

**Универзитет у Београду**  
**Машински факултет**



**Рачунарска интелигенција**  
**Adaptive simulated annealing**

**Бојана Стамболић 4011/2023**  
**Маша Малишић 4012/2023**

**31.10.2025.**

## Садржај

1. Увод .....	4
2. Генетички алгоритам (GA).....	5
2.1. Елементи генетичког алгоритма .....	5
3. Алгоритам симулираног жарења (SA).....	7
3.1. Принцип рада алгоритма симулираног жарења .....	8
3.2. Специфичне карактеристике алгоритма .....	9
3.3. Примена .....	9
4. Алгоритам адаптивног симулираног жарења (ASA).....	10
4.1. Основне идеје ASA алгоритма .....	10
4.2. Температурни распоред .....	10
4.3. Генерисање нових кандидата .....	11
4.4. Температура прихватања .....	11
4.5. Поновно каљење (Reannealing) .....	12
4.6. Главни ток ASA алгоритма .....	12
4.7. Предности ASA алгоритма .....	12
5. Минимизација тест функција .....	14
5.1. Функција Rosenbrock .....	14
5.1.1. Генетички алгоритам – анализа резултата.....	15
5.1.2. Алгоритам симулираног жарења – анализа резултата .....	16
5.1.3. Алгоритам адаптивног симулираног жарења – анализа резултата.....	18
5.1.4. Поређење алгоритама – минимизација Rosenbrock функције .....	18
5.2. Функција Rastrigin .....	19
5.2.1. Генетички алгоритам – анализа резултата.....	20
5.2.2. Алгоритам симулираног жарења – анализа резултата .....	21
5.2.3. Алгоритам адаптивног симулираног жарења – анализа резултата.....	22
5.2.4. Поређење алгоритама – минимизација Rastrigin функције.....	23
5.3. Функција Schwefel .....	24
5.3.1. Генетички алгоритам – анализа резултата.....	25
5.3.2. Алгоритам симулираног жарења – анализа резултата .....	26
5.3.3. Алгоритам адаптивног симулираног жарења – анализа резултата.....	27
5.3.4. Поређење алгоритама – минимизација Schwefel функције .....	27
6. Закључак.....	28
7. Литература.....	29

Слика 1 – Структура GA.....	5
Слика 2 – Принцип рада SA .....	7
Слика 3 – Приказ опадања температуре.....	9
Слика 4 – Rosenbrock функција .....	14
Слика 5 – Функција Rastrigin.....	19
Слика 6 – Функција Schwefel .....	24

Табела 1 – Оператори GA .....	6
Табела 2 – Резултати минимизације Rosenbrock функције применом GA .....	15
Табела 3 – Резултати минимизације Rosenbrock функције применом SA.....	16
Табела 4 - Резултати минимизације Rosenbrock функције применом ASA.....	18
Табела 5 - Резултати минимизације Rastrigin функције применом GA .....	20
Табела 6 - Резултати минимизације Rastrigin функције применом SA.....	21
Табела 7 - Резултати минимизације Rastrigin функције применом ASA .....	22
Табела 8 - Резултати минимизације Schwefel функције применом GA.....	25
Табела 9 - Резултати минимизације Schwefel функције применом SA.....	26
Табела 10 - Резултати минимизације Schwefel функције применом ASA.....	27

## 1. Увод

Савремена индустрија и инжењерство све више зависе од примене рачунарских метода које омогућавају решавање сложених проблема оптимизације. Са растом комплексности система, појављује се потреба за алгоритмима који могу ефикасно претраживати велики простор могућих решења и проналазити глобални минимум или максимум у условима бројних ограничења. Управо из тог разлога, рачунарска интелигенција све чешће користи стохастичке и еволутивне алгоритме који имитирају природне процесе учења, еволуције и физичких појава.

Оптимизација представља процес проналажења најбољег решења за одређени проблем у складу са постављеним циљевима и ограничењима. У пракси, то може бити минимизација трошкова, времена или енергије, као и максимизација ефикасности или прецизности. Класичне аналитичке методе често нису применљиве на сложене нелинеарне проблеме са више екстрема, па се уместо њих користе адаптивни алгоритми који омогућавају глобалну претрагу простора решења.

У овом раду примењена су три алгоритма оптимизације: **генетички алгоритам (GA)**, **алгоритам симулираног жарења (SA)** и **адаптивни алгоритам симулираног жарења (ASA)**. Генетички алгоритам је инспирисан процесом природне еволуције и користи операторе селекције, укрштања и мутације за постепено побољшавање решења. Алгоритам симулираног жарења заснива се на физичком процесу хлађења метала и користи концепт температуре за контролу вероватноће прихватања лошијих решења, што омогућава избегавање локалних минимума. ASA представља унапређену верзију SA алгоритма, која адаптивно мења параметре претраге у зависности од тренутног стања система, чиме се убрзава конвергенција и побољшава прецизност решења.

Да би се тестирала ефикасност и стабилност наведених алгоритама, коришћене су познате стандардне тест функције за оптимизацију: **Rosenbrock**, **Rastrigin** и **Schwefel**. Ове функције су одабране, јер представљају различите типове оптимизационих изазова.

Применом GA, SA и ASA алгоритама на наведеним функцијама омогућена је упоредна анализа њихових перформанси, брзине конвергенције и способности проналажења глобалног оптимума. Добијени резултати пружају увид у предности и ограничења сваког приступа и представљају основу за њихову даљу примену у реалним инжењерским и индустријским задацима оптимизације.

## 2. Генетички алгоритам (GA)

**Генетички алгоритми** представљају једну од најпознатијих и најстаријих метода за решавање оптимизационих проблема, које се базирају на природној селекцији и принципима *Дарвинове еволуције*. Засновани су на идеји опстанка најприлагођенијих јединки, где боља решења имају већу вероватноћу да „преживе“ и стварају нова решења.

Оригинална идеја о генетичким алгоритмима потиче из 1975. године од Џона Холанда, где се први пут предлаже кодирање генетске информације у виду хромозома фиксне дужине и на тај хромозом се примењују оператори генетичких алгоритама.

1.1 На слици приказана је структура генетичког алгоритама.



Слика 1 – Структура GA

### 2.1. Елементи генетичког алгоритама

1. **Иницијализација популације** - процес еволуције започиње формирањем популације случајно генерисаних јединки и извршава се кроз генерације. Вњличина популације је један од главних параметара алгоритама.

2. **Функција циља** - прилагођена изабраној шеми кодирања и користи се за процену квалитета сваке јединке, односно за одређивање критеријума шта представља боље решење. Задатак алгоритма је да максимизује или минимизује вредност ове функције.
3. **Оператори генетичких алгоритама** – у табели приказани су оператори генетичких алгоритама.

