Benchmark solvers for solving whole body metabolic models

Authors: Ronan M.T. Fleming, University of Galway

Reviewers:

Introduction

Compare the time taken to solve different formulations of constraint-based modelling problems involving whole body metabolic models with different solvers and different methods for each solver with the option to repeat the analysis to compute mean and variance of solution times.

EQUIPMENT SETUP

Initialize the COBRA Toolbox.

Please ensure that The COBRA Toolbox has been properly installed, and initialized using the `initCobraToolbox` function.

```
if 0 %set to true if your toolbox has not been initialised
    initCobraToolbox(false) % false, as we don't want to update
end
```

PROCEDURE

Define the location to save your results

```
if 1
    resultsFolder = '~/drive/sbgCloud/projects/variationalKinetics/
results/WBM/';
else
    resultsFolder = pwd;
end
```

Load whole body metabolic model - change this to suit your own setup.

```
modelToUse ='Harvey';
modelToUse ='Harvetta';
driver_loadBenchmarkWBMsolvers
```

Define the methods (algorithms) available for different solvers

```
% CPLEX
% 0    CPX_ALG_AUTOMATIC     Automatic: let CPLEX choose; default
% 1    CPX_ALG_PRIMAL     Primal simplex
% 2    CPX_ALG_DUAL     Dual simplex
% 3    CPX_ALG_NET     Network simplex
% 4    CPX_ALG_BARRIER     Barrier
```

```
% 5    CPX_ALG_SIFTING     Sifting
% 6    CPX_ALG_CONCURRENT     Concurrent (Dual, Barrier, and Primal in
opportunistic parallel mode; Dual and Barrier in deterministic parallel mode)
%https://www.ibm.com/docs/en/icos/12.10.0?topic=parameters-algorithm-
continuous-linear-problems
cplexLPMethods
={'AUTOMATIC','PRIMAL','DUAL','NETWORK','BARRIER','SIFTING','CONCURRENT'};
% 0    CPX_ALG_AUTOMATIC    Automatic: let CPLEX choose; default
% 1    CPX_ALG_PRIMAL    Use the primal simplex optimizer.
% 2    CPX_ALG_DUAL    Use the dual simplex optimizer.
% 3    CPX_ALG_NET    Use the network optimizer.
% 4    CPX_ALG_BARRIER    Use the barrier optimizer.
% 6    CPX_ALG_CONCURRENT    Use the concurrent optimizer.
% https://www.ibm.com/docs/en/icos/12.10.0?topic=parameters-algorithm-
continuous-quadratic-optimization
cplexQPMethods
={'AUTOMATIC','PRIMAL','DUAL','NETWORK','BARRIER','CONCURRENT'};

% Mosek

% MSK_IPAR_OPTIMIZER
%    The parameter controls which optimizer is used to optimize the task.
%    Default "FREE"
%    Accepted "FREE", "INTPNT", "CONIC", "PRIMAL_SIMPLEX", "DUAL_SIMPLEX",
"FREE_SIMPLEX", "MIXED_INT"
%    Example param.MSK_IPAR_OPTIMIZER = 'MSK_OPTIMIZER_FREE'
mosekMethods ={'MSK_OPTIMIZER_FREE', 'MSK_OPTIMIZER_INTPNT',
'MSK_OPTIMIZER_CONIC', 'MSK_OPTIMIZER_PRIMAL_SIMPLEX',
'MSK_OPTIMIZER_DUAL_SIMPLEX', 'MSK_OPTIMIZER_FREE_SIMPLEX'};

% Gurobi
% https://www.gurobi.com/documentation/current/refman/method.html
% Algorithm used to solve continuous models
% Algorithm used to solve continuous models or the initial root relaxation
of a MIP model. Options are:
gurobiLPMethods =
{'AUTOMATIC','PRIMAL','DUAL','BARRIER','CONCURRENT','DETERMINISTIC_CONCURRENT
'};
gurobiQPMethods = {'AUTOMATIC','PRIMAL','DUAL','BARRIER'};
```

Set parameters for benchmark

COBRA toolbox parameters

```
printLevel=1; %  {(0),1,2} 1 output from optimiseVKmode, 2 also output from
solver
changeOK = changeCobraSolverParams('LP', 'printLevel', printLevel);
feasTol=1e-6;
changeOK = changeCobraSolverParams('LP', 'printLevel', feasTol);
```

Select whether to compare one or a set of solvers

```
compareSolvers = 1;
```

Select whether to compare one or a set of different formulations of constraint-based modelling problems involving whole body metabolic models.

```
compareSolveWBMmethods = 0;
```

Select whether to compare one or a set of available methods (algorithms) for each solver

```
compareSolverMethods = 0;
```

Define the number of times to replicate the same formulation, solver, method combination.

```
nReplicates = 1;
```

Set the maximum time limit allowed to solve a single instance. Useful for eliminating slow instances in a large batch of trials.

```
secondsTimeLimit = 60;
```

Display and (optionally) modify properties of the whole body model that may effect solve time

```
[nMet,nRxn]=size(model.S)
```

```
nMet =
58095
nRxn =
83395
```

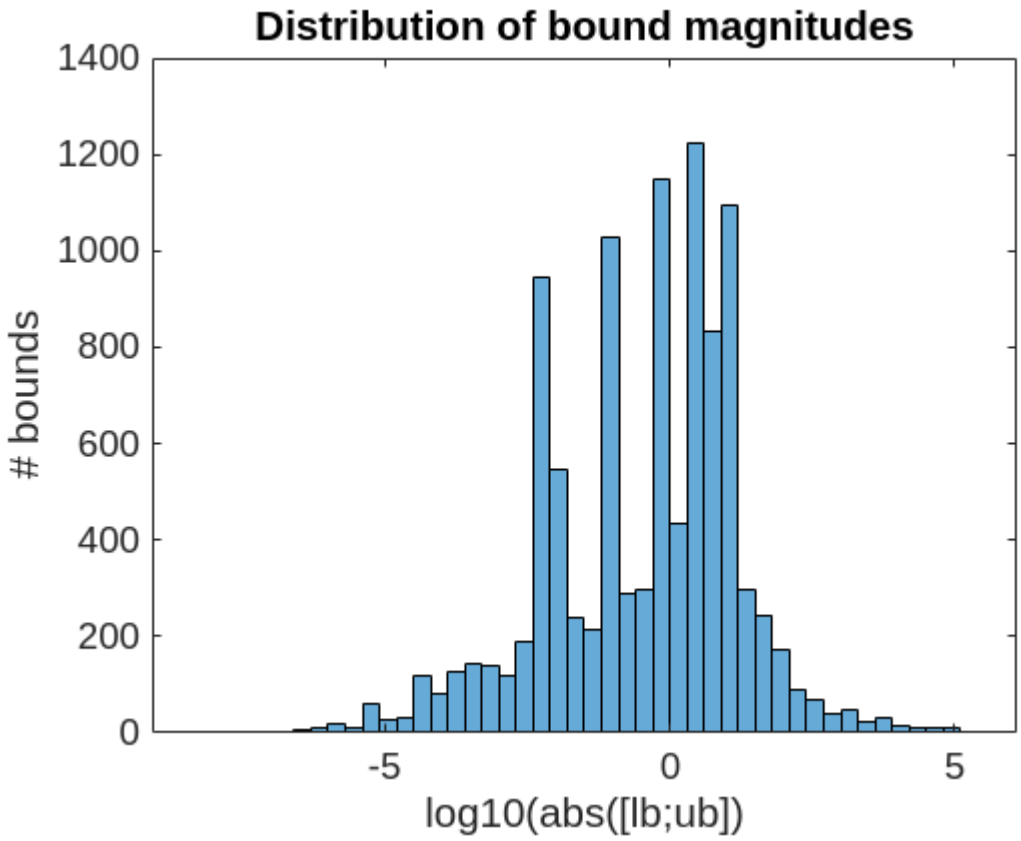
Identify large bounds not at the maximum

```
boundMagnitudes = [abs(model.lb);abs(model.ub)];
largestMagnitudeBound = max(boundMagnitudes);
fprintf('%g%s\n',
(nnz(largestMagnitudeBound==[abs(model.lb);abs(model.ub)])*100)/
(length(model.lb)*2),' = percent of bounds at maximum')
```

```
62.2112 = percent of bounds at maximum
```

Display bounds that are not at the maximum

```
if 1
    figure
    histogram(log10(boundMagnitudes(boundMagnitudes~=largestMagnitudeBound &
boundMagnitudes~=0)))
    xlabel('log10(abs([lb;ub])')
    ylabel('# bounds')
    title('Distribution of bound magnitudes')
end
```



