

Приложение на Whale Optimization Algorithm (WOA) за параметрична идентификация на модел на полупериодична култивация на *Escherichia coli* MC4110

Любомир Чолаков, ф.н. 8MI3400693

Цветан Цветанов, ф.н. 4MI3400570

Изкуствен интелект

29 май 2025 г.



Съдържание

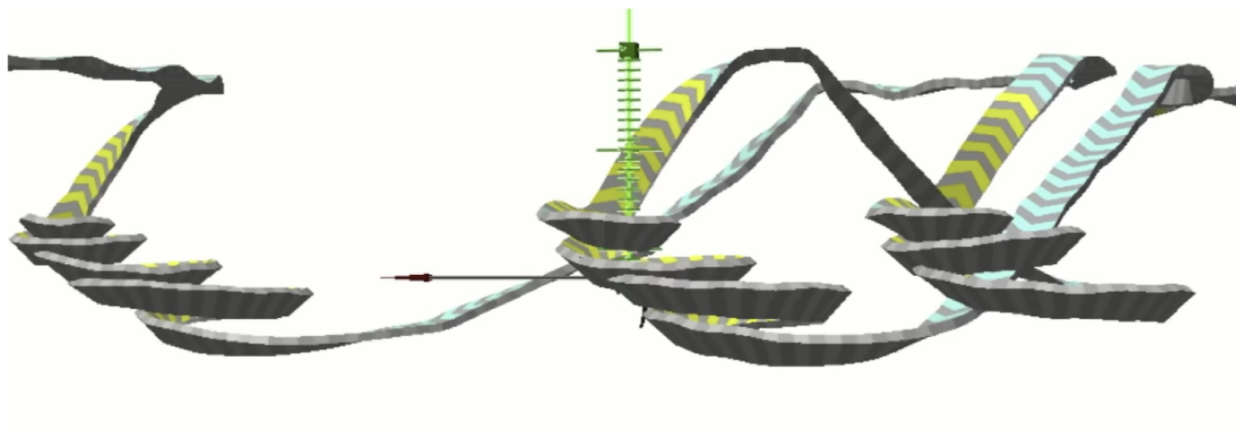
1	Описание на Whale Optimization Algorithm (WOA)	3
1.1	Псевдокод	4
1.2	Формули	4
1.2.1	Формула за обкръжаване на плячката (промяна на позицията на агента)	4
1.2.2	Формула за експлоатация (атакуване на плячка)	4
1.2.3	Формула за ескплорация (търсене на плячка)	5
2	Описание на задачата за параметрична идентификация	6
2.1	Култивационен процес и налични експериментални данни	6
2.2	Математически модел	6
2.3	Коефициенти на идентификация	6
2.4	Целева функция	7
3	Приложение на WOA за задачата за параметрична идентификация	8
3.1	Представяне на решението в термините на метаевристичния алгоритъм	8
3.2	Избор на параметри на метаевристичния алгоритъм	8
3.3	Запис на резултатите	9
3.4	Представяне на резултатите	10
3.4.1	Таблично	10
3.4.2	Графично	11
3.5	Анализ на резултатите	14
3.5.1	Постигнат най-добър (най-точен) модел	14
4	Сравнение на получените резултати от WOA с генетичен алгоритъм (GA), алгоритъм на симулираното закаляване (SA) и алгоритъм на гарвана (CSA)	15
4.1	Представяне на резултатите	15
4.2	Анализ на получените резултати	19
4.2.1	Оптимални стойности на целевата функция	19
4.2.2	Оценка на статистическата значимост на разликите в резултатите на алгоритмите	19
5	Интеркритериален анализ (ИКА) на получените оценки на параметрите на модела от идентификацията с различните алгоритми	20
5.1	Представяне на резултатите от ИКА за всеки алгоритъм	20
5.1.1	Genetic Algorithm (GA)	20
5.1.2	Simulated Annealing (SA)	21
5.1.3	Crow Search Algorithm (CSA)	22
5.1.4	Whale Optimization Algorithm (WOA)	23
5.2	Анализ на резултатите	24
6	Източници и използвана литература	25

1 Описание на Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm, или китовият оптимизационен алгоритъм, е популяционен алгоритъм вдъхновен от природата, който цели да моделира поведението за лов на гърбатите китове за намиране на решението на оптимизационни задачи. Алгоритъмът е представен за пръв път през 2015г от Seyedali Mirjalili и Andrew Lewis в 95-ото издание на журнала Advances in Engineering Software.

Гърбатите китове проявяват поведение, което не се среща другаде в природата. То се изразява в създаването на мрежи от балончета около плячката, за да оптимизират храненето си. Главните характеристики в поведението на гърбатите китове по време на лов са:

- Ловуват пасажи крил и дребни риби
- Гмуркат се до 12м дълбочина
- Създават мрежи от балончета под формата на кръгове или фигура 9 (спирала)
- Мрежите от балончета обграждат пасажа в ограничено пространство
- Китовете изплуват по средата с отворена уста



