

Dataset 1

x1	x2	x3	y
0.5	0.1	0.7	1
0.8	0.2	0.3	1
0.1	0.3	0.4	-1
0.7	0.4	0.2	1
0.2	0.5	0.5	1
0.4	0.6	0.6	-1
0.1	0.7	0.9	-1
0.9	0.8	0.1	-1
0.6	0.9	0.8	1
0.8	0.1	0.4	1

1. Primal Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset1\Primal.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset1\Primal.lg4

C++ code output (including LINGO code script):

```
// Dataset:
```

x1	x2	x3	y
0.5	0.1	0.7	1
0.8	0.2	0.3	1
0.1	0.3	0.4	-1
0.7	0.4	0.2	1
0.2	0.5	0.5	1
0.4	0.6	0.6	-1
0.1	0.7	0.9	-1
0.9	0.8	0.1	-1
0.6	0.9	0.8	1
0.8	0.1	0.4	1

```
c = 5
```

```
// LINGO input code script:
```

```
MIN = 0.5 * ( w1 * w1 + w2 * w2 + w3 * w3 ) + 5 * ( q1 + q2 + q3 + q4 + q5 + q6 + q7 + q8 + q9 + q10 );
```

```
1 * ( w1 * 0.5 + w2 * 0.1 + w3 * 0.7 + b ) + q1 >= 1;  
1 * ( w1 * 0.8 + w2 * 0.2 + w3 * 0.3 + b ) + q2 >= 1;  
-1 * ( w1 * 0.1 + w2 * 0.3 + w3 * 0.4 + b ) + q3 >= 1;  
1 * ( w1 * 0.7 + w2 * 0.4 + w3 * 0.2 + b ) + q4 >= 1;  
1 * ( w1 * 0.2 + w2 * 0.5 + w3 * 0.5 + b ) + q5 >= 1;  
-1 * ( w1 * 0.4 + w2 * 0.6 + w3 * 0.6 + b ) + q6 >= 1;  
-1 * ( w1 * 0.1 + w2 * 0.7 + w3 * 0.9 + b ) + q7 >= 1;  
-1 * ( w1 * 0.9 + w2 * 0.8 + w3 * 0.1 + b ) + q8 >= 1;  
1 * ( w1 * 0.6 + w2 * 0.9 + w3 * 0.8 + b ) + q9 >= 1;  
1 * ( w1 * 0.8 + w2 * 0.1 + w3 * 0.4 + b ) + q10 >= 1;
```

```
@FREE(w1);
```

```
@FREE(w2);
```

```
@FREE(w3);
```

```
@FREE(b);
```

// LINGO result:

w1 = 2.28571

w2 = -2.17857

w3 = 1.10714

b = -0.7

q1 = 0

q2 = 0

q4 = 0.75

q5 = 1.77857

q6 = 0.571429

q7 = 0

q8 = 0.725

q9 = 1.40357

q10 = 0

2. Dual Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset1\Dual.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset1\Dual.lg4

C++ code output (including LINGO code script):

```
// Dataset:
```

x1	x2	x3	y
0.5	0.1	0.7	1
0.8	0.2	0.3	1
0.1	0.3	0.4	-1
0.7	0.4	0.2	1
0.2	0.5	0.5	1
0.4	0.6	0.6	-1
0.1	0.7	0.9	-1
0.9	0.8	0.1	-1
0.6	0.9	0.8	1
0.8	0.1	0.4	1