Replace large bounds with inf or -inf. This is a good idea. Better to leave this option on.

```
if 1
    model.lb(-largestMagnitudeBound==model.lb)=-inf;
    model.ub(largestMagnitudeBound==model.ub)= inf;
end
boolMagnitudes = boundMagnitudes~=largestMagnitudeBound & boundMagnitudes~=0
& boundMagnitudes<1e-4;
boolRxns = boolMagnitudes(1:nRxn) | boolMagnitudes(nRxn+1:2*nRxn);
```

Optionally, print the bounds for reactions with small magnitude

```
if 0
    printFluxBounds(model,model.rxns(boolRxns))
end
fprintf('%g%s\n',nnz(boolRxns)*100/length(boolRxns),' = percent of bounds
with magnutide less than 1e-4')
```

```
0.389712 = percent of bounds with magnutide less than 1e-4
```

Optionally, print the bounds for reactions with small difference

```
boundDifference = model.ub - model.lb;
bool = length(model.rxns);
Z = table(boundDifference,model.rxns,model.rxnNames,'VariableNames',
{'boundDifference','rxns','rxnNames'});
if any(boundDifference<0)
```

```
    error(['lb > ub for ' num2str(nnz(boundDifference)) ' reactions'])
end
boolDifference = boundDifference<1e-5 & boundDifference~=0;
Z = sortrows(Z(boolDifference,:),'boundDifference');
if 0
    printFluxBounds(model,Z.rxns,1)
end
fprintf('%g%s\n',nnz(boolRxns)*100/length(boolRxns), ' = percent of bounds
with difference (ub - lb) less than 1e-5')
```

```
0.389712 = percent of bounds with difference (ub - lb) less than 1e-5
```

```
forwardBoolDifference = boolDifference & model.lb>=0 & model.ub>0;
reverseBoolDifference = boolDifference & model.lb<0 & model.ub<=0;
reversibleBoolDifference = boolDifference & model.lb<0 & model.ub>0;

if any((forwardBoolDifference | reverseBoolDifference |
reversibleBoolDifference)~=boolDifference)
    error('missing bool difference')
end
```

Optionally relax bounds that are very tight

```
if relaxTightBounds
    modelOld=model;
    done=false(nRxn,1);
    for x=higherExponent:-1:lowerExponent
        %calulate the difference between the bounds each time
        boundDifference = model.ub - model.lb;

        %forward
        bool = forwardBoolDifference & (boundDifference <= 10^(-x));
        model.ub(bool & ~done) = model.ub(bool & ~done)*(10^(x-
lowerExponent+1));
        done = done | bool;

        %reverse
        bool = reverseBoolDifference & (boundDifference <= 10^(-x));
        model.lb(bool & ~done) = model.lb(bool & ~done)*(10^(x-
lowerExponent+1));
        done = done | bool;

        %reversible
        bool = reversibleBoolDifference & (boundDifference <= 10^(-x));
        model.lb(bool & ~done) = model.lb(bool & ~done)*(10^((x-
lowerExponent+1)/2));
        model.ub(bool & ~done) = model.ub(bool & ~done)*(10^((x-
lowerExponent+1)/2));
        done = done | bool;

        %reset
```

```
        %done=false(nRxn,1);
    end
    if 1
        printFluxBounds(model,Z.rxns,1)
    end
end
```

Prepare a benchmark table, choose the solver and solve

```
VariableNames={'interface','solver','method','problem','model','stat','origSt
at','time','f','f1','f2','f0'};
% Define the corresponding variable types
VariableTypes = {'string', 'string',
'string','string','string','double','string','double','double','double','doub
le','double'};
T = table('Size', [0
length(VariableNames)],'VariableNames',VariableNames,'VariableTypes',
VariableTypes);
```

Select the solvers to compare

```
if compareSolvers
    solvers = { 'gurobi','ibm_cplex','mosek'};
    %solvers = { 'ibm_cplex','mosek','gurobi'};
    %solvers = { 'mosek','ibm_cplex','gurobi'};
else
    % Choose the solver
    % solvers = {'gurobi'};
    solvers = {'ibm_cplex'};
    solvers = {'mosek'};
end
```

Select the formulations to compare

```
if compareSolveWBMmethods
    %solveWBMmethods = {'LP','QP','QRLP','QRQP','zero','oneInternal'};
    solveWBMmethods = {'LP','QP'};%,'zero','oneInternal'};
    %solveWBMmethods = {'LP','oneInternal'};
else
    % Choose type of problem to solve
    % solveWBMmethods = {'LP'};
    solveWBMmethods = {'QP'};
    %solveWBMmethods = {'QRLP'};
    %solveWBMmethods = {'QRQP'};
end
```

Set the min norm weight for QP problems

```
minNormWeight = 1e-4;
```

```
%model.c(:)=0;
```