Фигура 1: Графика на спираловидното движение на китовете

1.1 Псевдокод

Алгоритъм 1 Псевдокод на китовия оптимизационен алгоритъм (WOA)

```
1: Инициализирай популацията от китове  $X_i, i = 1, 2, \dots, n$ 
2: Изчисли пригодността на всеки агент за търсене
3:  $X^* \leftarrow$  най-добрият агент за търсене (кит)
4: докато  $t <$  максимален брой итерации направи
5:   за всеки агент за търсене направи
6:     Актуализирай  $a, A, C, l$  и  $p$ 
7:     ако  $p < 0.5$  тогава
8:       ако  $|A| < 1$  тогава
9:         Актуализирай позицията на текущия агент по уравнение (2)
10:      в противен случай
11:        Избери случаен агент за търсене  $X_{\text{rand}}$ 
12:        Актуализирай позицията на текущия агент по уравнение (8)
13:      приключи ако
14:    в противен случай
15:      Актуализирай позицията на текущия агент по уравнение (5)
16:    приключи ако
17:  приключи за
18:  Провери дали някой агент излиза извън пространството за търсене и го коригирай
19:  Изчисли пригодността на всеки агент за търсене
20:  Актуализирай  $X^*$ , ако има по-добро решение
21:   $t \leftarrow t + 1$ 
22: приключи докато
23: върни  $X^*$ 
```

1.2 Формули

1.2.1 Формула за обкръжаване на плячката (промяна на позицията на агента)

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}^*(t)| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (2)$$

1.2.2 Формула за експлоатация (атакуване на плячка)

$$\vec{X}(t+1) = \vec{D}' e^{bl} \cos(2\pi l) + \vec{X}^*(t) \quad (5)$$

1.2.3 Формула за ескplorация (търсене на плячка)

$$\vec{X}(t+1) = \overrightarrow{X_{\text{rand}}} - \vec{A} \cdot \vec{D} \quad (8)$$

Където означенията имат следния смисъл:

- n - брой китове от популацията
- t - текуща итерация
- l - скаларен параметър за спиралата на обкръжаване
- b - константа за формата на спиралата
- \vec{C} - вектор, който контролира колко далеч вижда текущия кит
- \vec{A} - вектор, който контролира позицията на даден кит спрямо оптималното решение
- \vec{D} - вектор, който пази позицията между текущия кит и друг кит или най-добрата позиция
- $\overrightarrow{X_{\text{rand}}}$ - случайно избран кит
- \vec{X} - текущо избран кит

2 Описание на задачата за параметрична идентификация

Настоящият труд разглежда производителността на Whale Optimization Algorithm, съпоставена с тази на три широко използвани алгоритъма - Genetic Algorithm (GA), Simulated Annealing (SA), Crow Search Algorithm (CSA) при идентифициране на параметри на математически модел. Изследва математическия модел на процес на култивиране на *Escherichia coli* с периодично подхранване.

2.1 Култивационен процес и налични експериментални данни

Разглежда се процес на култивация на *E. coli* като са налични експериментални данни представени в файлове es1807ac (ацетат), es1807gs (газове), es1807k и es1807kf (калмачни филтри).

2.2 Математически модел

Разглежда се прост математически модел на ферментацията на бактерията *Escherichia coli*. Скоростите на клетъчен растеж и консумацията на глюкоза (субстрат) обикновено се описват съгласно баланс на масите, както следва:

$$\begin{aligned}\frac{dX}{dt} &= \mu_{\max} \frac{S}{k_S + S} X - \frac{F}{V} X, \\ \frac{dS}{dt} &= -\frac{1}{Y_{S/X}} \mu_{\max} \frac{S}{k_S + S} X + \frac{F}{V} (S_{\text{in}} - S), \\ \frac{dV}{dt} &= F, \\ \mu &= \mu_{\max} \frac{S}{k_S + S}\end{aligned}$$

2.3 Коефициенти на идентификация

Трите параметъра на модела са:

- μ_{\max} максимална скорост на растеж;
- k_s константа на насищане;
- $Y_{S/X}$ икономически коефициент.

2.4 Целева функция

$$J = \sum_{i=1}^n \{ [X_{\text{exp}}(i) - X_{\text{mod}}(i)] \}^2 + \{ [S_{\text{exp}}(i) - S_{\text{mod}}(i)] \}^2$$

Тази функция дефинира грешката, която съответства на разликата между модела и експерименталните данни. X_{exp} отговаря на биомасата в експерименталните данни, X_{mod} е биомасата в нашия модел, а S_{exp} и S_{mod} са съответно субстрата от експерименталните данни и от модела. Целта на метаевристичната оптимизация е да минимизира стойността на целевата функция.

3 Приложение на WOA за задачата за параметрична идентификация

3.1 Представяне на решението в термините на метаевристичния алгоритъм

WOA е стохастичен метаевристичен алгоритъм. Чрез методи на експлоатация и експлорация той намира решения, за които целевата функция приема минимални стойности. Всеки един от китовите съответства на точка от n -мерно пространство, като в тази задача $n = 3$. Всяка итерация е пълно обновяване позициите на всички китове. Всеки кит отговаря на едно потенциално решение, а с всяка итерация се тестват нови решения.

3.2 Избор на параметри на метаевристичния алгоритъм

WOA е сравнително опростен алгоритъм без голям набор от параметри за настройване. Входните параметри, които се използват от кода, изпълняващ алгоритъма ([WOA/main.m](#)), са два - размер на популацията и брой итерации. Разгледани са четири по-специфични постановки, при които може да сравнят както най-бърза сходимост до минимална грешка, така и бързодействие на изпълнение. Проведени са тестове на лаптопи с различни параметри и операционни системи - *Ubuntu Linux 22.04 LTS* и *Mac OSX 15.5*. Алгоритъмът се изпълнява с 30 реализации с цел събиране на необходимите данни за сравнение с други алгоритми и за статистически анализ.

