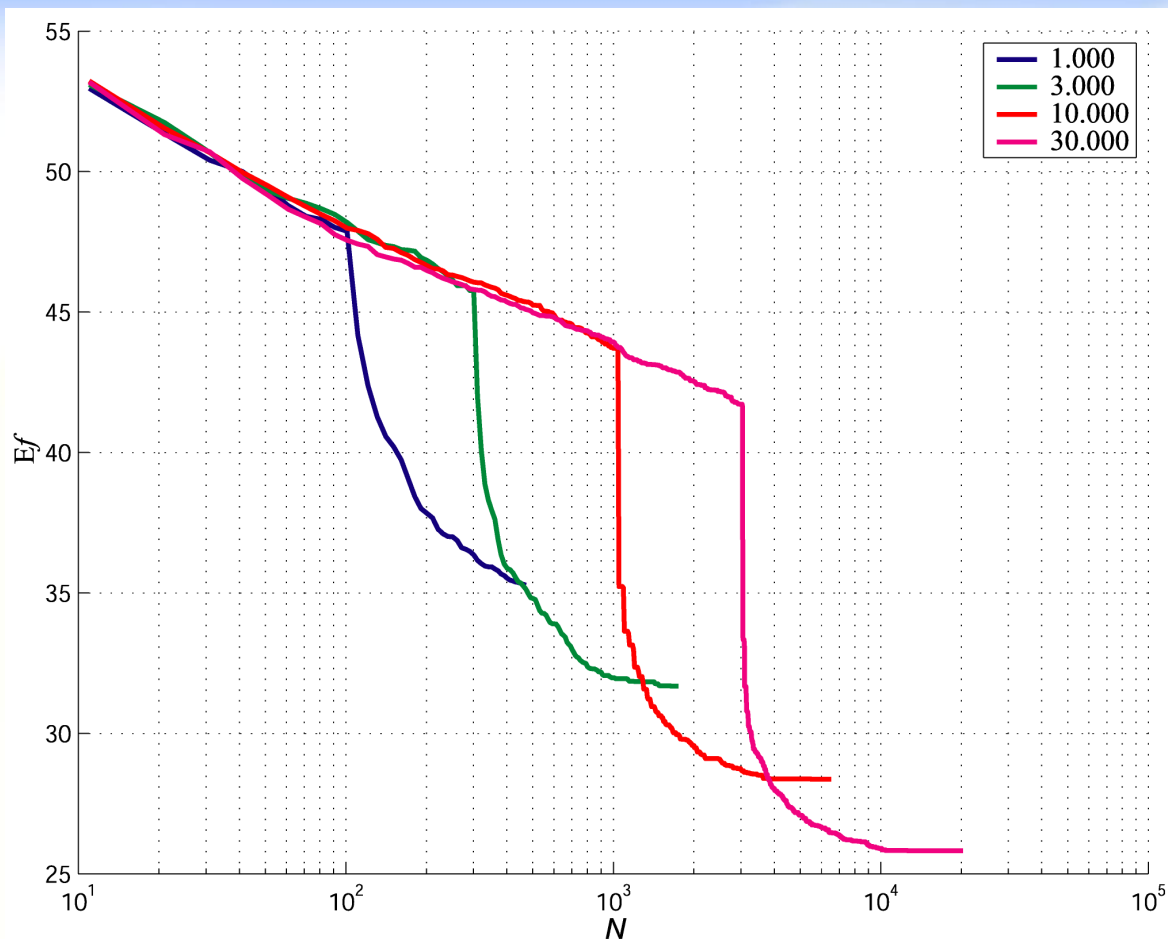
Микро ГА (популација)