Solve the ensemble of instances

```
for ind = 1:nReplicates
    for i = 1:length(solveWBMmethods)
        solveWBMmethod = solveWBMmethods{i};
        for j = 1:length(solvers)
            solver = solvers{j};

            clear param;
            param.printLevel=0;
            param.timelimit=secondsTimeLimit;
            param.solver=solver;

            switch solveWBMmethod
                case 'LP'
                    param.solveWBMmethod='LP';
                    param.minNorm = 0;
                    param.solver=solver;
                case 'QP'
                    param.solveWBMmethod='QP';
                    param.minNorm = minNormWeight;
                case 'QRLP'
                    param.solveWBMmethod='QRLP';
                    param.minNorm = 0;
                case 'QRQP'
                    param.solveWBMmethod='QRQP';
                    param.minNorm = minNormWeight;
                case 'zero'
                    param.solveWBMmethod='zero';
                    param.minNorm = 'zero';
                case 'oneInternal'
                    param.solveWBMmethod='oneInternal';
                    if isfield(model,'SConsistentRxnBool')
                        param.minNorm = 'oneInternal';
                    else
                        error('param.solveWBMmethod= oneInternal cannot be
implemented as model.SConsistentRxnBool is missing')
                    end
            end

            switch solver
                case 'gurobi'
                    % Model scaling
                    %      Type:    int
                    %      Default value:    -1
                    %      Minimum value:    -1
                    %      Maximum value:    3
```

```
                    % Controls model scaling. By default, the rows and
columns of the model are scaled in order to improve the numerical
                    % properties of the constraint matrix. The scaling is
removed before the final solution is returned. Scaling typically
                    % reduces solution times, but it may lead to larger
constraint violations in the original, unscaled model. Turning off
                    % scaling (ScaleFlag=0) can sometimes produce smaller
constraint violations. Choosing a different scaling option can
                    % sometimes improve performance for particularly
numerically difficult models. Using geometric mean scaling (ScaleFlag=2)
                    % is especially well suited for models with a wide range
of coefficients in the constraint matrix rows or columns.
                    % Settings 1 and 3 are not as directly connected to any
specific model characteristics, so experimentation with both
                    % settings may be needed to assess performance impact.
                    param.scaleFlag=0;
                    param.timelimit = secondsTimeLimit;

                case 'ibm_cplex'
                    % https://www.ibm.com/docs/en/icos/
12.10.0?topic=infeasibility-coping-ill-conditioned-problem-handling-unscaled-
infeasibilities
                    param.minNorm = 0;

                    % Decides how to scale the problem matrix.
                    % Value  Meaning
                    % -1    No scaling
                    % 0    Equilibration scaling; default
                    % 1    More aggressive scaling
                    % https://www.ibm.com/docs/en/icos/12.10.0?
topic=parameters-scale-parameter
                    param.scaind = -1;

                    % Emphasizes precision in numerically unstable or
difficult problems.
                    % This parameter lets you specify to CPLEX that it
should emphasize precision in
                    % numerically difficult or unstable problems, with
consequent performance trade-offs in time and memory.
                    % Value Meaning
                    % 0   Do not emphasize numerical precision; default
                    % 1    Exercise extreme caution in computation
                    % https://www.ibm.com/docs/en/icos/12.10.0?
topic=parameters-numerical-precision-emphasis
                    param.emphasis_numerical=1;

                    param.timelimit = secondsTimeLimit;
                case 'mosek'
                    param.MSK_DPAR_OPTIMIZER_MAX_TIME=secondsTimeLimit;
                    param.MSK_IPAR_WRITE_DATA_PARAM='MSK_ON';
```

```
                    param.MSK_IPAR_LOG_INTPNT=10;
                    param.MSK_IPAR_LOG_PRESOLVE=10;

                    % MSK_IPAR_INTPNT_SCALING
                    % Controls how the problem is scaled before the interior-
point optimizer is used.
                    % Default
                    % "FREE"
                    % Accepted
                    % "FREE", "NONE"
                    % param..MSK_IPAR_INTPNT_SCALING = 'MSK_SCALING_FREE';
                    param.MSK_IPAR_INTPNT_SCALING='MSK_SCALING_NONE';

                    % MSK_IPAR_SIM_SCALING
                    % Controls how much effort is used in scaling the
problem before a simplex optimizer is used.
                    % Default
                    % "FREE"
                    % Accepted
                    % "FREE", "NONE"
                    % Example
                    % param.MSK_IPAR_SIM_SCALING = 'MSK_SCALING_FREE'
                    param.MSK_IPAR_SIM_SCALING='MSK_SCALING_NONE';

                    % MSK_IPAR_SIM_SCALING_METHOD
                    % Controls how the problem is scaled before a simplex
optimizer is used.
                    % Default
                    % "POW2"
                    % Accepted
                    % "POW2", "FREE"
                    % Example
                    % param.MSK_IPAR_SIM_SCALING_METHOD =
'MSK_SCALING_METHOD_POW2'
                    %
param.MSK_IPAR_SIM_SCALING_METHOD='MSK_SCALING_METHOD_FREE';
            end

            if compareSolverMethods
                % Solve a problem with selected solver and each method
available to that solver
                % solveWBMmethod =
{'LP','QP','QRLP','QRQP','zero','oneInternal'};
                switch param.solveWBMmethod
                    case {'LP','zero','oneInternal'}
                        switch solver
                            case 'gurobi'
                                solverMethods = gurobiLPMethods;
                            case 'ibm_cplex'
                                solverMethods = cplexLPMethods;
```

```
                        case 'mosek'
                            solverMethods = mosekMethods;
                    end
                case {'QP','QRLP','QRQP'}
                    switch solver
                        case 'gurobi'
                            solverMethods = gurobiQPMethods;
                        case 'ibm_cplex'
                            solverMethods = cplexQPMethods;
                        case 'mosek'
                            solverMethods = mosekMethods;
                    end
            end
            for k=1:length(solverMethods)
                switch solver
                    case 'gurobi'
                        param.lpmethod=solverMethods{k};
                        param.qpmethod=solverMethods{k};
                    case 'ibm_cplex'
                        param.lpmethod=solverMethods{k};
                        param.qpmethod=solverMethods{k};
                    case 'mosek'
                        %https://docs.mosek.com/latest/toolbox/
parameters.html#mosek.iparam.optimizer
                        %The parameter controls which optimizer is used
to optimize the task.
                        param.MSK_IPAR_OPTIMIZER=solverMethods{k};
                end

                tic
                solution = optimizeCbModel(model,'min',
param.minNorm,1,param);
                T = [T; {'optimizeCbModel', solver, solverMethods{k},
param.solveWBMmethod, modelToUse, solution.stat,{solution.origStat},toc,
{solution.f},{solution.f1},{solution.f2},{solution.f0}}];