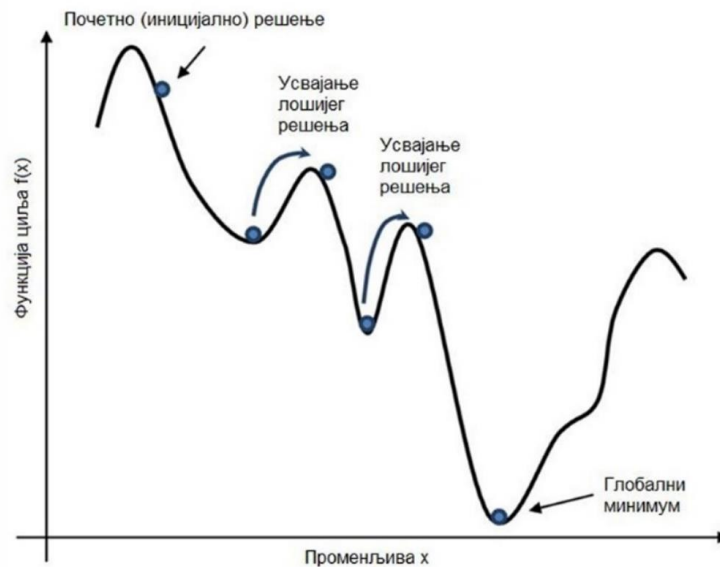
**Табела 1 – Оператори GA**

Оператор	Опис
Кодирање	Процес представљања јединке у облику погодном за генетички алгоритама. Најчешће се користи бинарно кодирање (хромозом као низ бита), али постоје и реално, пермутационо и друго кодирање. Циљ је да проблем буде што једноставније и ефикасније представљен.
Селекција	Процес одабира појединаца из популације на основу њихове прилагодљивости. Циљ је да боље јединке имају већу вероватноћу да буду изабране за стварање нових генерација. Најчешће технике: рулет селекција, турнирска селекција и рангирање. У пракси се често користи и <b>елитизам</b> , који осигурава да један или више најбољих појединаца из популације директно пређу у наредну генерацију без измена.
Укрштање	Комбинује генетички материјал два родитеља како би се добили потомци. Идеја је да се искористе добре особине оба родитеља и тако створи јединка са потенцијално бољим решењем. Најчешћи типови: једнопозиционо, вишепозиционо и униформно укрштање.
Мутација	Уношење случајних промена у генетички материјал појединца (на пример, промена једног бита у бинарној репрезентацији). Циљ је да се задржи разноврсност у популацији и спречи прерано конверговање ка локалном минимуму/максимуму.

4. **Услов заустављања** – алгоритам се зауставља када буде испуњен неки од критеријума, као што су достизање задате вредности функције циља, максималног броја генерација, временског ограничења, стагнације решења или ручног прекида.

### 3. Алгоритам симулираног жарења (SA)

Алгоритам симулираног жарења (енгл. Simulated Annealing – SA) представља стохастичку оптимизациону методу која се користи за решавање како дискретних (нпр. комбинаторних), тако и континуалних оптимизационих проблема. За разлику од детерминистичких алгоритама, SA поседује могућност да избегне заробљавање у локалне оптимуме тако што у одређеним фазама процеса дозвољава прихватање и лошијих решења. На тај начин алгоритам повећава вероватноћу проналажења глобалног оптимума. Принцип рада алгоритма илустрован је на слици 2.



Слика 2 – Принцип рада SA

Алгоритам симулираног жарења добио је назив и основну идеју из процеса жарења метала. У реалном процесу метал се најпре загрева до високе температуре, а затим се постепено хлади. Током овог хлађења атоми метала спонтано формирају правилну кристалну структуру без деформација, односно структуру минималне енергије.

У аналогiji са оптимизационим проблемом:

1. стање система одговара једном решењу проблема,
2. енергија система одговара вредности функције циља,
3. температура представља контролни параметар који одређује вероватноћу прихватања лошијих решења.



### 3.1. Принцип рада алгоритма симулираног жарења

Као што је приказано на слици 2, похлепни алгоритам често конвергира ка локалном минимуму и остаје заробљен у њему, док алгоритам симулираног жарења захваљујући прихватању и лошијих решења у одређеним фазама претраге може да изађе из локалног минимума и да настави претрагу простора све док не достигне глобални оптимум. На тај начин алгоритам показује већу поузданост у односу на традиционалне методе оптимизације.

У алгоритамском смислу, SA почиње избором почетног решења и постављањем почетне темпе ратуре на високу вредност. У свакој итерацији бира се ново решење из околине тренутног, након чега се израчунава разлика у енергији:

$$\Delta E = E_2 - E_1$$

где је  $E_1$  енергија тренутног решења, а  $E_2$  енергија новог решења.

Уколико је  $\Delta E < 0$ , ново решење се прихвата као боље. У случају да је  $\Delta E > 0$ , лошије решење се ипак може прихватити са одређеном вероватноћом:

$$P(\Delta E) = e^{-\Delta E/kT}$$

где је  $k$  Болцманова константа, а  $T$  тренутна температура.

Овакво прихватање са одређеном вероватноћом лошијих решења омогућава алгоритму да избегне локалне минимуме и да врши глобалну претрагу простора решења.

Како температура опада, вероватноћа прихватања лошијих решења се смањује.

Када  $T \rightarrow \infty$ :

$$\lim_{T \rightarrow \infty} e^{-\Delta E/kT} = 1$$

што значи да се на високој температури прихватају и лошија решења готово са сигурношћу.

Са друге стране, када  $T \rightarrow 0$ :

$$\lim_{T \rightarrow 0} e^{-\Delta E/kT} = 0$$

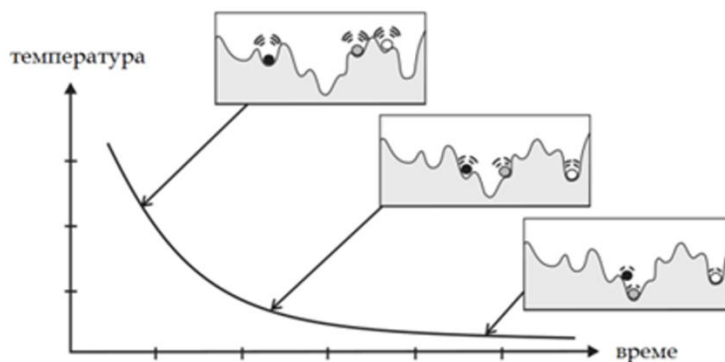
па се на ниским температурама прихватају само боља решења.

Температура најчешће опада експоненцијално, по правилу:

$$T_{t+1} = T_c \cdot T_t$$

где је  $T_c \in (0, 1)$  тзв. фактор хлађења.

На овај начин, како итерације напредују, температура постепено опада све док вероватноћа при хватања лошијих решења не постане једнака нули, чиме алгоритам завршава оптимизацију.



Слика 3 – Приказ опадања температуре

Као што је приказано на слици 3, температура у алгоритму симулираног жарења опада током времена формирајући експоненцијалну путању. У почетним корацима, при високим вредности ма температуре, алгоритам има већу вероватноћу прихватања лошијих решења, што омогућава излазак из локалних минимума. Како температура постепено опада, вероватноћа прихватања лошијих решења се смањује, па алгоритам тежи да се стабилизује у глобалном оптимуму.

### 3.2. Специфичне карактеристике алгоритма

- SA је недетерминистички алгоритам, па исти проблем може дати различита решења при више покретања.
- Алгоритам је способан да избегне велики број локалних оптимума, али не гарантује да ће увек достићи глобално оптимално решење.
- У односу на традиционалне алгоритме, SA је процесорски захтевнији, али се његова предност огледа у флексибилности и могућности проналажења бољих решења у сложеним просторима претраге.

### 3.3. Примена

Захваљујући својој једноставности и снажној способности глобалне претраге, SA је нашао примену у великом броју области:

- решавање проблема трговачког путника (TSP),
- терминирање у производним системима,
- проблеми рутирања и планирања путање,
- дигитална обрада сигнала и пројектовање филтера,
- телекомуникационе мреже,
- обрада слике и оптимизација у електронским колима.

## 4. Алгоритам адаптивног симулираног жарења (ASA)

**Simulated Annealing (SA)** је стохастички алгоритам за глобалну оптимизацију који имитира процес физичког каљења метала — споро хлађење омогућава систему да постигне стабилно, минимално енергетско стање. Класични SA користи јединствену „температуру“ која постепено опада, али је тај процес често спор и неефикасан код сложених или вишедимензионалних проблема.

**Adaptive Simulated Annealing (ASA)**, који је развио **Lester Ingber (1989)**, представља побољшану верзију SA. ASA омогућава прилагођавање температуре за сваку димензију посебно, користи **адаптивно хлађење** и **реанеалинг** (reannealing) како би се процес оптимизације убрзао, уз очување способности проналажења глобалног минимума.

