

Министерство науки и высшего образования  
Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра теоретической и прикладной информатики

Практическое задание № 3  
по дисциплине «Методы оптимизации»

### МЕТОД ШТРАФНЫХ ФУНКЦИЙ



Факультет:	ПМИ
Группа:	ПМИ-62
Бригада:	4
Студенты:	Блинов Александр, Ковалёв Никита
Преподаватель:	Постовалов Сергей Николаевич

Новосибирск  
2019

## 1. ЦЕЛЬ РАБОТЫ

Ознакомиться с методами штрафных функций при решении задач нелинейного программирования. Изучить типы штрафных и барьерных функций, их особенности, способы и области применения, влияние штрафных функций на сходимость алгоритмов, зависимость точности решения задачи нелинейного программирования от величины коэффициента штрафа.

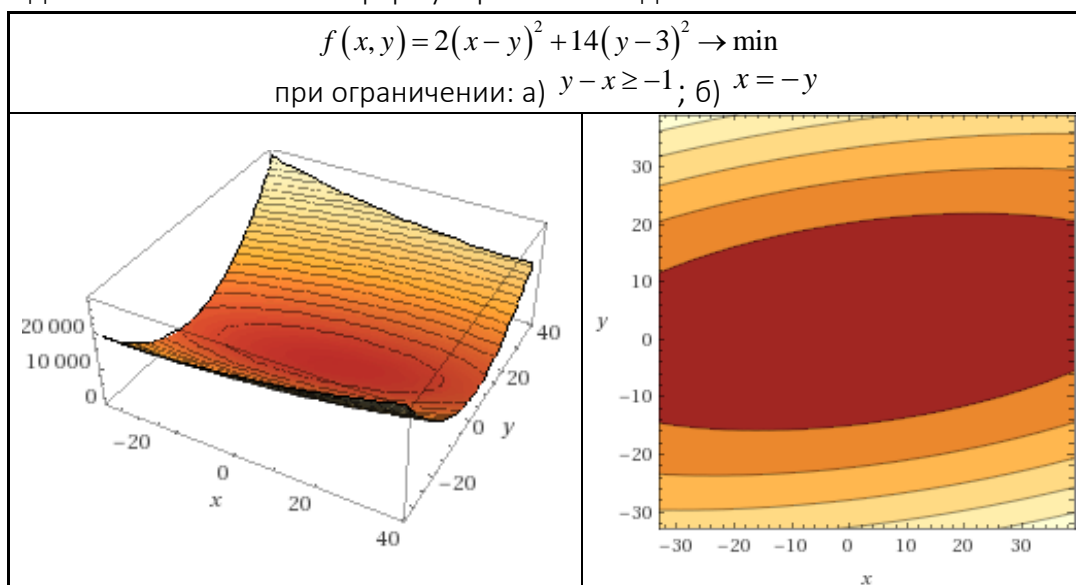
## 2. ЗАДАНИЕ

Применяя методы поиска минимума 0-го порядка, реализовать программу для решения задачи нелинейного программирования с использованием метода штрафных функций.

Исследовать сходимость **метода штрафных функций** в зависимости от выбора штрафных функций, начальной величины коэффициента штрафа, стратегии изменения коэффициента штрафа, начальной точки, задаваемой точности  $\varepsilon$ . Сформулировать выводы.

Применяя методы поиска минимума 0-го порядка, реализовать программу для решения задачи нелинейного программирования с ограничением типа неравенства (только пункт а) с использованием **метода барьерных функций**.

Исследовать сходимость **метода барьерных функций** (только пункт а) в зависимости от выбора барьерных функций, начальной величины коэффициента штрафа, стратегии изменения коэффициента штрафа, начального приближения, задаваемой точности  $\varepsilon$ . Сформулировать выводы.



