Министерство образования и науки Российской Федерации

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Лабораторная работа №4

Метод штрафных функций

по дисциплине

«Методы оптимизации»

Факультет прикладной математики и информатики

Группа ПМ-63

Студенты Майер В.А.

Курбангалиева В.Т.

Назарова Т.А.

Преподаватели Чимитова Е. В.

Вариант 3

Новосибирск 2019

1. Цель работы

Ознакомиться со статистическими методами поиска при реше­нии задач нелинейного программирования. Изучить методы случайного поиска при определении глобального экстремума функции.

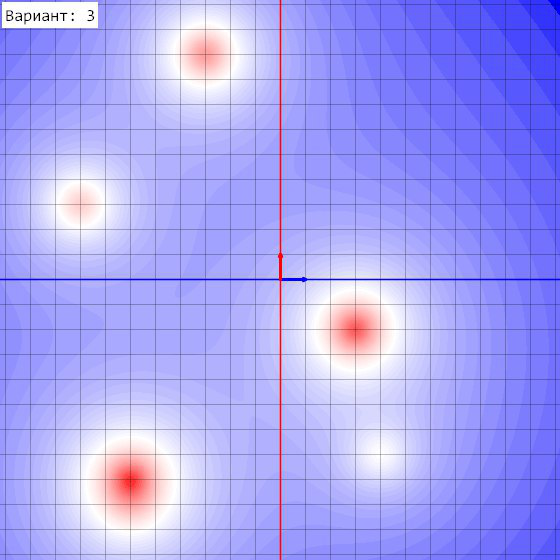
1. Задание

*Найти максимум заданной функции*

На области , .







1. Результаты

* Метод простого случайного поиска.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 5 | 0,8 | 25 | -5,91840646970943  -7,44438295133616 | -6,81470556339712 |
| 1 | 0,8 | 643 | -5,95544446071398  -8,03348832672159 | -9,14752936101062 |
| 0,5 | 0,8 | 2575 | -6,20180270457724  -8,08606455013438 | -8,83399492591295 |
| 0,1 | 0,8 | 64377 | -5,99137457832292  -8,04526988325886 | -9,16795049433735 |
| 0,05 | 0,8 | 257510 | -5,99745536967994  -8,06975889861107 | -9,17159086280062 |
| 0,01 | 0,8 | 6437751 | -6,00092529133005  -8,06345155372445 | -9,17162333922771 |
| 0,005 | 0,8 | 25751006 | -5,99846197105873  -8,06372769086795 | -9,17165443555103 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 0,01 | 0,3 | 1426700 | -5,99846343323517  -8,07674095876363 | -9,17090249254065 |
| 0,01 | 0,4 | 2043303 | -5,99896894581568  -8,06005269198681 | -9,17142370950286 |
| 0,01 | 0,5 | 2772589 | -6,00947310962224  -8,07180010623848 | -9,17058787139558 |
| 0,01 | 0,6 | 3665163 | -5,9927600976046  -8,05659968315465 | -9,17066919019681 |
| 0,01 | 0,7 | 4815891 | -5,99811651557596  -8,07106642894031 | -9,17152778768681 |
| 0,01 | 0,8 | 6437751 | -5,99753712117557  -8,06280429384802 | -9,17159497928147 |
| 0,01 | 0,9 | 9210340 | -5,99938886985154  -8,06575495659642 | -9,17170395113145 |
| 0,01 | 0,95 | 11982928 | -6,00099484249996  -8,0679074386451 | -9,17165926879386 |

* Алгоритм 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 5 | 0,8 | 25 | -5,99920897069525  -8,06626269802098 | -9,17170610839818 |
| 1 | 0,8 | 643 | -5,999208822709  -8,06626271388425 | -9,17170610839833 |
| 0,5 | 0,8 | 2575 | -5,99920884784293  -8,06626269637166 | -9,17170610839833 |
| 0,1 | 0,8 | 64377 | -5,99920883385118  -8,06626269631907 | -9,17170610839834 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 1 | 0,3 | 143 | -5,99920885418519  -8,06626286403881 | -9,17170610839814 |
| 1 | 0,4 | 205 | -5,99920884442156  -8,06626268340182 | -9,17170610839833 |
| 1 | 0,5 | 277 | -5,99920861329283  -8,06626263562255 | -9,17170610839788 |
| 1 | 0,6 | 367 | -5,99920877340544  -8,0662626071775 | -9,17170610839824 |
| 1 | 0,7 | 481 | -5,99920895756479  -8,06626285049539 | -9,17170610839804 |
| 1 | 0,8 | 643 | -5,99920906216863  -8,06626269340552 | -9,1717061083979 |
| 1 | 0,9 | 920 | -5,99920883809552  -8,06626272651229 | -9,17170610839833 |
| 1 | 0,95 | 1197 | -5,99920885813737  -8,06626268003108 | -9,17170610839833 |