### 4.1. Основне идеје ASA алгоритма

ASA користи две главне температуре:

- **Параметарску температуру** (*parameter temperature*) за контролу величине корака у свакој димензији,
- **Температуру прихватања** (*acceptance temperature*) која контролише вероватноћу прихватања лошијих решења.

Поред тога, ASA уводи:

- **quenching фактор**  $Q_i$ , који омогућава убрзано хлађење,
- **reannealing** адаптацију, која прилагођава температуре на основу локалних градијената функције циља.

### 4.2. Температурни распоред

Параметарска температура по димензији  $i$  дефинисана је експоненцијалним опадањем:

$$T_i(k) = T_{0i} \cdot e^{-c_i \cdot k^{\frac{1}{D \cdot Q_i}}}$$

где је  $D$  број димензија проблема,  $T_{0i}$  почетна температура за димензију  $i$ , а  $c_i$  параметар који контролише брзину опадања температуре,  $Q_i$  — *quenching фактор* (ако је  $Q_i > 1$ , температура опада брже). Када је  $Q_i = 1$ , добија се стандардни ASA без убрзаног хлађења. Овај експоненцијални закон омогућава ефикасније смањивање температуре у односу на класични SA, који користи логаритамско хлађење.

#### 4.3. Генерисање нових кандидата

За сваку димензију  $i$ , нова вредност се добија према формули:

$$y_i = \text{sgn}(u_i - 0.5) \cdot T_i(k) \cdot \left[ \left( 1 + \frac{1}{T_i(k)} \right)^{|2u_i - 1|} - 1 \right]$$

где је  $u_i \in U(0,1)$ , случајна вредност уз умерене дистрибуције. Ови поремећаји омогућавају истраживање простора решења у складу са тренутном температуром — већа температура дозвољава веће одступање, док мања постепено усмерава алгоритам ка локалним минимумима.

Затим се нови кандидат израчунава као:

$$x_i^{new} = x_i^{old} + y_i(UB_i - LB_i)$$

са ограничењем  $UB_i \leq x_i^{new} \leq LB_i$ .

Ова дистрибуција омогућава да систем може да направи и велике и мале кораке, што побољшава могућност изласка из локалних минимума.

#### 4.4. Температура прихватања

Температура прихватања се такође експоненцијално смањује, али зависи од броја прихваћених решења (препоручено је да се користи број прихватања уместо броја итерација):

$$T_{cost}(a) = T_{0,cost} \cdot e^{-c_{cost}(a+1)^{\frac{1}{D \cdot Q_i}}}$$

где је  $a$  број прихваћених решења.

Прихватање новог решења одређује се Metropolis правилом:

$$P_{accept} = f(x) = \begin{cases} 1, & \text{ако } \Delta f \leq 0 \\ \exp\left(-\frac{\Delta f}{T_{cost}}\right), & \text{ако } \Delta f > 0 \end{cases}$$

где је  $\Delta f = f(x_{new}) - f(x_{old})$ , ако је случајна вредност  $r \in [0,1]$  мања од  $P_{accept}$  решење се прихвата. Ово омогућава да алгоритам повремено прихвати и лошије решење, чиме се избегава заглављивање у локалним минимумима.

#### 4.5. Поновно каљење (Reannealing)

Да би се прилагодила стопа промена по димензијама, ASA периодично рачуна **приближне градијенте** функције циља у најбољем познатом решењу:

$$s_i = \left| \frac{\partial f}{\partial x_i} \right|$$

Затим се све вредности нормализују:

$$s_{max} = \max(s_i) + \varepsilon$$

$$scale_i = \frac{c_i}{s_i + \varepsilon}$$

и параметри температуре се прилагођавају:

$$c_i \leftarrow \frac{c_i}{scale_i}, c_{cost} \leftarrow \frac{c_{cost}}{scale_i}$$

Овим се постиже да се у димензијама где је функција „равнија“ (мали градијент) температура смањује спорије — алгоритам више истражује те димензије.

#### 4.6. Главни ток ASA алгоритма

1. **Иницијализација:** Постављање почетног решења  $x_0$  и температура  $T_{0,i}$ .
2. **Генерисање кандидата:** Израчунавање  $y_i$  и  $x_i^{new}$ .
3. **Процена:** Израчунавање  $f(x_{new})$ .
4. **Прихватање/одбијање:** Примена Метрополисовог критеријума.
5. **Ажурирање температура:** Израчунавање  $T_i(k)$  и  $T_{cost}$ .
6. **Поновно каљење:** Повремено подешавање  $c_i$  према градијентима.
7. **Завршетак:** Након задатог броја итерација или стабилизације решења.

#### 4.7. Предности ASA алгоритма

- Прилагођава температуру *по димензији*, што омогућава брже прилагођавање сложеним функцијама.
- Може да користи **quenching** да би се убрзало хлађење без великих губитака у квалитету.
- **Reannealing** омогућава да се алгоритам „поново загреје“ и настави истраживање ако се приближи локалном минимуму.
- Због своје експоненцијалне природе, ASA има **врло брзу конвергенцију** у односу на класични SA.

На овај начин ASA комбинује адаптивно хлађење, реанилинг и контролисано квање како би постигла баланс између глобалне претраге и локалне експлоатације простора решења. Овај алгоритам је снажан и флексибилан алгоритам за глобалну оптимизацију. У односу на класични Simulated Annealing, он омогућава адаптивно прилагођавање по свакој димензији, што доводи до брже и поузданије конвергенције.

## 5. Минимизација тест функција

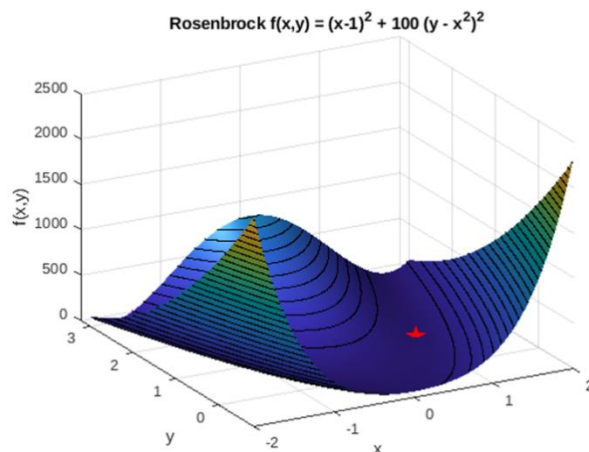
Рад претходно поменута три алгоритма, GA, SA и ASA, тестирали смо на минимизацији три тест функције. У даљем раду представљене су тест функције и приложени су резултати извршених експеримената.

### 5.1. Функција Rosenbrock

Розенброкова функција је непрекидна, нековексна и детерминистичка функција са израженом нелинеарношћу. Њен уски и издужени минимум, познат као Розенброкова долина, представља типичан случај за испитивање ефикасности оптимизационих алгоритма, укључујући генетичке алгоритме. Математички функција се дефинише као:

$$f(x, y) = (x - 1)^2 + 100 \cdot (y - x^2)^2$$

На слици 4 се налази 3D приказ Розенброкове функције.



Слика 4 – Rosenbrock функција

У овом поглављу приказана је употреба генетичког алгоритма, алгоритма симулираног жарења и алгоритма адаптивног симулираног жарења за проналажење глобалног минимума функције, са анализом утицаја различитих параметара алгоритма са аспекта брзине конвергенције, оствареног оптималног решења и остварене грешке.

### 5.1.1. Генетички алгоритам – анализа резултата

Приликом тестирања генетичког алгоритма на минимизацији Rosenbrock функције изведено је 25 експеримената који су приказани у табели 2. Приликом покретања мењани су параметри величина популације, максималан број генерација, вероватноћа укрштања, вероватноћа мутације и број елитних хромозома и разматран је њихов утицај на резултате.