## 3. ИССЛЕДОВАНИЯ

РЕЗУЛЬТАТ РЕШЕНИЯ БЕЗУСЛОВНОЙ ЗАДАЧИ ДЛЯ ЗАДАННОЙ ФУНКЦИИ:

ОТВЕТ	Вычислений $f$	Начальное ПРИБЛИЖЕНИЕ	$\varepsilon$
[2.99999998, 2.99999997]	8204	[0, 1]	1E-7

### Метод штрафных функции

Функция штрафа для ограничения вида  $y - x + 1 \geq 0$  имеет вид:

$$G = \begin{cases} 0, & y - x + 1 \geq 0 \\ -(y - x + 1), & \text{иначе} \end{cases}$$

Изменение результата в зависимости от задаваемой точности  $\varepsilon$  (начальная величина коэффициента штрафа – 10, коэффициент изменения штрафа – 10):

Ответ	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[2.9999, 3.0000]	1	4006	[0, 1]	1E-3
[2.99999, 3.00000]	1	4681	[0, 1]	1E-4
[2.999991, 2.999999]	1	5314	[0, 1]	1E-5
[3.0000000, 2.9999997]	1	7421	[0, 1]	1E-6
[2.99999998, 2.99999997]	1	8204	[0, 1]	1E-7

Изменение результата в зависимости от начальной точки (начальная величина коэффициента штрафа – 10, коэффициент изменения штрафа – 10):

Ответ	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[2.99999998, 2.99999997]	1	8204	[0, 1]	1E-7
[2.99999997, 2.99999998]	1	2591	[0, 2]	1E-7
[3.00000001, 3.00000000]	1	416	[0, 3]	1E-7
[2.99993895, 2.9999235]	1	11816	[1, 1]	1E-7
[2.99999997, 2.99999997]	1	4892	[1, 2]	1E-7
[2.99999999, 3.00000000]	1	408	[1, 3]	1E-7
[2.99999997, 3.00000000]	1	408	[2, 3]	1E-7
[2.99999998, 3.00000000]	1	124	[3, 3]	1E-7

Изменения результата в зависимости от стратегии изменения коэффициента штрафа (начальная величина коэффициента штрафа – 1):

Ответ	Итераций	Вычислений $f$	Начальное приближение	$C_0$	$\varepsilon$
[2.99999998, 2.99999997]	1	8204	[1, -3]	2	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	3	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	4	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	5	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	6	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	7	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	8	1E-7

[2.99999998, 2.99999997]	1	8204	[1, -3]	9	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	10	1E-7

Изменения результата в зависимости от начальной величины коэффициента штрафа (коэффициент изменения штрафа – 1):

Ответ	Итераций	Вычислений $f$	Начальное приближение	$r_0$	$\varepsilon$
[2.99999998, 2.99999997]	1	8204	[1, -3]	1	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	2	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	3	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	4	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	5	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	6	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	7	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	8	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	9	1E-7
[2.99999998, 2.99999997]	1	8204	[1, -3]	10	1E-7

#### Исследование стратегии изменения функции штрафа

Для изменения функции штрафа была выбрана функция:

$$G = \left( \frac{g + |g|}{2} \right)^{2n}, \text{ где } g = x - y - 1$$

Изменения результата в зависимости от выбора штрафных функций (начальная величина коэффициента штрафа – 10, коэффициент изменения штрафа – 10):

Ответ	Итераций	Вычислений $f$	Начальное приближение	n	$\varepsilon$
[2.99999998, 2.99999997]	1	8204	[0, 1]	1	1E-7
[2.99999997, 2.99999997]	1	1742	[0, 1]	2	1E-7
[2.99999996, 2.99999997]	1	1890	[0, 1]	3	1E-7
[2.99999995, 2.99999997]	1	2038	[0, 1]	4	1E-7
[3.00000000, 2.99999997]	1	2186	[0, 1]	5	1E-7
[2.99999999, 2.99999997]	1	2334	[0, 1]	6	1E-7
[2.99999998, 2.99999997]	1	2482	[0, 1]	7	1E-7

Функция штрафа для ограничения вида  $x = -y$  имеет вид:

$$G = |x + y|$$

Изменение результата в зависимости от задаваемой точности  $\varepsilon$  (начальная величина коэффициента штрафа – 10, коэффициент изменения штрафа – 10):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
$[-2.28421756, 2.28427219]$	$5.463121e-05$	100	7859	$[0, 1]$	$1E-3$
$[-2.28598598, 2.28596869]$	$1.728103e-05$	100	9992	$[0, 1]$	$1E-4$
$[-2.28570857, 2.28570882]$	$2.51011048e-07$	100	12239	$[0, 1]$	$1E-5$
$[-2.28570499, 2.28570475]$	$2.48234803e-07$	100	14185	$[0, 1]$	$1E-6$
$[-2.28571108, 2.28571106]$	$2.197705351e-08$	100	16122	$[0, 1]$	$1E-7$