Агенти (бр.)	Итерации (бр.)	Минимална грешка	Средна грешка	Време за изп.(сек)
1000	30	4.7325	4.7376	99
500	100	4.7324	4.7496	190
50	100	4.7331	7.8085	41
300	30	4.7327	5.1474	29

Таблица 1: Резултати с различни входни хиперпараметри

В стандартната си форма WOA се използва в комбинация от 50 агента и 100 итерации. Тестването показва, че дефиниране на повече агенти в дадения проблем не води до значително подобрение на решението. От друга страна, използването на повече агенти води до значително повишаване на консистентността на получаваните резултати, както се вижда от осреднените стойности в таблица 1. След проведени тестове се счита, че комбинация от 300 агента и 30 итерации дава значително подобрение средната грешка и във времето за изпълнение. С оглед на това, са представени за сравнение и анализ тези резултати получени от 30 реализации с 300 агента и 30 итерации.

3.3 Запис на резултатите

След изпълнение на програмата, резултатите от 30-те реализации на алгоритъма се записват в матрица 5 на 30 в матричен файл `.mat`. Запазват се следните стойности:

- минималната грешка от всяка реализация J
- стойностите за всеки от трите параметъра μ_{max} , k_s и $(Y_{S/X})$
- време за изпълнение

Данните от изпълнението както на алгоритъм WOA, така и на алгоритмите GA, SA и CSA, с които се прави съпоставка, са копирани в стандартна таблица `.xlsx` за улеснено сортиране и въвеждане в програмата за анализ.