Табела 2 – Резултати минимизације Rosenbrock функције применом GA

Параметри	Максимум	# Минимум	Средња вредно	Стандардна дс
n=50,gens=100,cross=90%,mut=0.020,elite=5	0.37443269	0.002661126657	0.09188825642	0.1213685954
n=100,gens=100,cross=90%,mut=0.020,elite=5	0.2367898263	0.0002797792238	0.0282932096	0.07343428509
n=200,gens=100,cross=90%,mut=0.020,elite=5	0.07162645323	0.000241713614	0.01578560786	0.02160003273
n=400,gens=100,cross=90%,mut=0.020,elite=5	0.009017938503	0.0004355854874	0.004166521959	0.002836410935
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.003980620992	0.0000577211614	0.0007956246249	0.001170938637
n=800,gens=50,cross=90%,mut=0.020,elite=5	0.007400880978	0.000101616893	0.002448171106	0.002679817576
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.003818971893	0.000299162121	0.001251376035	0.00103947927
n=800,gens=200,cross=90%,mut=0.020,elite=5	0.001268996108	0.0000112916859	0.0005497034878	0.0003496405538
n=800,gens=400,cross=90%,mut=0.020,elite=5	0.001628708919	0.00002085027769	0.0004161309351	0.0005603795767
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.0003567653931	0.000007708904134	0.0001786868593	0.0001362093423
n=800,gens=800,cross=50%,mut=0.020,elite=5	0.000503369167	0.00003087390121	0.0002110577471	0.0001608922205
n=800,gens=800,cross=70%,mut=0.020,elite=5	0.000422968079	0.000006439773618	0.00008123372933	0.0001233830413
n=800,gens=800,cross=80%,mut=0.020,elite=5	0.0007303551563	0.000001080000489	0.0001912171113	0.0002340946603
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.0005838172006	0.00002585635431	0.0002031499686	0.0001660495803
n=800,gens=800,cross=100%,mut=0.020,elite=5	0.0004930203953	0.000002418617751	0.00009292381568	0.0001464928053
n=800,gens=800,cross=70%,mut=0.010,elite=5	0.001130663316	0.000002548161458	0.0003254476097	0.0004087278764
n=800,gens=800,cross=70%,mut=0.030,elite=5	0.0006959281793	0.00001950119092	0.0003062263633	0.0002402511054
n=800,gens=800,cross=70%,mut=0.050,elite=5	0.0006072421344	0.000009428011726	0.0001171061778	0.0001859661789
n=800,gens=800,cross=70%,mut=0.100,elite=5	0.000437801795	0.000002087093109	0.0001287335725	0.0001261951785
n=800,gens=800,cross=70%,mut=0.200,elite=5	0.0006099910949	0.000001993207451	0.0001656740269	0.0001936222252
n=800,gens=800,cross=70%,mut=0.050,elite=1	0.001376504982	0.00002478368209	0.0003877468573	0.0004726144427
n=800,gens=800,cross=70%,mut=0.050,elite=2	0.0004517305532	0.00001146593042	0.0001847147392	0.0001553745501
n=800,gens=800,cross=70%,mut=0.050,elite=5	0.001351406427	0.00002333170247	0.0003776401539	0.0004768137352
n=800,gens=800,cross=70%,mut=0.050,elite=10	0.0006619116608	0.0000005045909936	0.0001148195159	0.0002000738566
n=800,gens=800,cross=70%,mut=0.050,elite=20	0.0005269980418	0.000001039184475	0.0001779839246	0.0001717956442

Повећање величине популације позитивно је утицало на решења, односно повећањем броја јединки добијене су мање вредности параметара, такође је и смањена стандардна девијација што указује на стабилизацију решења до којих је алгоритам долази. Повећање максималног броја генерација такође је позитивно утицало на рад алгоритма, добијане су још ниже вредности и алгоритам се додатно стабилизовао. Следећи разматран параметар јесте вероватноћа укрштања, најнижи минимум је добијен када је вредност параметра crossover била 80%, док су приликом



crossover-a од 70% добијене најстабилније вредности. Приликом мењања параметра вероватноће мутације, најбоље вредности су добијене када је за параметар мутације узета вредност 0.2. Најнижу вредност приликом 25 изведених експеримената добили смо када је за број елитних јединки узет број 10.

Приликом сваког од изведених експеримената добијене су задовољавајуће вредности мера од значаја, чиме можемо закључити да је генетички алгоритам погодан за минимизацију Rosenbrock функције.

### 5.1.2. Алгоритам симулираног жарења – анализа резултата

Приликом тестирања алгоритма симулираног жарења на минимизацији Rosenbrock функције изведено је 15 експеримената који су приказани у табели 3. Приликом покретања мењани су параметри иницијална температура  $T_0$ , минимална температура, фактор хлађења и разматран је њихов утицај на резултате.

Табела 3 – Резултати минимизације Rosenbrock функције применом SA

Параметри	✓	#	Најбоља	✓	#	Средња	✓	Стандардна д	✓
$T_0=50, T_{min}=0.1, T_c=0.993$			0.003842973311			0.05539497271		0.05815940796	
$T_0=100, T_{min}=0.1, T_c=0.993$			0.002699018747			0.06352302616		0.0435995764	
$T_0=200, T_{min}=0.1, T_c=0.993$			0.001677851131			0.08226520638		0.1118863668	
$T_0=350, T_{min}=0.1, T_c=0.993$			0.001012861733			0.04740695293		0.05471983569	
$T_0=700, T_{min}=0.1, T_c=0.993$			0.0009722769195			0.05532708785		0.1049452615	
$T_0=350, T_{min}=0.001, T_c=0.993$			0.003079473263			0.02296972252		0.01569593764	
$T_0=350, T_{min}=0.01, T_c=0.993$			0.0005574277122			0.04372286612		0.06054973158	
$T_0=350, T_{min}=0.05, T_c=0.993$			0.0006963264645			0.05011568135		0.06045822599	
$T_0=350, T_{min}=0.1, T_c=0.993$			0.001081368895			0.02043661497		0.02544469153	
$T_0=350, T_{min}=0.5, T_c=0.993$			0.001840695861			0.06000750223		0.06325454272	
$T_0=350, T_{min}=0.1, T_c=0.9$			0.1622135794			1.52562314		1.0618193	
$T_0=350, T_{min}=0.1, T_c=0.95$			0.01785473055			1.006841463		1.751816113	
$T_0=350, T_{min}=0.1, T_c=0.98$			0.02079540403			0.1112161502		0.1004885455	
$T_0=350, T_{min}=0.1, T_c=0.993$			0.0002692983103			0.06407798417		0.0781728861	
$T_0=350, T_{min}=0.1, T_c=0.997$			0.0005144875441			0.007624456206		0.009472591417	

Резултати показују да већа иницијална температура ( $T_0$ ) побољшава резултате јер омогућава алгоритму да ефикасније избегне локалне минимуме карактеристичне за Rosenbrock функцију, али прекомерно повећање (нпр.  $T_0 = 700$ ) не доноси значајно побољшање у односу на  $T_0$

= 350. Код минималне температуре ( $T_{\min}$ ) најбољи резултати су добијени за вредности око 0.01–0.05, јер прениске вредности успоравају процес, а превисоке доводе до прераног заустављања алгорита. Највећи утицај има фактор хлађења ( $T_c$ ) — ниске вредности (0.9–0.95) узрокују нагло хлађење и слаба решења, док спорије хлађење ( $T_c = 0.993$ –0.997) омогућава стабилнију и прецизнију конвергенцију ка глобалном минимуму. Оптимална комбинација параметара за Rosenbrock функцију је  $T_0 = 350$ ,  $T_{\min} = 0.01$ ,  $T_c = 0.997$ , што обезбеђује најнижу фитнес вредност и најбољу стабилност алгорита.

### 5.1.3. Алгоритам адаптивног симулираног жарења – анализа резултата

Приликом тестирања ASA алгоритма на минимизацији Rosenbrock функције обављено је 12 експеримената приликом којих су мењани параметри почетна температура прихватања, брзина хлађења, интервал за reannealing и максималан број итерација и анализиран је њихов утицај на резултате.