Изменение результата в зависимости от начальной точки (начальная величина коэффициента штрафа – 10, коэффициент изменения штрафа – 10):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
$[-2.28571108, 2.28571106]$	$2.197705351e-08$	100	16122	$[0, 1]$	$1E-7$
$[-2.28571102, 2.28571100]$	$2.15510627e-08$	100	15780	$[0, 2]$	$1E-7$
$[-2.28571098, 2.28571096]$	$2.10968646e-08$	100	16170	$[0, 3]$	$1E-7$
$[-2.28571107, 2.28571104]$	$2.274998411e-08$	100	15826	$[1, 1]$	$1E-7$
$[-2.28571115, 2.28571113]$	$2.281753985e-08$	100	15456	$[1, 2]$	$1E-7$
$[-2.28571114, 2.28571115]$	$9.2818610575e-09$	100	16000	$[1, 3]$	$1E-7$
$[-2.28571077, 2.28571075]$	$2.174469626e-08$	100	15622	$[2, 3]$	$1E-7$
$[-2.28571094, 2.28571092]$	$2.005990751e-08$	100	15618	$[3, 3]$	$1E-7$

Изменения результата в зависимости от стратегии изменения коэффициента штрафа (начальная величина коэффициента штрафа – 1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$C_0$	$\varepsilon$
$([-1.89599997, 1.89599997])$	$4.81295669985e-10$	8	4808	$[0, 1]$	2	$1E-7$
$[-2.35713975, 2.35713973]$	$2.060205162e-08$	100	17614	$[0, 1]$	3	$1E-7$
$[-1.89647610, 1.89647608]$	$2.172803514e-08$	100	17480	$[0, 1]$	4	$1E-7$
$[-2.64285427, 2.64285425]$	$2.007805965e-08$	100	17064	$[0, 1]$	5	$1E-7$
$[-2.57142517, 2.57142515]$	$2.153328404e-08$	100	17108	$[0, 1]$	6	$1E-7$
$[-2.49999655, 2.49999653]$	$2.052630245e-08$	100	17118	$[0, 1]$	7	$1E-7$
$[-2.42856806, 2.428568044]$	$2.143875388e-08$	100	17122	$[0, 1]$	8	$1E-7$
$[-2.35713969, 2.357139674]$	$2.004257604e-08$	100	17268	$[0, 1]$	9	$1E-7$
$[-2.285710899, 2.2857108877]$	$1.199440102e-08$	100	17274	$[0, 1]$	10	$1E-7$

Изменения результата в зависимости от начальной величины коэффициента штрафа (коэффициент изменения штрафа – 1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$r_0$	$\varepsilon$
[2.67857143, 2.92857142]	5.6071428624	100	16436	[0,1]	1	1E-7
[2.35714281, 2.85714284]	5.2142856637	100	16960	[0,1]	2	1E-7
[2.03571428, 2.78571430]	4.8214285889	100	18136	[0,1]	3	1E-7
[1.71428570, 2.71428571]	4.4285714169	100	16520	[0,1]	4	1E-7
[1.39285707, 2.642857112]	4.0357141925	100	16134	[0,1]	5	1E-7
[1.07142854, 2.571428560]	3.6428571008	100	16900	[0,1]	6	1E-7
[0.74999999, 2.500000000]	3.2499999921	100	16396	[0,1]	7	1E-7
[0.42857137, 2.428571406]	2.8571427804	100	16010	[0,1]	8	1E-7
[0.10714276, 2.357142820]	2.4642855830	100	15996	[0,1]	9	1E-7
[-0.21428577, 2.28571429]	2.0714285211	100	15974	[0,1]	10	1E-7

#### Метод барьерных функций

Одна из возможных функций штрафа для ограничения вида  $y - x + 1 \geq 0$  имеет вид:

$$G = f(x) = \begin{cases} -\ln(-g), & g \leq 0 \\ \text{infinity}, & \text{иначе} \end{cases}$$

Изменение результата в зависимости от задаваемой точности  $\varepsilon$  (начальная величина коэффициента штрафа – 0.1, коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[2.9759, 3.0000]	-0.0238721339	1	4045	[0,1]	1E-3
[2.97557, 2.99999]	-0.0241257046	1	4720	[0,1]	1E-4
[2.975594, 2.999998]	-0.0241117166	1	6620	[0,1]	1E-5
[2.9755956, 2.9999997]	-0.0241110888	1	7462	[0,1]	1E-6
[2.97559558, 2.99999997]	-0.0241113596	1	8240	[0,1]	1E-7

