EQUILIBRIUM SETS AND RELATED ASPECTS IN NETWORK INTERACTIONS

N. Hemachandra
(Based on joint work with Tushar K. Satish)

Industrial Engineering and Operations Research
Indian Institute of Technology Bombay

March, 2016

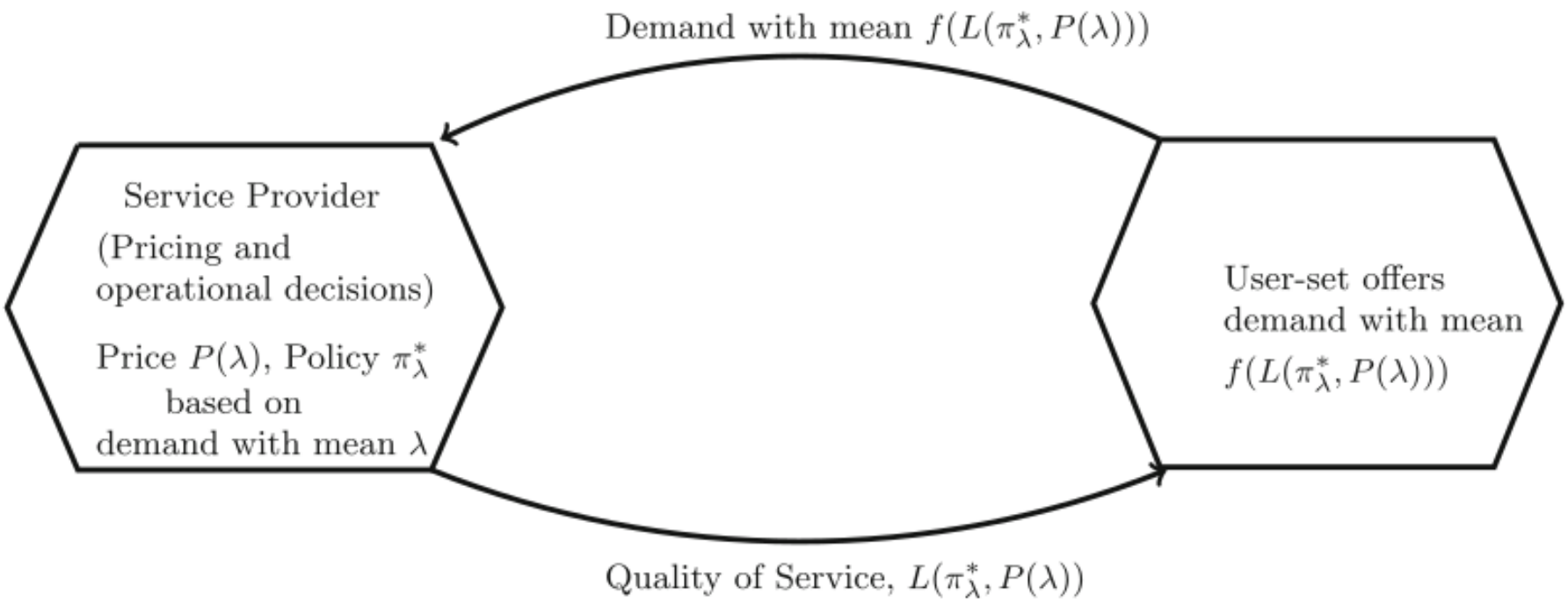
Outline

Background I

- Tech/Engg systems offer various services
- Demand/Usage depends on offered Quality of Service
- Offered QoS depends on demand (finite resources like capacities, etc.)
- Equilibrium points, Equilibrium sets, etc. ?
- Network based interactions?
- Characterization of Equilibrium sets, etc.?

Background II

A schematic for service-provider user-set interaction



Related Work

- Equilibrium sets when resources are allocated as Markov decision models[4]
- Equilibrium sets when queue resources are shared[5]
- Equilibrium points when queue is priced[7]

Introduction

- Network Based Interactions
  - Users : The user is identified as a route, so every route is treated as a different consumer. This helps us differentiate users easily as a person might be using multiple resources on the network simultaneously.
  - Routes : A bundle of links having a unique origin and destination pair as a route.
  - Firms :A firm owns a certain amount of links in the network, and has a certain amount of capacity (bandwidth to offer on each of those links).

Network Market Design[6]

Though Network market design can solve the problems of Network utility maximisation, we need to consider

- Mechanism design can handle bundles but the mechanism becomes too complex in terms of computational costs, very quickly.
- Most theory available currently is for single sided auctions.
- Auction theory deals with mostly indivisible goods
- Network resource allocation frequently involves exchange between agents (some being buyers, others sellers), non-budget balanced mechanism are untenable.

Network Second Price Mechanism

- Each player submits his bid as a tuple (β_i, d_i)
- The network determines the allocation by solving

Model

$$\begin{aligned} \text{maximize} \quad & \sum_i \beta_i x_i \\ \text{subject to} \quad & x_i \quad \epsilon \quad [0, d_i] \end{aligned}$$

- The amount to be paid by each player is

Payment

$$P_i(b_i, b_{-i}) = \sum_{j \neq i} \beta_j (x_j^{-i*} - x_j^*)$$

Model with indivisible goods

Combinatorial Seller's Double auction[6]

Consider a network with N nodes and L links. Let R be the set of routes identified with the user set. Let J be the set of service providers.