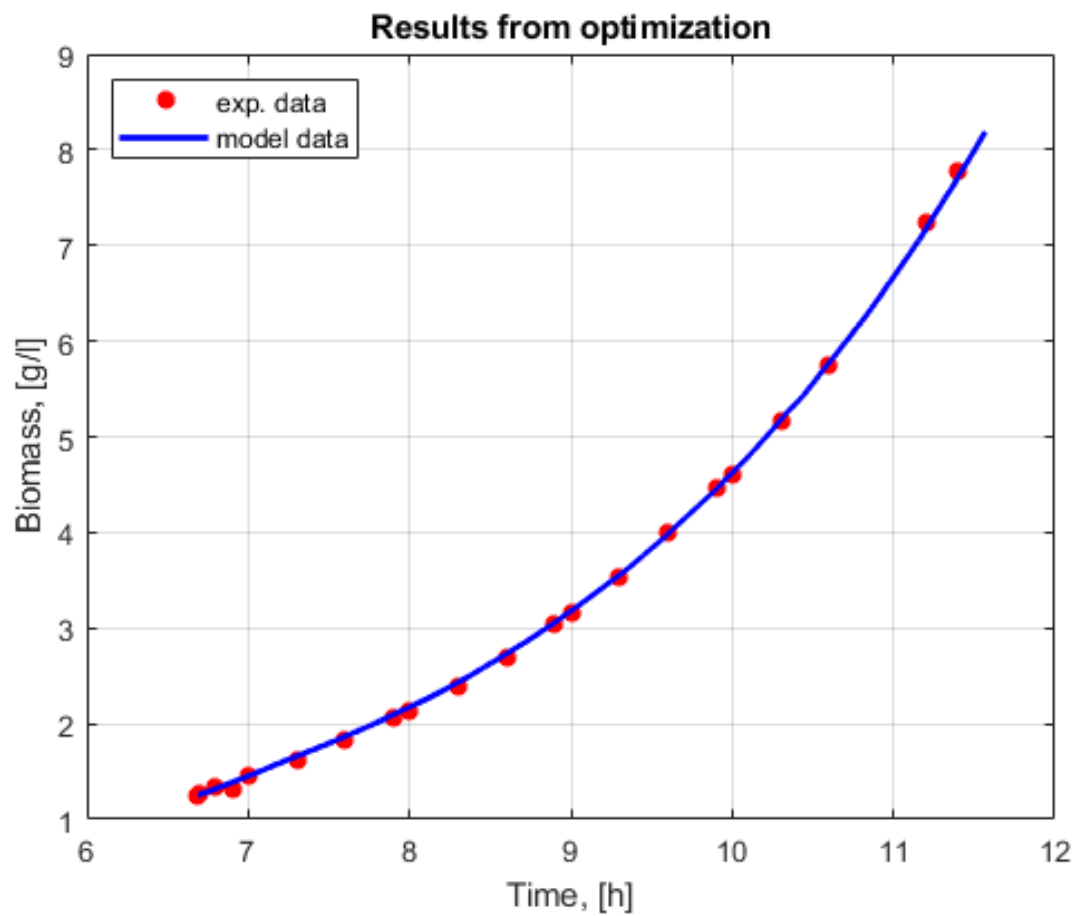
3.4 Представяне на резултатите

3.4.1 Таблично

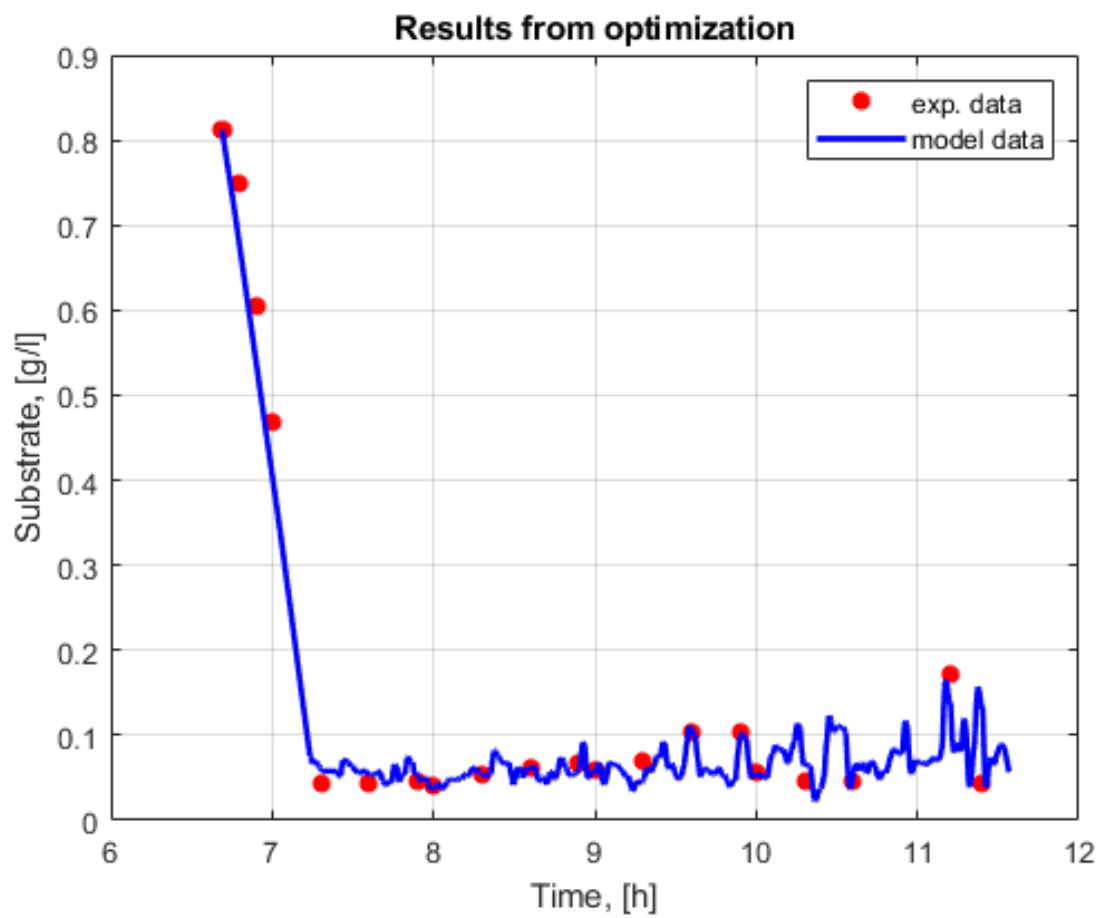
№	J	μ_{max}	k_s	$Y_{S/X}$	Време за изп.(сек)
r01	4.8636	0.50433	0.01562	2.02187	65
r02	7.2802	0.59837	0.03795	2.0186	33
r03	4.7524	0.47737	0.01006	2.02315	33
r04	5.4866	0.54218	0.0213	2.02204	34
r05	5.0931	0.51777	0.01886	2.0208	36
r06	9.3538	0.6478	0.05381	2.0169	33
r07	4.7432	0.48885	0.01241	2.02253	33
r08	4.7327	0.485	0.01144	2.02288	32
r09	4.7466	0.4907	0.01267	2.02254	32
r10	4.7347	0.4871	0.01188	2.02276	32
r11	4.7653	0.49513	0.0133	2.02259	33
r12	5.05	0.46032	0.00648	2.02456	35
r13	4.8654	0.50006	0.01529	2.02161	33
r14	5.666	0.54677	0.02472	2.02013	33
r15	5.1157	0.51898	0.01915	2.02094	32
r16	4.9979	0.51313	0.01767	2.0214	33
r17	5.0389	0.51914	0.01788	2.0219	39
r18	4.7347	0.48728	0.01184	2.02282	33
r19	5.0252	0.51847	0.01738	2.02235	32
r20	4.7885	0.49674	0.01403	2.02223	33
r21	5.1428	0.5213	0.0195	2.02101	31
r22	4.7552	0.49329	0.01296	2.02252	30
r23	4.9208	0.49788	0.01537	2.02182	29
r24	4.7609	0.49326	0.01324	2.0224	30
r25	4.7352	0.48728	0.01175	2.02293	29
r26	4.9844	0.51416	0.01748	2.02167	32
r27	4.9021	0.50593	0.01621	2.02165	31
r28	4.8258	0.49722	0.01458	2.02183	32
r29	4.779	0.47469	0.0093	2.02351	34
r30	4.7824	0.49645	0.01389	2.02226	33
Avg.	5.1474				

