```
% This script file computes the optimal investment levels TAopt and
 TBopt
% for the two firms (Firm A and Firm B) of the Simplified
 Oligopolistic
% Optimal Influence model which also involves a single consumer C. The
 rest
% of the model parameters (Sopt,Xopt,Popt,Qopt,Fopt) are also
 computed.

% Specifically, the internal model parameters whose optimal values are
 to
% be determined during the grid searching optimization process are the
% follwing:
% (i):   [TAopt,TBopt]
% (ii):  [SAopt,SCopt,SBopt]
% (iii): [XAopt,XBopt]
% (iv):  [PAopt,PBopt] (These are the optimal prices!!!)
% (v):   [QAopt,QBopt]
% (vi):  [FAopt FBopt FA_rev_opt FB_rev_opt FA_cost_opt FB_rev_opt]

% The grid searching process will be conducted within the 8-
dimensional
% space defined by the Cartesian product of the following external
 model
% parameters:
% (i):    LA (direct influnce exerted by consumer C on Firm A).
% (ii):   LB (direct influnce exerted by consumer C on Firm B).
% (iii):  PA (initial belief consumer C holds for product A).
% (iv):   PB (initial belief consumer C holds for product B).
% (v):    M  (sensitivity coefficient).
% (vi):   K  (sensitivity coefficient).
% (vii):  C  (marginal cost).
% (viii): G  (marginal influence cost or Gamma).

% IMPORTANT NOTE!!!
% Mind that the underlying continuous game may not have an equlibrium
 point
% for any given configuration of the external parameters. Since all
 the
% internal parameters to be determined accept positive values, the
 value of
% (-1) will be utilized in order to indicate the absence of an
 equilibrium
% point for a particular configuration of the external parameters.

% Define ranges and corresponding increment step for each external
 parameter
% of the Simplified Oligopolistic Optimal Influence model.

clc
clear all
```

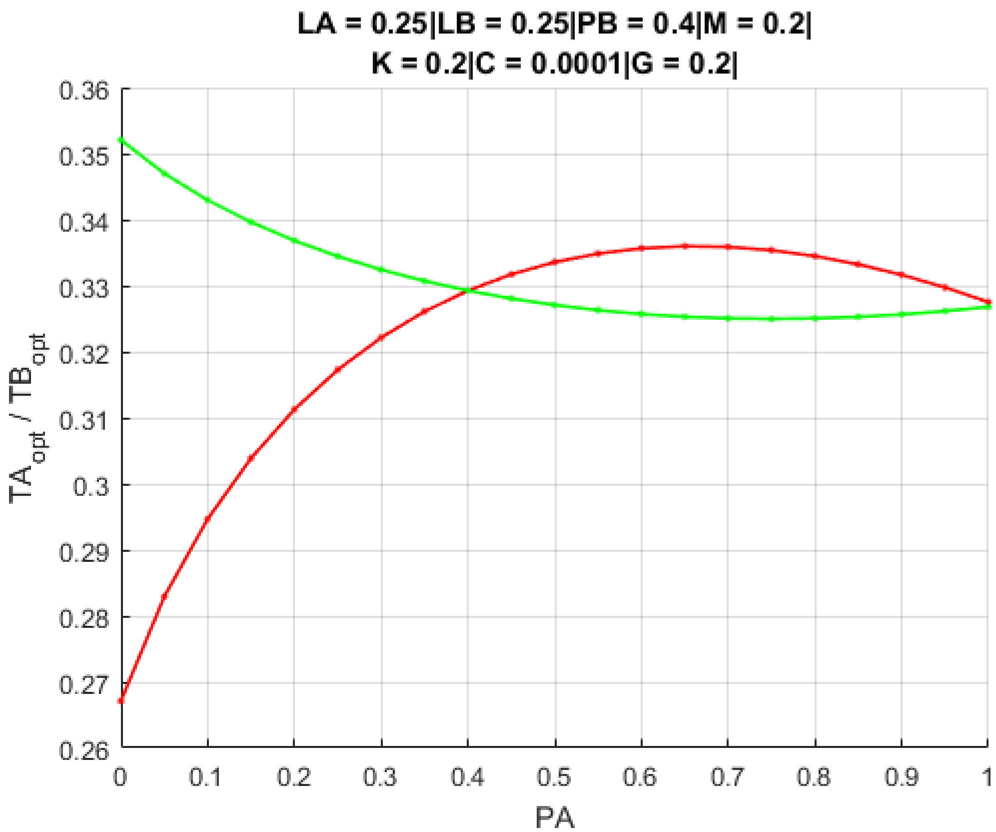
```
% Construct a cell array storing the names of model parameters.