                %display(T)
            end
        else
            % Solve a problem with selected solver and one method
available to that solver
            switch solver
                case 'gurobi'
                    param.lpmethod='BARRIER';
                    param.qpmethod='BARRIER';
                    method = param.lpmethod;
                case 'ibm_cplex'
                    param.lpmethod='BARRIER';
                    param.qpmethod='BARRIER';
                    method = param.lpmethod;
                case 'mosek'
```

```
                        method = 'FREE';
                        method = 'INTPNT';
                        method = 'CONIC';
                        param.MSK_IPAR_OPTIMIZER=['MSK_OPTIMIZER_' method];

                end
                tic
                solution = optimizeCbModel(model,'min',
param.minNorm,1,param);
                T = [T; {'optimizeCbModel', solver, method,
param.solveWBMmethod, modelToUse, solution.stat,{solution.origStat},toc,
{solution.f},{solution.f1},{solution.f2},{solution.f0}}];
            end
        end
    end
end
```

```
MOSEK Version 10.2.5 (Build date: 2024-9-17 12:12:35)
Copyright (c) MOSEK ApS, Denmark WWW: mosek.com
Platform: Linux/64-X86

Problem
  Name                   :
  Objective sense        : minimize
  Type                   : QO (quadratic optimization problem)
  Constraints            : 269909
  Affine conic cons.     : 0
  Disjunctive cons.      : 0
  Cones                  : 0
  Scalar variables       : 189302
  Matrix variables       : 0
  Integer variables      : 0

Optimizer started.
Quadratic to conic reformulation started.
Quadratic to conic reformulation terminated. Time: 0.02
Presolve started.
Linear dependency checker started.
Linear dependency checker terminated.
Eliminator started.
Freed constraints in eliminator : 1882
Eliminator terminated.
Eliminator started.
Freed constraints in eliminator : 16
Eliminator terminated.
Eliminator - tries                  : 2                 time                   : 0.00
Lin. dep.  - tries                  : 1                 time                   : 0.11
Lin. dep.  - primal attempts        : 1                 successes              : 1
Lin. dep.  - dual attempts          : 0                 successes              : 0
Lin. dep.  - primal deps.           : 406               dual deps.             : 0
MOSEK warning 803 (MSK_RES_WRN_PRESOLVE_PRIMAL_PERTUBATIONS): The bounds of the constraints and variables
Presolve terminated. Time: 1.10
Optimizer  - threads                : 18
Optimizer  - solved problem         : the primal
Optimizer  - Constraints            : 135333
Optimizer  - Cones                  : 1
Optimizer  - Scalar variables       : 261138            conic                  : 159874
Optimizer  - Semi-definite variables: 0                 scalarized             : 0
Factor     - setup time             : 0.82
Factor     - dense det. time        : 0.00              GP order time          : 0.00
```

```
Factor     - nonzeros before factor : 1.34e+06         after factor          : 4.39e+06
Factor     - dense dim.             : 2                flops                 : 2.97e+09
ITE PFEAS    DFEAS    GFEAS    PRSTATUS   POBJ              DOBJ              MU       TIME
0   2.0e+03  2.9e-01  2.4e+00  0.00e+00   1.707177492e+00   2.928225081e-01   1.0e+00  2.10
1   1.9e+03  2.8e-01  2.3e+00  -9.58e-01  2.494732423e+02   2.481141587e+02   9.5e-01  2.57
2   1.7e+03  2.5e-01  2.2e+00  -9.65e-01  1.005170936e+03   1.003940660e+03   8.4e-01  2.92
3   1.4e+03  2.1e-01  2.0e+00  -9.71e-01  2.324646009e+03   2.323618897e+03   7.2e-01  3.27
4   2.7e+02  4.0e-02  8.1e-01  -9.55e-01  8.268862961e+04   8.269247778e+04   1.4e-01  4.07
5   1.8e+02  2.7e-02  6.3e-01  -8.13e-01  1.639776242e+05   1.639837149e+05   9.1e-02  4.42
6   7.7e+01  1.1e-02  3.5e-01  -7.46e-01  5.184215865e+05   5.184337577e+05   3.9e-02  4.91
7   6.5e+01  9.5e-03  3.0e-01  -5.16e-01  6.539201030e+05   6.539331677e+05   3.2e-02  5.29
8   2.8e+01  4.1e-03  1.4e-01  -4.52e-01  1.592220292e+06   1.592236383e+06   1.4e-02  5.93
9   2.3e+01  3.3e-03  1.1e-01  -6.39e-02  1.872467690e+06   1.872483291e+06   1.1e-02  6.29
10  1.8e+01  2.7e-03  8.9e-02  1.98e-02   2.185431520e+06   2.185446433e+06   9.2e-03  6.64
11  1.2e+01  1.8e-03  5.7e-02  1.01e-01   2.754471035e+06   2.754484587e+06   6.2e-03  7.12
12  1.1e+01  1.6e-03  5.0e-02  2.13e-01   2.916842535e+06   2.916855746e+06   5.5e-03  7.47
13  9.1e+00  1.3e-03  4.1e-02  2.18e-01   3.172004312e+06   3.172017029e+06   4.6e-03  7.91
14  7.7e+00  1.1e-03  3.3e-02  2.52e-01   3.417938475e+06   3.417950623e+06   3.9e-03  8.25
15  6.6e+00  9.8e-04  2.8e-02  2.97e-01   3.618477941e+06   3.618489558e+06   3.3e-03  8.60
16  5.6e+00  8.2e-04  2.3e-02  3.33e-01   3.857190320e+06   3.857201277e+06   2.8e-03  8.95
17  4.3e+00  6.4e-04  1.7e-02  3.75e-01   4.192651723e+06   4.192661671e+06   2.2e-03  9.40
18  3.5e+00  5.1e-04  1.3e-02  4.32e-01   4.465596472e+06   4.465605559e+06   1.8e-03  9.75
19  3.2e+00  4.7e-04  1.2e-02  4.71e-01   4.581968112e+06   4.581976890e+06   1.6e-03  10.09
20  2.1e+00  3.1e-04  7.0e-03  4.94e-01   5.039170284e+06   5.039177425e+06   1.1e-03  10.76
21  1.8e+00  2.7e-04  5.8e-03  5.87e-01   5.186011199e+06   5.186017824e+06   9.2e-04  11.11
22  1.7e+00  2.5e-04  5.4e-03  5.88e-01   5.247971548e+06   5.247978030e+06   8.6e-04  11.45
23  1.6e+00  2.4e-04  5.1e-03  5.77e-01   5.288378501e+06   5.288384860e+06   8.2e-04  11.79
24  1.5e+00  2.2e-04  4.4e-03  5.71e-01   5.390966667e+06   5.390972730e+06   7.3e-04  12.14
25  1.3e+00  1.8e-04  3.7e-03  5.79e-01   5.523683490e+06   5.523689136e+06   6.3e-04  12.49
26  1.1e+00  1.7e-04  3.2e-03  5.93e-01   5.604336231e+06   5.604341636e+06   5.7e-04  12.84
27  7.8e-01  1.2e-04  2.0e-03  5.99e-01   5.876891951e+06   5.876896449e+06   3.9e-04  13.39
28  6.5e-01  1.0e-04  1.6e-03  6.16e-01   5.992506624e+06   5.992510743e+06   3.3e-04  13.88
29  5.9e-01  1.3e-04  1.4e-03  6.49e-01   6.052827528e+06   6.052831436e+06   3.0e-04  14.24
30  5.3e-01  1.6e-04  1.2e-03  6.64e-01   6.126499978e+06   6.126503603e+06   2.7e-04  14.59
31  4.0e-01  1.8e-04  8.6e-04  6.84e-01   6.275455393e+06   6.275458464e+06   2.0e-04  15.04
32  3.7e-01  2.1e-04  7.7e-04  7.16e-01   6.317526783e+06   6.317529691e+06   1.8e-04  15.40
33  3.1e-01  2.4e-04  6.1e-04  7.25e-01   6.397647861e+06   6.397650469e+06   1.5e-04  15.75
34  2.7e-01  2.8e-04  5.2e-04  7.37e-01   6.449522508e+06   6.449524917e+06   1.4e-04  16.10
35  2.5e-01  3.0e-04  4.5e-04  7.60e-01   6.487523747e+06   6.487526005e+06   1.2e-04  16.46
36  2.1e-01  3.6e-04  3.8e-04  7.64e-01   6.538613889e+06   6.538615950e+06   1.1e-04  16.80
37  1.9e-01  4.8e-04  3.1e-04  7.58e-01   6.582776221e+06   6.582778117e+06   9.3e-05  17.15
38  1.6e-01  7.2e-04  2.5e-04  7.53e-01   6.635747596e+06   6.635749301e+06   7.8e-05  17.60
39  1.5e-01  7.9e-04  2.4e-04  7.25e-01   6.644959565e+06   6.644961239e+06   7.5e-05  17.95
40  1.3e-01  9.9e-04  2.0e-04  7.43e-01   6.682845967e+06   6.682847503e+06   6.5e-05  18.31
41  1.2e-01  1.2e-03  1.8e-04  7.46e-01   6.704895312e+06   6.704896768e+06   6.0e-05  18.68
42  9.4e-02  2.1e-03  1.3e-04  7.41e-01   6.751169057e+06   6.751170382e+06   4.7e-05  19.24
43  8.2e-02  3.2e-03  1.1e-04  5.73e-01   6.772240698e+06   6.772241993e+06   4.1e-05  19.69
44  7.7e-02  3.7e-03  1.1e-04  5.05e-01   6.781163505e+06   6.781164794e+06   3.9e-05  20.04
45  7.2e-02  4.4e-03  1.0e-04  4.61e-01   6.790395600e+06   6.790396888e+06   3.6e-05  20.42
46  6.8e-02  4.8e-03  9.4e-05  4.34e-01   6.797930779e+06   6.797932058e+06   3.4e-05  20.78
47  6.8e-02  4.8e-03  9.3e-05  4.48e-01   6.800858510e+06   6.800859777e+06   3.4e-05  21.13
48  6.7e-02  4.9e-03  9.3e-05  4.83e-01   6.800783287e+06   6.800784557e+06   3.4e-05  21.48
49  6.4e-02  5.2e-03  8.7e-05  4.60e-01   6.810006223e+06   6.810007470e+06   3.2e-05  21.92