Табела 4 - Резултати минимизације Rosenbrock функције применом ASA

Параметри	Максимум	Минимум	Средња вре	Стандардна
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	0.07112837553	0.002770648091	0.02199831723	0.02023877963
T0=1.00, c=1.00, T0_cost=1.00, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	0.1303874669	0.005307666008	0.03222667121	0.03651299792
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	0.2009703391	0.001356787406	0.03837226904	0.0605007145
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	0.09523674218	0.001285221177	0.03524343579	0.03332462144
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=10000	0.0799259442	0.000986898418	0.02680399772	0.02299795054
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=10000	0.1029326792	0.000379200922	0.03368926791	0.03442096311
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=100, max_iter=10000	0.08788157185	0.0001066562031	0.03259355292	0.03047952239
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=10000	0.06830659043	0.0005606260345	0.02648103333	0.02420877147
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=500, max_iter=10000	0.09667507756	0.00152731671	0.03594307526	0.02715597654
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=5000	0.1778859607	0.002951697142	0.06619471367	0.06871684518
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=10000	0.1240508084	0.004477541901	0.0523168863	0.04300223441
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=20000	0.03811940771	0.0004114841181	0.01546116364	0.01222547709

Први параметар који је разматран јесте почетна температура прихватања. Најнижа осцилација у решењима добијена је када је за вредност почетне температуре прихватања узета вредност 0.5, док је најнижа вредност добијена када је за почетну температуру прихватања стављена вредност 2. Након тога разматран је фактор који утиче на брзину хлађења, његовим повећањем добијене су боље вредности мера од значаја. Следећи разматран параметар јесте интервал за reannealing, најбоље вредности добијане су када је био постављен на 100 и 200. Напошетку посматран је максималан број итерација и његовим повећањем побољшале су се и добијени резултати.

Свеукупно коришћењем ASA алгоритма на минимизацији Rosenbrock функције добијене су задовољавајуће вредности.

### 5.1.4. Поређење алгоритама – минимизација Rosenbrock функције

Rosenbrock функција је представљала добар тест за упоређивање прецизности и стабилности три алгорита – GA, SA и ASA. Генетички алгоритам (GA) се показао као врло поуздан у постизању стабилних решења: повећањем броја генерација, величине популације и броја елитних јединки добијане су ниже вредности функције и мања стандардна девијација. Алгоритам симулираног жарења (SA) такође је дао задовољавајуће резултате, при чему су најбољи исходи постигнути при умереној почетној температури ( $T_0 = 350$ ) и спором хлађењу ( $T_c = 0.997$ ), што је омогућило избегавање локалних минимума и стабилну конвергенцију. Адаптивни SA (ASA) дао је најбржу конвергенцију захваљујући адаптивном подешавању температуре и reannealing механизма,

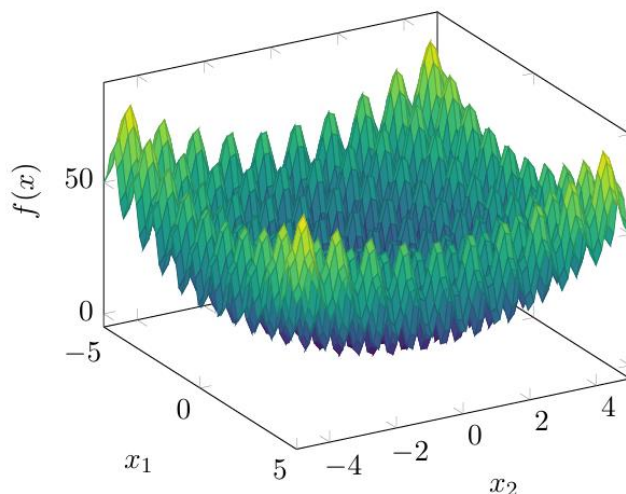
чиме је побољшана прецизност у односу на класични SA. Свеукупно, ASA је био најпрецизнији, SA најробуснији, а GA најстабилнији у смислу конзистентности резултата.

## 5.2. Функција Rastrigin

Математички се дефинише као:

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$$

Rastrigin функција је посебно популарна због свог великог броја локалних минимума, што ствара „таласасту“ површину и отежава алгоритмима да избегну заглављивање у локалним минимумима.



Слика 5 – Функција Rastrigin

### 5.2.1. Генетички алгоритам – анализа резултата

Приликом тестирања генетичког алгоритма на минимизацији Rastrigin функције изведено је 25 експеримената који су приказани у табели 5. Приликом покретања мењани су параметри величина популације, максималан број генерација, вероватноћа укрштања, вероватноћа мутације и број елитних хромозома и разматран је њихов утицај на резултате.

Табела 5 - Резултати минимизације Rastrigin функције применом GA

Параметри	Максимум	#	Минимум	Средња вреднос	Стандардна деви
n=50,gens=100,cross=90%,mut=0.020,elite=5	0.3655424535	0.003459132416	0.1300581681	0.1278749064	
n=100,gens=100,cross=90%,mut=0.020,elite=5	0.2903532592	0.0001197604758	0.0561023757	0.09131754366	
n=200,gens=100,cross=90%,mut=0.020,elite=5	0.08067842186	0.000002936941755	0.02583190746	0.03142187386	
n=400,gens=100,cross=90%,mut=0.020,elite=5	0.0095613453	0.0001997493907	0.002005238715	0.002755913677	
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.003423829327	0.00002346101832	0.001403830637	0.001177515536	
n=800,gens=50,cross=90%,mut=0.020,elite=5	0.0371980171	0.0001570187872	0.007127279374	0.01124729046	
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.00370904059	0.0001935067482	0.001006608006	0.001096938519	
n=800,gens=200,cross=90%,mut=0.020,elite=5	0.001374954958	0.000004281839068	0.0003331813473	0.0004370797752	
n=800,gens=400,cross=90%,mut=0.020,elite=5	0.0002280087049	0.000004699609541	0.00007573121471	0.00007437224466	
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.00001116951929	0.000002296036001	0.000005455172183	0.000003303085881	
n=800,gens=800,cross=50%,mut=0.020,elite=5	0.0001755928384	0.000001811041738	0.00006339972856	0.00006611859075	
n=800,gens=800,cross=70%,mut=0.020,elite=5	0.00004906295084	0.000002220699201	0.00002542704246	0.00001807254296	
n=800,gens=800,cross=80%,mut=0.020,elite=5	0.00001096970621	0.0000001920476791	0.000004461093407	0.000003535092594	
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.00002115003948	0.0000001832355494	0.000008653258395	0.000007890688698	
n=800,gens=800,cross=100%,mut=0.020,elite=5	0.0002037842632	0.0000007872429997	0.00004762326978	0.00006590931667	
n=800,gens=800,cross=80%,mut=0.010,elite=5	0.00006696608749	0.0000001192789121	0.00001281113015	0.00002092470561	
n=800,gens=800,cross=80%,mut=0.030,elite=5	0.00001688708748	0.00000101667213	0.000006304718676	0.00000475592771	
n=800,gens=800,cross=80%,mut=0.050,elite=5	0.00002356695462	0.0000001913002983	0.000007425392829	0.000007071260469	
n=800,gens=800,cross=80%,mut=0.100,elite=5	0.0000358897361	0.0000001251417352	0.00001346095051	0.00001270772958	
n=800,gens=800,cross=80%,mut=0.200,elite=5	0.00007170750899	0.000001340142866	0.00002025728801	0.00002509020072	
n=800,gens=800,cross=80%,mut=0.030,elite=1	0.0001700826798	0.000003662647526	0.00003846431764	0.00005638740274	
n=800,gens=800,cross=80%,mut=0.030,elite=2	0.0002175411065	0.0000001009261581	0.00003030250475	0.00006688707351	
n=800,gens=800,cross=80%,mut=0.030,elite=5	0.00003234445569	0.0000005586701306	0.00001438433148	0.00001108939275	
n=800,gens=800,cross=80%,mut=0.030,elite=10	0.00002270819834	0.00000000700595848	0.000005065147305	0.000006762701263	
n=800,gens=800,cross=80%,mut=0.030,elite=20	0.00002205030294	0.0000001257262952	0.000004695114927	0.000006626744154	