Таблица 2: Стойности за 30 реализации за 300 кита и 30 итерации

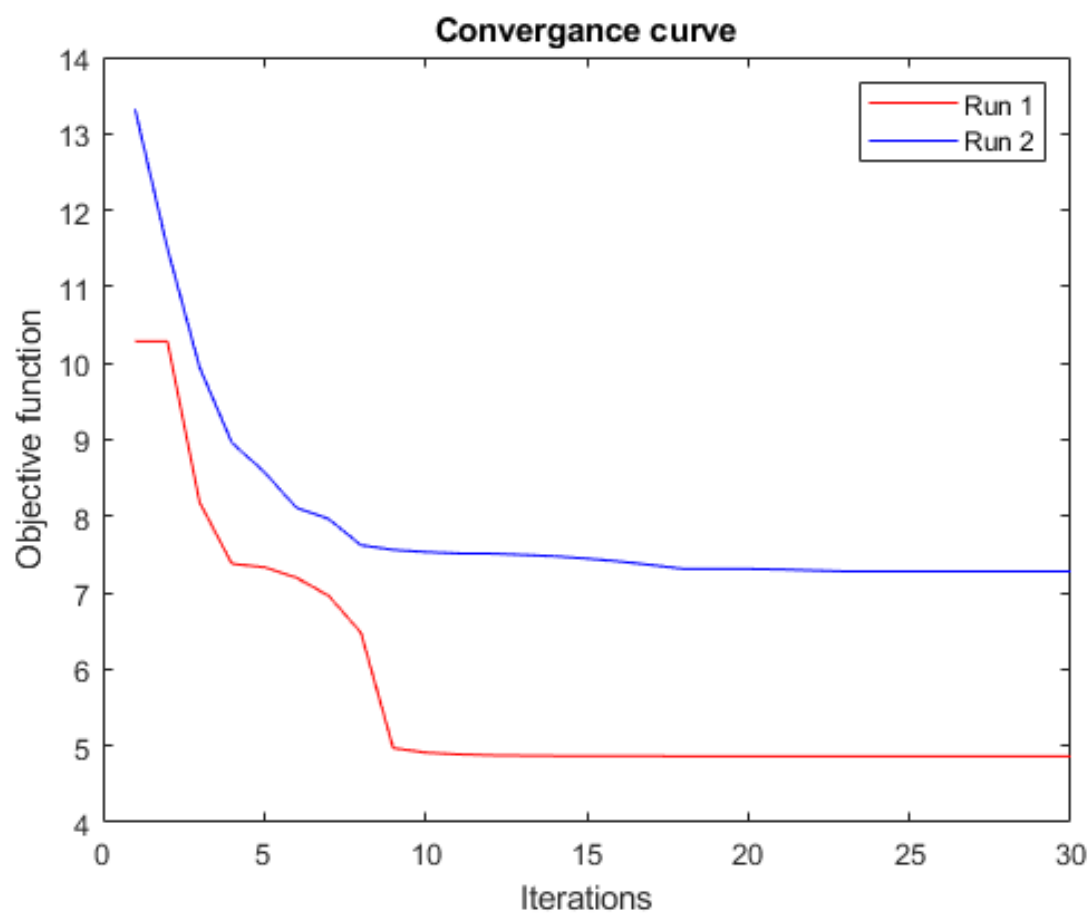
3.4.2 Графічно



Фигура 2: Биомаса при 300 агента и 30 итерации



Фигура 3: Субстрат при 300 агента и 30 итерации



Фигура 4: Сходимость при 300 агента и 30 итерации

3.5 Анализ на резултатите

3.5.1 Постигнат най-добър (най-точен) модел

Всички проведени експерименти от таблица 1 достигат $J \approx 4.733$. По-значителни разлики се наблюдават във времето за изпълнение и средната грешка. В най-бързия случай времето е около 6 пъти по-малко, отколкото при най-бавното изпълнение на метаевристичния алгоритъм. Средната грешка варира в интервала (4.7376, 7.8085).

В таблица 2 е разгледано най-бързото изпълнение (с 300 агента и 30 итерации), при което минималната грешка е 4.7327, а средната грешка е 5.1474. За всички изчисления, въведени в този документ, е използван по-бавният компютър. Най-малка разлика между средна и минимална грешка дава изпълнението с 1000 агента и 30 итерации - минималната грешка е 4.7325, а средната грешка е 4.7376. С тези стойности на хиперпараметрите алгоритъмът работи много по-бавно, отнемайки три пъти повече време, отколкото с 300 агента и същия брой итерации. Изпълнението с 50 агента и 100 итерации също е бързо, но там средната грешка е най-голяма и не дава достатъчно добри решения на задачата, въпреки че тези стойности са дадени за пример от авторите на алгоритъма.

4 Сравнение на получените резултати от WOA с генетичен алгоритъм (GA), алгоритъм на симулираното закаляване (SA) и алгоритъм на гарвана (CSA)

4.1 Представяне на резултатите

WOA е сравнен с алгоритми GA, SA и CSA със следните параметри:

GA

- Number of individuals = 40
- Maximal number of generations = 15
- Number of variables = 3
- Precision of binary representation = 20
- Mutation rate = 0.2
- Crossover rate = 0.8
- Generation gap = 0.97