|  |  |  |
| --- | --- | --- |
| N | Точность решения | Func Iter |
| 10 | -1,27652555192981E-10 | 39136 |
| 100 | -1,56319401867222E-13 | 258341 |
| 1e3 | -2,8421709430404E-14 | 2340808 |
| 1e4 | -3,5527136788005E-15 | 239398315 |

* Алгоритм 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eps | P | N | | (x\*, y\*) | f(x\*, y\*) |
| 5 | 0,8 | 25 | | 2,99913105452065  -2,00180346002244 | -8,22235545489173 |
| 1 | 0,8 | 643 | | -5,99920822428085  -8,06625242849377 | -9,17170610762526 |
| 0,5 | 0,8 | 2575 | | -5,99920646822377  8,06626507671738 | -9,17170610830939 |
| 0,1 | 0,8 | 64377 | | -5,99921243380007  -8,06626905803874 | -9,17170610799332 |
| 0,05 | 0,8 | 257510 | | -5,99920618837186  -8,06626180835388 | -9,17170610833291 |
| 0,01 | 0,8 | 6437751 | | -5,999208821254  -8,0662623149921 | -9,17170610839725 |
| 0,005 | 0,8 | 25751006 | | -5,99922034299701  -8,06627432946425 | -9,17170610628605 |
| 0,001 | 0,8 | 643775169 | | -5,99921578946188  -8,0662595224744 | -9,17170610791383 |
| Eps | P | | N | (x\*, y\*) | f(x\*, y\*) |
| 0,01 | 0,3 | | 1426700 | -5,99921180907528  -8,06626208996721 | -9,17170610832055 |
| 0,01 | 0,4 | | 2043303 | -5,99920879254701  -8,06626136787055 | -9,17170610838538 |
| 0,01 | 0,5 | | 2772589 | -5,99921394852464  -8,06626817685751 | -9,17170610795732 |
| 0,01 | 0,6 | | 3665163 | -5,99920962069764  -8,06626313317483 | -9,17170610839174 |
| 0,01 | 0,7 | | 4815891 | -5,99920743129817  -8,06625889395543 | -9,17170610827585 |
| 0,01 | 0,8 | | 6437751 | -5,99921237084495  -8,066266815006 | -9,17170610816859 |
| 0,01 | 0,9 | | 9210340 | -5,9992010001715  -8,06625837898279 | -9,1717061077401 |
| 0,01 | 0,95 | | 11982928 | -5,99921736472896  -8,06626458024319 | -9,17170610775442 |

|  |  |  |
| --- | --- | --- |
| N | Точность решения | Func Iter |
| 10 | -0,949350654558135 | 245 |
| 100 | -5,7530513686288E-10 | 511 |
| 1e3 | -6,01266592070715E-10 | 11927 |
| 1e4 | -5,52283552224253E-10 | 10506 |
| 1e5 | -6,62373267346084E-10 | 100794 |
| 1e6 | -5,25801624462474E-13 | 1000536 |
| 1e7 | -1,1601386518123E-11 | 10003656 |