Први разматран параметар јесте мењање величине популације, њеним повећањем добијане су све бољи резултати. Следећи параметар чији смо утицај пратили јесте број генерација, као и код броја јединки, тако је и код броја генерација њихово повећање позитивно утицало на рад алгоритма и побољшало добијене вредности. Затим је разматран параметар crossover, приликом вредности crossover параметра од 80% добијени су најбољи резултати. Када је за параметар мутација узета вредност 0.01 добијена је најнижа вредност минимума, док је алгоритам био најстабилнији када је

вредност мутације била 0.03. На послетку разматран је број елитних јединки, када је за број елитних јединки узета вредност 10 добијена је најнижа вредност приликом свих 25 експеримената.

Генерално су резултати добијени тестирањем генетичког алгоритма на Rastrigin функцији добри, чиме можемо да закључимо да је алгоритам погодан за њену минимизацију.

### 5.2.2. Алгоритам симулираног жарења – анализа резултата

Приликом тестирања алгоритма симулираног жарења на минимизацији Rastrigin функције изведено је 15 експеримената који су приказани у табели 6. Приликом покретања мењани су параметри иницијална температура  $T_0$ , минимална температура, фактор хлађења и разматран је њихов утицај на резултате.

Табела 6 - Резултати минимизације Rastrigin функције применом SA

Параметри	Најбоља	Средња вред	Стандардн
$T_0=50, T_{min}=0.1, T_c=0.993$	0.2179876631	0.5851196764	0.2649224158
$T_0=100, T_{min}=0.1, T_c=0.993$	0.02431538079	0.746045157	0.6029963939
$T_0=200, T_{min}=0.1, T_c=0.993$	0.2453500981	0.7916419424	0.414598179
$T_0=350, T_{min}=0.1, T_c=0.993$	0.3461008403	0.989730691	0.6256638584
$T_0=700, T_{min}=0.1, T_c=0.993$	0.05364669286	0.8321382454	0.5765440626
$T_0=50, T_{min}=0.001, T_c=0.993$	0.02228591145	0.553985433	0.4144871033
$T_0=50, T_{min}=0.01, T_c=0.993$	0.1064912466	0.5346874261	0.427683104
$T_0=50, T_{min}=0.05, T_c=0.993$	0.1170579924	0.6424153337	0.4119787309
$T_0=50, T_{min}=0.1, T_c=0.993$	0.08385854607	0.6421758342	0.6693446039
$T_0=50, T_{min}=0.5, T_c=0.993$	0.9139552865	1.290286547	0.342905637
$T_0=50, T_{min}=0.01, T_c=0.9$	0.06819948933	4.634909441	3.166738045
$T_0=50, T_{min}=0.01, T_c=0.95$	0.02142732836	2.034494551	1.530230946
$T_0=50, T_{min}=0.01, T_c=0.98$	0.1717988381	0.7939437131	0.5605863461
$T_0=50, T_{min}=0.01, T_c=0.993$	0.2720428972	0.8399483063	0.3262758556
$T_0=50, T_{min}=0.01, T_c=0.997$	0.07211928782	0.267177018	0.1267999328

Резултати за Rastrigin функцију показују да избор параметара у алгоритму симулираног жарења (SA) значајно утиче на коначну тачност и стабилност решења. Код промене иницијалне температуре ( $T_0$ ), најниже вредности фитнес функције постигнуте су за  $T_0 = 100$  (најбоља  $\approx 0.024$ ), што указује да умерена почетна температура омогућава добро истраживање простора решења без



превелике случајности. Прениска температура ( $T_0 = 50$ ) смањује могућност изласка из локалних минимума, док превисоке вредности ( $T_0 = 350, 700$ ) уводе сувише шума и успоравају конвергенцију. При промени минималне температуре ( $T_{min}$ ) најбољи резултат добијен је за  $T_{min} = 0.001$  (најбоља  $\approx 0.022$ ), што значи да спорије хлађење до ниже граничне температуре омогућава алгоритму даfino доради решење. Превелике вредности  $T_{min}$  (нпр. 0.5) прекидају процес пре него што се постигне стабилна конвергенција. Најзначајнији утицај има фактор хлађења ( $T_c$ ) — ниске вредности (0.9–0.95) доводе до наглог хлађења и високих средњих фитнес вредности, што значи да алгоритам „замрзава“ прерано, док спорије хлађење ( $T_c = 0.997$ ) омогућава најбољу прецизност и стабилност (најбоља  $\approx 0.072$ , средња  $\approx 0.267$ ). Укупно гледано, оптимална комбинација параметара за Rastrigin функцију је  $T_0 = 100$ ,  $T_{min} = 0.001$ , и  $T_c = 0.997$ , јер обезбеђује најнижу фитнес вредност и најравномерније смањење варијације решења.

### 5.2.3. Алгоритам адаптивног симулираног жарења – анализа резултата

Приликом тестирања ASA алгоритма на минимизацији Rastrigin функције обављено је 12 експеримената приликом којих су мењани параметри су почетна температура прихватања, брзину хлађења, интервал за reannealing и максималан број итерација и анализиран је њихов утицај на резултате.

Табела 7 - Резултати минимизације Rastrigin функције применом ASA

Параметар	Максимум	Минимум	Средња вр	Стандардн
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=200, max\_iter=10000$	1.131568652	0.106790455	0.5733202261	0.4105169412
$T_0=1.00, c=1.00, T_0\_cost=1.00, c\_cost=1.00, Q_i=1.00, reanneal=200, max\_iter=10000$	1.224772328	0.01840829158	0.6202484605	0.4041363967
$T_0=1.00, c=1.00, T_0\_cost=2.00, c\_cost=1.00, Q_i=1.00, reanneal=200, max\_iter=10000$	1.623482792	0.2209669874	0.7467034831	0.5211932746
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=200, max\_iter=10000$	1.166181089	0.01502273046	0.405757462	0.4066834535
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=2.00, reanneal=200, max\_iter=10000$	1.316338865	0.05323489365	0.7560570226	0.4794707446
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=5.00, reanneal=200, max\_iter=10000$	1.123517658	0.0766141983	0.550402075	0.4029860822
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=100, max\_iter=10000$	1.546141603	0.009123322018	0.6224371588	0.5327508489
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=200, max\_iter=10000$	1.106537753	0.01228788696	0.4480301266	0.3415926757
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=500, max\_iter=10000$	0.9568926021	0.1574458065	0.3688169437	0.2653958696
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=500, max\_iter=5000$	1.473058385	0.4827329408	1.0642117	0.2959839208
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=500, max\_iter=10000$	1.110649858	0.01008760603	0.4719121012	0.410666047
$T_0=1.00, c=1.00, T_0\_cost=0.50, c\_cost=1.00, Q_i=1.00, reanneal=500, max\_iter=20000$	0.8409849226	0.006183670078	0.2108649687	0.2468944804

Уколико посматрамо утицај почетне температуре прихватања на резултате добијене приликом минимизације Rastrigin функције, можемо уочити да је најнижи минимум достигнут када је вредност параметра била 1, док је средња вредност била најнижа када је вредност параметра била 0.5. Следећи разматран параметар јесте брзина хлађења, његовим повећањем благо су се погоршали и добијени резултати. Повећањем интервала за reannealing по стандардној девијацији можемо уочити да се алгоритам све више стабилизовао. Најнижу вредност приликом свих покретања алгоритма добили смо разматрајући максималан број итерација и то у експерименту у који смо као максималан број итерација усвојили 20000.