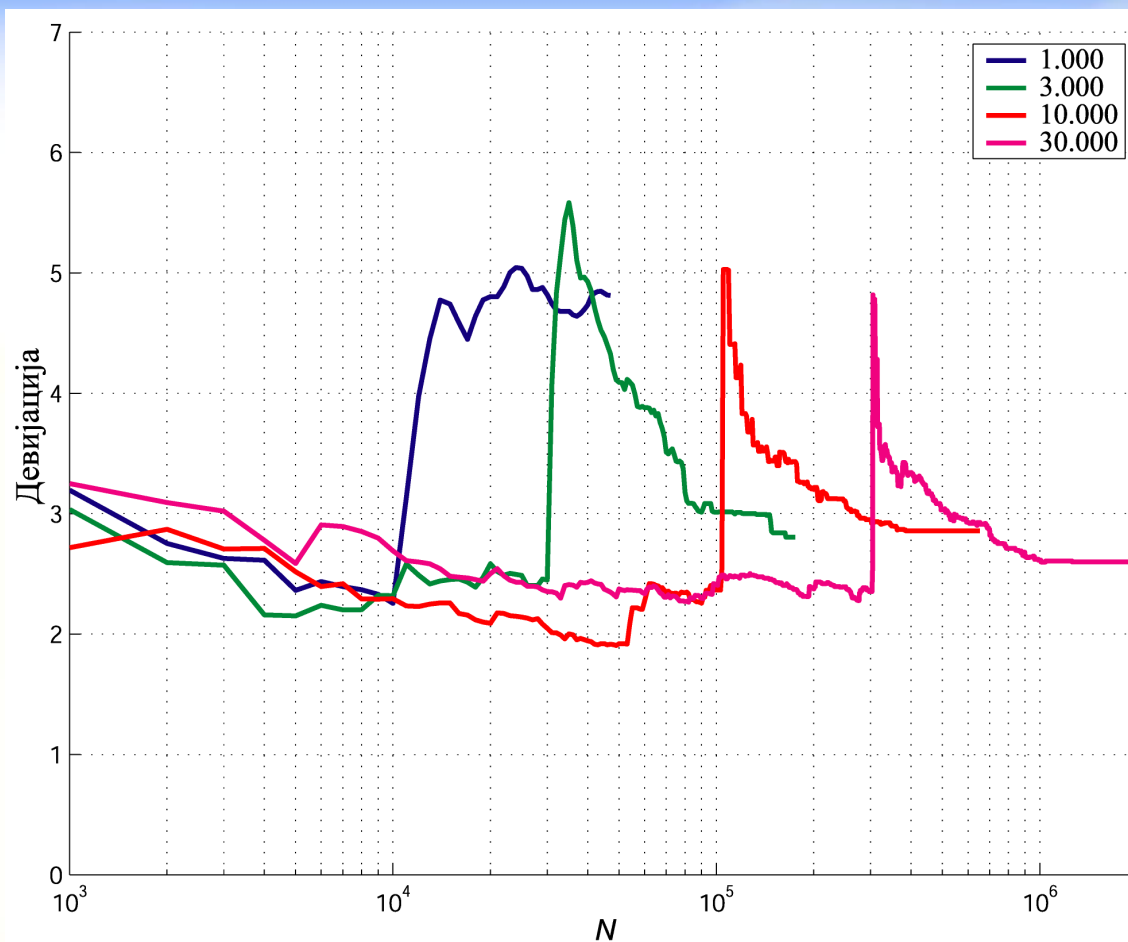
Микро ГА: девијација



Метод процене локалних минимума (полазни скуп)