ParamsNames = {'LA','LB','PA','PB','M','K','C','G'};

% Set the varying parameter index.
ParamIndex = 3;

% Parameter #1
LA_MIN = 0.25;
LA_MAX = 0.25;
LA_STEP = 0.01;
% Parameter #2
LB_MIN = 0.25;
LB_MAX = 0.25;
LB_STEP = 0.01;
% Parameter #3
PA_MIN = 0.00;
PA_MAX = 1.00;
PA_STEP = 0.05;
% Parameter #4
PB_MIN = 0.40;
PB_MAX = 0.40;
PB_STEP = 0.01;
% Parameter #5
M_MIN = 0.2;
M_MAX = 0.2;
M_STEP = 0.05;
% Parameter #6
K_MIN = 0.2;
K_MAX = 0.2;
K_STEP = 0.05;
% Parameter #7
C_MIN = 0.0001;
C_MAX = 0.0001;
C_STEP = 0.00001;
% Parameter #8 (G or Gamma!!!)
G_MIN = 0.2;
G_MAX = 0.2;
G_STEP = 0.01;

% Set the corresponding ranges for each parameter of the model.
LA_RANGE = [LA_MIN:LA_STEP:LA_MAX];
LB_RANGE = [LB_MIN:LB_STEP:LB_MAX];
PA_RANGE = [PA_MIN:PA_STEP:PA_MAX];
PB_RANGE = [PB_MIN:PB_STEP:PB_MAX];
M_RANGE = [M_MIN:M_STEP:M_MAX];
K_RANGE = [K_MIN:K_STEP:K_MAX];
C_RANGE = [C_MIN:C_STEP:C_MAX];
G_RANGE = [G_MIN:G_STEP:G_MAX];

% Initialize matrices that store fundamental intermediate quantities
 of the
% underlying optimization problem within the simplified oligopolistic
% enviroment as well as the values of the external optimization
 parameters
```

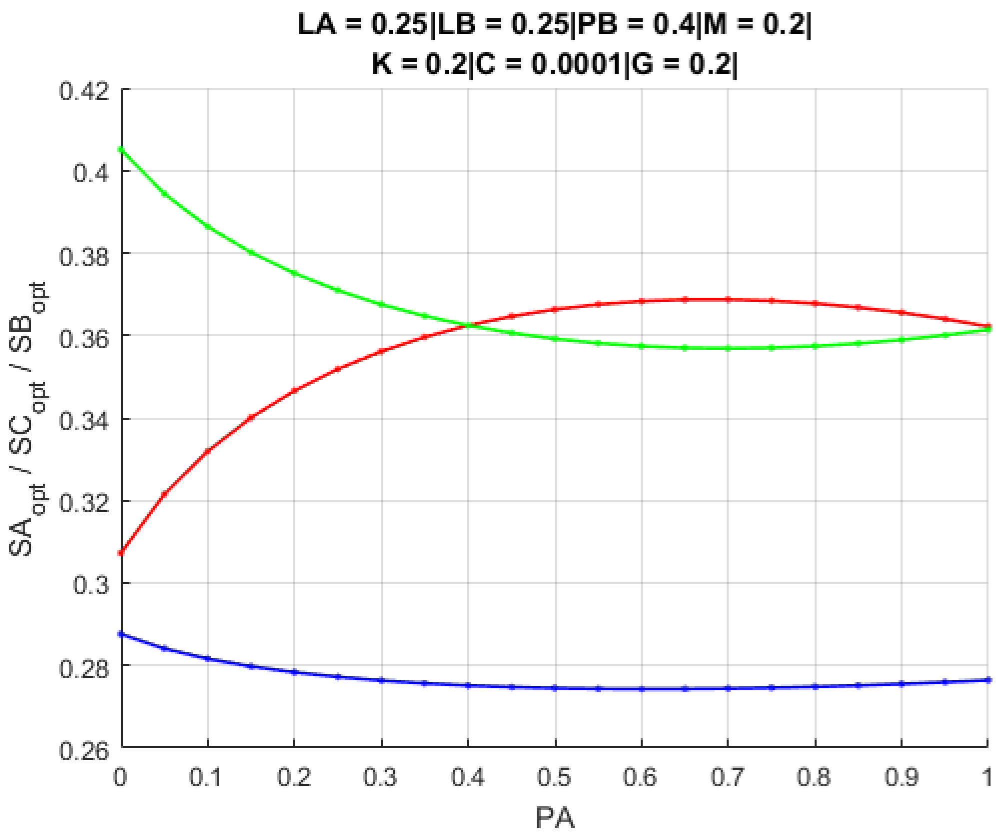
```
% for each step of the multi-dimensional grid searching process.