Тестирањем ASA алгоритма на минимизацији Rastrigin функције добијени су мало лошији резултати, него његовом применом на минимизацији Rosenbrock функције.

#### 5.2.4. Поређење алгоритама – минимизација Rastrigin функције

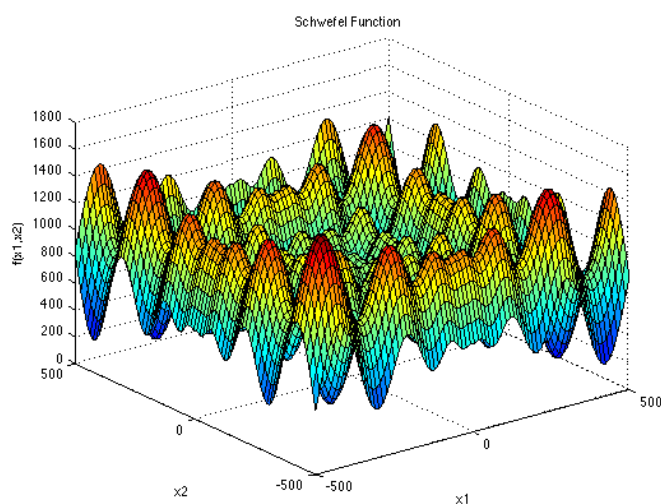
Rastrigin функција, због великог броја локалних минимума, представља тежак тест за алгоритме глобалне оптимизације. GA је успешно минимизовао функцију, показујући најбољи баланс између истраживања и експлоатације простора решења. Најбољи резултати постигнути су са crossover параметром од 80%, мутацијом 0.01 и 10 елитних јединки. SA је показао добре перформансе када су параметри били пажљиво изабрани – најбоља комбинација је  $T_0 = 100$ ,  $T_{min} = 0.001$  и  $T_c = 0.997$ , што је омогућило најпрецизније и најстабилније резултате. ASA је, иако задржао стабилност, постигао нешто слабије резултате у односу на Rosenbrock функцију, при чему је утицај параметра хлађења био осетљивији. Генерално, GA је дао најбољи укупан резултат на Rastrigin функцији, док је SA био ефикасан при пажљиво подешеним параметрима, а ASA је имао нешто већу осцилацију вредности.



### 5.3. Функција Schwefel

Schwefel функција је типична мултимодална бенчмарк функција са великим бројем локалних минимума. Математички се функција дефинише као:

$$f(x) = 418.9820d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|})$$



Слика 6 – Функција Schwefel

### 5.3.1. Генетички алгоритам – анализа резултата

Приликом тестирања генетичког алгоритма на минимизацији Schwefel функције изведено је 25 експеримената који су приказани у табели 8. Приликом покретања мењани су параметри величина популације, максималан број генерација, вероватноћа укрштања, вероватноћа мутације и број елитних хромозома и разматран је њихов утицај на резултате.

Табела 8 - Резултати минимизације Schwefel функције применом GA

Параметри	Максимум	# Минимум	Средња вредно	Стандардна девија
n=50,gens=100,cross=90%,mut=0.020,elite=5	0.3529926791	0.00002599136997	0.06311161549	0.1204415214
n=100,gens=100,cross=90%,mut=0.020,elite=5	0.1114498616	0.0000314731567	0.01156156437	0.03510097793
n=200,gens=100,cross=90%,mut=0.020,elite=5	0.0006520729681	0.00002574988639	0.0002153648149	0.000239971911
n=400,gens=100,cross=90%,mut=0.020,elite=5	0.0006655367257	0.00005715659893	0.000216134085	0.0002036978599
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.0003733544172	0.00002591361033	0.00007779027279	0.0001053749363
n=800,gens=50,cross=90%,mut=0.020,elite=5	0.007919237011	0.00003608273528	0.00228328438	0.002839314599
n=800,gens=100,cross=90%,mut=0.020,elite=5	0.0002083746705	0.00002570227491	0.00008064676568	0.00005797771613
n=800,gens=200,cross=90%,mut=0.020,elite=5	0.0000474712424	0.00002631714585	0.00003438712359	0.000006984032553
n=800,gens=400,cross=90%,mut=0.020,elite=5	0.00002683869729	0.00002546140365	0.00002584741123	0.00000045554955
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.0000264821366	0.00002547875897	0.000025807389	0.0000003649778177
n=800,gens=800,cross=50%,mut=0.020,elite=5	0.0001119927624	0.00002815430366	0.00006688468143	0.00002798956456
n=800,gens=800,cross=70%,mut=0.020,elite=5	0.00006146474186	0.00002566105718	0.00003359027427	0.0000120939457
n=800,gens=800,cross=80%,mut=0.020,elite=5	0.00003150709745	0.00002546715439	0.00002630280247	0.000001856070042
n=800,gens=800,cross=90%,mut=0.020,elite=5	0.00002707513522	0.00002549382486	0.00002589281415	0.0000005299225914
n=800,gens=800,cross=100%,mut=0.020,elite=5	0.00002554910748	0.00002547034467	0.00002549891343	0.00000003101877056
n=800,gens=800,cross=100%,mut=0.010,elite=5	0.00002694027592	0.00002545838197	0.00002574838697	0.000000459604614
n=800,gens=800,cross=100%,mut=0.030,elite=5	0.00002598720653	0.00002545554935	0.00002557363356	0.0000001835767893
n=800,gens=800,cross=100%,mut=0.050,elite=5	0.00002559558231	0.00002545557618	0.00002548248284	0.00000004657746976
n=800,gens=800,cross=100%,mut=0.100,elite=5	0.00002561326778	0.00002545717473	0.0000254850194	0.00000004670970783
n=800,gens=800,cross=100%,mut=0.200,elite=5	0.00002556346806	0.00002545531856	0.00002548465377	0.00000004008400752
n=800,gens=800,cross=100%,mut=0.050,elite=1	0.00002731835366	0.00002546509279	0.00002592918855	0.0000005612969947
n=800,gens=800,cross=100%,mut=0.050,elite=2	0.00002575761846	0.00002545547977	0.00002549589835	0.00000009261898398
n=800,gens=800,cross=100%,mut=0.050,elite=5	0.00002562644397	0.00002545579946	0.00002549816929	0.00000005127359491
n=800,gens=800,cross=100%,mut=0.050,elite=10	0.00002546945097	0.00002545703296	0.00002546102182	0.00000000423145057
n=800,gens=800,cross=100%,mut=0.050,elite=20	0.00002549171018	0.00002545520158	0.00002546050804	0.00000001107220513

Повећањем величине популације добијане су бољи резултати. Мањи број генерација је лоше утицао на резултате, док су се вредности поправљале његовим повећањем. Мењање вероватноће укрштања, мутације и броја елитних јединки није у великој мери утицало на резултате, вредности су остале приближно сличне.

Генетички алгоритам се показао погодним за минимизацију још једне тест функције.

### 5.3.2. Алгоритам симулираног жарења – анализа резултата

Приликом тестирања алгоритма симулираног жарења на минимизацији Schwefel функције изведено је 15 експеримената који су приказани у табели 9. Приликом покретања мењани су параметри иницијална температура  $T_0$ , минимална температура, фактор хлађења и разматран је њихов утицај на резултате.