c = 5

```
// LINGO input code script:
```

```
MIN = 0.5 * ( lambda_1 * lambda_1 * 0.75 + lambda_1 * lambda_2 * 0.63 + lambda_1 *  
lambda_3 * -0.36 + lambda_1 * lambda_4 * 0.53 + lambda_1 * lambda_5 * 0.5 + lambda_1 *  
lambda_6 * -0.68 + lambda_1 * lambda_7 * -0.75 + lambda_1 * lambda_8 * -0.6 +  
lambda_1 * lambda_9 * 0.95 + lambda_1 * lambda_10 * 0.69 + lambda_2 * lambda_1 *  
0.63 + lambda_2 * lambda_2 * 0.77 + lambda_2 * lambda_3 * -0.26 + lambda_2 * lambda_4  
* 0.7 + lambda_2 * lambda_5 * 0.41 + lambda_2 * lambda_6 * -0.62 + lambda_2 *  
lambda_7 * -0.49 + lambda_2 * lambda_8 * -0.91 + lambda_2 * lambda_9 * 0.9 + lambda_2  
* lambda_10 * 0.78 + lambda_3 * lambda_1 * -0.36 + lambda_3 * lambda_2 * -0.26 +  
lambda_3 * lambda_3 * 0.26 + lambda_3 * lambda_4 * -0.27 + lambda_3 * lambda_5 * -  
0.37 + lambda_3 * lambda_6 * 0.46 + lambda_3 * lambda_7 * 0.58 + lambda_3 * lambda_8  
* 0.37 + lambda_3 * lambda_9 * -0.65 + lambda_3 * lambda_10 * -0.27 + lambda_4 *  
lambda_1 * 0.53 + lambda_4 * lambda_2 * 0.7 + lambda_4 * lambda_3 * -0.27 + lambda_4  
* lambda_4 * 0.69 + lambda_4 * lambda_5 * 0.44 + lambda_4 * lambda_6 * -0.64 +  
lambda_4 * lambda_7 * -0.53 + lambda_4 * lambda_8 * -0.97 + lambda_4 * lambda_9 *  
0.94 + lambda_4 * lambda_10 * 0.68 + lambda_5 * lambda_1 * 0.5 + lambda_5 * lambda_2  
* 0.41 + lambda_5 * lambda_3 * -0.37 + lambda_5 * lambda_4 * 0.44 + lambda_5 *  
lambda_5 * 0.54 + lambda_5 * lambda_6 * -0.68 + lambda_5 * lambda_7 * -0.82 +  
lambda_5 * lambda_8 * -0.63 + lambda_5 * lambda_9 * 0.97 + lambda_5 * lambda_10 *
```

$$\begin{aligned}
&0.41 + \lambda_6 * \lambda_1 * -0.68 + \lambda_6 * \lambda_2 * -0.62 + \lambda_6 * \\
&\lambda_3 * 0.46 + \lambda_6 * \lambda_4 * -0.64 + \lambda_6 * \lambda_5 * -0.68 + \\
&\lambda_6 * \lambda_6 * 0.88 + \lambda_6 * \lambda_7 * 1 + \lambda_6 * \lambda_8 * 0.9 + \\
&\lambda_6 * \lambda_9 * -1.26 + \lambda_6 * \lambda_{10} * -0.62 + \lambda_7 * \lambda_1 * - \\
&0.75 + \lambda_7 * \lambda_2 * -0.49 + \lambda_7 * \lambda_3 * 0.58 + \lambda_7 * \lambda_4 \\
&* -0.53 + \lambda_7 * \lambda_5 * -0.82 + \lambda_7 * \lambda_6 * 1 + \lambda_7 * \lambda_7 \\
&* 1.31 + \lambda_7 * \lambda_8 * 0.74 + \lambda_7 * \lambda_9 * -1.41 + \lambda_7 * \\
&\lambda_{10} * -0.51 + \lambda_8 * \lambda_1 * -0.6 + \lambda_8 * \lambda_2 * -0.91 + \\
&\lambda_8 * \lambda_3 * 0.37 + \lambda_8 * \lambda_4 * -0.97 + \lambda_8 * \lambda_5 * - \\
&0.63 + \lambda_8 * \lambda_6 * 0.9 + \lambda_8 * \lambda_7 * 0.74 + \lambda_8 * \lambda_8 * \\
&1.46 + \lambda_8 * \lambda_9 * -1.34 + \lambda_8 * \lambda_{10} * -0.84 + \lambda_9 * \\
&\lambda_1 * 0.95 + \lambda_9 * \lambda_2 * 0.9 + \lambda_9 * \lambda_3 * -0.65 + \lambda_9 \\
&* \lambda_4 * 0.94 + \lambda_9 * \lambda_5 * 0.97 + \lambda_9 * \lambda_6 * -1.26 + \\
&\lambda_9 * \lambda_7 * -1.41 + \lambda_9 * \lambda_8 * -1.34 + \lambda_9 * \lambda_9 * \\
&1.81 + \lambda_9 * \lambda_{10} * 0.89 + \lambda_{10} * \lambda_1 * 0.69 + \lambda_{10} * \\
&\lambda_2 * 0.78 + \lambda_{10} * \lambda_3 * -0.27 + \lambda_{10} * \lambda_4 * 0.68 + \\
&\lambda_{10} * \lambda_5 * 0.41 + \lambda_{10} * \lambda_6 * -0.62 + \lambda_{10} * \lambda_7 * - \\
&0.51 + \lambda_{10} * \lambda_8 * -0.84 + \lambda_{10} * \lambda_9 * 0.89 + \lambda_{10} * \\
&\lambda_{10} * 0.81) - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \\
&\lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10});
\end{aligned}$$