Mathematical Model

$$\begin{aligned} \text{maximize} \quad & \sum_i \beta_i x_i - \sum_l \sum_j a_{jl} y_{jl} \\ \text{subject to} \quad & \sum_{i:l \in \mathcal{R}_i} x_i \leq \sum_{j:l \in L_j} y_{jl} \quad \forall \quad l \in L \\ & x_i \in \{0, \ldots, d_i\} \quad \forall \quad i \in R \\ & y_{jl} \in \{0, \ldots, c_{jl}\} \quad \forall \quad j \in J \quad , l \in L \end{aligned} \tag{1}$$

Notations

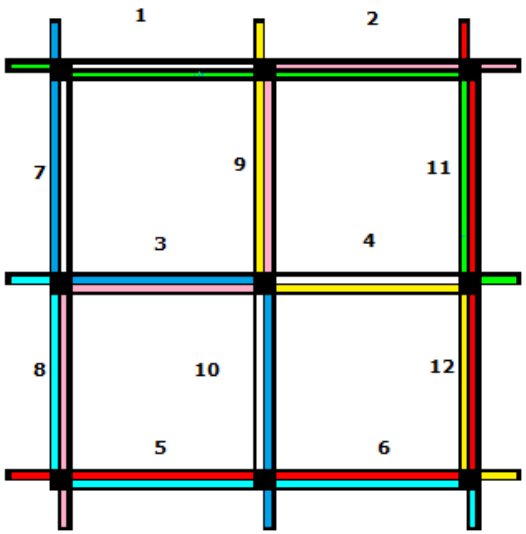
- β_i is the amount upto which user i is willing to pay
- a_{jl} is the amount required by service provider j to provide unit resource on link l.
- y_{jl} is the amount of resource provided by service provider j on link l
- d_i is the amount of resource demanded by route i
- c_{jl} is the capacity provided by service provider j on link l
- The first constraint implies that the total usage on a link must be less than or equal to the total allocation on the link.

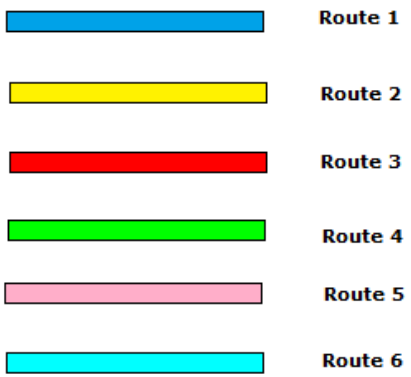
The Algorithm

1. Players first submit their bids which is a 2 tuple containing the amount they are willing to pay β_i and their demand d_i.
2. Sellers indicate the capacity they own on each link l,c_{lj} and the minimum amount they require to provide allocation on that link a_{lj}
3. We now run the matching problem in order to maximise the trading surplus.
4. After the allocation, the cost of each link is decided as $p_l = sup\{a_j : y_j > 0, l \epsilon L_j\}$
5. The cost of each route is charged as the sum total of all the links on the route. Players are then charged accordingly

Network Model - Example 1

Network with uniform links





Network Properties - Example 1

Network Characteristics

- Number of links, l is 12.
- Number of routes,r_i is 6.
- Number of firms, j are 3.
- Each link is owned by all 3 firms , having a capacity of 1 each.
- The firms charge \$1,2 and 3 respectively.

Demands and Allocation

- Each player submits a 2 - tuple as a bid. β_i represents how much he is willing to pay and d_i represents his demand. Below is the

Buyers(i)	1	2	3	4	5	6
Bids β_i	1	2	3	4	5	6
Demand d_i	2	3	2	3	2	3

Bid submitted by each player i

Buyers(i)	1	2	3	4	5	6
Allocation x_i	0	0	0	1	0	1

User Allocation:Example 1

Quality of Service: Single attribute based

- Quality of service is a way of capturing how satisfied or dissatisfied consumers are with a service or product provided by the firm.
- The measure of quality of service is

QoS

$$QoS = \sum_i \beta_i \frac{x_i - d_i}{d_i} \tag{2}$$

- We also investigated the effect of growing willingness to pay on the QoS for network in example 1 above.

Quality of Service: Single attribute based

Variation of QoS with increase in willingness to pay β_i

Buyers(i)	1	2	3	4	5	6	QoS
Bids β_i	1	2	3	4	5	6	-
Allocation x_i	0	0	0	0	0	1	-19
Bids β_i	2	3	4	5	6	7	-
Allocation x_i	0	0	0	0	0	1	-24.67
Bids β_i	3	4	5	6	7	8	-
Allocation x_i	0	1	0	1	0	1	-27
Bids β_i	4	5	6	7	8	9	-
Allocation x_i	0	1	0	1	0	1	-32
Bids β_i	5	6	7	8	9	10	-
Allocation x_i	1	1	0	1	0	2	-31.17