Изменение результата в зависимости от начальной точки (начальная величина коэффициента штрафа – 0.1, коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[2.97559558, 2.99999997]	-0.0241113596	1	8240	[0,1]	1E-7
[2.97559550, 2.99999999]	-0.0241114542	1	2612	[0,2]	1E-7
[2.97559555, 3.000000000]	-0.0241114203	1	416	[0,3]	1E-7

[2.97559554, 2.99999998]	-0.0241114095	1	19046342	[1, 1]	1E-7
[2.97559556, 3.00000001]	-0.0241114184	1	4892	[1, 2]	1E-7
[2.97559554, 3.00000000]	-0.0241114210	1	408	[1, 3]	1E-7
[2.97559557, 3.00000000]	-0.0241113955	1	408	[2, 3]	1E-7
[2.97559555, 3.00000000]	-0.0241114206	1	396	[3, 3]	1E-7

Изменения результата в зависимости от стратегии изменения коэффициента штрафа  
(начальная величина коэффициента штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$C_0$	$\varepsilon$
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-1	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-2	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-3	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-4	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-5	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-6	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-7	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-8	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-9	1E-7
[2.97559558, 2.99999997]	-0.02411135964	1	8240	[0, 1]	1E-10	1E-7

Изменения результата в зависимости от начальной величины коэффициента штрафа  
(коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$r_0$	$\varepsilon$
[2.97559558, 2.99999998]	-0.024111359	1	8240	[0, 1]	1E-1	1E-7
[2.99750623, 3.00000002]	-0.002490689	1	8214	[0, 1]	1E-2	1E-7
[2.99975005, 2.99999999]	-0.000249903	1	8204	[0, 1]	1E-3	1E-7
[2.99997500, 2.99999999]	-2.499014949 e-05	1	8204	[0, 1]	1E-4	1E-7
[2.99999748, 2.99999999]	-2.514888587 e-06	1	8204	[0, 1]	1E-5	1E-7
[2.99999972, 3.00000001]	-2.827817255 e-07	1	8204	[0, 1]	1E-6	1E-7
[2.99999996, 3.00000001]	-4.903098429 e-08	1	8204	[0, 1]	1E-7	1E-7
[2.99999998, 2.99999998]	7.033122997 e-09	1	8204	[0, 1]	1E-8	1E-7
[2.99999998, 2.99999998]	7.033122997 e-09	1	8204	[0, 1]	1E-9	1E-7

[2.99999998, 2.99999998]	7.033122997 e-09	1	8204	[0,1]	1E-10	1E-7
-----------------------------	---------------------	---	------	-------	-------	------

Одна из возможных функций штрафа для ограничения вида  $y - x + 1 \geq 0$  имеет вид:

$$G = f(x) = \begin{cases} -\frac{1}{g(x)}, & g \leq 0 \\ \text{infinity}, & \text{иначе} \end{cases}$$

Изменение результата в зависимости от задаваемой точности  $\varepsilon$  (начальная величина коэффициента штрафа – 0.1, коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[3.0000, 3.0001]	0.9998858377	3	1079	[0,1]	1E-3
[2.99998, 3.00001]	0.9999620734	4	1476	[0,1]	1E-4
[2.999995, 2.999998]	0.9999970703	5	2059	[0,1]	1E-5
[2.9999999, 3.0000002]	0.9999996655	6	2485	[0,1]	1E-6
[2.999999955, 2.999999978]	0.9999999776	7	3230	[0,1]	1E-7

Изменение результата в зависимости от начальной точки (начальная величина коэффициента штрафа – 0.1, коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$\varepsilon$
[2.999999955, 2.999999978]	0.9999999776	7	3230	[0,1]	1E-7
[2.99999997, 2.99999999]	0.9999999798	7	2480	[0,2]	1E-7
[2.99999997, 3.00000000]	0.9999999767	7	1910	[0,3]	1E-7
[2.99999996, 2.99999999]	0.9999999643	7	63660	[1,1]	1E-7
[2.99999999, 3.00000001]	0.9999999786	7	2818	[1,2]	1E-7
[2.99999997, 3.00000000]	0.9999999759	7	1906	[1,3]	1E-7
[2.99999998, 3.00000000]	0.9999999786	7	1906	[2,3]	1E-7
[2.99999998, 3.00000000]	0.9999999764	7	1900	[3,3]	1E-7