$$\begin{aligned}
&\lambda_1 * 1 + \lambda_2 * 1 + \lambda_3 * -1 + \lambda_4 * 1 + \lambda_5 * 1 + \lambda_6 * \\
&-1 + \lambda_7 * -1 + \lambda_8 * -1 + \lambda_9 * 1 + \lambda_{10} * 1 = 0;
\end{aligned}$$

```

@BND(0, lambda_1, 5);
@BND(0, lambda_2, 5);
@BND(0, lambda_3, 5);
@BND(0, lambda_4, 5);
@BND(0, lambda_5, 5);
@BND(0, lambda_6, 5);
@BND(0, lambda_7, 5);
@BND(0, lambda_8, 5);
@BND(0, lambda_9, 5);
@BND(0, lambda_10, 5);

```

// LINGO result:

```

lambda_1 = 4.46428
lambda_2 = 7.16208e-07
lambda_3 = 5
lambda_4 = 5
lambda_5 = 5
lambda_6 = 5
lambda_7 = 4.46428
lambda_8 = 5

```

$\lambda_9 = 5$
 $\lambda_{10} = 2.81948e-08$

// Post-processing of LINGO result:

$w_1 = 2.28571$
 $w_2 = -2.17857$
 $w_3 = 1.10714$
 $b = -0.7$
 $q_1 = 0$
 $q_2 = 0$
 $q_3 = 0.317857$
 $q_4 = 0.75$
 $q_5 = 1.77857$
 $q_6 = 0.571429$
 $q_7 = 0$
 $q_8 = 0.725$
 $q_9 = 1.40357$
 $q_{10} = 0$

Dataset 2

x1	x2	x3	y
1	2	3	1
5	4	2	1
8	6	2	1
9	5	7	1
6	8	6	1
5	4	5	-1
2	6	4	-1
4	7	1	-1
3	5	8	-1
8	1	6	-1

1. Primal Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset2\Primal.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset2\Primal.lg4

C++ code output (including LINGO code script):

// Dataset:

x1	x2	x3	y
1	2	3	1
5	4	2	1
8	6	2	1
9	5	7	1
6	8	6	1
5	4	5	-1
2	6	4	-1
4	7	1	-1
3	5	8	-1
8	1	6	-1

c = 5

// LINGO input code script:

MIN = 0.5 * (w1 * w1 + w2 * w2 + w3 * w3) + 5 * (q1 + q2 + q3 + q4 + q5 + q6 + q7 + q8 + q9 + q10);

1 * (w1 * 1 + w2 * 2 + w3 * 3 + b) + q1 >= 1;
1 * (w1 * 5 + w2 * 4 + w3 * 2 + b) + q2 >= 1;
1 * (w1 * 8 + w2 * 6 + w3 * 2 + b) + q3 >= 1;
1 * (w1 * 9 + w2 * 5 + w3 * 7 + b) + q4 >= 1;
1 * (w1 * 6 + w2 * 8 + w3 * 6 + b) + q5 >= 1;
-1 * (w1 * 5 + w2 * 4 + w3 * 5 + b) + q6 >= 1;
-1 * (w1 * 2 + w2 * 6 + w3 * 4 + b) + q7 >= 1;
-1 * (w1 * 4 + w2 * 7 + w3 * 1 + b) + q8 >= 1;
-1 * (w1 * 3 + w2 * 5 + w3 * 8 + b) + q9 >= 1;
-1 * (w1 * 8 + w2 * 1 + w3 * 6 + b) + q10 >= 1;

@FREE(w1);

@FREE(w2);

@FREE(w3);

@FREE(b);