Quality of Service: Single attribute based

Variation of QoS with increase in willingness to pay β_i

Bids β_i	6	7	8	9	10	11	-
Allocation x_i	1	1	0	1	0	2	-35.33
Bids β_i	7	8	9	10	11	12	-
Allocation x_i	1	1	0	1	0	2	-39.5
Bids β_i	8	9	10	11	12	13	-
Allocation x_i	1	1	0	1	0	2	-43.67
Bids β_i	9	10	11	12	13	14	-
Allocation x_i	1	1	0	1	0	2	-47.83
Bids β_i	10	11	12	13	14	15	-
Allocation x_i	1	1	0	1	0	2	-52

Quality of Service: Tuple I

- As our model has two input values viz. price and quantity , single attribute based QoS cannot complete the feedback loop. This led us to investigate two tuple Quality of Service.
- The Quality of service is

QoS_1 : Demand

$$QoS_1 = \sum_i \frac{x_i}{d_i} \tag{3}$$

QoS_2: Payment

$$QoS_2 = \sum_i p_i \tag{4}$$

Quality of Service: Tuple II

- Clearly, both the QoS values are monotone with increasing price bids of the players. Hence, it can be used as a measure of allocations that the players receive.

2 tuple Quality of Service

Variation of 2 tuple QoS with increase in willingness to pay β_i

Buyers(i)	1	2	3	4	5	6	QoS_1	QoS_2
Bids β_i	1	2	3	4	5	6	-	-
Allocation x_i	0	0	0	0	0	1	0.33	5
Bids β_i	2	3	4	5	6	7	-	-
Allocation x_i	0	0	0	0	0	1	0.33	5
Bids β_i	3	4	5	6	7	8	-	-
Allocation x_i	0	1	0	1	0	1	1	13
Bids β_i	4	5	6	7	8	9	-	-
Allocation x_i	0	1	0	1	0	1	1	13
Bids β_i	5	6	7	8	9	10	-	-

2 tuple Quality of Service

Variation of 2 tuple QoS with increase in willingness to pay β_i

Allocation x_i	1	1	0	1	0	2	1.83	21
Bids β_i	6	7	8	9	10	11	-	-
Allocation x_i	1	1	0	1	0	2	1.83	21
Bids β_i	7	8	9	10	11	12	-	-
Allocation x_i	1	1	0	1	0	2	1.83	21
Bids β_i	8	9	10	11	12	13	-	-
Allocation x_i	1	1	0	1	0	2	1.83	21
Bids β_i	9	10	11	12	13	14	-	-
Allocation x_i	1	1	0	1	0	2	1.83	21
Bids β_i	10	11	12	13	14	15	-	-
Allocation x_i	1	1	0	1	0	2	1.83	21

Why QoS_2 is not normalised

Consider the following example

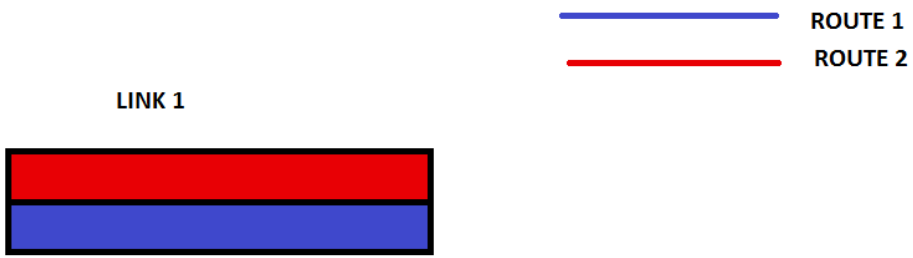
- Single link setting
- No. of Users : 2
- No. of Firms : 2
- User 1's bid : ($2,1) (willingness to pay, demand)
- User 2's bid : ($8,1)
- Firm 1's cost: $1
- Firm 2's cost: $6

Results

- Allocation will happen for both players in this case with a trading surplus of 3.
- If we consider using a normalised QoS_2 user 2 would have $\frac{p_2}{\beta_2} = 0.75$, but for user 1 $\frac{p_1}{\beta_1} = 3$.
- Thus it can be seen that it is not possible to normalise QoS_2 as the value is dependent on the prices charged by sellers.

Single O-D Pair Network

- Network contains 2 routes
- There are 2 firms offering resource on the link



Integer program to determine the allocations in single O-D pair , 2 route network

The Model

$$
\begin{aligned}
\text{maximize} \quad & \beta_1 x_1 + \beta_2 x_2 - c_1 y_1 - c_2 y_2 \\
\text{subject to} \quad & x_1 + x_2 \leq y_1 + y_2 \\
& x_1 \leq d_1 \\
& x_2 \leq d_2 \\
& y_1 \leq \mathrm{S}_1 \\
& y_2 \leq \mathrm{S}_2 \\
& x_i \geq 0 \quad \forall i
\end{aligned}
$$

The coefficient matrix is

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We can decompose A into the form

$$\begin{bmatrix} B \\ I \end{bmatrix}$$

where B is

$$\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}$$

Theorem

All single O-D pair networks having 2 firms and 2 routes will have a totally unimodular coefficient matrix for the trading surplus maximization problem.

Computational Results for Single O-D pair

- Integer program and its relaxation were run for 4000 random bid combinations and following results were obtained.