50  6.0e-02  5.1e-03  8.0e-05  5.09e-01   6.824482548e+06   6.824483736e+06   3.0e-05  22.26
51  5.9e-02  5.3e-03  7.9e-05  5.35e-01   6.824782123e+06   6.824783317e+06   3.0e-05  22.61
52  5.6e-02  5.5e-03  7.4e-05  5.17e-01   6.834133863e+06   6.834135029e+06   2.8e-05  22.97
53  5.1e-02  6.1e-03  6.7e-05  5.09e-01   6.848324756e+06   6.848325890e+06   2.6e-05  23.46
54  4.6e-02  6.8e-03  5.8e-05  4.47e-01   6.866772066e+06   6.866773158e+06   2.3e-05  23.95
55  4.3e-02  7.4e-03  5.4e-05  4.27e-01   6.874540819e+06   6.874541902e+06   2.2e-05  24.29
56  4.2e-02  6.3e-03  5.2e-05  7.27e-01   6.884090727e+06   6.884091741e+06   2.1e-05  24.65
57  3.8e-02  5.8e-03  4.4e-05  6.35e-01   6.905546491e+06   6.905547409e+06   1.9e-05  25.00
58  3.4e-02  5.6e-03  3.9e-05  6.92e-01   6.921597910e+06   6.921598761e+06   1.7e-05  25.35
59  2.8e-02  5.7e-03  2.9e-05  7.21e-01   6.952782644e+06   6.952783370e+06   1.4e-05  25.86
60  2.5e-02  5.8e-03  2.5e-05  7.62e-01   6.967973069e+06   6.967973735e+06   1.3e-05  26.36
```

```
61  2.2e-02  5.6e-03  2.1e-05  8.04e-01   6.981557363e+06   6.981557973e+06   1.1e-05  26.72
62  2.1e-02  5.6e-03  2.0e-05  8.31e-01   6.986843872e+06   6.986844459e+06   1.1e-05  27.06
63  1.7e-02  5.5e-03  1.4e-05  8.41e-01   7.012851580e+06   7.012852061e+06   8.5e-06  27.45
64  1.6e-02  5.4e-03  1.3e-05  8.81e-01   7.017605979e+06   7.017606440e+06   8.1e-06  27.80
65  1.2e-02  5.6e-03  9.0e-06  8.89e-01   7.040733173e+06   7.040733541e+06   6.2e-06  28.29
66  1.1e-02  5.6e-03  8.1e-06  9.26e-01   7.045550751e+06   7.045551099e+06   5.8e-06  28.65
67  9.3e-03  6.2e-03  6.1e-06  9.31e-01   7.058251214e+06   7.058251510e+06   4.8e-06  29.05
68  9.0e-03  6.2e-03  5.9e-06  9.39e-01   7.059820413e+06   7.059820703e+06   4.7e-06  29.44
69  8.8e-03  6.0e-03  5.7e-06  9.44e-01   7.061731597e+06   7.061731879e+06   4.5e-06  29.80
70  8.3e-03  6.1e-03  5.4e-06  9.46e-01   7.064698114e+06   7.064698384e+06   4.3e-06  30.22
71  8.0e-03  6.0e-03  5.1e-06  9.48e-01   7.066897297e+06   7.066897558e+06   4.1e-06  30.58
72  7.5e-03  6.1e-03  4.7e-06  9.52e-01   7.070138232e+06   7.070138480e+06   3.9e-06  30.94
73  6.9e-03  6.3e-03  4.1e-06  9.52e-01   7.074565860e+06   7.074566090e+06   3.6e-06  31.29
74  6.7e-03  6.2e-03  4.0e-06  9.60e-01   7.076130811e+06   7.076131035e+06   3.5e-06  31.63
75  6.3e-03  6.2e-03  3.6e-06  9.60e-01   7.078670594e+06   7.078670808e+06   3.3e-06  31.98
76  5.8e-03  6.4e-03  3.2e-06  9.62e-01   7.082114208e+06   7.082114408e+06   3.1e-06  32.34
77  5.6e-03  6.1e-03  2.9e-06  8.59e-01   7.083937071e+06   7.083937264e+06   2.9e-06  33.17
78  5.5e-03  6.0e-03  2.9e-06  8.52e-01   7.084524828e+06   7.084525019e+06   2.9e-06  33.93
79  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  34.65
80  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  35.68
81  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  36.68
Optimizer terminated. Time: 37.88


Interior-point solution summary
  Problem status  : PRIMAL_AND_DUAL_FEASIBLE
  Solution status : OPTIMAL
  Primal.  obj: 7.0826522774e+06    nrm: 1e+05    Viol.  con: 1e+01    var: 1e+02
  Dual.    obj: 7.0914996991e+06    nrm: 3e+06    Viol.  con: 1e-13    var: 1e+04
Optimizer summary
  Optimizer                 -                        time: 37.88
    Interior-point          - iterations : 82        time: 37.86
      Basis identification  -                        time: 0.00
        Primal              - iterations : 0         time: 0.00
        Dual                - iterations : 0         time: 0.00
        Clean primal        - iterations : 0         time: 0.00
        Clean dual          - iterations : 0         time: 0.00
    Simplex                 -                        time: 0.00
      Primal simplex        - iterations : 0         time: 0.00
      Dual simplex          - iterations : 0         time: 0.00
    Mixed integer           - relaxations: 0         time: 0.00

Mosek returned an error or warning, open the following link in your browser:
https://docs.mosek.com/latest/toolbox/response-codes.html#mosek.rescode.trm_stall
     -12  min(sbl) = min(A*x - bl), (should be positive)
     -12  min(sbu) = min(bu - A*x), (should be positive)
[mosek] reports OPTIMAL but Primal optimality condition in solveCobraQP not satisfied, residual = 1.1173,

MOSEK Version 10.2.5 (Build date: 2024-9-17 12:12:35)
Copyright (c) MOSEK ApS, Denmark WWW: mosek.com
Platform: Linux/64-X86

Problem
  Name                   :
  Objective sense        : minimize
  Type                   : QO (quadratic optimization problem)
  Constraints            : 269909
  Affine conic cons.     : 0
  Disjunctive cons.      : 0
  Cones                  : 0
  Scalar variables       : 189302
  Matrix variables       : 0
  Integer variables      : 0
```

```
Optimizer started.
Quadratic to conic reformulation started.
Quadratic to conic reformulation terminated. Time: 0.02
Presolve started.
Linear dependency checker started.
Linear dependency checker terminated.
Eliminator started.
Freed constraints in eliminator : 1882
Eliminator terminated.
Eliminator started.
Freed constraints in eliminator : 16
Eliminator terminated.
Eliminator - tries                  : 2                 time                   : 0.00
Lin. dep.  - tries                  : 1                 time                   : 0.10
Lin. dep.  - primal attempts        : 1                 successes              : 1
Lin. dep.  - dual attempts          : 0                 successes              : 0
Lin. dep.  - primal deps.           : 406               dual deps.             : 0
MOSEK warning 803 (MSK_RES_WRN_PRESOLVE_PRIMAL_PERTUBATIONS): The bounds of the constraints and variables
Presolve terminated. Time: 0.94
Optimizer  - threads                : 18
Optimizer  - solved problem         : the primal
Optimizer  - Constraints            : 135333
Optimizer  - Cones                  : 1
Optimizer  - Scalar variables       : 261138            conic                  : 159874
Optimizer  - Semi-definite variables: 0                 scalarized             : 0
Factor     - setup time             : 0.90
Factor     - dense det. time        : 0.00              GP order time          : 0.00
Factor     - nonzeros before factor : 1.34e+06          after factor           : 4.39e+06
Factor     - dense dim.             : 2                 flops                  : 2.97e+09
ITE PFEAS    DFEAS    GFEAS    PRSTATUS   POBJ              DOBJ              MU       TIME
0   2.0e+03  2.9e-01  2.4e+00  0.00e+00   1.707177492e+00   2.928225081e-01   1.0e+00  2.02
1   1.9e+03  2.8e-01  2.3e+00  -9.58e-01  2.494732423e+02   2.481141587e+02   9.5e-01  2.49
2   1.7e+03  2.5e-01  2.2e+00  -9.65e-01  1.005170936e+03   1.003940660e+03   8.4e-01  2.84
3   1.4e+03  2.1e-01  2.0e+00  -9.71e-01  2.324646009e+03   2.323618897e+03   7.2e-01  3.19
4   2.7e+02  4.0e-02  8.1e-01  -9.55e-01  8.268862961e+04   8.269247778e+04   1.4e-01  3.96
5   1.8e+02  2.7e-02  6.3e-01  -8.13e-01  1.639776242e+05   1.639837149e+05   9.1e-02  4.32
6   7.7e+01  1.1e-02  3.5e-01  -7.46e-01  5.184215865e+05   5.184337577e+05   3.9e-02  4.83
7   6.5e+01  9.5e-03  3.0e-01  -5.16e-01  6.539201030e+05   6.539331677e+05   3.2e-02  5.18
8   2.8e+01  4.1e-03  1.4e-01  -4.52e-01  1.592220292e+06   1.592236383e+06   1.4e-02  5.81
9   2.3e+01  3.3e-03  1.1e-01  -6.39e-02  1.872467690e+06   1.872483291e+06   1.1e-02  6.16
10  1.8e+01  2.7e-03  8.9e-02  1.98e-02   2.185431520e+06   2.185446433e+06   9.2e-03  6.51
11  1.2e+01  1.8e-03  5.7e-02  1.01e-01   2.754471035e+06   2.754484587e+06   6.2e-03  7.00
12  1.1e+01  1.6e-03  5.0e-02  2.13e-01   2.916842535e+06   2.916855746e+06   5.5e-03  7.35
13  9.1e+00  1.3e-03  4.1e-02  2.18e-01   3.172004312e+06   3.172017029e+06   4.6e-03  7.79
14  7.7e+00  1.1e-03  3.3e-02  2.52e-01   3.417938475e+06   3.417950623e+06   3.9e-03  8.14