// LINGO result:

w1 = 0.333333

w2 = 0.333333

w3 = 0

b = -3.66667

q1 = 3.66667

q2 = 1.66667

q3 = 0

q4 = 0

q5 = 0

q6 = 0.333333

q7 = 0

q8 = 1

q9 = 0

q10 = 0.333333

2. Dual Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset2\Dual.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset2\Dual.lg4

C++ code output (including LINGO code script):

// Dataset:

x1	x2	x3	y
1	2	3	1
5	4	2	1
8	6	2	1
9	5	7	1
6	8	6	1
5	4	5	-1
2	6	4	-1
4	7	1	-1
3	5	8	-1
8	1	6	-1

c = 5

// LINGO input code script:

MIN = 0.5 * (lambda_1 * lambda_1 * 14 + lambda_1 * lambda_2 * 19 + lambda_1 *
lambda_3 * 26 + lambda_1 * lambda_4 * 40 + lambda_1 * lambda_5 * 40 + lambda_1 *
lambda_6 * -28 + lambda_1 * lambda_7 * -26 + lambda_1 * lambda_8 * -21 + lambda_1 *
lambda_9 * -37 + lambda_1 * lambda_10 * -28 + lambda_2 * lambda_1 * 19 + lambda_2 *
lambda_2 * 45 + lambda_2 * lambda_3 * 68 + lambda_2 * lambda_4 * 79 + lambda_2 *
lambda_5 * 74 + lambda_2 * lambda_6 * -51 + lambda_2 * lambda_7 * -42 + lambda_2 *
lambda_8 * -50 + lambda_2 * lambda_9 * -51 + lambda_2 * lambda_10 * -56 + lambda_3 *
lambda_1 * 26 + lambda_3 * lambda_2 * 68 + lambda_3 * lambda_3 * 104 + lambda_3 *
lambda_4 * 116 + lambda_3 * lambda_5 * 108 + lambda_3 * lambda_6 * -74 + lambda_3 *
lambda_7 * -60 + lambda_3 * lambda_8 * -76 + lambda_3 * lambda_9 * -70 + lambda_3 *
lambda_10 * -82 + lambda_4 * lambda_1 * 40 + lambda_4 * lambda_2 * 79 + lambda_4 *
lambda_3 * 116 + lambda_4 * lambda_4 * 155 + lambda_4 * lambda_5 * 136 + lambda_4 *
lambda_6 * -100 + lambda_4 * lambda_7 * -76 + lambda_4 * lambda_8 * -78 + lambda_4 *
lambda_9 * -108 + lambda_4 * lambda_10 * -119 + lambda_5 * lambda_1 * 40 + lambda_5
* lambda_2 * 74 + lambda_5 * lambda_3 * 108 + lambda_5 * lambda_4 * 136 + lambda_5 *
lambda_5 * 136 + lambda_5 * lambda_6 * -92 + lambda_5 * lambda_7 * -84 + lambda_5 *
lambda_8 * -86 + lambda_5 * lambda_9 * -106 + lambda_5 * lambda_10 * -92 + lambda_6 *
lambda_1 * -28 + lambda_6 * lambda_2 * -51 + lambda_6 * lambda_3 * -74 + lambda_6 *

```

lambda_4 * -100 + lambda_6 * lambda_5 * -92 + lambda_6 * lambda_6 * 66 + lambda_6 *
lambda_7 * 54 + lambda_6 * lambda_8 * 53 + lambda_6 * lambda_9 * 75 + lambda_6 *
lambda_10 * 74 + lambda_7 * lambda_1 * -26 + lambda_7 * lambda_2 * -42 + lambda_7 *
lambda_3 * -60 + lambda_7 * lambda_4 * -76 + lambda_7 * lambda_5 * -84 + lambda_7 *
lambda_6 * 54 + lambda_7 * lambda_7 * 56 + lambda_7 * lambda_8 * 54 + lambda_7 *
lambda_9 * 68 + lambda_7 * lambda_10 * 46 + lambda_8 * lambda_1 * -21 + lambda_8 *
lambda_2 * -50 + lambda_8 * lambda_3 * -76 + lambda_8 * lambda_4 * -78 + lambda_8 *
lambda_5 * -86 + lambda_8 * lambda_6 * 53 + lambda_8 * lambda_7 * 54 + lambda_8 *
lambda_8 * 66 + lambda_8 * lambda_9 * 55 + lambda_8 * lambda_10 * 45 + lambda_9 *
lambda_1 * -37 + lambda_9 * lambda_2 * -51 + lambda_9 * lambda_3 * -70 + lambda_9 *
lambda_4 * -108 + lambda_9 * lambda_5 * -106 + lambda_9 * lambda_6 * 75 + lambda_9 *
lambda_7 * 68 + lambda_9 * lambda_8 * 55 + lambda_9 * lambda_9 * 98 + lambda_9 *
lambda_10 * 77 + lambda_10 * lambda_1 * -28 + lambda_10 * lambda_2 * -56 + lambda_10
* lambda_3 * -82 + lambda_10 * lambda_4 * -119 + lambda_10 * lambda_5 * -92 +
lambda_10 * lambda_6 * 74 + lambda_10 * lambda_7 * 46 + lambda_10 * lambda_8 * 45 +
lambda_10 * lambda_9 * 77 + lambda_10 * lambda_10 * 101 ) - ( lambda_1 + lambda_2 +
lambda_3 + lambda_4 + lambda_5 + lambda_6 + lambda_7 + lambda_8 + lambda_9 +
lambda_10 );