Percentage when firms are allocated the same	100.0
Percentage when routes are allocated the same	99.2
Percentage when objectives are equal	100.0

Some observations

- In spite of objective value being same, the allocation of routes can be different when the model has multiple optimal solutions.
- When both customers are willing to pay the same price, and their total demand is more than the capacity offered by any individual firm, the firms will allocate the same resource while the customers are getting different amount of resources.
- The situation where one customer's entire demand is met and the other customer does not get anything, and the situation where both customer share the resource would produce the same objective value.
- An example of such a situation is depicted next.

Instance Showing Variance in Route and Firm Allocations

Customer 1's bid(Price bid, demand bid)	(3.5,3)
Customer 2's bid(Price bid, demand bid)	(3.5,4)

Linear program solution

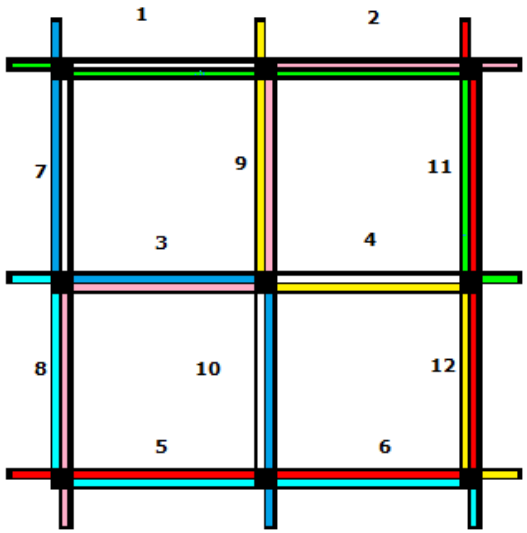
Customer 1's allocation	0
Customer 2's allocation	4

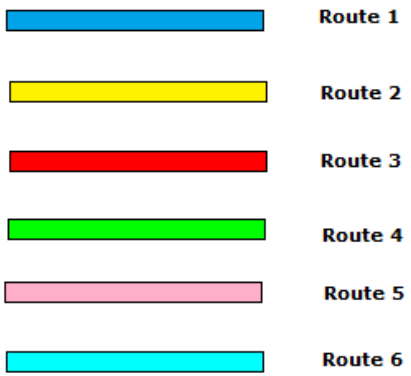
Integer program solution

Customer 1's allocation	3
Customer 2's allocation	1

Large Networks

- We extend the above ideas to a large network with 6 routes , 12 links and 3 firms.
- Every firm offers a capacity on each link and charges a price per unit capacity used.





The A matrix obtained from the model for a large network is a 54×42 matrix. It can be decomposed into the form

$$A = \left[\begin{array}{c|c|c|c} B & -I_{12} & -I_{12} & -I_{12} \\ \hline & & & \\ & & I_{42} & \end{array}\right]$$

Theorem

For any given network in which we apply the trading surplus maximization problem, the coefficient matrix can be decomposed into the form

$$A = \left[\begin{array}{c|c|c|c|c} B & -I_{1|L|} & -I_{2|L|} & \ldots & -I_{j|L|} \\ \hline & I_m & & & \end{array}\right]$$

where B is the route link incidence matrix, $|L|$ is the number of links in the network, j is the number of firms and
$m =$ number of routes $+ |L| \times j$.

Total unimodularity

- Discrete(integer) valed optimizations, as above, are computationally hard.
- Linear Programs (LP) are **easy**(polynomial time).
- When are above NMD's are easy? – *Unimodular matrices might help.*
- A submatrix of matrix A is any square matrix that evolves from A by deleting some columns and rows from A.
- A matrix A is called totally unimodular (TU), if and only if the determinants of all submatrices of A are either $-1, 0, 1$.
- Any totally unimodular matrix has only $0, +1$or 1 entries. The opposite is not true, i.e., a matrix with only $0, +1$ or 1 entries is not necessarily unimodular.

We know the following properties of totally unimodular matrices. If A is a totally unimodular matrix, then

- A^T is totally unimodular
- -A is totally unimodular
- $[A \quad I]$ is totally unimodular

Theorem

The total unimodularity of the coefficient matrix A is equivalent to the total unimodularity of the route-link incidence matrix in the trading surplus maximization problem. Hence, the total unimodularity of the route-link will ensure integer results to the linear programming relaxation of the trading surplus maximization problem.

Theorem

All single O-D pair networks will have a totally unimodular coefficient matrix A and thus the linear programming relaxation of all single O-D pair networks will have integer solutions.

Computational Results

The integer program and its linear programming relaxation were run for 4000 randomized setting. The following results were obtained.

Percentage when firms are allocated the same	94.9
Percentage when routes are allocated the same	94.9
Percentage when objectives are equal	100.0

Some observations on large network results

- Unlike single O-D pair setting, the allocation to firms and routes is different using integer optimization and linear relaxation.
- In large networks, with multiple customers and no routes having the same set of links, sharing of resource by bidding the same amount is not possible. Instead, a change in the resource allocation of one customer will affect the allocation of some or every customer in the route - *The Network Effect*.
- In a scenario where the allocations of a customer changes in a large network, the resource offered by the firms which is used will also change.
- An example of such a situation is depicted next.