Изменения результата в зависимости от стратегии изменения коэффициента штрафа (начальная величина коэффициента штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$C_0$	$\varepsilon$
[2.99999995, 2.99999997]	0.9999999776	7	3230	[0,1]	1E-1	1E-7
[2.99999995, 2.99999997]	0.9999999733	4	2638	[0,1]	1E-2	1E-7
[2.99999994, 2.99999998]	0.9999999568	3	2342	[0,1]	1E-3	1E-7
[3.000000002, 2.99999999]	1.00000000295	3	2046	[0,1]	1E-4	1E-7



[2.99999998, 2.99999999]	0.9999999915	3	2046	[0,1]	1E-5	1E-7
[2.99999997, 2.99999999]	0.9999999788	2	3648	[0,1]	1E-6	1E-7
[2.99999997, 2.99999999]	0.9999999788	2	3648	[0,1]	1E-7	1E-7
[2.99999997, 2.99999999]	0.9999999788	2	3648	[0,1]	1E-8	1E-7
[2.99999997, 2.99999999]	0.9999999788	2	3648	[0,1]	1E-9	1E-7
[2.99999997, 2.99999999]	0.9999999788	2	3648	[0,1]	1E-10	1E-7

Изменения результата в зависимости от начальной величины коэффициента штрафа (коэффициент изменения штрафа – 0.1):

Ответ	Последнее вычисленное значение $G(x)$	Итераций	Вычислений $f$	Начальное приближение	$r_0$	$\varepsilon$
[2.99999996, 2.99999998]	0.9999999776	7	3230	[0,1]	1E-1	1E-7
[3.00000002, 3.00000002]	0.9999999930	6	2782	[0,1]	1E-2	1E-7
[2.99999995, 2.99999999]	0.9999999689	5	2482	[0,1]	1E-3	1E-7
[2.99999998, 2.99999999]	0.9999999870	4	2186	[0,1]	1E-4	1E-7
[2.99999997, 2.99999999]	0.9999999758	3	2038	[0,1]	1E-5	1E-7
[2.99999998, 3.00000001]	0.9999999791	2	1890	[0,1]	1E-6	1E-7
[3.00000000, 3.00000001]	0.9999999910	2	1890	[0,1]	1E-7	1E-7
[2.99999997, 2.99999997]	0.9999999963	1	1742	[0,1]	1E-8	1E-7
[2.99999997, 2.99999997]	0.9999999963	1	1742	[0,1]	1E-9	1E-7
[2.99999997, 2.99999997]	0.9999999963	1	1742	[0,1]	1E-10	1E-7

#### 4. Выводы

- При увеличении задаваемой точности  $\varepsilon$  количество вычислений целевой функции возрастает вне зависимости от метода.
- Сходимость и количество вычислений целевой функции практически не зависит от начального приближения.
- Начальная величина коэффициента штрафа и стратегия изменения коэффициента штрафа практически не влияют на сходимость в целом, но значительно могут сократить или увеличить количество вычислений целевой функции.
- Выбор функции штрафа практически не влияет на сходимость, но на практике выбор удачной функции штрафа может значительно сократить количество вычислений целевой функции.