* Алгоритм 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 5 | 0,8 | 25 | 2,99913050332007  -2,00180361147645 | -8,22235545488799 |
| 1 | 0,8 | 643 | -5,99920547724527  -8,06625749087786 | -9,17170610810441 |
| 0,5 | 0,8 | 2575 | -5,99921247671091  -8,06625886490978 | -9,17170610817843 |
| 0,1 | 0,8 | 64377 | -5,99920692078337  -8,06626143778324 | -9,17170610835551 |
| 0,05 | 0,8 | 257510 | -5,9992082563801  -8,06626075622532 | -9,17170610836791 |
| 0,01 | 0,8 | 6437751 | -5,99921077418231  -8,06626496536744 | -9,17170610832897 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | P | N | (x\*, y\*) | f(x\*, y\*) |
| 1 | 0,3 | 143 | 2,99912470542725  -2,00181575655259 | -8,22235545328118 |
| 1 | 0,4 | 205 | -5,99920862196055  -8,06626217846007 | -9,17170610839596 |
| 1 | 0,5 | 277 | -5,99920950398118  -8,06626270795076 | -9,17170610839456 |
| 1 | 0,6 | 367 | -5,99921355491281  -8,06626488406219 | -9,17170610817432 |
| 1 | 0,7 | 481 | -5,99920691737327  -8,06626732349639 | -9,17170610821096 |
| 1 | 0,8 | 643 | -5,99920672354531  -8,06626167159146 | -9,17170610835266 |
| 1 | 0,9 | 920 | -5,99920920911524  -8,06625991925413 | -9,17170610834075 |
| 1 | 0,95 | 1197 | -5,99920116169307  -8,06626343609847 | -9,17170610789364 |

|  |  |  |
| --- | --- | --- |
| N | Точность решения | Func Iter |
| 10 | -2,32153851698058E-10 | 6155 |
| 100 | -1,23547394537127E-10 | 89871 |
| 1e3 | -1,03383968053095E-11 | 934496 |
| 1e4 | -2,54413379252583E-10 | 8823316 |
| 1e5 | -5,60255841719481E-10 | 88159386 |
| 1e6 | -7,73326291891863E-10 | 881814041 |
| 1e7 | -5,45927747452879E-10 | 44346654 |

* Вывод

1. **Текст программы**

#region lab4

/// <summary>

/// Алгоритм 2 из лабораторной работы #4

/// </summary>

/// <param name="func"> функция где ищится минимум </param>

/// <param name="m"> Количество попыток найти минимум </param>

/// <returns> Найденный минимум </returns>

static public Point RandomAlgaritm(Func<Point, double> func, int m)

{

Random rand = new Random();

Point min = new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10);

Point y;

int n = 0;

if (n < m)

{

y = new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10);

if (func(min) < func(y))

n++;

else

{

min = y;

n = 0;

}

}

Console.WriteLine($"Func: {func(min)}");

return min;

}

/// <summary>

/// Алгоритм 1 из лабораторной работы #4

/// </summary>

/// <param name="func"> функция где ищится минимум </param>

/// <param name="m"> Количество попыток найти минимум </param>

/// <param name="h"> Штрафные функции </param>

/// <returns> Найденный минимум </returns>

static public Point Algoritm1(Func<Point, double> func, int m, List<Func<Point, double>> h)

{

Random rand = new Random();

Point x = new Point(rand.NextDouble()\* 20 - 10, rand.NextDouble() \* 20 - 10);

Point min = Solve\_Penalty\_function(func, h, x);

//Point min = MSG\_Fletchera\_Riversa(func, x);

Point y;

double fmin = func(min);

FuncIters++;

int n = 0;

while(n < m)

{

y = Solve\_Penalty\_function(func, h, new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10));

//MSG\_Fletchera\_Riversa(func, new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10));

FuncIters += 2;

if ( fmin > func(y))

{

min = y;

fmin = func(min);

FuncIters++;

}

else

n++;

Console.Write($"\r {n} из {m} - {100.0 \* n / m}% F: {fmin} ");

}

Console.WriteLine();

Console.WriteLine($"Точность решения: {-9.17170610839834 - func(min)} Func iter: {FuncIters}");

return min;

}

/// <summary>

/// Алгоритм 2 из лабораторной работы #4

/// </summary>

/// <param name="func"> функция где ищится минимум </param>

/// <param name="m"> Количество попыток найти минимум </param>

/// <param name="h"> Штрафные функции </param>

/// <returns> Найденный минимум </returns>

static public Point Algoritm2(Func<Point, double> func, int m, List<Func<Point, double>> h)

{

Random rand = new Random();

Point x = new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10);

Point min = Solve\_Penalty\_function(func, h, x);

//Point min = MSG\_Fletchera\_Riversa(func, x);

Point y;

double fmin = func(min);

FuncIters++;

int n = 0;

while (n < m)

{

y = new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10);

FuncIters++;

if (fmin < func(y))

n++;

else

{

min = Solve\_Penalty\_function(func, h, y);

fmin = func(min);

FuncIters++;

n = 0;