Variation of Integer Program and LP relaxation for Large Network

Customer 1's bid(Price bid, demand bid)	(3.5,4)
Customer 2's bid(Price bid, demand bid)	(3.5,4)
Customer 3's bid(Price bid, demand bid)	(0.5,1)
Customer 4's bid(Price bid, demand bid)	(3.5,4)
Customer 5's bid(Price bid, demand bid)	(5.5,5)
Customer 6's bid(Price bid, demand bid)	(2.5,3)

Linear program solution

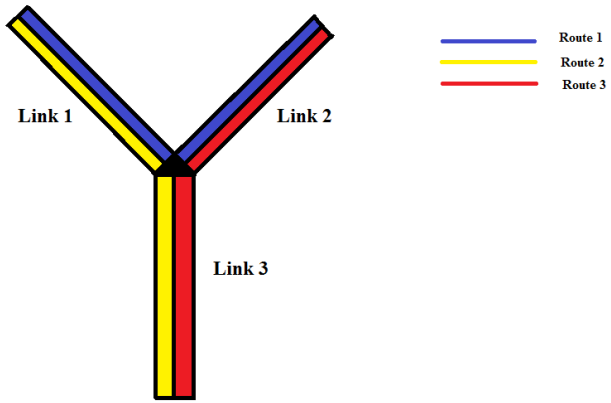
Customer 1's allocation	1
Customer 2's allocation	1
Customer 3's allocation	0
Customer 4's allocation	1
Customer 5's allocation	1
Customer 6's allocation	0

Integer program solution

Customer 1's allocation	2
Customer 2's allocation	2
Customer 3's allocation	0
Customer 4's allocation	2
Customer 5's allocation	0
Customer 6's allocation	0

Networks Where Linear Programming Relaxations may not be Possible

- B matrix defined above is the route link matrix used to describe the network where each column is a route and each row is a link.
- Thus certain networks with high network affects will not lead to unimodular matrices.
- Consider the following network with 3 links and 3 users, and each link is used exactly by 2 customers.
- Such networks are not totally unimodular and when costs and price bids are not integral can lead to fractional outputs.



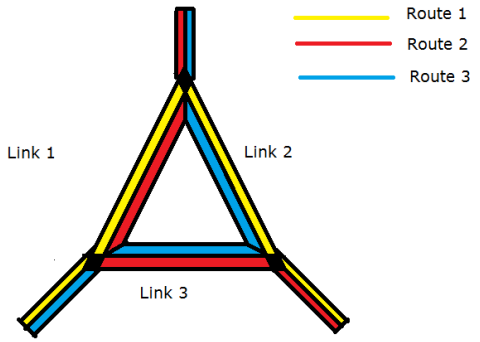
This is because the link route matrix of the network will be

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

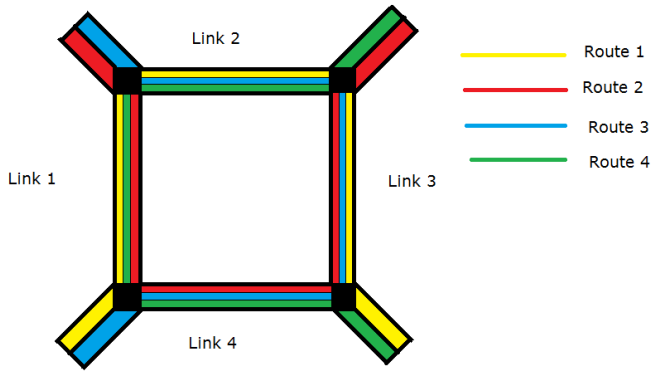
The determinant of such a matrix is -2 and hence the matrix is not totally unimodular..

Other network configurations which do not have coefficient matrices that are totally unimodular are

Cyclic triangular network



Cyclic square network



Some remarks

- Even though experiments were conducted with non totally unimodular matrices, the presence of a non integer optimal solution has not been seen.
- This could be because of the small sized networks being tested or the random cases had been ones without fractional optimal solution.
- Another property of these matrices is that , unlike usual node arc incidence matrices with $+1$ or -1 as elements, the elements here are 0 or 1. This might be a reason for total integer results even though the matrix is not totally unimodular.

Equilibrium in Systems Controlled by 2 Interacting Parameters

- In order to generate a random variable X_2, which represents the price bids, the value of X_1, which represents the demand, is generated first. A random variable ϵ is generated.
- $X_2 = X_1 + \epsilon$. This generates a price random variable that is correlated to the demand that is being bid.
- Having uncorrelated price bids and demand would lead to bids in which demand is high while price is low and vice versa.
- The value ϵ is generated such that
  - If $\lambda_1 = 0$, $\epsilon = 0$.
  - Else if $X_1 \neq 0$

$$\epsilon = \begin{cases} 0.5, & \text{w.p.} \quad p \\ -0.5, & \text{w.p.} \quad 1-p \end{cases}$$

- The demand is generated by generating 0 demand with probability α and generating a uniform distribution [1,2,3,4,5] with probability $1-\alpha$

$$\lambda_1 = E[X_1] = \alpha\times 0+(1-\alpha)\times(\frac{1}{5}\times 1+\frac{1}{5}\times 2+\frac{1}{5}\times 3+\frac{1}{5}\times 4++\frac{1}{5}\times 5)$$

$$\lambda_1 = E[X_1] = (1-\alpha)\times 3$$

$$\alpha = 1-\frac{\lambda_1}{3}$$

- For a given value of λ_1 we are able to determine an α, which we use to generate the random variable X_2 with mean λ_1.