Табела 9 - Резултати минимизације Schwefel функције применом SA

Параметри	#	Најбоља	Средња вред	Стандардн
$T_0=50, T_{min}=0.1, T_c=0.993$	0.000428142479	396.1900435	229.4192851	
$T_0=100, T_{min}=0.1, T_c=0.993$	118.4448076	390.1069901	162.2838195	
$T_0=200, T_{min}=0.1, T_c=0.993$	118.4384794	387.4762709	186.6672486	
$T_0=350, T_{min}=0.1, T_c=0.993$	118.4384467	303.6843313	157.1396137	
$T_0=700, T_{min}=0.1, T_c=0.993$	0.001405644905	349.4281077	223.3623591	
$T_0=350, T_{min}=0.001, T_c=0.993$	0.0003876010934	354.8177943	184.4456149	
$T_0=350, T_{min}=0.01, T_c=0.993$	217.1397732	539.5928068	152.6485119	
$T_0=350, T_{min}=0.05, T_c=0.993$	0.001239777992	319.804457	208.2636945	
$T_0=350, T_{min}=0.1, T_c=0.993$	118.4387109	417.3210147	178.6583016	
$T_0=350, T_{min}=0.5, T_c=0.993$	0.002741368202	404.7110179	182.6797454	
$T_0=350, T_{min}=0.05, T_c=0.9$	335.5288518	550.4583785	185.0052257	
$T_0=350, T_{min}=0.05, T_c=0.95$	57.19478579	576.8674782	295.3097969	
$T_0=350, T_{min}=0.05, T_c=0.98$	0.006451163603	428.036687	235.0689241	
$T_0=350, T_{min}=0.05, T_c=0.993$	236.8770529	485.3066555	116.8655514	
$T_0=350, T_{min}=0.05, T_c=0.997$	118.4385133	395.9626581	146.968014	

Резултати показују да иницијална температура ( $T_0$ ) има значајан утицај на успешност SA алгоритма – прениска ( $T_0=50$ ) и превисока ( $T_0=700$ ) вредност могу дати добра решења, али превелика температура повећава варијабилност. Најстабилнији резултати су постигнути за умерене вредности  $T_0$  (100–200). Минимална температура ( $T_{min}$ ) утиче на завршну фазу хлађења – најмање вредности (0.001–0.05) омогућавају прецизнију локалну претрагу и боља решења, док веће  $T_{min}$  (нпр. 0.5) доводе до слабије конвергенције. Фактор хлађења ( $T_c$ ) показује да спорије хлађење ( $T_c=0.98$ – $0.993$ ) даје најбоље резултате јер омогућава довољно итерација за истраживање простора решења, док пребрзо хлађење ( $T_c=0.9$ ) значајно погоршава квалитет. Укупно, најбољи баланс перформанси постиже се при  $T_0 \approx 200$ ,  $T_{min} \approx 0.001$ – $0.05$  и  $T_c \approx 0.98$ – $0.993$ .

### 5.3.3. Алгоритам адаптивног симулираног жарења – анализа резултата

Приликом тестирања ASA алгоритма на минимизацији Schwefel функције обављено је 12 експеримената приликом којих су мењани параметри су почетна температура прихватања, брзину хлађења, интервал за reannealing и максималан број итерација и анализиран је њихов утицај на резултате.

Табела 10 - Резултати минимизације Schwefel функције применом ASA

Параметри	Максимум	Минимум	Средња вр	Стандардн
T0=1.00, c=1.00, T0_cost=0.50, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	26.06570032	0.8728858921	4.937937191	7.634386006
T0=1.00, c=1.00, T0_cost=1.00, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	11.27400318	0.09341519449	5.692903785	4.058205216
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	6.441283472	0.1736753785	3.333750058	2.778571907
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=1.00, reanneal=200, max_iter=10000	25.7163318	0.04576619751	6.515525359	8.87418833
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=2.00, reanneal=200, max_iter=10000	12.57810186	1.057353579	4.569052379	3.363235837
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=10000	10.58658957	0.9600032501	3.319973336	2.99798561
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=100, max_iter=10000	19.52167	0.4639543885	5.427031956	6.09955775
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=10000	12.57617516	0.7286880771	4.03443197	3.879929054
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=500, max_iter=10000	14.04593827	0.01749903948	4.568227239	5.134619468
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=5000	23.1995993	0.3706479285	5.608443289	6.67355865
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=10000	15.42293858	0.1040787644	4.524798247	4.812179912
T0=1.00, c=1.00, T0_cost=2.00, c_cost=1.00, Qi=5.00, reanneal=200, max_iter=20000	9.459824044	0.1115649863	3.147776906	2.621669069

ASA алгоритам се нешто лошије показао приликом минимизације Schwefel функције, у односу на минимизацију осталих тест функција. По високој стандардној девијацији и великој разлици између максимума и минимума видимо да је алгоритам пуно осциловао и да није био толико стабилан.

### 5.3.4. Поређење алгоритама – минимизација Schwefel функције

Schwefel функција је мултимодална и веома изазовна за све три методе, због великог броја локалних минимума. Генетички алгоритам је и у овом случају био стабилан, са побољшањем резултата при већим популацијама и бројевима генерација. SA је постигао најбољи баланс између прецизности и стабилности при умереним вредностима параметара ( $T_0 \approx 200$ ,  $T_{min} \approx 0.001-0.05$ ,  $T_c \approx 0.98-0.993$ ), што му је омогућило прецизнију локалну претрагу без прераног заустављања. ASA, међутим, није постигао тако стабилну конвергенцију као код претходне две функције — резултати су имали високу стандардну девијацију и осцилације, што указује на већу осетљивост алгоритма на параметре. Укупно гледано, SA је био најуспешнији за Schwefel функцију, GA стабилан али мање прецизан, док је ASA показао мању поузданост.

## 6. Закључак

У овом раду извршена је компаративна анализа три алгорита стохастичке оптимизације — генетичког алгорита (GA), алгорита симулираног жарења (SA) и адаптивног алгорита симулираног жарења (ASA) — применом на три стандардне тест функције: Rosenbrock, Rastrigin и Schwefel. Циљ је био да се упореде њихове перформансе у смислу брзине конвергенције, стабилности и способности проналажења глобалног минимума.

Резултати су показали да сваки од алгорита има своје предности у зависности од типа функције. Генетички алгоритам се показао као најстабилнији и најпоузданији у свим случајевима, са конзистентним резултатима и добром прецизношћу при правилном избору параметара. Алгоритам симулираног жарења је показао изразиту флексибилност и способност избегавања локалних минимума, нарочито при пажљивом подешавању температуре и фактора хлађења. ASA алгоритам, као унапређена верзија SA, дао је најбоље резултате код глатких функција попут Rosenbrock-а, где је адаптивно хлађење убрзало конвергенцију, али је показао већу осетљивост и нестабилност на комплексним функцијама као што је Schwefel.

Свеукупно, може се закључити да нема универзално најбољег алгорита — њихова ефикасност зависи од природе проблема и облика функције циља. GA обезбеђује стабилност и робусност, SA добро балансира између глобалне и локалне претраге, док ASA постиже високу прецизност код континуалних функција уз бржу конвергенцију. Добијени резултати потврђују да комбиновање и прилагођавање ових метода може представљати најефикаснији приступ при решавању сложених инжењерских оптимизационих проблема у оквиру концепта Индустрије 4.0.

## 7. Литература

[1] Миљковић З., Петровић М., Интелигентни технолошки системи – са изводима из роботике и вештачке интелигенције, Универзитет у Београду, Машински факултет, 2021.

[2] Ingber, L. (1995). Adaptive Simulated Annealing (ASA): Lessons Learned. Control and Cybernetics, Polish Academy of Sciences. Lester Ingber Research, McLean, VA, USA. Available at:  
<https://arxiv.org/abs/cs/0001018>

[3] Surjanovic, S., & Bingham, D. (2013). Schwefel Function. Simon Fraser University. Available at:  
<https://www.sfu.ca/~ssurjano/schwef.html>

[4] Surjanovic, S., & Bingham, D. (2013). Rastrigin Function. Simon Fraser University. Available at:  
<https://www.sfu.ca/~ssurjano/rastr.html>

[5] Surjanovic, S., & Bingham, D. (2013). Rosenbrock Function. Simon Fraser University. Available at:  
<https://www.sfu.ca/~ssurjano/rosen.html>