Topts = []; % Optimal investment levels.
Sopts = []; % Optimal limiting influences.
Xopts = []; % Optimal limiting beliefs.
Popts = []; % Optimal prices.
Qopts = []; % Optimal quantities.
Fopts = []; % Optimal profits.
FilterFlags = []; % Solution filtering flag which may be indicative of
 a non-existing solution.
DigitAccuracies = []; % Length of maximal sequence of identical digits
 within the obtained optimal solutions.
Params = []; % 8-tuple of the varying optimization parameters.

% Initialize internal solver parameters.
% Set the number of initial solution points.
N = 50;
% Set the tolerance value for the minimizer. (Preferable value =
 1e-10)
Tolerance = 1e-15;
% Set the Fvals tolerance value for filtering the obtained solutions.
 (Preferable value = 1e-15)
FvalsTolerance = 1e-15;
% Set the derivative tolerance value for filering the obtained
 solutions. (Preferable value = 1e-08).
DerivativeTolerance = 1e-10;
% Set the maximum number of iterations to be conducted by the
 optimizer.
MaxIterations = 1000;
% Set the maximum number of function evaluations to be conducted by
 the
% optimizer.
MaxFunctionEvaluations = 10000;
% Set the display flag parameter to 'off'.
DisplayFlag = 'off';

fprintf('Grid Evaluation Process in Progess...\n');
% Perform the actual grid searching.
for LA = LA_RANGE
    for LB = LB_RANGE
        for PA = PA_RANGE
            for PB = PB_RANGE
                for M = M_RANGE
                    for K = K_RANGE
                        for C = C_RANGE
                            for G = G_RANGE
                                % Additional parameters definition.
                                alpha = (K*M - 2) / (M^2 - 4);
                                beta = (2*K - M) / (M^2 - 4);
                                gamma = C / (M - 2);
                                gamma_prime = gamma * (M - 1); %
 gamma_prime is the gamma' parameter.
                                % Set the current params vector.