Generating X_2 about the mean λ_2

- According to the definition $X_2 = X_1 + \epsilon$, we get

$$\begin{aligned} \lambda_2 &= E[X_2] = E[X_1 + \epsilon] \\ \lambda_2 &= (1-\alpha) \times (3 + 0.5 \times p + -0.5 \times (1-p)) \\ \lambda_2 &= (1-\alpha) \times (3 + 0.5 \times (2p-1) \end{aligned}$$

p as a function of λ_2

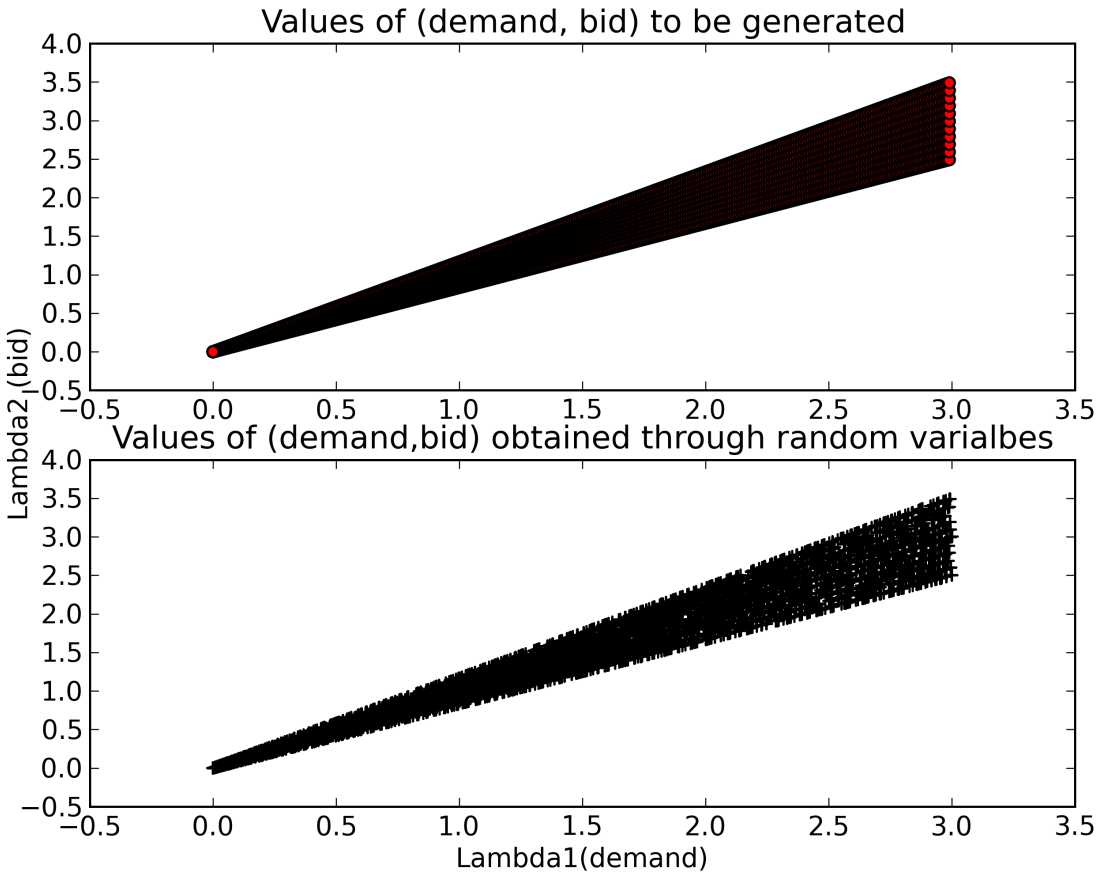
$$p = \frac{\lambda_2}{1-\alpha} - 2.5$$

- The values of p being probability has to lie between 0 and 1.
- Thus the values of λ_2 we can generate for a given λ_1 are

$$\text{Upper bound} = (1-\alpha) \times 3.5$$

$$\text{Lower bound} = (1-\alpha) \times 2.5$$

Comparison of values to be generated and the average of values that were generated



The Algorithm

1: — Beginning of pre-solving —
2: Generate all possible bids for a customer
3: **for all** Bids of customer 1 **do**
4: **for all** Bids of customer 2 **do**
5: Generate Bid Pair Combination
6: **end for**
7: **end for**
8: **for all** Bid Pair Combinations **do**
9: Determine Allocations
10: Determine QoS of System
11: **end for**
12: —-End of Pre-solving—-

—-Monte Carlo Simulations——
for all $\lambda_1 \geq 0$ and $\lambda_1 \leq$ highest mean **do**
    Determine upper and lower bound for given λ_1
    for all $\lambda_2 \geq$ lower bound and $\lambda_2 \leq$ upper bound **do**
        for total repetitions **do**
            for all Routes **do**
                Generate random bids about λ_1
            end for
            Determine QoS_1 and QoS_2 based on bids
        end for
        Determine average QoS_1 and QoS_2 for (λ_1, λ_2)
    end for
end for

Continuity of QoS With Respect To Mean

- In order to determine an equilibrium point, we must first determine the continuity of QoS_1 with the means λ_1 and λ_2.