}

Console.Write($"\r {n} из {m} - {100.0 \* n / m}% F: {fmin} ");

}

Console.WriteLine();

Console.WriteLine($"Точность решения: {-9.17170610839834 - func(min)} Func iter: {FuncIters}");

return min;

}

/// <summary>

/// Алгоритм 3 Из лаюораторной работы #4

/// </summary>

/// <param name="func"> Функция где ищится минимум </param>

/// <param name="m"> Количество попыток </param>

/// <param name="h"> Штрафные функции </param>

/// <param name="dx"> Длина шага </param>

/// <returns> Найденную точку минимума </returns>

static public Point Algoritm3(Func<Point, double> func, int m, List<Func<Point, double>> h, double dx)

{

Random rand = new Random();

Point x = new Point(rand.NextDouble() \* 20 - 10, rand.NextDouble() \* 20 - 10);

Point min = Solve\_Penalty\_function(func, h, x);

Point S = new Point(); // Квадрат x найправления в котором ищим минимум

//Point min = MSG\_Fletchera\_Riversa(func, x);

Point y;

double fmin = func(min); //Значение функции в найденном минимуме

FuncIters++;

int w;

bool f; //Показывает вышла ли точка за пределы области

int n = 0;

while (n < m)

{

f = true;

S.x = rand.NextDouble();

S.y = dx \* Math.Sqrt(1 - S.x);

S.x = dx \* Math.Sqrt(S.x);

w = 1;

do

{

// Console.WriteLine($"Func: {fmin}, {func(min + w \* S)}");

FuncIters++;

if(fmin > func(min + w \* S))

{

min = Solve\_Penalty\_function(func, h, min + w \* S);

fmin = func(min);

FuncIters++;

n = 0;

break;

}

for (int i = 0; i < h.Count; i++)

if (h[i](min + w \* S) > 0)

{

n++;

f = false;

break;

}

w++;

}while (f);

if (f)

n++;

Console.Write($"\r {n} из {m} - {100.0 \* n / m}% F: {fmin} ");

//Console.WriteLine($"Func: {func(min)}, {min.x}, {min.y}");

}

Console.WriteLine();

Console.WriteLine($"Точность решения: {-9.17170610839834 - func(min)} Func iter: {FuncIters}");

//Console.WriteLine($"Func: {func(min)}");

return min;

}

#endregion

#region 4 lab

Solver.EPS = 1e-5; //НЕ вводим, это зависит как точно у нас будут считать алгоритмы

//Ограничения

List<Func<Solver.Point, double>> h = new List<Func<Solver.Point, double>>();

h.Add(p1 => Math.Max(p1.y - 10, 0));

h.Add(p1 => Math.Max(-p1.y - 10, 0));

h.Add(p1 => Math.Max(p1.x - 10, 0));

h.Add(p1 => Math.Max(-p1.x - 10, 0));

//Наща функция

double[] C = { 6.0, 2.0, 4.0, 2.0, 8.0, 8.0 };

double[] A = { -3.0, 4.0, -8.0, -6.0, 3.0, -6.0 };

double[] B = { 9.0, -7.0, 3.0, -9.0, -2.0, -8.0 };

Func<Solver.Point, double> func = p1 =>

{

double f = 0;

for (int i = 0; i < 6; i++)

f -= C[i] / (1 + Math.Pow(p1.x - A[i], 2) + Math.Pow(p1.y - B[i], 2));

return f;

};

//Это мы вводим в 1 исследовании

double[] EPS = { 5, 1, 0.5, 1e-1, 5e-2, 1e-2, 5e-3, 1e-3};

double[] pp = { 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95 };

Console.WriteLine("Алгоритм 2");

for (int i = 0; i < EPS.Length; i++)

{

double eps = EPS[i];

double P = 0.8;

//Чисто испытаний

int m = (int)Math.Log(1.0 - P, 1.0 - eps \* eps / 400.0) + 1; //1 исследование

//int m = 1000; //Вводим во 2 исследовании

//Solver.Point p = Solver.RandomAlgaritm(func, m);

//Раскоменичваем ОДИН из этих методов

//Solver.Point p = Solver.Algoritm1(func, m, h);

Solver.Point p = Solver.Algoritm2(func, m, h);

//Solver.Point p = Solver.Algoritm3(func, m, h, 0.5);

int pad = 22;