                                params = [LA LB PA PB M K C G];
                                Params = [Params;params];
```

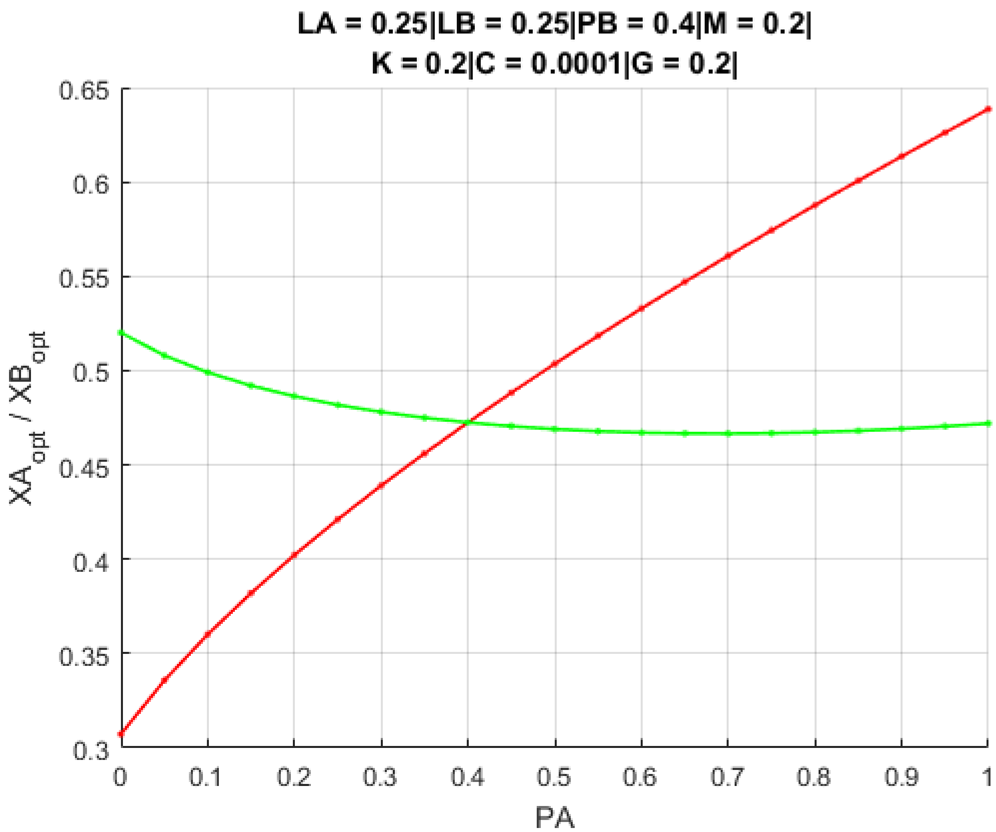
```
 [TAopt,TBopt,FilterFlag,DigitsAccuracy] =
 SimplifiedOligopolisticOptimalInfluences(N,DisplayFlag,Tolerance,FvalsTolerance,D
                                T = [TAopt TBopt];
                                if(FilterFlag==0)
                                    [S,X,P,Q,F] =
 RetrieveOptimalModelParameters(T,C,G,LA,LB,PA,PB,alpha,beta,gamma,gamma_prime);
                                else
                                    S = -1*ones(1,3);
                                    X = -1*ones(1,2);
                                    P = -1*ones(1,2);
                                    Q = -1*ones(1,2);
                                    F = -1*ones(1,6);
                                end;
                                    % Print current solution.
                                param_value_string =
 strcat([ParamsNames{ParamIndex} ' = ' num2str(params(ParamIndex))]);
                                fprintf('%s TAopt =
 %f TBopt = %f FilterFlag = %d DigitsAccuracy = %d
\n',param_value_string,TAopt,TBopt,FilterFlag,DigitsAccuracy);
                                Topts = [Topts;T];
                                FilterFlags =
 [FilterFlags;FilterFlag];
                                DigitsAccuracies =
 [DigitAccuracies;DigitsAccuracy];
                                Sopts = [Sopts;S];
                                Xopts = [Xopts;X];
                                Popts = [Popts;P];
                                Qopts = [Qopts;Q];
                                Fopts = [Fopts;F];
                            end
                        end
                    end
                end
            end
        end
    end
end

% Set parameters' indices and corresponding values for the parameters
 that
% remain constant.
ConstIndices = setdiff(1:length(params),ParamIndex);
ConstValues = params(ConstIndices);
% Perform plotting operations.