SA

- Initial temperature = 100
- Reanneal interval = 100

CSA

- Flock = 100
- Awareness = 0.1
- Flight length = 2

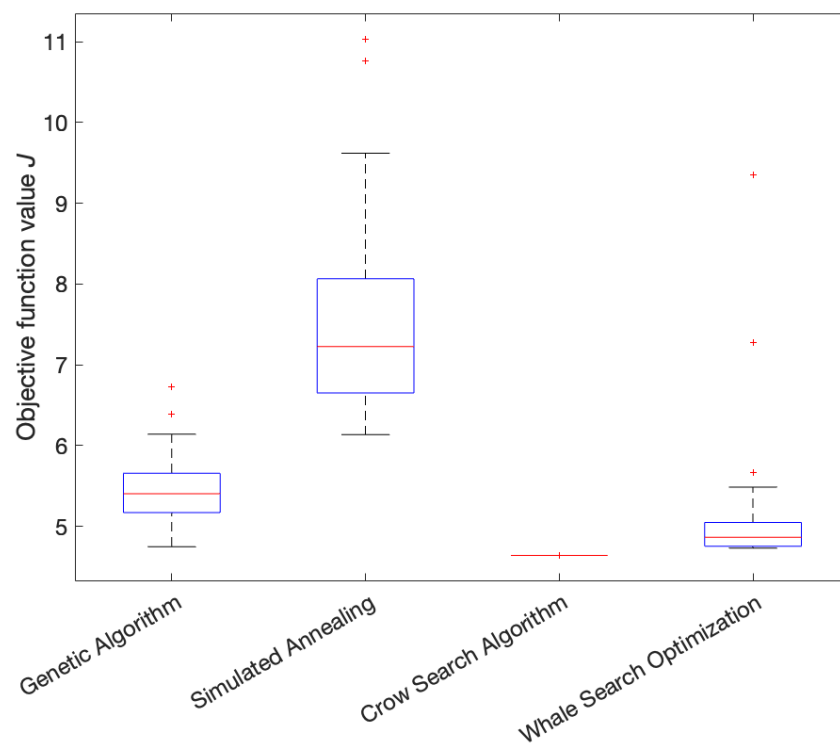
WOA

- Agents = 300
- Iterations = 30

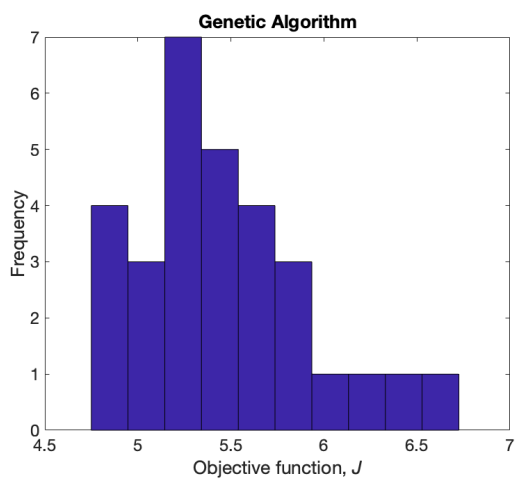
Алгоритъм	Avg J	Min J	μ_{max}	k_s	$Y_{S/X}$
GA	7.6484	4.80705	0.47812	0.01066	2.01979
SA	7.5815	6.13686	0.48825	0.01895	2.01362
CSA	4.64	4.63995	0.48392	0.01121	2.02473
WOA	5.1474	4.73266	0.485	0.01144	2.02288

Таблица 3: Сравнение на резултати от GA, SA, CSA, WOA

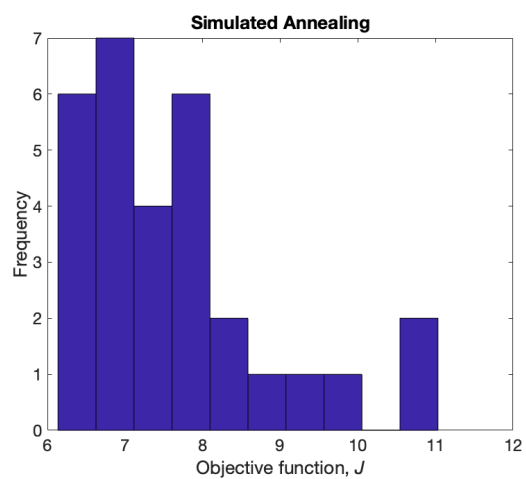
Сравнение на box-plots на резултатите за стойността на целевата функция от 30 реализации на всеки от алгоритмите:



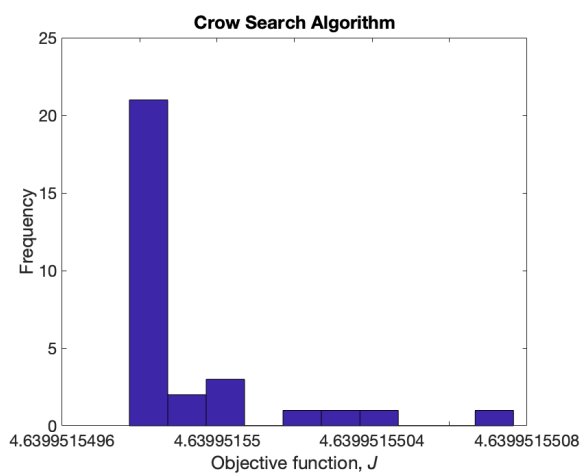
Фигура 5: Box-plot на 30 реализации на GA, SA, CSA, WOA



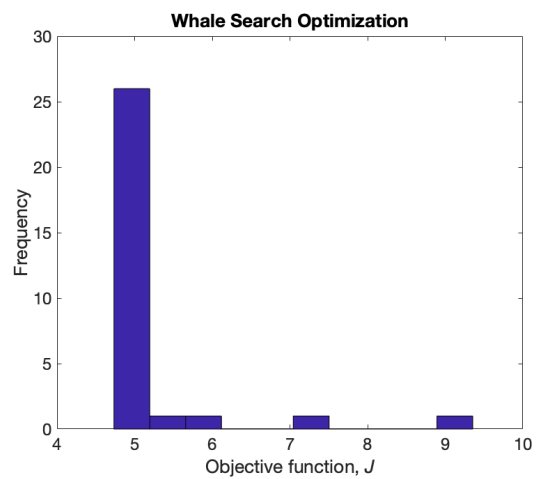
(a) GA



(б) SA



(в) CSA



(г) WOA

Фигура 6: Сравнение на хистограмите за GA, SA, CSA и WOA

Алгоритъм 1	Алгоритъм 2	P	H
Genetic Algorithm	Whale Search Optimization	4.08×10^{-5}	1
Simulated Annealing	Whale Search Optimization	1.69×10^{-9}	1
Crow Search Algorithm	Whale Search Optimization	3.02×10^{-11}	1
Genetic Algorithm	Simulated Annealing	1.33×10^{-10}	1
Genetic Algorithm	Crow Search Algorithm	3.02×10^{-11}	1
Simulated Annealing	Crow Search Algorithm	3.02×10^{-11}	1

Таблица 4: Wilcoxon тест между всяка двойка отделни алгоритми

Алгоритъм 1	Алгоритъм 2	P
Genetic Algorithm	Whale Search Optimization	0.441
Simulated Annealing	Whale Search Optimization	0.959
Crow Search Algorithm	Whale Search Optimization	0.414
Genetic Algorithm	Simulated Annealing	0.603
Genetic Algorithm	Crow Search Algorithm	0.265
Simulated Annealing	Crow Search Algorithm	0.310

Таблица 5: Friedman тест между всяка двойка отделни алгоритми

4.2 Анализ на получените резултати

4.2.1 Оптимални стойности на целевата функция

Crow Search Algorithm постига както най-малка стойност $J=4.64$, така и най-малка средна стойност 4.64. CSA показва и напълно консистентна стойност на решенията, което показва надеждна сходимост.

