Assignment 2

Ran Liu rliu361@gatech.edu
903515184

Problem 1

question 1

Indices: $i = 1, 2, ..., m$ for different boundary ids of the polyhedron P. $j = 1, 2, ..., n$ for different dimensions in space R^n.

Data: a_i and b_i together form the ith boundary of the polyhedron P. $A_{i,j}$ is a matrix where $a_i = A_{i,}$ and b_i is a vector.

Variables: $y \in R^n$ is the center of the ball we want that is inside the polyhedron. r is the radius of that ball.

Objective: maximize r

Constraints: $a_i y + r <= b_i$ for every i.

Bounds: r needs not to be negative numbers.

question 2

Result as shown in Figure 1.

```
Coefficient statistics:
  Matrix range     [3e-06, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+01]
Presolve time: 0.03s
Presolved: 101 rows, 1000 columns, 101000 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.000000e+00   9.999945e+08      0s
     542    1.4853767e+00   0.000000e+00   0.000000e+00      0s

Solved in 542 iterations and 0.16 seconds
Optimal objective  1.485376717e+00
```

Figure 1

Problem 2

question 1

Indices: $i = 1, 2, ..., M$ are the indices for oil fields. $j = 1, 2, ..., N$ are the indices for processing plants.

Data: c_i is the cost of extracting crude oil from oil field i, its unit: dollars per barrel. a_i is the percentage of sulfur from oil field i, each barrel of crude oil has a_i barrels of sulfur. m_i is the maximum of total crude extracted at field i, its unit: barrels. $f_{i,j}$ is the cost of transporting from oil field i to processing plant j, its unit: dollars per barrel. u_j is the maximum percentage of sulfur in the final product after blending in processing plant j. d_j is the minimum of total crude transported to processing plant j, its unit: barrels.

Variables: $x_{i,j}$ is how much oil to transport from field i to plant j, its unit: barrels.

Objective: minimize $\sum_{i=1}^{M}\sum_{j=1}^{N}(c_i + f_{i,j})x_{i,j}$

Constraints: $\sum_{i=1}^{M} x_{i,j}a_i <= u_j \sum_{i=1}^{M} x_{i,j}$ for every j, $\sum_{i=1}^{M} x_{i,j} >= d_j$ for every j, $\sum_{j=1}^{N} x_{i,j} <= m_i$ for every i.

Nonnegativity: $x_{i,j}$ needs not to be negative numbers.

question 2

Result as shown in Figure 2.

```
Barrier statistics:
 AA' NZ     : 1.991e+05
 Factor NZ  : 2.489e+05 (roughly 40 MBytes of memory)
 Factor Ops : 5.705e+07 (less than 1 second per iteration)
 Threads    : 3

                  Objective                Residual
Iter       Primal          Dual         Primal    Dual     Compl     Time
   0   7.12025725e+06  0.00000000e+00  7.04e+03 0.00e+00  1.28e+02     0s
   1   2.58562855e+05 -3.41859297e+04  1.86e+02 2.80e-01  5.02e+00     0s

Barrier performed 1 iterations in 0.24 seconds
Barrier solve interrupted - model solved by another algorithm


Solved with dual simplex
Solved in 538 iterations and 0.25 seconds
Optimal objective  1.362806158e+03
```

Figure 2

Problem 3

question 1

Indices: $t = 1, 2, ..., T$ for different months. $s = 1, 2, ..., T-1$, also for different months. Here $T = 12$. t has the same meaning with s except that $s = T$ does not exist.

Data: D_t is the demand for the additive in month t, its unit: lbs, L_s is the maximum of absolute difference between production in month s and month (s + 1), its unit: lbs, C_t is the cost of production in month t, its unit: dollars per lb, H_t is the cost of holding in month t, its unit: dollars per lb.

Variables: p_t is how much additive to produce in month t. s_t is how much additive that they still have in month t.

Objective: minimize $\sum_{t=1}^{T} p_t C_t + s_t H_t$

Constraints: $s_1 = 100 + p_1 - D_1$, $h_t = h_{t-1} * 0.95 + p_t - D_t$ for t except $t = 1$, $p_{s+1} - p_s <= L_s$, and $t = 1$, $p_{s+1} - p_s >= -L_s$.

Bounds: p_t and s_t all need not to be negative numbers.

question 2

Result as shown in Figure 3.

```
Model fingerprint: 0x0e1ce084
Coefficient statistics:
  Matrix range     [9e-01, 1e+00]
  Objective range  [7e-02, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [5e+00, 2e+02]
Presolve removed 23 rows and 12 columns
Presolve time: 0.00s
Presolved: 11 rows, 23 columns, 43 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    6.5536160e+02   3.430187e+01   0.000000e+00      0s
      12    8.0984260e+02   0.000000e+00   0.000000e+00      0s

Solved in 12 iterations and 0.00 seconds
Optimal objective  8.098426008e+02
```

Figure 3

Problem 4

question 1

Indices: $i = 1, 2, ..., m$ are the different indices/ids of boundaries. $j = 1, 2, ..., n$ is the dimension of vector in R^n space.

Data: a_i and b_i together form the center of the ith boundary of the polyhedron P. $A_{i,j}$ is a matrix where $a_i = A_{i,}$ and b_i is a vector. δ_i is the uncertainty of each element of the vector a_i.

Variables: $x \in R^n$ is a vector representing the center of that ball, where x_j is the nth item. $posix \in R^n$ is all positive items of x and $negx \in R^n$ is all negative items of x. For $posix$ and $negx$, all items that does not satisfy the positive or negative criterion equal to zero.

Objective: No objective since we are trying to find feasible point.

Constraints: $a_i x + \delta_i(posix - negx) <= b_i$ for every i. $posix_j + negx_j = x_j$ for every dimension j.

Bounds: $posix_j >= 0$ and $negx_j <= 0$ for every dimension j.

question 2

With the same indices, data, bounds, and variables as above, to have a numerical measure of feasibility, we change the objective and constraints as below:

New variables: k is a number, which is a numerical measure of feasibility.

Objective: minimize k

Constraints: $a_i x + \delta(posix - negx) <= b_i + k$ for every i. $posix_j + negx_j = x_j$ for every dimension j.

Explanation: when k equals to zero or is less than zero, the previous model is feasible at the beginning. When k is greater than zero, the previous model is not feasible since it needs a k greater than zero to satisfy all the constraints. Hence, the k can be a numerical measure of feasibility, where the greater the k is the less feasible the model is. Now, the point x that is as feasible as possible is the variable x.

question 3

Result as shown in Figure 4.

```
Using license file /Users/ranliu/gurobi.lic
Academic license - for non-commercial use only
Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (mac64)
Optimize a model with 1100 rows, 300 columns and 300300 nonzeros
Model fingerprint: 0xa51dd784
Coefficient statistics:
  Matrix range     [9e-05, 3e+00]
  Objective range  [5e+01, 2e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+02, 1e+02]
Presolve time: 0.06s
Presolved: 1100 rows, 300 columns, 300300 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0      handle free variables                          0s
     689   -5.0555330e+03   0.000000e+00   0.000000e+00      0s

Solved in 689 iterations and 0.41 seconds
Optimal objective -5.055532984e+03
```

Figure 4