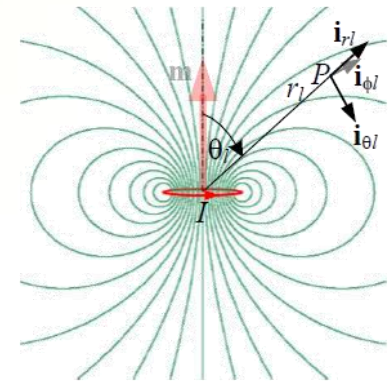
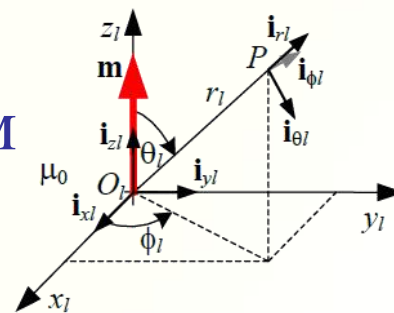
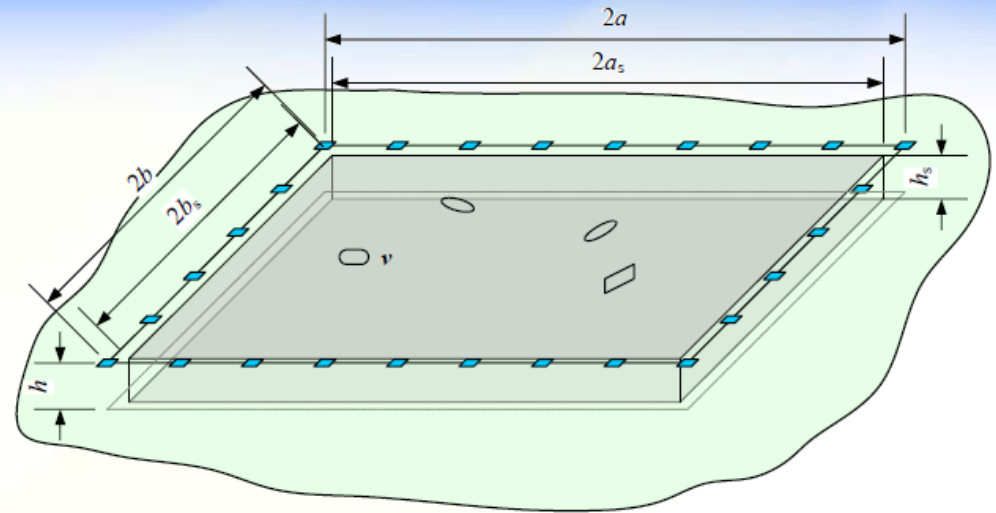


ПЛМ: девијација



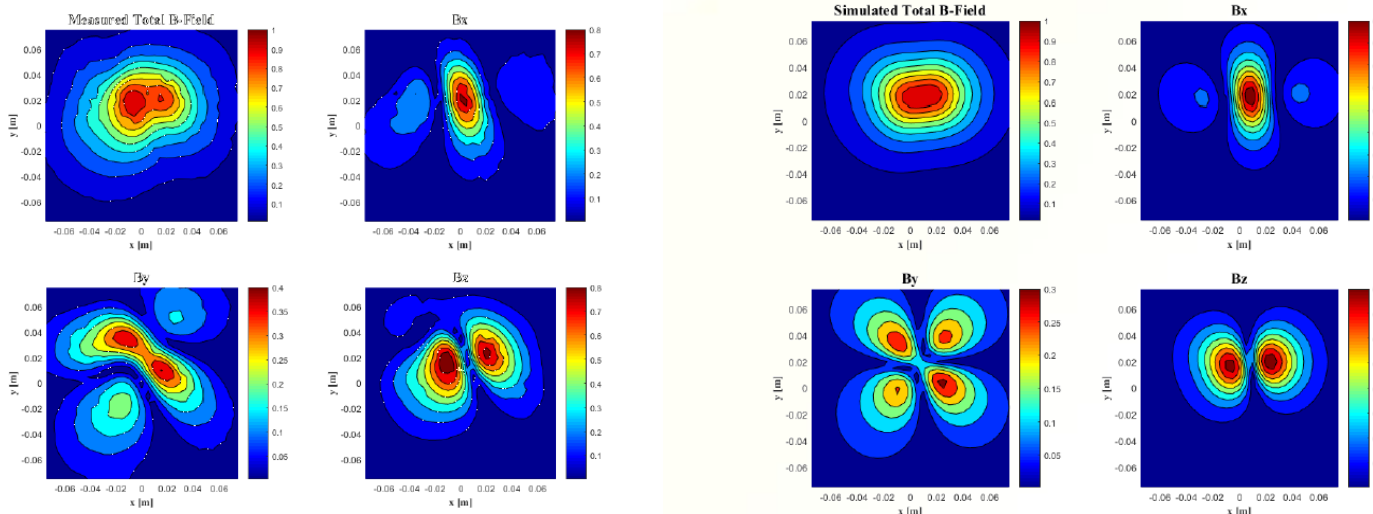
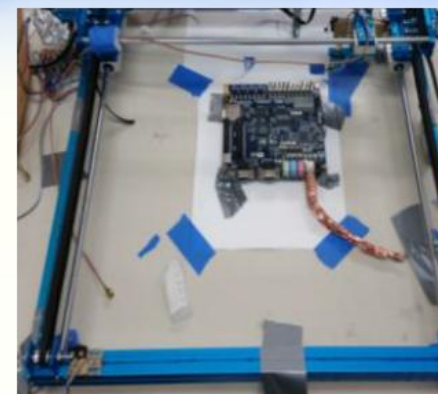
Пример #4: Одређивање процесорског кода на основу ЕМ