```

```

lambda_1 * 1 + lambda_2 * 1 + lambda_3 * 1 + lambda_4 * 1 + lambda_5 * 1 + lambda_6 * -
1 + lambda_7 * -1 + lambda_8 * -1 + lambda_9 * -1 + lambda_10 * -1 = 0;

```

```

@BND(0, lambda_1, 5);
@BND(0, lambda_2, 5);
@BND(0, lambda_3, 5);
@BND(0, lambda_4, 5);
@BND(0, lambda_5, 5);
@BND(0, lambda_6, 5);
@BND(0, lambda_7, 5);
@BND(0, lambda_8, 5);
@BND(0, lambda_9, 5);
@BND(0, lambda_10, 5);

```

```

// LINGO result:

```

```

lambda_1 = 5
lambda_2 = 5
lambda_3 = 0.796425
lambda_4 = 4.56552
lambda_5 = 2.24917
lambda_6 = 5
lambda_7 = 2.2106
lambda_8 = 5
lambda_9 = 0.40051
lambda_10 = 5

```

// Post-processing of LINGO result:

w1 = 0.333328

w2 = 0.33333

w3 = -5.72205e-06

b = -3.66659

q1 = 3.66662

q2 = 1.66664

q3 = 0

q4 = 0

q5 = 0

q6 = 0.333344

q7 = 0

q8 = 1.00003

q9 = 0

q10 = 0.333332

Dataset 3

x1	x2	x3	y
14	47	92	1
29	65	24	-1
63	92	14	1
85	24	85	-1
54	59	37	1
56	77	12	-1
43	18	68	1
74	66	79	-1
37	83	51	1
99	34	47	-1

1. Primal Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset3\Primal.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset3\Primal.lg4

C++ code output (including LINGO code script):

```
// Dataset:
```

x1	x2	x3	y
14	47	92	1
29	65	24	-1
63	92	14	1
85	24	85	-1
54	59	37	1
56	77	12	-1
43	18	68	1
74	66	79	-1
37	83	51	1
99	34	47	-1

```
c = 5
```

```
// LINGO input code script:
```

```
MIN = 0.5 * ( w1 * w1 + w2 * w2 + w3 * w3 ) + 5 * ( q1 + q2 + q3 + q4 + q5 + q6 + q7 + q8 + q9 + q10 );
```

```
1 * ( w1 * 14 + w2 * 47 + w3 * 92 + b ) + q1 >= 1;  
-1 * ( w1 * 29 + w2 * 65 + w3 * 24 + b ) + q2 >= 1;  
1 * ( w1 * 63 + w2 * 92 + w3 * 14 + b ) + q3 >= 1;  
-1 * ( w1 * 85 + w2 * 24 + w3 * 85 + b ) + q4 >= 1;  
1 * ( w1 * 54 + w2 * 59 + w3 * 37 + b ) + q5 >= 1;  
-1 * ( w1 * 56 + w2 * 77 + w3 * 12 + b ) + q6 >= 1;  
1 * ( w1 * 43 + w2 * 18 + w3 * 68 + b ) + q7 >= 1;  
-1 * ( w1 * 74 + w2 * 66 + w3 * 79 + b ) + q8 >= 1;  
1 * ( w1 * 37 + w2 * 83 + w3 * 51 + b ) + q9 >= 1;  
-1 * ( w1 * 99 + w2 * 34 + w3 * 47 + b ) + q10 >= 1;
```

```
@FREE(w1);
```

```
@FREE(w2);
```

```
@FREE(w3);
```

```
@FREE(b);
```

```
// LINGO result:  
w1 = -0.0414132  
w2 = -0.00973778  
w3 = -0.0226163  
b = 4.49395  
q1 = 0  
q2 = 3.11722  
q3 = 0.327581  
q4 = 0  
q5 = 0.15369  
q6 = 2.15361  
q7 = 0  
q8 = 0  
q9 = 0  
q10 = 0
```