Whale Optimization Algorithm също постига ниска стойност на J , но с малко по-голяма вариация и следователно средната стойност 5.147 е по-висока от тази на CSA, но значително по-ниска от тези на GA и SA. WOA постига тези резултати с много по-висока бързина от CSA.

Genetic Algorithm показва по-ниска производителност от горните два, но постига резултати за значително по-кратко време от тях.

Simulated Annealing показва най-слаба стабилност от избраните алгоритми и често достига до сравнително високи стойности на J .

Резултати показват, че ако няма ограничение във времето за реализация, CSA е най-подходящ за дадената задача. Ако времето за реализация се счита за важен параметър, то WOA предоставя най-добро съотношение между бързодействие и стабилност в намиране на решение.

4.2.2 Оценка на статистическата значимост на разликите в резултатите на алгоритмите

При дадените стойности на Wilcoxon теста, нулевата хипотеза е отхвърлена във всички случаи. При теста на Friedman няма случаи на $p < 0.05$. Тези данни доказват, че четирите алгоритъма не са еднакви и всеки от тях се различава значимо от всеки друг.

5 Интеркритериален анализ (ИКА) на получените оценки на параметрите на модела от идентификацията с различните алгоритми

5.1 Представяне на резултатите от ИКА за всеки алгоритъм

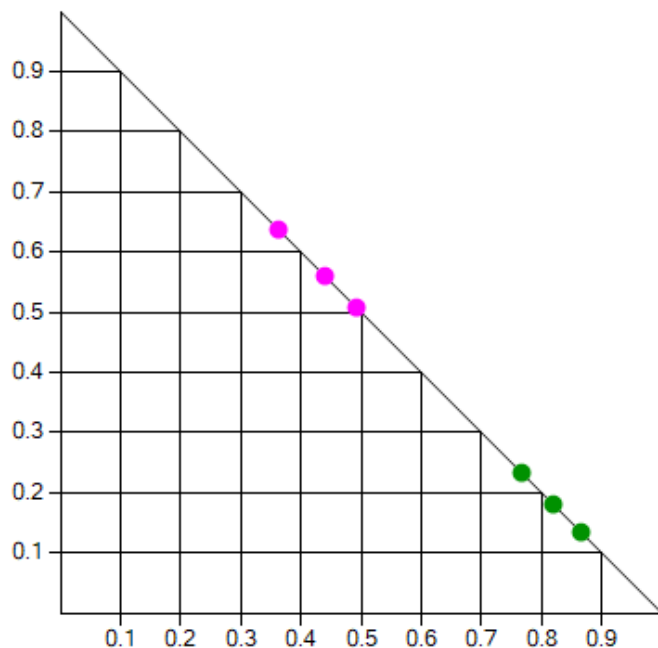
5.1.1 Genetic Algorithm (GA)

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.8184	0.7655	0.4897
μ_{max}	0.8184	-	0.8644	0.4368
k_s	0.7655	0.8644	-	0.3609
$Y_{S/X}$	0.4897	0.4368	0.3609	-

Таблица 6: μ -table GA

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.1816	0.2345	0.5103
μ_{max}	0.1816	-	0.1356	0.5632
k_s	0.2345	0.1356	-	0.6391
$Y_{S/X}$	0.5103	0.5632	0.6391	-

Таблица 7: ν -table GA



Фигура 7: Интуиционистки размит триъгълник GA

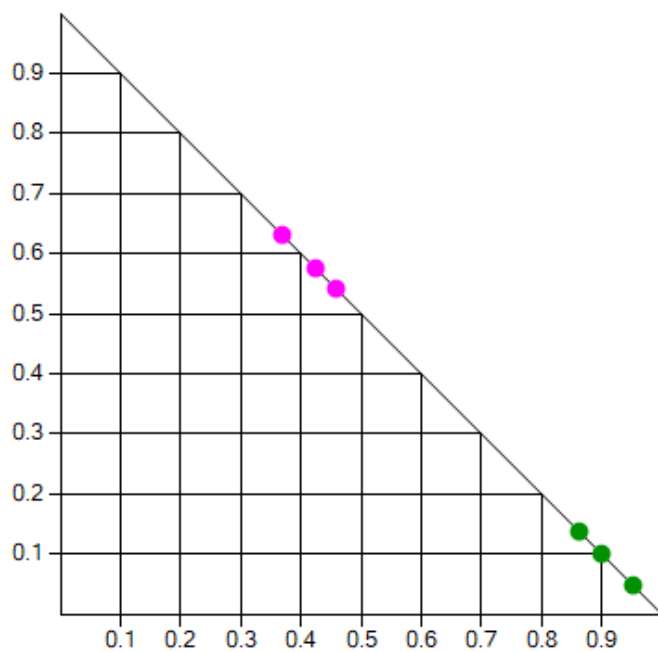
5.1.2 Simulated Annealing (SA)

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.8598	0.8989	0.3678
μ_{max}	0.8598	-	0.9517	0.4575
k_s	0.8989	0.9517	-	0.4230
$Y_{S/X}$	0.3678	0.4575	0.4230	-

Таблица 8: μ -table SA

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.1402	0.1011	0.6322
μ_{max}	0.1402	-	0.0483	0.5425
k_s	0.1011	0.0483	-	0.5770
$Y_{S/X}$	0.6322	0.5425	0.5770	-