plot_parameters_tuples(Topts,Sopts,Xopts,Popts,Qopts,Fopts,Params,ParamIndex,Const
```

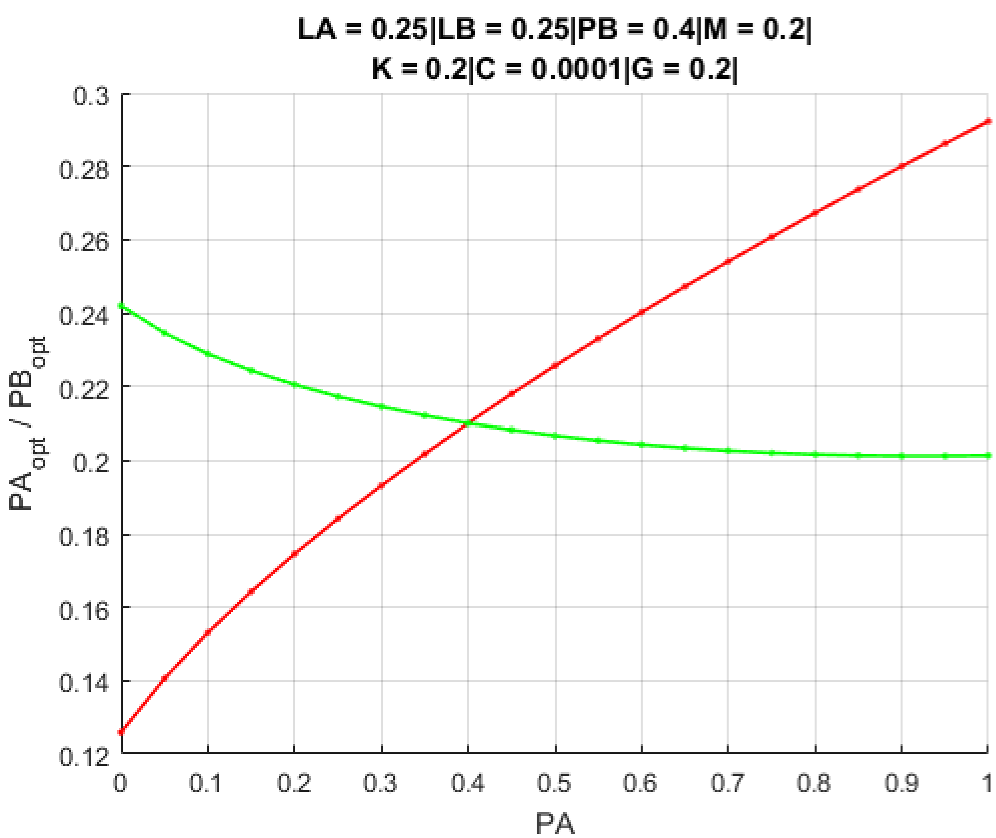
Grid Evaluation Process in Progess...
PA = 0 TAopt = 0.267093 TBopt = 0.352071 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.05 TAopt = 0.282917 TBopt = 0.346988 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.1 TAopt = 0.294675 TBopt = 0.342955 FilterFlag = 0 DigitsAccuracy = 11

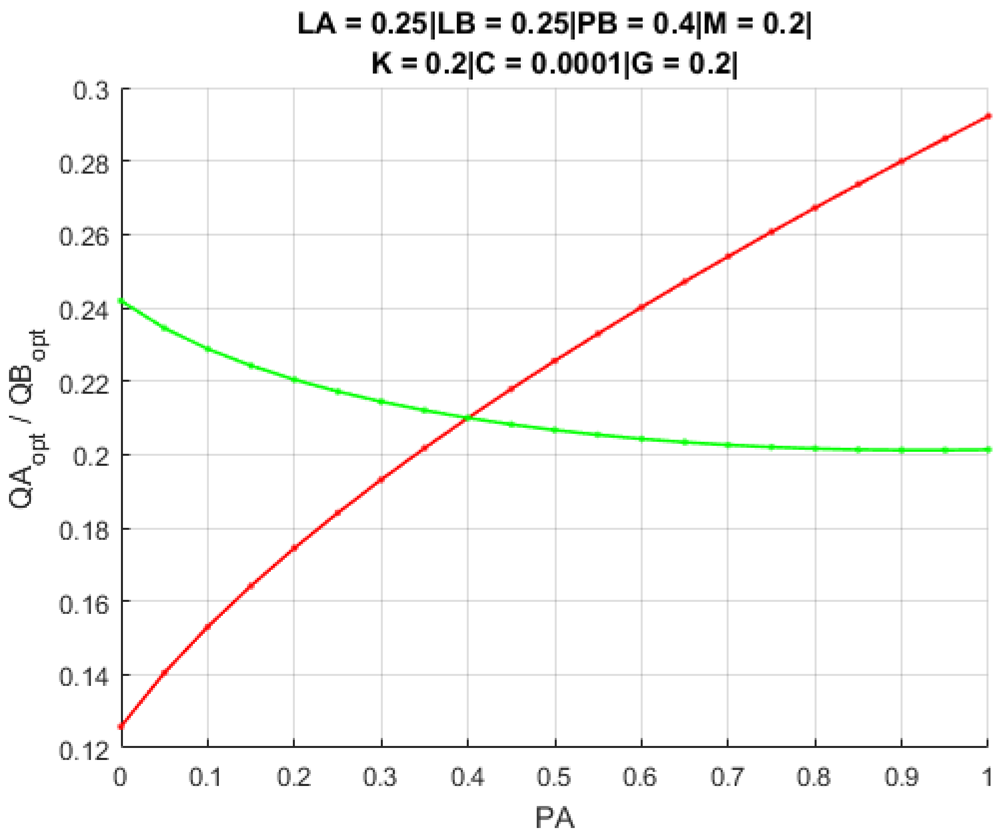
PA = 0.15 TAopt = 0.303879 TBopt = 0.339626 FilterFlag = 0 DigitsAccuracy = 14
PA = 0.2 TAopt = 0.311266 TBopt = 0.336824 FilterFlag = 0 DigitsAccuracy = 13
PA = 0.25 TAopt = 0.317264 TBopt = 0.334446 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.3 TAopt = 0.322145 TBopt = 0.332422 FilterFlag = 0 DigitsAccuracy = 11