- У зависности од инструкција које се извршавају, различити делови матичне плоче су активни
- Сваки струјни пут моделује се магнетском контуром
- Мерење ЕМ поља око процесора



Експериментална поставка и мерење поља

- На FPGA процесору се извршава DIV/ADD пар инструкција
- Измерено поље и оптимизацијом пронађени извори

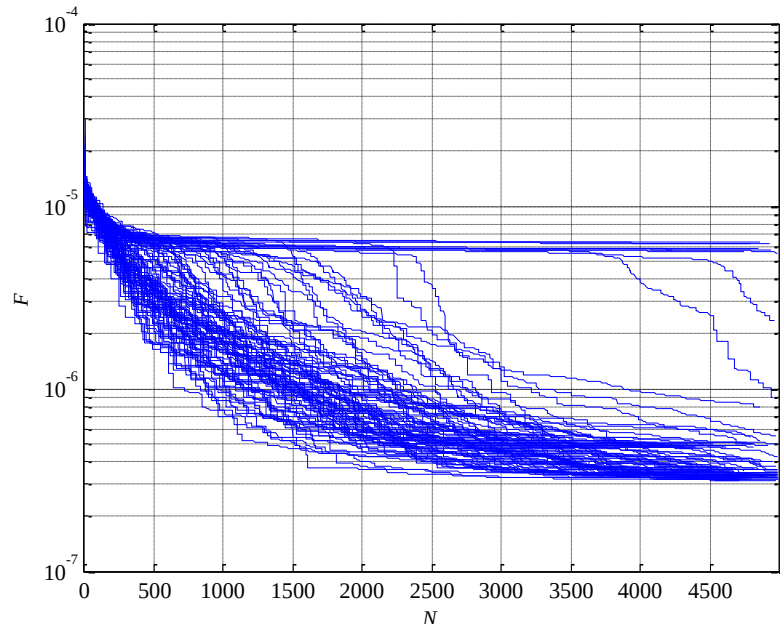
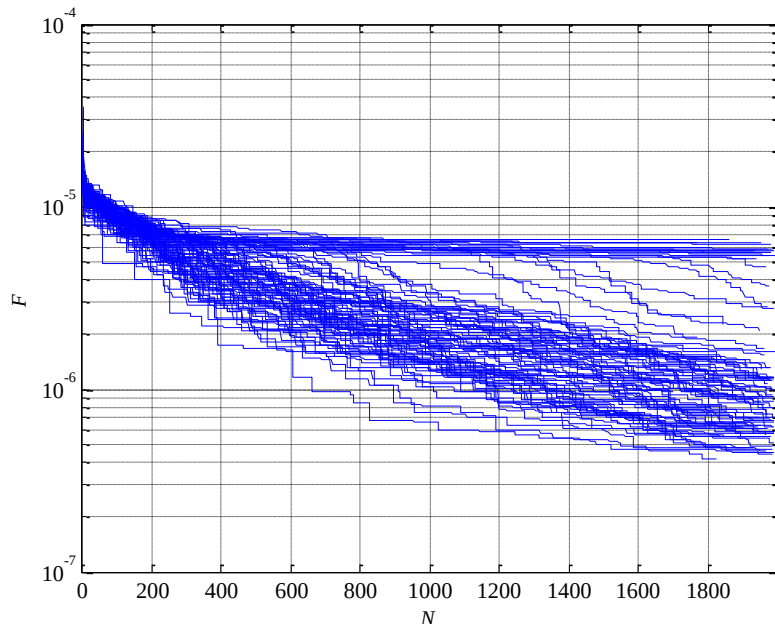


Оптимизациони проблем

- Број извора је оптимизациона променљива и тај број је дискретан
- Позиције извора поља и њихове струје су реални бројеви
- За сваки извор по 7 променљивих: x , y , z , углови ротације ϕ и θ , ефективна вредност струје и фаза (укупно 7 до 21 променљивих)
- У питању је комбинација SAT и NLP оптимизационих проблема!
- Коришћени су PSO, NM симплекс и случајно претраживање