2. Dual Formulation:

C++ code file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset3\Dual.cpp

LINGO file: \Pulkit_Singal_2023AIB2064\Using_LINGO_and_cpp\Dataset3\Dual.lg4

C++ code output (including LINGO code script):

// Dataset:

x1	x2	x3	y
14	47	92	1
29	65	24	-1
63	92	14	1
85	24	85	-1
54	59	37	1
56	77	12	-1
43	18	68	1
74	66	79	-1
37	83	51	1
99	34	47	-1

c = 5

// LINGO input code script:

MIN = 0.5 * (lambda_1 * lambda_1 * 10869 + lambda_1 * lambda_2 * -5669 + lambda_1 * lambda_3 * 6494 + lambda_1 * lambda_4 * -10138 + lambda_1 * lambda_5 * 6933 + lambda_1 * lambda_6 * -5507 + lambda_1 * lambda_7 * 7704 + lambda_1 * lambda_8 * -11406 + lambda_1 * lambda_9 * 9111 + lambda_1 * lambda_10 * -7308 + lambda_2 * lambda_1 * -5669 + lambda_2 * lambda_2 * 5642 + lambda_2 * lambda_3 * -8143 + lambda_2 * lambda_4 * 6065 + lambda_2 * lambda_5 * -6289 + lambda_2 * lambda_6 * 6917 + lambda_2 * lambda_7 * -4049 + lambda_2 * lambda_8 * 8332 + lambda_2 * lambda_9 * -7692 + lambda_2 * lambda_10 * 6209 + lambda_3 * lambda_1 * 6494 + lambda_3 * lambda_2 * -8143 + lambda_3 * lambda_3 * 12629 + lambda_3 * lambda_4 * -8753 + lambda_3 * lambda_5 * 9348 + lambda_3 * lambda_6 * -10780 + lambda_3 * lambda_7 * 5317 + lambda_3 * lambda_8 * -11840 + lambda_3 * lambda_9 * 10681 + lambda_3 * lambda_10 * -10023 + lambda_4 * lambda_1 * -10138 + lambda_4 * lambda_2 * 6065 + lambda_4 * lambda_3 * -8753 + lambda_4 * lambda_4 * 15026 + lambda_4 * lambda_5 * -9151 + lambda_4 * lambda_6 * 7628 + lambda_4 * lambda_7 * -9867 + lambda_4 * lambda_8 * 14589 + lambda_4 * lambda_9 * -9472 + lambda_4 * lambda_10 * 13226 + lambda_5 * lambda_1 * 6933 + lambda_5 * lambda_2 * -6289 + lambda_5 * lambda_3 * 9348 + lambda_5 * lambda_4 * -9151 + lambda_5 * lambda_5 * 7766 + lambda_5 * lambda_6 * -8011 + lambda_5 * lambda_7 * 5900 + lambda_5 * lambda_8 * -