$$QoS = E[\text{realisedQoS}] \tag{5}$$

$$QoS = \begin{bmatrix} E[\text{realisedQoS}_1] \\ E[\text{realisedQoS}_2] \end{bmatrix} \tag{6}$$

- Probability Law used to generate demand

$$\sim \begin{cases} 0 & \text{w.p.} \quad \alpha \\ U[a,b] & \text{w.p.} \quad 1-\alpha \end{cases} \tag{7}$$

- Bernoulli Random variable to generate price

$$\sim \begin{cases} 0.5 & \text{w.p.} \quad p \\ -0.5 & \text{w.p.} \quad 1-p \end{cases} \tag{8}$$

Quality of Service Tuple

- As we are using a bid tuple there are two Quality of Services which are being measured.
  - The first measure is QoS_1 which captures how much a persons bid is being met.

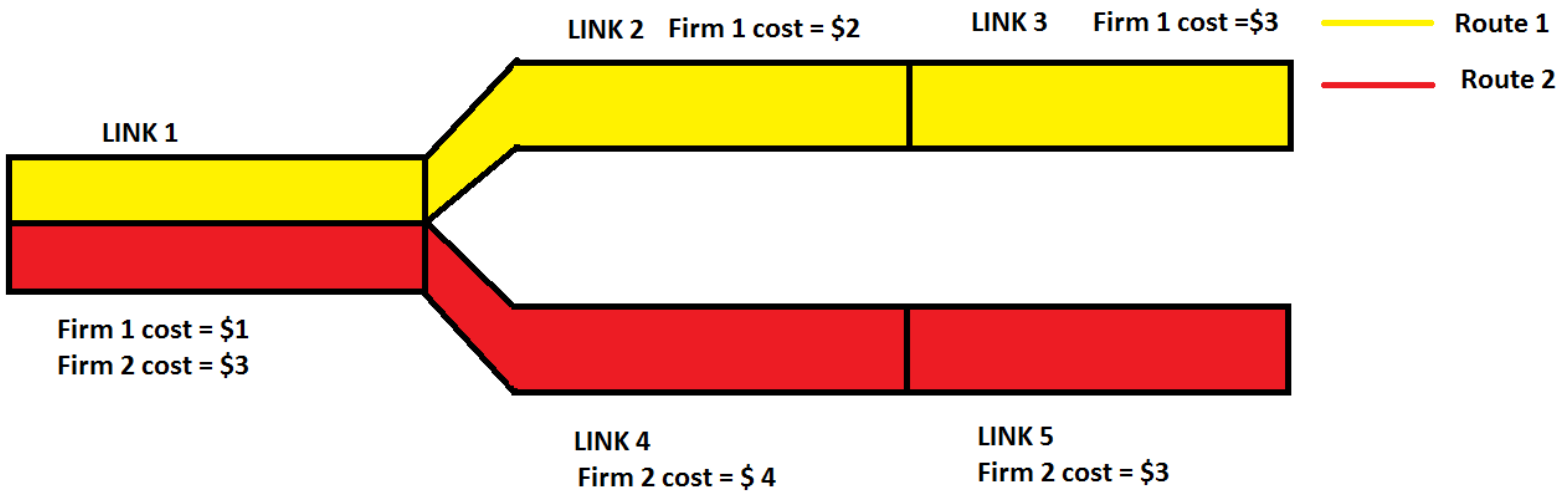
$$QoS_1 = \sum_i 1 - \frac{x_i}{d_i} \tag{9}$$

  - The second measure is QoS_2 which captures the price submitted by the customer.

$$QoS_2 = \sum_i 1 - \frac{p_i}{\text{Maximum Chargeable price}_i} \tag{10}$$

- Unlike QoS_1, we normalise QoS_2 by Maximum Chargeable price instead of the customers bid β_i. This is because there are situations where a customer can be charged more than what he has bid.

Network where players are charged over their bid



- Customer 1's bid is $(8, 2)$ and Customer 2's bid is $(12, 2)$.
- Both customers get an allocation of 2 units each.
- Thus, Customer 1 will now be charged 9 and Customer 2 will be charged 13. Clearly, customer 1 is being charged over his current bid of 8. Thus, we cannot use β_i to normalise it.

Continuity with respect to parameters I

- The output allocation to customers and the capacity allocated by each firm can be seen as a function of the bids of all the customers.

$$(x_1(\omega), x_2(\omega), y_1(\omega), y_2(\omega)) = f(\beta_1(\omega), \beta_2(\omega), d_1(\omega), d_2(\omega))$$

- Allocations are used to determine the Quality of Service.