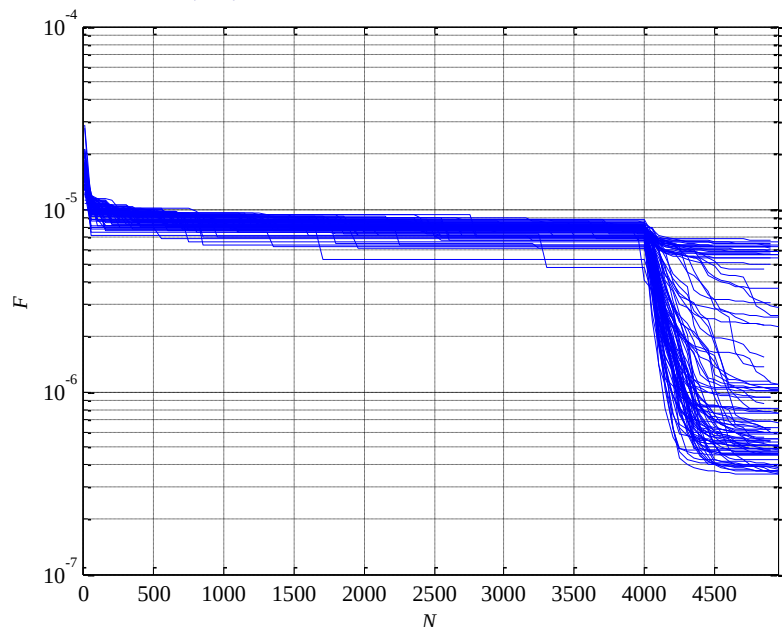
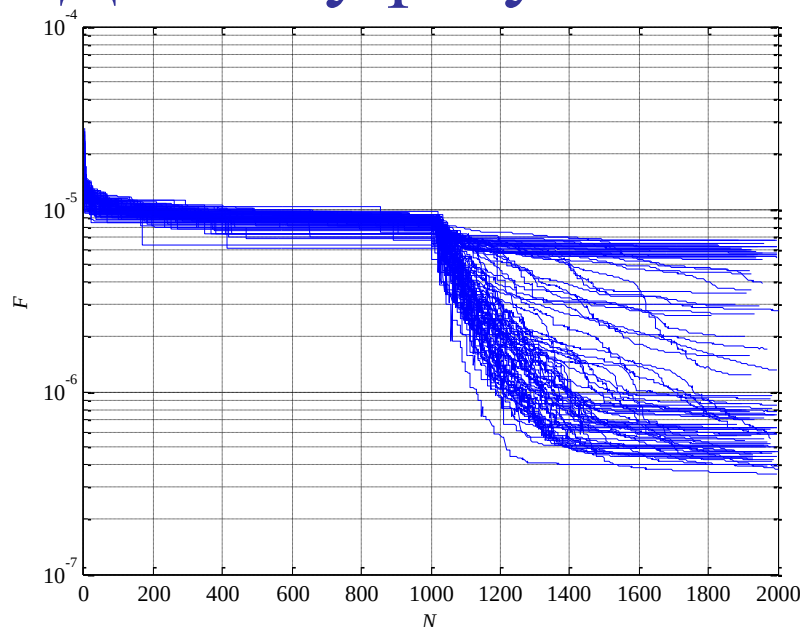
PSO максималан број итерација

- Број агената 25, итерација 2000 и 5000 (100 независних опт.)
- Неке оптимизације и са 2k пронађу добро решење, али са 5k се у већини случајева пронађе добро решење