Console.WriteLine("Eps:".PadLeft(pad) + "P:".PadLeft(pad) + "N:".PadLeft(pad) + "".PadLeft(pad) + "(x\*, y\*)".PadRight(pad) + "f(x\*, y\*)".PadLeft(pad));

Console.WriteLine(eps.ToString().PadLeft(pad) + P.ToString().PadLeft(pad) + m.ToString().PadLeft(pad)

+ p.x.ToString().PadLeft(pad) + p.y.ToString().PadLeft(pad) + func(p).ToString().PadLeft(pad));

}

Console.WriteLine("--------------------------------");

for (int j = 0; j < pp.Length; j++)

{

double P = pp[j];

double eps = 1e-2;

//Чисто испытаний

int m = (int)Math.Log(1.0 - P, 1.0 - eps \* eps / 400.0) + 1; //1 исследование

//int m = 1000; //Вводим во 2 исследовании

//Solver.Point p = Solver.RandomAlgaritm(func, m);

//Раскоменичваем ОДИН из этих методов

//Solver.Point p = Solver.Algoritm1(func, m, h);

Solver.Point p = Solver.Algoritm2(func, m, h);

//Solver.Point p = Solver.Algoritm3(func, m, h, 0.5);

int pad = 22;

Console.WriteLine("Eps:".PadLeft(pad) + "P:".PadLeft(pad) + "N:".PadLeft(pad) + "".PadLeft(pad) + "(x\*, y\*)".PadRight(pad) + "f(x\*, y\*)".PadLeft(pad));

Console.WriteLine(eps.ToString().PadLeft(pad) + P.ToString().PadLeft(pad) + m.ToString().PadLeft(pad)

+ p.x.ToString().PadLeft(pad) + p.y.ToString().PadLeft(pad) + func(p).ToString().PadLeft(pad));

}

Console.WriteLine("Алгоритм 3");

for (int i = 0; i < EPS.Length; i++)

{

double eps = EPS[i];

double P = 0.8;

//Чисто испытаний

int m = (int)Math.Log(1.0 - P, 1.0 - eps \* eps / 400.0) + 1; //1 исследование

//int m = 1000; //Вводим во 2 исследовании

//Solver.Point p = Solver.RandomAlgaritm(func, m);

//Раскоменичваем ОДИН из этих методов

//Solver.Point p = Solver.Algoritm1(func, m, h);

//Solver.Point p = Solver.Algoritm2(func, m, h);

Solver.Point p = Solver.Algoritm3(func, m, h, 0.5);

int pad = 22;

Console.WriteLine("Eps:".PadLeft(pad) + "P:".PadLeft(pad) + "N:".PadLeft(pad) + "".PadLeft(pad) + "(x\*, y\*)".PadRight(pad) + "f(x\*, y\*)".PadLeft(pad));

Console.WriteLine(eps.ToString().PadLeft(pad) + P.ToString().PadLeft(pad) + m.ToString().PadLeft(pad)

+ p.x.ToString().PadLeft(pad) + p.y.ToString().PadLeft(pad) + func(p).ToString().PadLeft(pad));

}

Console.WriteLine("--------------------------------");

for (int j = 0; j < pp.Length; j++)

{

double P = pp[j];

double eps = 1e-2;

//Чисто испытаний

int m = (int)Math.Log(1.0 - P, 1.0 - eps \* eps / 400.0) + 1; //1 исследование

//int m = 1000; //Вводим во 2 исследовании

//Solver.Point p = Solver.RandomAlgaritm(func, m);

//Раскоменичваем ОДИН из этих методов

//Solver.Point p = Solver.Algoritm1(func, m, h);

//Solver.Point p = Solver.Algoritm2(func, m, h);

Solver.Point p = Solver.Algoritm3(func, m, h, 0.5);

int pad = 22;

Console.WriteLine("Eps:".PadLeft(pad) + "P:".PadLeft(pad) + "N:".PadLeft(pad) + "".PadLeft(pad) + "(x\*, y\*)".PadRight(pad) + "f(x\*, y\*)".PadLeft(pad));

Console.WriteLine(eps.ToString().PadLeft(pad) + P.ToString().PadLeft(pad) + m.ToString().PadLeft(pad)

+ p.x.ToString().PadLeft(pad) + p.y.ToString().PadLeft(pad) + func(p).ToString().PadLeft(pad));

}

#endregion