$$\begin{aligned}
&10813 + \lambda_5 * \lambda_9 * 8782 + \lambda_5 * \lambda_{10} * -9091 + \lambda_6 * \\
&\lambda_1 * -5507 + \lambda_6 * \lambda_2 * 6917 + \lambda_6 * \lambda_3 * -10780 + \\
&\lambda_6 * \lambda_4 * 7628 + \lambda_6 * \lambda_5 * -8011 + \lambda_6 * \lambda_6 * \\
&9209 + \lambda_6 * \lambda_7 * -4610 + \lambda_6 * \lambda_8 * 10174 + \lambda_6 * \\
&\lambda_9 * -9075 + \lambda_6 * \lambda_{10} * 8726 + \lambda_7 * \lambda_1 * 7704 + \\
&\lambda_7 * \lambda_2 * -4049 + \lambda_7 * \lambda_3 * 5317 + \lambda_7 * \lambda_4 * - \\
&9867 + \lambda_7 * \lambda_5 * 5900 + \lambda_7 * \lambda_6 * -4610 + \lambda_7 * \\
&\lambda_7 * 6797 + \lambda_7 * \lambda_8 * -9742 + \lambda_7 * \lambda_9 * 6553 + \\
&\lambda_7 * \lambda_{10} * -8065 + \lambda_8 * \lambda_1 * -11406 + \lambda_8 * \lambda_2 * \\
&8332 + \lambda_8 * \lambda_3 * -11840 + \lambda_8 * \lambda_4 * 14589 + \lambda_8 * \\
&\lambda_5 * -10813 + \lambda_8 * \lambda_6 * 10174 + \lambda_8 * \lambda_7 * -9742 + \\
&\lambda_8 * \lambda_8 * 16073 + \lambda_8 * \lambda_9 * -12245 + \lambda_8 * \lambda_{10} * \\
&13283 + \lambda_9 * \lambda_1 * 9111 + \lambda_9 * \lambda_2 * -7692 + \lambda_9 * \\
&\lambda_3 * 10681 + \lambda_9 * \lambda_4 * -9472 + \lambda_9 * \lambda_5 * 8782 + \\
&\lambda_9 * \lambda_6 * -9075 + \lambda_9 * \lambda_7 * 6553 + \lambda_9 * \lambda_8 * - \\
&12245 + \lambda_9 * \lambda_9 * 10859 + \lambda_9 * \lambda_{10} * -8882 + \lambda_{10} * \\
&\lambda_1 * -7308 + \lambda_{10} * \lambda_2 * 6209 + \lambda_{10} * \lambda_3 * -10023 + \\
&\lambda_{10} * \lambda_4 * 13226 + \lambda_{10} * \lambda_5 * -9091 + \lambda_{10} * \lambda_6 * \\
&* 8726 + \lambda_{10} * \lambda_7 * -8065 + \lambda_{10} * \lambda_8 * 13283 + \lambda_{10} * \\
&\lambda_9 * -8882 + \lambda_{10} * \lambda_{10} * 13166) - (\lambda_1 + \lambda_2 + \\
&\lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \\
&\lambda_{10});
\end{aligned}$$

$$\begin{aligned}
&\lambda_1 * 1 + \lambda_2 * -1 + \lambda_3 * 1 + \lambda_4 * -1 + \lambda_5 * 1 + \lambda_6 * \\
&* -1 + \lambda_7 * 1 + \lambda_8 * -1 + \lambda_9 * 1 + \lambda_{10} * -1 = 0;
\end{aligned}$$

```

@BND(0, lambda_1, 5);
@BND(0, lambda_2, 5);
@BND(0, lambda_3, 5);
@BND(0, lambda_4, 5);
@BND(0, lambda_5, 5);
@BND(0, lambda_6, 5);
@BND(0, lambda_7, 5);
@BND(0, lambda_8, 5);
@BND(0, lambda_9, 5);
@BND(0, lambda_10, 5);

```

```

// LINGO result:
lambda_1 = 7.88398e-08
lambda_2 = 5
lambda_3 = 5
lambda_4 = 7.76932e-07
lambda_5 = 5
lambda_6 = 5

```

$\lambda_7 = 2.03843$
 $\lambda_8 = 3.97505$
 $\lambda_9 = 2.34299$
 $\lambda_{10} = 0.406372$

// Post-processing of LINGO result:

$w_1 = -0.0412827$
 $w_2 = -0.00965118$
 $w_3 = -0.022541$
 $b = 4.47715$
 $q_1 = 0$
 $q_2 = 3.11164$
 $q_3 = 0.327139$
 $q_4 = 0$
 $q_5 = 0.15555$
 $q_6 = 2.15169$
 $q_7 = 0$
 $q_8 = 0$
 $q_9 = 0$
 $q_{10} = 0$