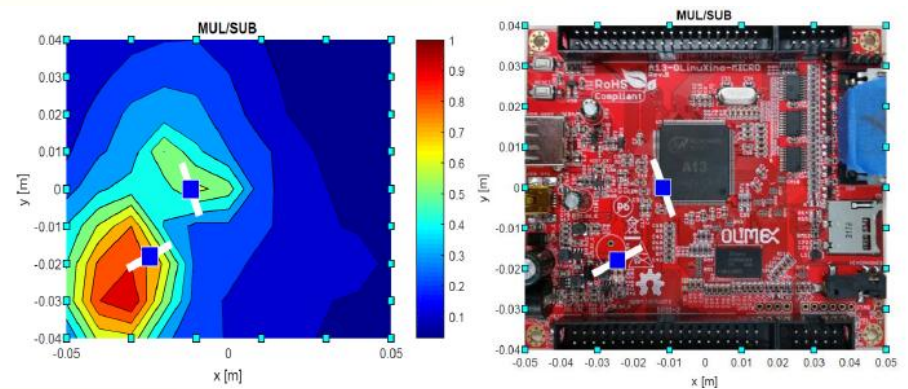
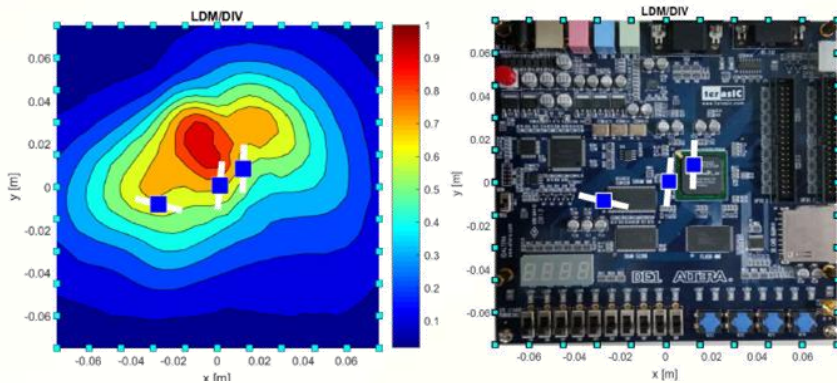
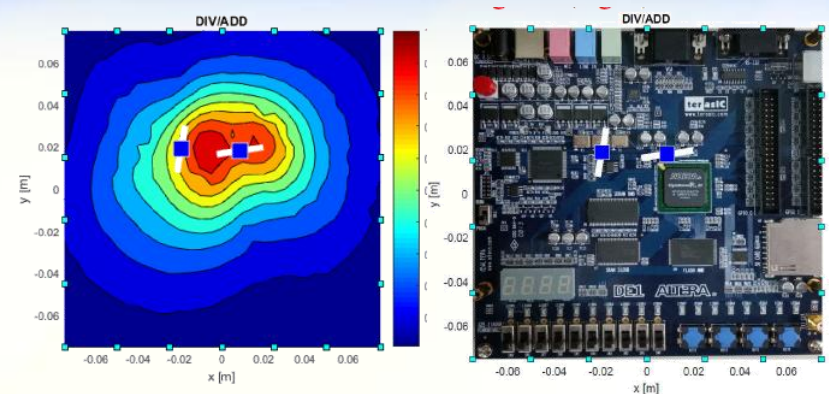
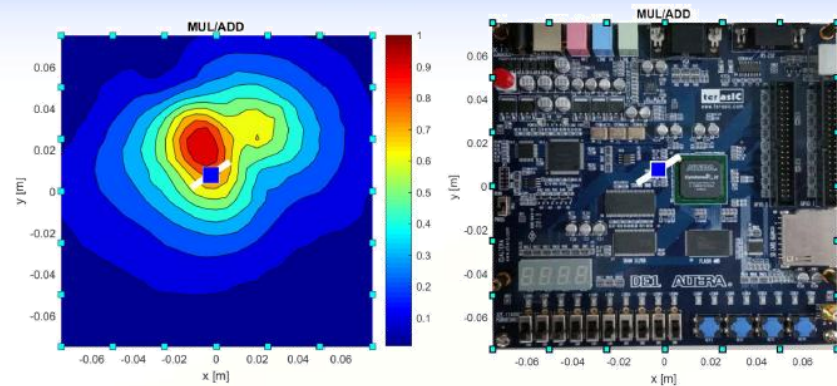


Случајно претраживање и NM симплекс

- Random(1k)+NMsimplex(1k) и Random(4k)+NMsimplex(1k)
- Да ли су резултати бољи од PSO?