15  6.6e+00  9.8e-04  2.8e-02  2.97e-01   3.618477941e+06   3.618489558e+06   3.3e-03  8.49
16  5.6e+00  8.2e-04  2.3e-02  3.33e-01   3.857190320e+06   3.857201277e+06   2.8e-03  8.85
17  4.3e+00  6.4e-04  1.7e-02  3.75e-01   4.192651723e+06   4.192661671e+06   2.2e-03  9.33
18  3.5e+00  5.1e-04  1.3e-02  4.32e-01   4.465596472e+06   4.465605559e+06   1.8e-03  9.68
19  3.2e+00  4.7e-04  1.2e-02  4.71e-01   4.581968112e+06   4.581976890e+06   1.6e-03  10.03
20  2.1e+00  3.1e-04  7.0e-03  4.94e-01   5.039170284e+06   5.039177425e+06   1.1e-03  10.74
21  1.8e+00  2.7e-04  5.8e-03  5.87e-01   5.186011199e+06   5.186017824e+06   9.2e-04  11.09
22  1.7e+00  2.5e-04  5.4e-03  5.88e-01   5.247971548e+06   5.247978030e+06   8.6e-04  11.44
23  1.6e+00  2.4e-04  5.1e-03  5.77e-01   5.288378501e+06   5.288384860e+06   8.2e-04  11.79
24  1.5e+00  2.2e-04  4.4e-03  5.71e-01   5.390966667e+06   5.390972730e+06   7.3e-04  12.14
25  1.3e+00  1.8e-04  3.7e-03  5.79e-01   5.523683490e+06   5.523689136e+06   6.3e-04  12.49
26  1.1e+00  1.7e-04  3.2e-03  5.93e-01   5.604336231e+06   5.604341636e+06   5.7e-04  12.84
27  7.8e-01  1.2e-04  2.0e-03  5.99e-01   5.876891951e+06   5.876896449e+06   3.9e-04  13.37
28  6.5e-01  1.0e-04  1.6e-03  6.16e-01   5.992506624e+06   5.992510743e+06   3.3e-04  13.87
29  5.9e-01  1.3e-04  1.4e-03  6.49e-01   6.052827528e+06   6.052831436e+06   3.0e-04  14.22
30  5.3e-01  1.6e-04  1.2e-03  6.64e-01   6.126499978e+06   6.126503603e+06   2.7e-04  14.57
31  4.0e-01  1.8e-04  8.6e-04  6.84e-01   6.275455393e+06   6.275458464e+06   2.0e-04  15.01
32  3.7e-01  2.1e-04  7.7e-04  7.16e-01   6.317526783e+06   6.317529691e+06   1.8e-04  15.38
33  3.1e-01  2.4e-04  6.1e-04  7.25e-01   6.397647861e+06   6.397650469e+06   1.5e-04  15.74
```

```
34  2.7e-01  2.8e-04  5.2e-04  7.37e-01   6.449522508e+06   6.449524917e+06   1.4e-04  16.08
35  2.5e-01  3.0e-04  4.5e-04  7.60e-01   6.487523747e+06   6.487526005e+06   1.2e-04  16.44
36  2.1e-01  3.6e-04  3.8e-04  7.64e-01   6.538613889e+06   6.538615950e+06   1.1e-04  16.78
37  1.9e-01  4.8e-04  3.1e-04  7.58e-01   6.582776221e+06   6.582778117e+06   9.3e-05  17.13
38  1.6e-01  7.2e-04  2.5e-04  7.53e-01   6.635747596e+06   6.635749301e+06   7.8e-05  17.58
39  1.5e-01  7.9e-04  2.4e-04  7.25e-01   6.644959565e+06   6.644961239e+06   7.5e-05  17.95
40  1.3e-01  9.9e-04  2.0e-04  7.43e-01   6.682845967e+06   6.682847503e+06   6.5e-05  18.30
41  1.2e-01  1.2e-03  1.8e-04  7.46e-01   6.704895312e+06   6.704896768e+06   6.0e-05  18.65
42  9.4e-02  2.1e-03  1.3e-04  7.41e-01   6.751169057e+06   6.751170382e+06   4.7e-05  19.18
43  8.2e-02  3.2e-03  1.1e-04  5.73e-01   6.772240698e+06   6.772241993e+06   4.1e-05  19.62
44  7.7e-02  3.7e-03  1.1e-04  5.05e-01   6.781163505e+06   6.781164794e+06   3.9e-05  19.97
45  7.2e-02  4.4e-03  1.0e-04  4.61e-01   6.790395600e+06   6.790396888e+06   3.6e-05  20.32
46  6.8e-02  4.8e-03  9.4e-05  4.34e-01   6.797930779e+06   6.797932058e+06   3.4e-05  20.68
47  6.8e-02  4.8e-03  9.3e-05  4.48e-01   6.800858510e+06   6.800859777e+06   3.4e-05  21.03
48  6.7e-02  4.9e-03  9.3e-05  4.83e-01   6.800783287e+06   6.800784557e+06   3.4e-05  21.37
49  6.4e-02  5.2e-03  8.7e-05  4.60e-01   6.810006223e+06   6.810007470e+06   3.2e-05  21.82
50  6.0e-02  5.1e-03  8.0e-05  5.09e-01   6.824482548e+06   6.824483736e+06   3.0e-05  22.16
51  5.9e-02  5.3e-03  7.9e-05  5.35e-01   6.824782123e+06   6.824783317e+06   3.0e-05  22.51
52  5.6e-02  5.5e-03  7.4e-05  5.17e-01   6.834133863e+06   6.834135029e+06   2.8e-05  22.86
53  5.1e-02  6.1e-03  6.7e-05  5.09e-01   6.848324756e+06   6.848325890e+06   2.6e-05  23.35
54  4.6e-02  6.8e-03  5.8e-05  4.47e-01   6.866772066e+06   6.866773158e+06   2.3e-05  23.83
55  4.3e-02  7.4e-03  5.4e-05  4.27e-01   6.874540819e+06   6.874541902e+06   2.2e-05  24.19
56  4.2e-02  6.3e-03  5.2e-05  7.27e-01   6.884090727e+06   6.884091741e+06   2.1e-05  24.53
57  3.8e-02  5.8e-03  4.4e-05  6.35e-01   6.905546491e+06   6.905547409e+06   1.9e-05  24.87
58  3.4e-02  5.6e-03  3.9e-05  6.92e-01   6.921597910e+06   6.921598761e+06   1.7e-05  25.23
59  2.8e-02  5.7e-03  2.9e-05  7.21e-01   6.952782644e+06   6.952783370e+06   1.4e-05  25.72
60  2.5e-02  5.8e-03  2.5e-05  7.62e-01   6.967973069e+06   6.967973735e+06   1.3e-05  26.21
61  2.2e-02  5.6e-03  2.1e-05  8.04e-01   6.981557363e+06   6.981557973e+06   1.1e-05  26.55
62  2.1e-02  5.6e-03  2.0e-05  8.31e-01   6.986843872e+06   6.986844459e+06   1.1e-05  26.89
63  1.7e-02  5.5e-03  1.4e-05  8.41e-01   7.012851580e+06   7.012852061e+06   8.5e-06  27.29
64  1.6e-02  5.4e-03  1.3e-05  8.81e-01   7.017605979e+06   7.017606440e+06   8.1e-06  27.64
65  1.2e-02  5.6e-03  9.0e-06  8.89e-01   7.040733173e+06   7.040733541e+06   6.2e-06  28.13
66  1.1e-02  5.6e-03  8.1e-06  9.26e-01   7.045550751e+06   7.045551099e+06   5.8e-06  28.47
67  9.3e-03  6.2e-03  6.1e-06  9.31e-01   7.058251214e+06   7.058251510e+06   4.8e-06  28.87
68  9.0e-03  6.2e-03  5.9e-06  9.39e-01   7.059820413e+06   7.059820703e+06   4.7e-06  29.26
69  8.8e-03  6.0e-03  5.7e-06  9.44e-01   7.061731597e+06   7.061731879e+06   4.5e-06  29.60
70  8.3e-03  6.1e-03  5.4e-06  9.46e-01   7.064698114e+06   7.064698384e+06   4.3e-06  30.00
71  8.0e-03  6.0e-03  5.1e-06  9.48e-01   7.066897297e+06   7.066897558e+06   4.1e-06  30.35
72  7.5e-03  6.1e-03  4.7e-06  9.52e-01   7.070138232e+06   7.070138480e+06   3.9e-06  30.70
73  6.9e-03  6.3e-03  4.1e-06  9.52e-01   7.074565860e+06   7.074566090e+06   3.6e-06  31.04
74  6.7e-03  6.2e-03  4.0e-06  9.60e-01   7.076130811e+06   7.076131035e+06   3.5e-06  31.39
75  6.3e-03  6.2e-03  3.6e-06  9.60e-01   7.078670594e+06   7.078670808e+06   3.3e-06  31.73
76  5.8e-03  6.4e-03  3.2e-06  9.62e-01   7.082114208e+06   7.082114408e+06   3.1e-06  32.07
77  5.6e-03  6.1e-03  2.9e-06  8.59e-01   7.083937071e+06   7.083937264e+06   2.9e-06  32.87
78  5.5e-03  6.0e-03  2.9e-06  8.52e-01   7.084524828e+06   7.084525019e+06   2.9e-06  33.64
79  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  34.35
80  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  35.36
81  5.1e-03  5.6e-03  2.6e-06  8.47e-01   7.087077489e+06   7.087077672e+06   2.7e-06  36.37
Optimizer terminated. Time: 37.57


Interior-point solution summary
  Problem status  : PRIMAL_AND_DUAL_FEASIBLE
  Solution status : OPTIMAL
  Primal.  obj: 7.0826522774e+06    nrm: 1e+05    Viol.  con: 1e+01    var: 1e+02
  Dual.    obj: 7.0914996991e+06    nrm: 3e+06    Viol.  con: 1e-13    var: 1e+04
Optimizer summary
  Optimizer                 -                        time: 37.57
    Interior-point          - iterations : 82        time: 37.55
      Basis identification  -                        time: 0.00
        Primal              - iterations : 0         time: 0.00
        Dual                - iterations : 0         time: 0.00
        Clean primal        - iterations : 0         time: 0.00
        Clean dual          - iterations : 0         time: 0.00
```

```
    Simplex                 -                       time: 0.00
      Primal simplex        - iterations : 0        time: 0.00
      Dual simplex          - iterations : 0        time: 0.00
    Mixed integer           - relaxations: 0        time: 0.00

Mosek returned an error or warning, open the following link in your browser:
https://docs.mosek.com/latest/toolbox/response-codes.html#mosek.rescode.trm_stall
     -12  min(sbl) = min(A*x - bl), (should be positive)
     -12  min(sbu) = min(bu - A*x), (should be positive)
[mosek] reports OPTIMAL but Primal optimality condition in solveCobraQP not satisfied, residual = 1.1173,
```

```
T.method = replace(T.method,'MSK_OPTIMIZER_','');
T.method = replace(T.method,'_',' ');
T.solver = replace(T.solver,'ibm_','');
T.approach = append(T.solver, ' ', T.method);
T =sortrows(T,{'stat','time'},{'ascend','ascend'});
display(T)
```