## 5. ЛИСТИНГ РАЗРАБОТАННОЙ ПРОГРАММЫ

### Метод золотого сечения

```
import math

FUNC_CALC = 0

# Метод золотого сечения для одномерного поиска
def goldenSection(function, l0, X, S, EPS):
    global FUNC_CALC
    interval = searchSection(function, l0, X, S)
    if interval[1] >= interval[0]:
        b = interval[1]
        a = interval[0]
    else:
        a = interval[1]
        b = interval[0]
    lenOfSection = b - a
    l1 = a + (3 - math.sqrt(5)) / 2 * (b - a)
    l2 = a + (math.sqrt(5) - 1) / 2 * (b - a)
    f1 = function(X[0] + l1 * S[0], X[1] + l1 * S[1])
    FUNC_CALC += 1
    f2 = function(X[0] + l2 * S[0], X[1] + l2 * S[1])
    FUNC_CALC += 1
    while lenOfSection > EPS:
        if f1 > f2:
            a = l1
            l1 = l2
            l2 = a + (math.sqrt(5) - 1) / 2 * (b - a)
            f1 = f2
            f2 = function(X[0] + l2 * S[0], X[1] + l2 * S[1])
            FUNC_CALC += 1
        else:
            b = l2
            l2 = l1
            l1 = a + (3 - math.sqrt(5)) / 2 * (b - a)
            f2 = f1
            f1 = function(X[0] + l1 * S[0], X[1] + l1 * S[1])
            FUNC_CALC += 1
        lenOfSection = b - a
    return (b + a) / 2, FUNC_CALC

# Поиск отрезка, содержащий минимум
def searchSection(function, l0, X, S):
    global FUNC_CALC
    delta = 0.001
    l1 = 0
    h = 0
    f0 = function(X[0] + l0 * S[0], X[1] + l0 * S[1])
    FUNC_CALC += 1
    if f0 > function(X[0] + (l0 + delta) * S[0], X[1] + (l0 + delta) * S[1]):
        FUNC_CALC += 1
        l1 = l0 + delta
        h = delta
    elif f0 > function(X[0] + (l0 - delta) * S[0], X[1] + (l0 - delta) * S[1]):
        FUNC_CALC += 1
        l1 = l0 - delta
        h = - delta
    else:
        return l0 - delta, l0 + delta
    h = 2 * h
    f1 = function(X[0] + l1 * S[0], X[1] + l1 * S[1])
    FUNC_CALC += 1
    l2 = l1 + h
    f2 = function(X[0] + l2 * S[0], X[1] + l2 * S[1])
    FUNC_CALC += 1
    while f1 > f2:
        l0 = l1
        f0 = f1
        l1 = l2
        f1 = f2
        h = 2 * h
        l2 = l1 + h
        f2 = function(X[0] + l2 * S[0], X[1] + l2 * S[1])
        FUNC_CALC += 1
    return l0 - h / 2, l2
```

## Метод Розенброка

```
import copy
import math
import golden_section_method

# Вычисление нормы
def findNorm(V):
    return math.sqrt(V[0] ** 2 + V[1] ** 2)

# Условие выхода
def exitCondition(function, X1, X0, EPS):
    if math.fabs(function(X1[0], X1[1]) - function(X0[0], X0[1])) < EPS:
        return True
    elif math.fabs(X1[0] - X0[0]) < EPS and math.fabs(X1[1] - X0[1]) < EPS:
        return True
    return False

# Метод Розенброка
def RosenbrockMethod(function, X, S1, S2, EPS):
    FUNC_CALC = 0
    A = [[0, 0], [0, 0]]
    B = [0, 0]
    while True:
        X0 = copy.copy(X)

        L1, func_calc_L1 = golden_section_method.goldenSection(function, -1, X, S1, EPS)
        FUNC_CALC += func_calc_L1
        X[0] = X0[0] + L1 * S1[0]
        X[1] = X0[1] + L1 * S1[1]

        L2, func_calc_L2 = golden_section_method.goldenSection(function, 0, X, S2, EPS)
        FUNC_CALC += func_calc_L2
        X[0] = X[0] + L2 * S2[0]
        X[1] = X[1] + L2 * S2[1]

        A[0][0] = L1 * S1[0] + L2 * S2[0]
        A[0][1] = L1 * S1[1] + L2 * S2[1]

        if math.fabs(L1) >= math.fabs(L2):
            A[1][0] = L2 * S2[0]
            A[1][1] = L2 * S2[1]
        else:
            A[1][0] = L1 * S2[0]
            A[1][1] = L1 * S2[1]

        S1[0] = A[0][0] / findNorm(A[0])
        S1[1] = A[0][1] / findNorm(A[0])

        K = A[1][0] * S1[0] + A[1][1] * S1[1]
        B[0] = A[1][0] - K * S1[0]
        B[1] = A[1][1] - K * S1[1]

        S2[0] = B[0] / findNorm(B)
        S2[1] = B[1] / findNorm(B)

        if exitCondition(function, X, X0, EPS):
            break
    return X, FUNC_CALC
```

## Метод штрафных/барьерных функций

```
import math
import rosenbrock_method

# поиск минимума с ограничением
def findMinWithRestriction(f, restriction, x_0, r_0, C_0, eps):
    for i in range(100):
        res, FUNC_CALC = rosenbrock_method.RosenbrockMethod(lambda x, y: f(x, y) + r_0 *
restriction(x, y), x_0, [1, 0], [0, 1], eps)
        if r_0 * restriction(res[0], res[1]) < eps:
            return res, restriction(res[0], res[1]), FUNC_CALC, i + 1
        r_0 *= C_0
    return res, restriction(res[0], res[1]), FUNC_CALC, 100
```