Успешно идентификоване ситуације са 1, 2, 3 активна дела



Оптимизациона хеуристика

Алгоритам	D	Коментар	Главни параметар
Случајно претраживање	-	Грубо претраживање простора. Спора конвергенција. Само у комбинацији са другим алгоритмима.	Униформна или Гаусова расподела
Систематско претраживање	1-3	Детаљно претраживање простора. Неефикасно за просторе са више од ~ 3 димензије	Корак за претрагу може бити исти или различит за сваку димензију.
Градијентни алгоритам	1-7	Најбржа конвергенција, уколико је познато добро полазно решење.	Корак за процену градијента.
Симплекс	1-7	Најробуснији локални оптимизациони алгоритам. Може да пронађе најбољи од неколико околних минимума (статистички).	Димензије полазног симплекса.
Симулирано каљење	10+	Глобални оптимизациони алгоритам са спором конвергенцијом.	Почетна температура.
Генетички алгоритам	10+	Тренутно најефикаснији глобални оптимизациони алгоритам, уколико се посматрају проблеми са више од 10 димензија.	Величина популације.
PSO и DE	5-15	Алгоритми који су између глобалних и локалних оптимизационих алгоритама.	Број решења у јату (популацији).

Задатак за вежбе

- Дата је бинарна секвенца дужине 31:
 $X_0 = 1010\ 0001\ 0001\ 1011\ 0000\ 1100\ 1110\ 011$
- Пронаћи бинарну секвенцу X , за коју важе следећи услови
 - Секвенца је дужине 31 бит
 - Број нула и јединица у секвенци разликује се за тачно 1
 - Кроскорелација секвенци X_0 и X је већа од -4 и мања од 6 , за сваки циклички померај $k = 0, 1, 2, \dots, 30$
 - Аутокорелација секвенце је већа од -18 и мања од 12 , за сваки циклички померај $k = 1, 2, \dots, 30$
- **Кроскорелација** секвенци X_0 и X , једнаких дужина, дефинише се као разлика броја позиција са истим битима и броја позиција са различитим битима, за свако циклично померање секвенце X_0
- **Аутокорелација** секвенце X је кроскорелација секвенце са самом собом

Задатак: пример

- Кроскорелација секвенци

$\mathbf{x}_0 = 1010 \ 0001 \ 0001 \ 1011 \ 0000 \ 1100 \ 1110 \ 011$

$\mathbf{x} = 1001 \ 1101 \ 0010 \ 1010 \ 1010 \ 1001 \ 0101 \ 101$

за циклички померај $k = 0$ једнака је -1

- За циклички померај $k = 5$, секвенце $\mathbf{x}_0$ на десну страну, кроскорелација је једнака 11

$\mathbf{x}_0' = 1001 \ 1101 \ 0000 \ 1000 \ 1101 \ 1000 \ 0110 \ 011$

$\mathbf{x} = 1001 \ 1101 \ 0010 \ 1010 \ 1010 \ 1001 \ 0101 \ 101$

- Аутокорелација секвенце $\mathbf{x}$, за циклички померај $k = 3$ у десно, је -5

$\mathbf{x}' = 1011 \ 0011 \ 1010 \ 0101 \ 0101 \ 0101 \ 0010 \ 101$

$\mathbf{x} = 1001 \ 1101 \ 0010 \ 1010 \ 1010 \ 1001 \ 0101 \ 101$

Задатак: детаљи

- Задатак је испитног типа, по обиму и поставци
 - Запис решења (кодовање оптимизационих променљивих у вектор решења $\mathbf{x}$) не мора бити експлицитно задат, стога је потребно изабрати један могући запис
 - Описан је оптимизациони проблем, али оптимизациона функција није усвојена (потребно је изабрати једну могућу дефиницију)
 - Оптимизациони алгоритам за решавање није наведен, стога је потребно изабрати један алгоритам или комбинацију алгоритама са курса
- При избору записа најједноставније је изабрати природан запис (SAT, TSP или NLP) за задати оптимизациони проблем
- Проценити број различитих решења у оптимизационом простору и време израчунавања оптимизационе функције, и на основу тога изабрати оптимизациони алгоритам
 - Уколико потпуна претрага траје недопустиво дуго, изабрати оптимизациони алгоритам (искористити кодове и приступе са предавања и вежби) и решити задати проблем
 - Ако је могућа потпуна претрага, урадити је!
- Потребно је пронаћи **једно решење** овог проблема и записати га у пратећи ASCII/TXT фајл
- Решење представити **и у бинарном и у децималном запису!**
- Поред ASCII/TXT фајла проследити и код коришћен за решавање