T = 3×13 table

...

	interface	solver	method	problem	model	stat	origStat
1	"optimizeCbModel"	"cplex"	"BARRIER"	"QP"	"Harvey"	1	"optimal"
2	"optimizeCbModel"	"mosek"	"CONIC"	"QP"	"Harvey"	1	"OPTIMAL"
3	"optimizeCbModel"	"gurobi"	"BARRIER"	"QP"	"Harvey"	1	"OPTIMAL"

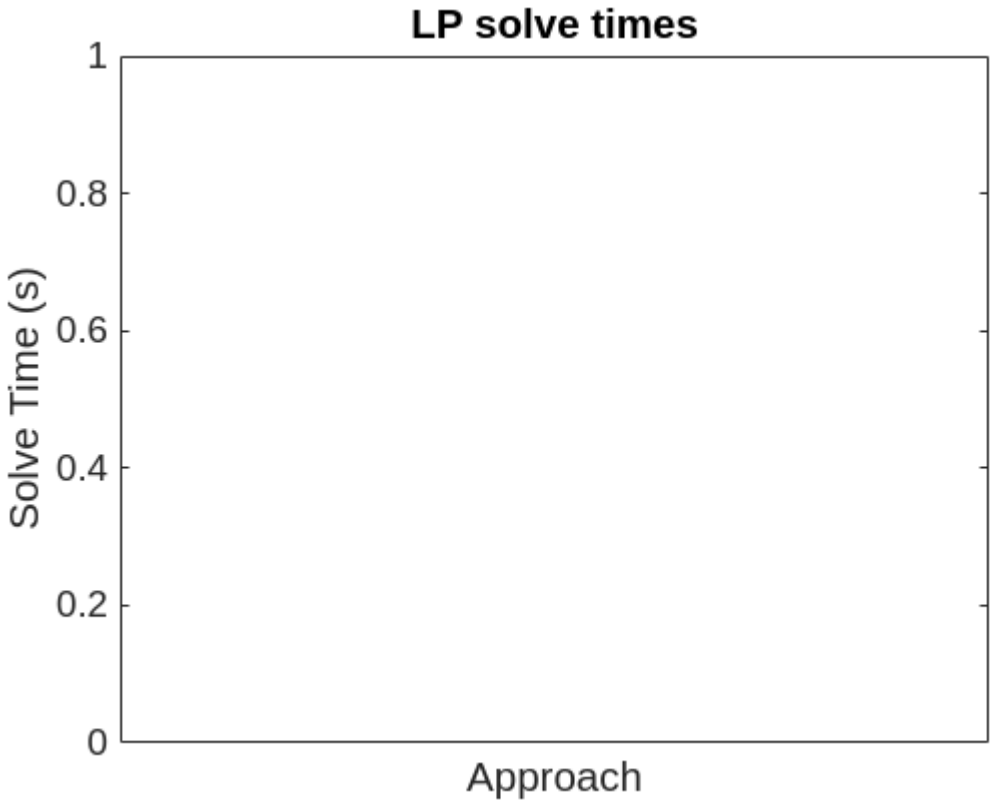
```
save([resultsFolder 'results_benchmarkWBMsolvers.mat'],'T')
```

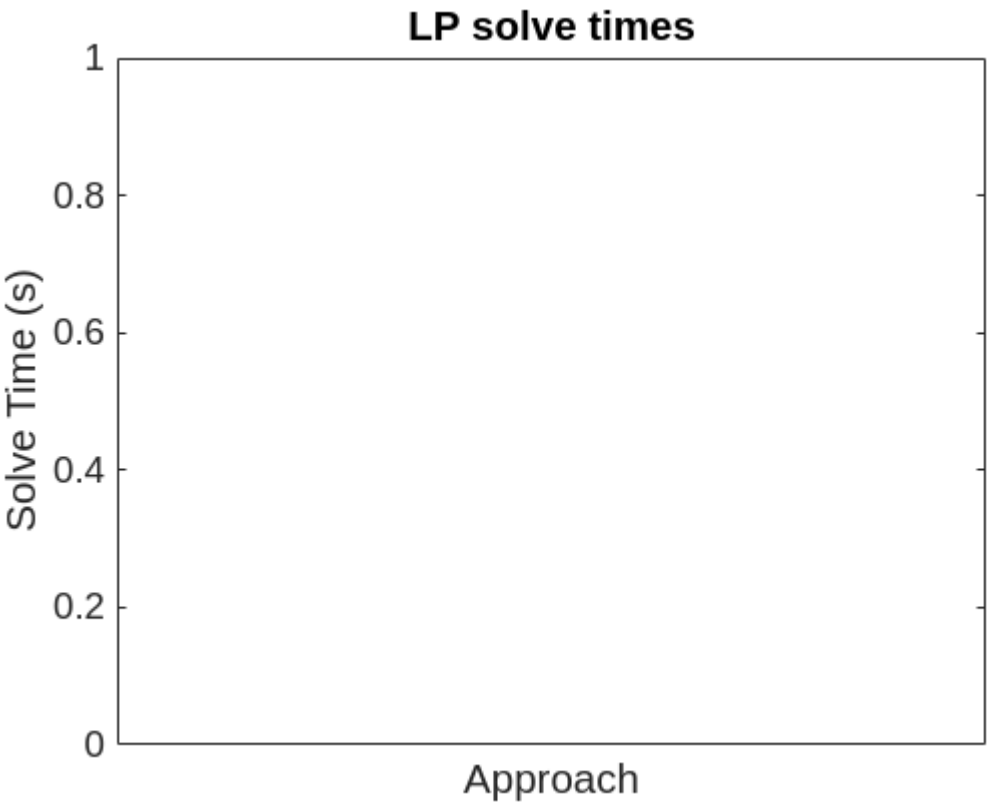
```
if 1
    if 0
        % Create the first histogram
        histogram(T.time(T.stat==1 &
strcmp(T.problem,'LP')),'NumBins',100,'FaceColor', 'r','FaceAlpha', 0.5);  %
'r' sets the color to red
        hold on;  % Keep the current plot so that the second histogram is
overlaid

        % Create the second histogram
        histogram(T.time(T.stat==1 &
strcmp(T.problem,'QP')),'NumBins',100,'FaceColor', 'b','FaceAlpha', 0.5);  %
'r' sets the color to red
        xlabel({'Whole body metabolic model LP solution time (seconds)',
[int2str(nMet) ' metabolites, ' int2str(nRxn) ' reactions.']})
        ylabel('Number of solutions')
        title('Solution time depends on solver, method and problem');
        legend('LP', 'QP');
        hold off;  % Release the hold for future plots

    else
        if ~exist('T0','var')
```