PA = 0.35 TAopt = 0.326097 TBopt = 0.330703 FilterFlag = 0 DigitsAccuracy = 11
PA = 0.4 TAopt = 0.329255 TBopt = 0.329255 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.45 TAopt = 0.331718 TBopt = 0.328049 FilterFlag = 0 DigitsAccuracy = 10
PA = 0.5 TAopt = 0.333567 TBopt = 0.327066 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.55 TAopt = 0.334861 TBopt = 0.326287 FilterFlag = 0 DigitsAccuracy = 10
PA = 0.6 TAopt = 0.335650 TBopt = 0.325699 FilterFlag = 0 DigitsAccuracy = 13
PA = 0.65 TAopt = 0.335976 TBopt = 0.325291 FilterFlag = 0 DigitsAccuracy = 11
PA = 0.7 TAopt = 0.335870 TBopt = 0.325052 FilterFlag = 0 DigitsAccuracy = 12
PA = 0.75 TAopt = 0.335361 TBopt = 0.324975 FilterFlag = 0 DigitsAccuracy = 13
PA = 0.8 TAopt = 0.334474 TBopt = 0.325053 FilterFlag = 0 DigitsAccuracy = 14
PA = 0.85 TAopt = 0.333228 TBopt = 0.325278 FilterFlag = 0 DigitsAccuracy = 11
PA = 0.9 TAopt = 0.331640 TBopt = 0.325646 FilterFlag = 0 DigitsAccuracy = 13
PA = 0.95 TAopt = 0.329725 TBopt = 0.326152 FilterFlag = 0 DigitsAccuracy = 9
PA = 1 TAopt = 0.327496 TBopt = 0.326792 FilterFlag = 0 DigitsAccuracy = 11

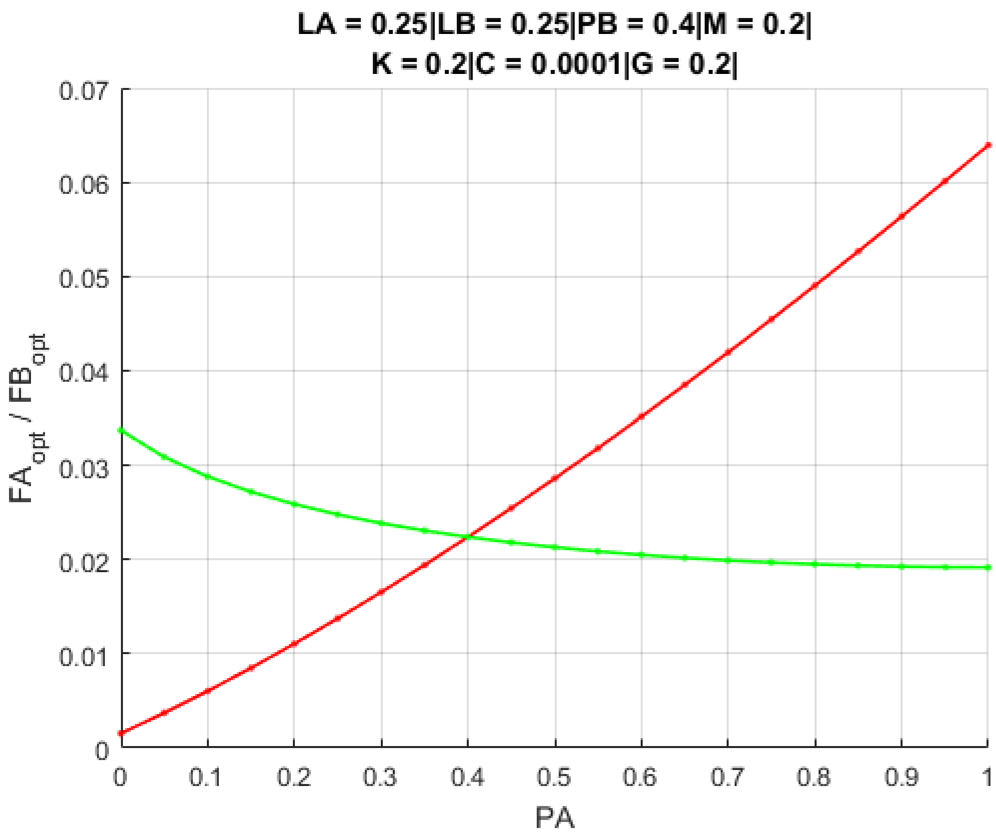

LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
TA_opt / TB_opt
PA

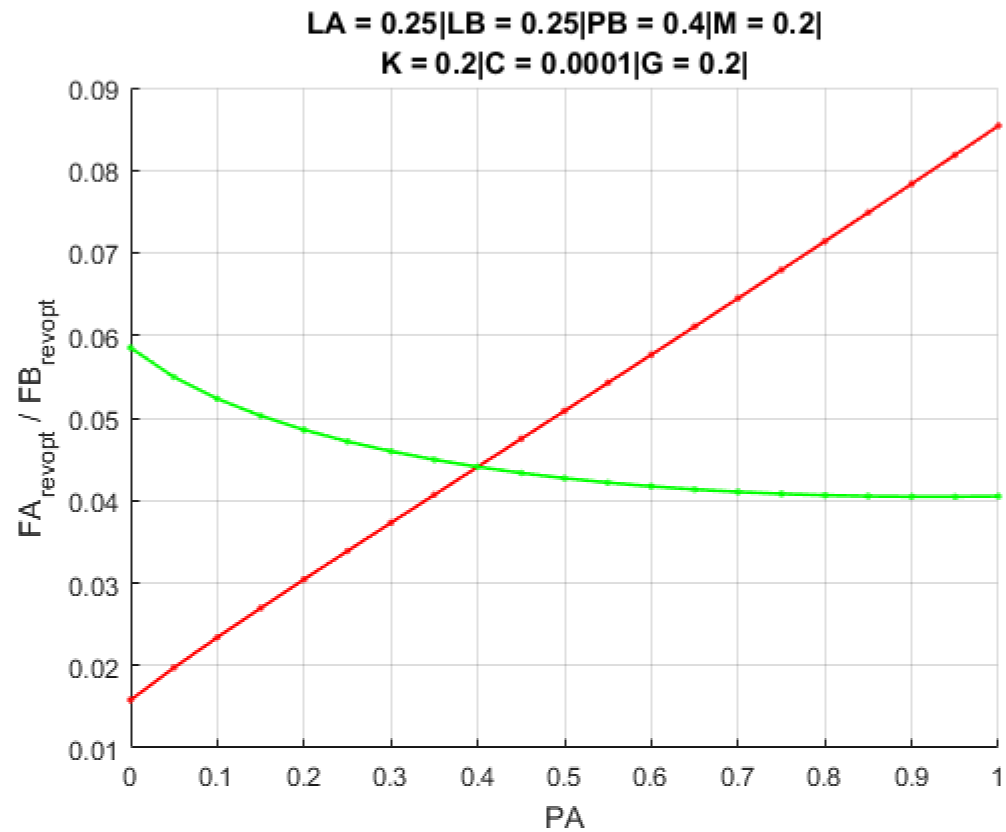

LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
SA_opt / SC_opt / SB_opt
PA

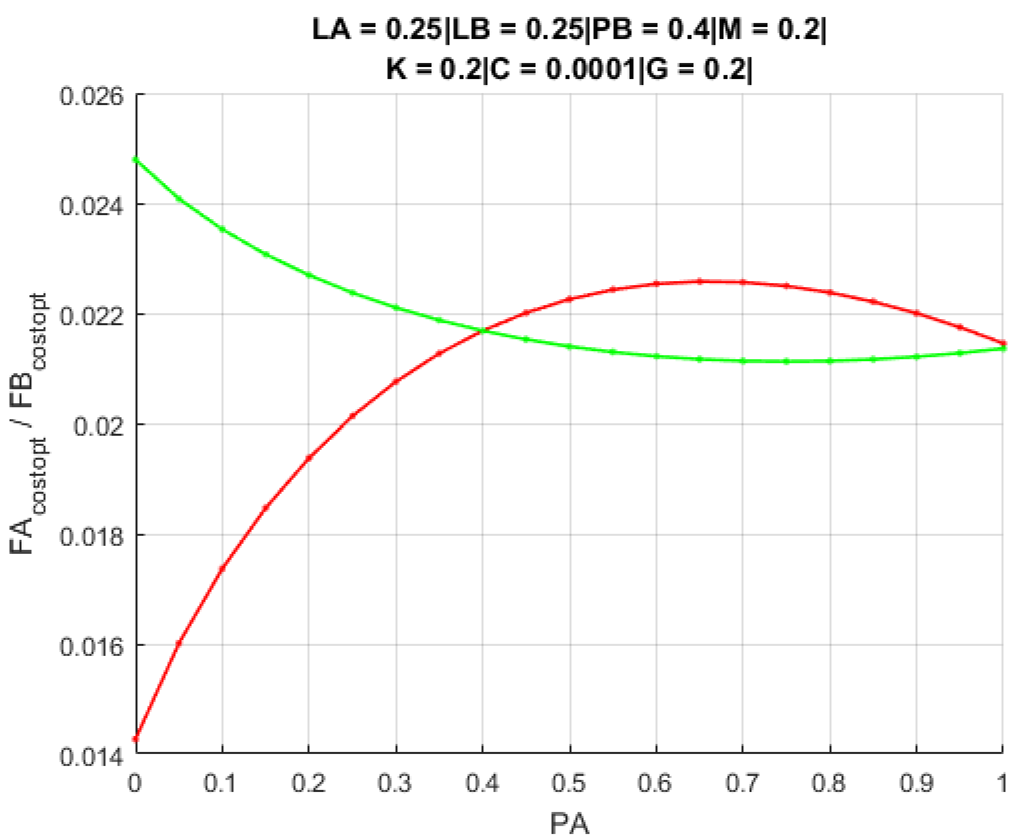

LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
XA$_{opt}$ / XB$_{opt}$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
PA


LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
PA$_{opt}$ / PB$_{opt}$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
PA


LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
QA$_{opt}$ / QB$_{opt}$
PA
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
FA$_{opt}$ / FB$_{opt}$
PA
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
FA_{revopt} / FB_{revopt}
PA
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


LA = 0.25|LB = 0.25|PB = 0.4|M = 0.2|
K = 0.2|C = 0.0001|G = 0.2|
$FA_{costopt}$ / $FB_{costopt}$
PA
0.026
0.024
0.022
0.02
0.018
0.016
0.014
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Published with MATLAB® R2017a