Таблица 9: ν -table SA



Фигура 8: Интуиционистки размит триъгълник SA

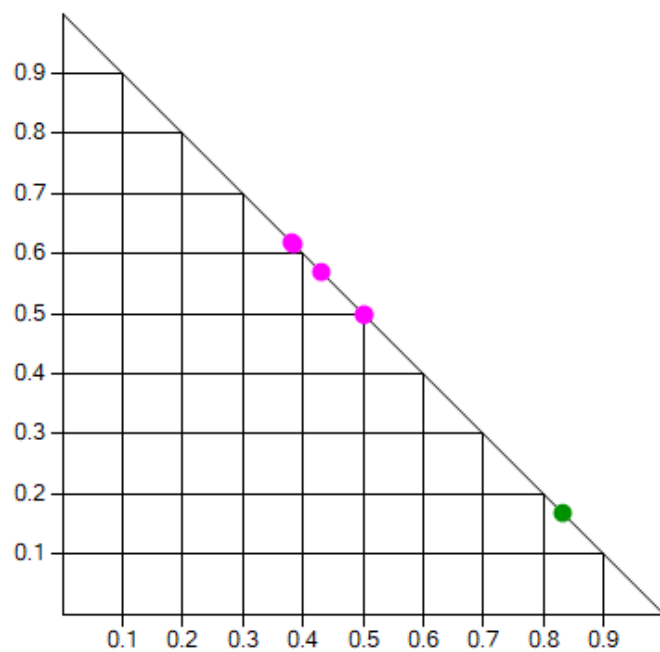
5.1.3 Crow Search Algorithm (CSA)

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.3793	0.3839	0.4299
μ_{max}	0.3793	-	0.8299	0.4989
k_s	0.3839	0.8299	-	0.4989
$Y_{S/X}$	0.4299	0.4989	0.4989	-

Таблица 10: μ -table CSA

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.6207	0.6161	0.5701
μ_{max}	0.6207	-	0.1701	0.5011
k_s	0.6161	0.1701	-	0.5011
$Y_{S/X}$	0.5701	0.5011	0.5011	-

Таблица 11: ν -table CSA



Фигура 9: Интуиционистки размит триъгълник CSA

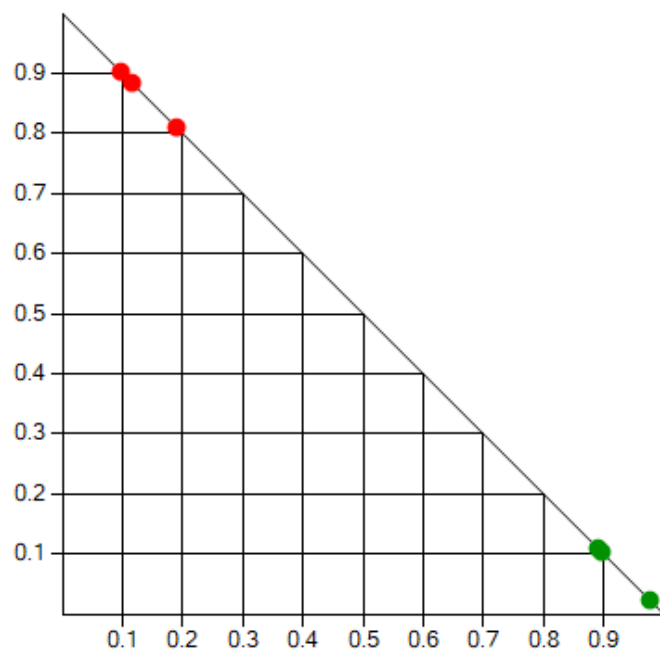
5.1.4 Whale Optimization Algorithm (WOA)

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.8897	0.8966	0.1885
μ_{max}	0.8897	-	0.9747	0.1149
k_s	0.8966	0.9747	-	0.0943
$Y_{S/X}$	0.1885	0.1149	0.0943	-

Таблица 12: μ -table WOA

	J	μ_{max}	k_s	$Y_{S/X}$
J	-	0.1103	0.1034	0.8115
μ_{max}	0.1103	-	0.0253	0.8851
k_s	0.1034	0.0253	-	0.9057
$Y_{S/X}$	0.8115	0.8851	0.9057	-

Таблица 13: ν -table WOA



Фигура 10: Интуиционистски размит триъгълник WOA

5.2 Анализ на резултатите

При μ резултати, близко до 1 може да се твърди за силна връзка между параметрите на даден модел. В случай на ν стойности около 1, може да се заключи, че такава връзка няма. Стойности по средата, например много близо до 0.5, не могат да дадат категоричен извод дали има връзка между параметрите на модела.

Таблицата за WOA и съответстващият интуиционистки размит триъгълник показват изключително висока зависимост между μ_{max} и k_s от 0.9747. SA показва също много висока зависимост от 0.9517, а GA и CSA показват съответно 0.8644 и 0.8299.

Предвид тази взаимовръзка между двата параметъра при използването на WOA, в дадена специфична ситуация, този алгоритъм би могъл да бъде оптимален избор, защото дава възможност да се зададе константа стойност за един от параметрите μ_{max} или k_s .

6 Източници и използвана литература

1. Документи и материали от курса "Въведение в метаевристичните алгоритми"
2. Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments [Nadim Rana, Muhammad Shafie Abd Latiff, Shafi'i Muhammad Abdulhamid, Haruna Chiroma]
3. The Whale Optimization Algorithm, Advances in Engineering Software 95 (2016) 51–67 [Seyedali Mirjalili, Andrew Lewis]
4. Improvement of genetic algorithm performance for identification of cultivation process models [Olympia Roeva]