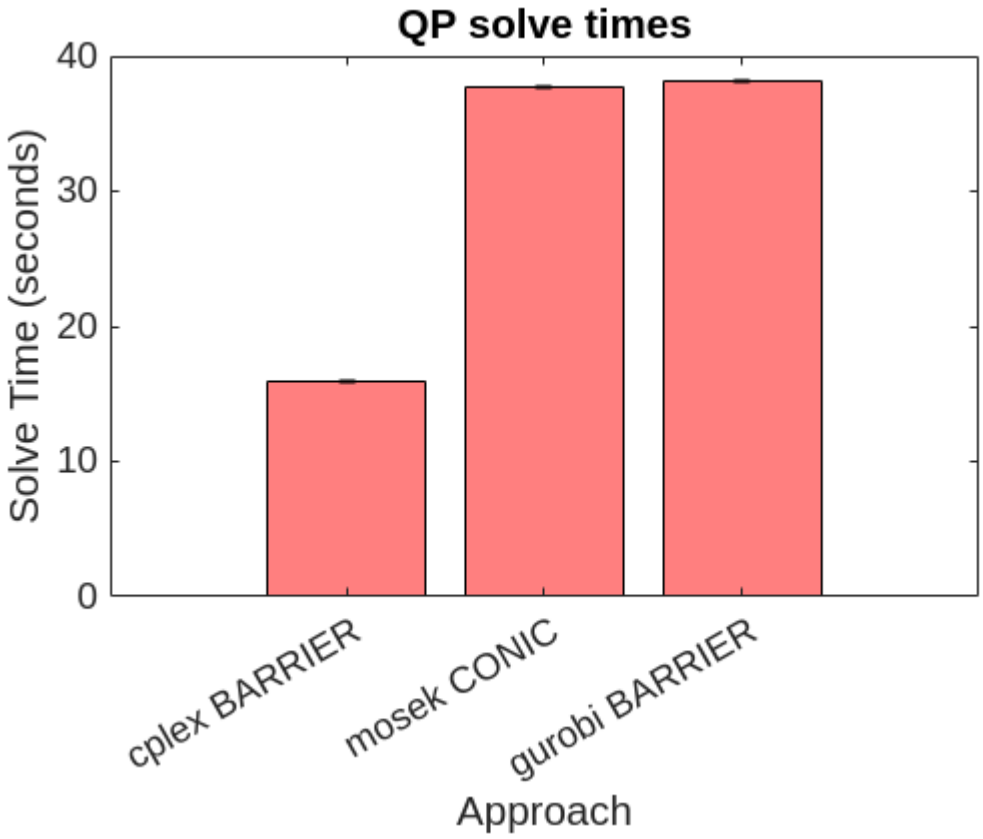
```
            T0 = T;
        else
            T = T0;
        end
        figure
        % Concatenate solver and method into 'approach'
        T.approach = append(T.solver, ' ', T.method);

        T = T(strcmp(T.problem,'LP'),:);
        % Calculate the mean solve time and standard deviation for each
approach
        avg_times = varfun(@mean, T, 'InputVariables', 'time',
'GroupingVariables', 'approach');
        std_times = varfun(@std, T, 'InputVariables', 'time',
'GroupingVariables', 'approach');

        times = avg_times;
        times.std_time = std_times.std_time;
        % Sort both the avg_times and std_times by the mean solve time
        [times, sort_idx] = sortrows(times, 'mean_time');

        % Create a bar plot with the sorted data
        b = bar(times.mean_time, 'FaceColor', 'b', 'FaceAlpha', 0.5);
        hold on;

        % Add error bars using the sorted standard deviations
        errorbar(times.mean_time, times.std_time, 'k', 'linestyle', 'none',
'LineWidth', 1.5);
        xticks(1:length(times.approach))
        xticklabels(times.approach)

        % Add labels and title
        xlabel('Approach', 'Interpreter', 'none');
        ylabel('Solve Time (s)');
        title('LP solve times', 'Interpreter', 'none');

        figure
        % fastest times
        times = times(times.mean_time<mean(times.mean_time),:);
        % Create a bar plot with the sorted data
        b = bar(times.mean_time, 'FaceColor', 'b', 'FaceAlpha', 0.5);
        hold on;

        % Add error bars using the sorted standard deviations
        errorbar(times.mean_time, times.std_time, 'k', 'linestyle', 'none',
'LineWidth', 1.5);
        xticks(1:length(times.approach))
        xticklabels(times.approach)

        % Add labels and title
```

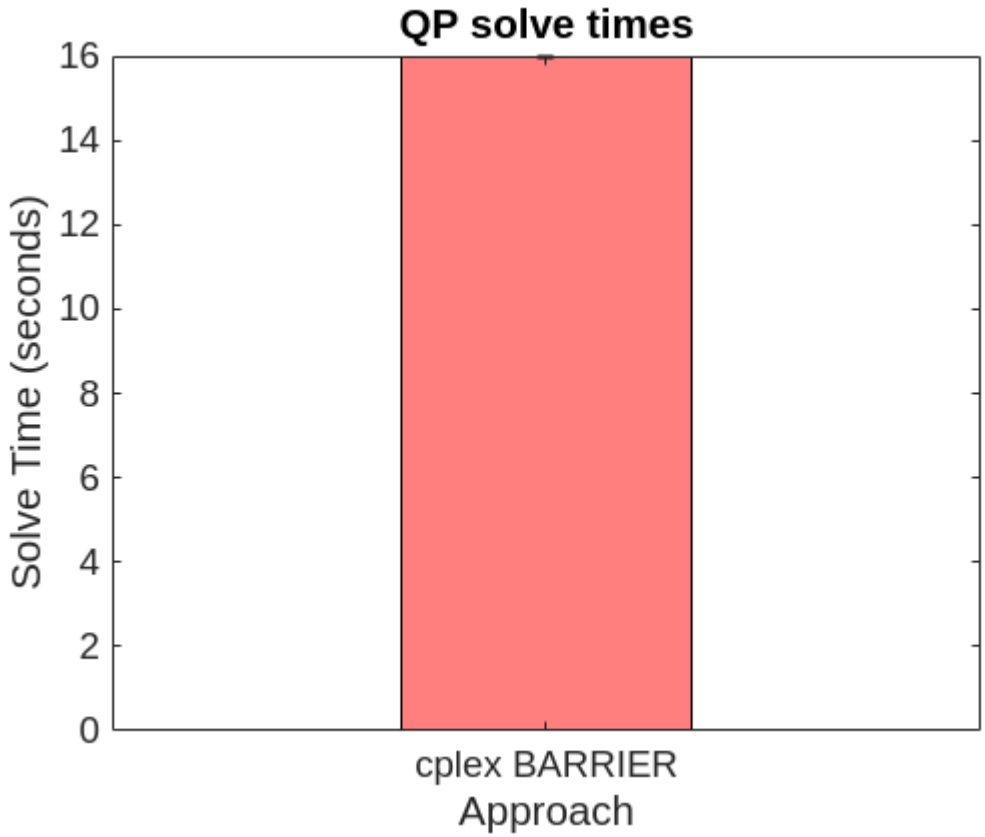
```
        xlabel('Approach', 'Interpreter', 'none');
        ylabel('Solve Time (s)');
        title('LP solve times', 'Interpreter', 'none');


        T = T0;
        figure
        T = T(strcmp(T.problem,'QP'),:);
        % Calculate the mean solve time and standard deviation for each
approach
        avg_times = varfun(@mean, T, 'InputVariables', 'time',
'GroupingVariables', 'approach');
        std_times = varfun(@std, T, 'InputVariables', 'time',
'GroupingVariables', 'approach');

        times = avg_times;
        times.std_time = std_times.std_time;
        % Sort both the avg_times and std_times by the mean solve time
        [times, sort_idx] = sortrows(times, 'mean_time');

        % Create a bar plot with the sorted data
        b = bar(times.mean_time, 'FaceColor', 'r', 'FaceAlpha', 0.5);
        hold on;

        % Add error bars using the sorted standard deviations
        errorbar(times.mean_time, times.std_time, 'k', 'linestyle', 'none',
'LineWidth', 1.5);
        xticks(1:length(times.approach))
        xticklabels(times.approach)

        % Add labels and title
        xlabel('Approach', 'Interpreter', 'none');
        ylabel('Solve Time (seconds)');
        title('QP solve times', 'Interpreter', 'none');

        T = T0;
        figure
        % fastest times
        times = times(times.mean_time<mean(times.mean_time),:);
        % Create a bar plot with the sorted data
        b = bar(times.mean_time, 'FaceColor', 'r', 'FaceAlpha', 0.5);
        hold on;
        % Add error bars using the sorted standard deviations
        errorbar(times.mean_time, times.std_time, 'k', 'linestyle', 'none',
'LineWidth', 1.5);
        xticks(1:length(times.approach))
        xticklabels(times.approach)

        % Add labels and title
        xlabel('Approach', 'Interpreter', 'none');
```

```
        ylabel('Solve Time (seconds)');
        title('QP solve times', 'Interpreter', 'none');
    end
end
```




LP solve times
1
0.8
0.6
0.4
0.2
0
Solve Time (s)
Approach


QP solve times
40
30
20
10
0
Solve Time (seconds)
cplex BARRIER
mosek CONIC
gurobi BARRIER
Approach


QP solve times
Solve Time (seconds)
16
14
12
10
8
6
4
2
0
cplex BARRIER
Approach