Continuity with respect to parameters II

- Functions $g_1(\cdot)$ and $g_2(\cdot)$ which takes the customer and firm allocations and maps it to a Quality of Service tuple are defined below.

$$\begin{aligned} QoS_1(\omega) &= g_1(x_1(\omega), x_2(\omega), y_1(\omega), y_2(\omega)) \\ QoS_2(\omega) &= g_2(x_1(\omega), x_2(\omega), y_1(\omega), y_2(\omega)) \\ QoS_1(\omega) &= g_1(f(\beta_1(\omega), \beta_2(\omega), d_1(\omega), d_2(\omega))) \\ QoS_2(\omega) &= g_2(f(\beta_1(\omega), \beta_2(\omega), d_1(\omega), d_2(\omega))) \end{aligned}$$

- QoS can be determined by finding the expected values of these functions

$$\begin{bmatrix} QoS_1 \\ QoS_2 \end{bmatrix} = \begin{bmatrix} E[g_1(f(\beta_1(\omega), \beta_2(\omega), d_1(\omega), d_2(\omega)))] \\ E[g_2(f(\beta_1(\omega), \beta_2(\omega), d_1(\omega), d_2(\omega)))] \end{bmatrix} \quad (11)$$

In a two customer model, we can categorise the bid pairs obtained into 6 different categories.

Categories of bid pairs and their probabilities

No.	Category	Probability
1	Both customers have [0,0]	α^2
2	One customer has bid [0,0] other customer has price above demand	$\alpha(1-\alpha)\frac{p}{n}$
3	One customer has bid [0,0] other customer has price below demand	$\alpha(1-\alpha)\frac{1-p}{n}$
4	Both customers have non zero bid, both customers have price below	$(1-\alpha)^2(\frac{1-p}{n})^2$
5	Both customers have non zero bid, both customers have price above	$(1-\alpha)^2(\frac{p}{n})^2$
6	Both customers have non zero bid,but opposite prices	$(1-\alpha)^2(\frac{(1-p)p}{n^2})$

Where n is the number of discrete elements in the uniform distribution. We can determine the allocation for all possible bid pair combinations that players can have and segregate them into the 7 different categories.

Determining Coefficients

- **Example** : Let us consider Customer 1's bid to be $(2, 2.5)$ and Customer 2's bid to be $(1, 1.5)$. We can see that this bid combination falls into category 5 where both players have a price higher than their demand.Solve the trading surplus problem to get
  - Customer 1's allocation is 2 units.
  - Customer 2's allocation is 0 units.
  - Firm 1 is offering 2 units of resource.
  - Firm 2 is offering 0 units of resource.
- The QoS_1 for this allocation is $1 - \frac{2}{2} + 1 - \frac{0}{3}$ which is equal to 2. Similarly, we see that the firm which is allocating a resource is firm 1. Thus the price of the link is \$1. The maximum payable price is \$2.
- QoS_2 can be computed as $1 - \frac{1}{2} + 1 - \frac{0}{2}$ which is 1.5.
- We do this for all possible bid pair combinations, segregate them into categories and determine their summations.

$$QoS_1 = \alpha^2 \times 2 + \alpha(1-\alpha)\frac{p}{3} \times 6 + \alpha(1-\alpha)\frac{1-p}{3} \times 8$$
$$+(1-\alpha)^2(\frac{1-p}{3})^2 \times 9\frac{2}{3} + (1-\alpha)^2(\frac{p}{3})^2 \times 5\frac{2}{3} \quad (12)$$
$$+(1-\alpha)^2(\frac{(1-p)p}{3^2}) \times 15\frac{2}{3}$$

$$QoS_2 = \alpha^2 \times 2 + \alpha(1-\alpha)\frac{p}{3} \times 8 + \alpha(1-\alpha)\frac{1-p}{3} \times 9$$
$$+(1-\alpha)^2(\frac{1-p}{3})^2 \times 10.5 + (1-\alpha)^2(\frac{p}{3})^2 \times 6 \quad (13)$$
$$+(1-\alpha)^2(\frac{(1-p)p}{3^2}) \times 17.5$$

- Equations for curve generated by smaller supports

$$\begin{bmatrix} QoS_1 \\ QoS_2 \end{bmatrix} = \begin{bmatrix} 2 - 0.0162\lambda_1^2 - 0.1667\lambda_1 + 0.185\lambda_1\lambda_2 - 0.6667\lambda_2 - 0.037\lambda_2^2 \\ -0.111\lambda_2^2 + 0.1389\lambda_1\lambda_2 - 0.25\lambda_1 - 0.33\lambda_2 + 2 \end{bmatrix} \tag{14}$$

- Similarly, for larger supports we get

$$\begin{bmatrix} QoS_1 \\ QoS_2 \end{bmatrix} = \begin{bmatrix} \frac{-7\lambda_1^2+6\lambda_1(34\lambda_2-65)-12\left(\lambda_2^2+75\lambda_2-375\right)}{2250} \\ \frac{1}{150}\left(\lambda_1^2 + \lambda_1(13\lambda_2 - 45) - 6\left(\lambda_2^2 + 5\lambda_2 - 50\right)\right) \end{bmatrix} \tag{15}$$

Determining Theoretical Fixed Point I

- Equation to solve to determine the fixed point

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} QoS_1(\lambda_1, \lambda_2) \\ QoS_2(\lambda_1, \lambda_2) \end{bmatrix} \tag{16}$$

- On solving the previous equation for the smaller and larger supports, we get

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1.14922 \\ 1.29937 \end{bmatrix}$$

- The solution obtained i.e. $(1.14922, 1.29937)$ maps to $(1.13481, 1.29937)$ which is slightly different from itself.

Determining Theoretical Fixed Point II

- We do a search in the vicinity of the point to obtain

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1.134 \\ 1.13 \end{bmatrix}$$

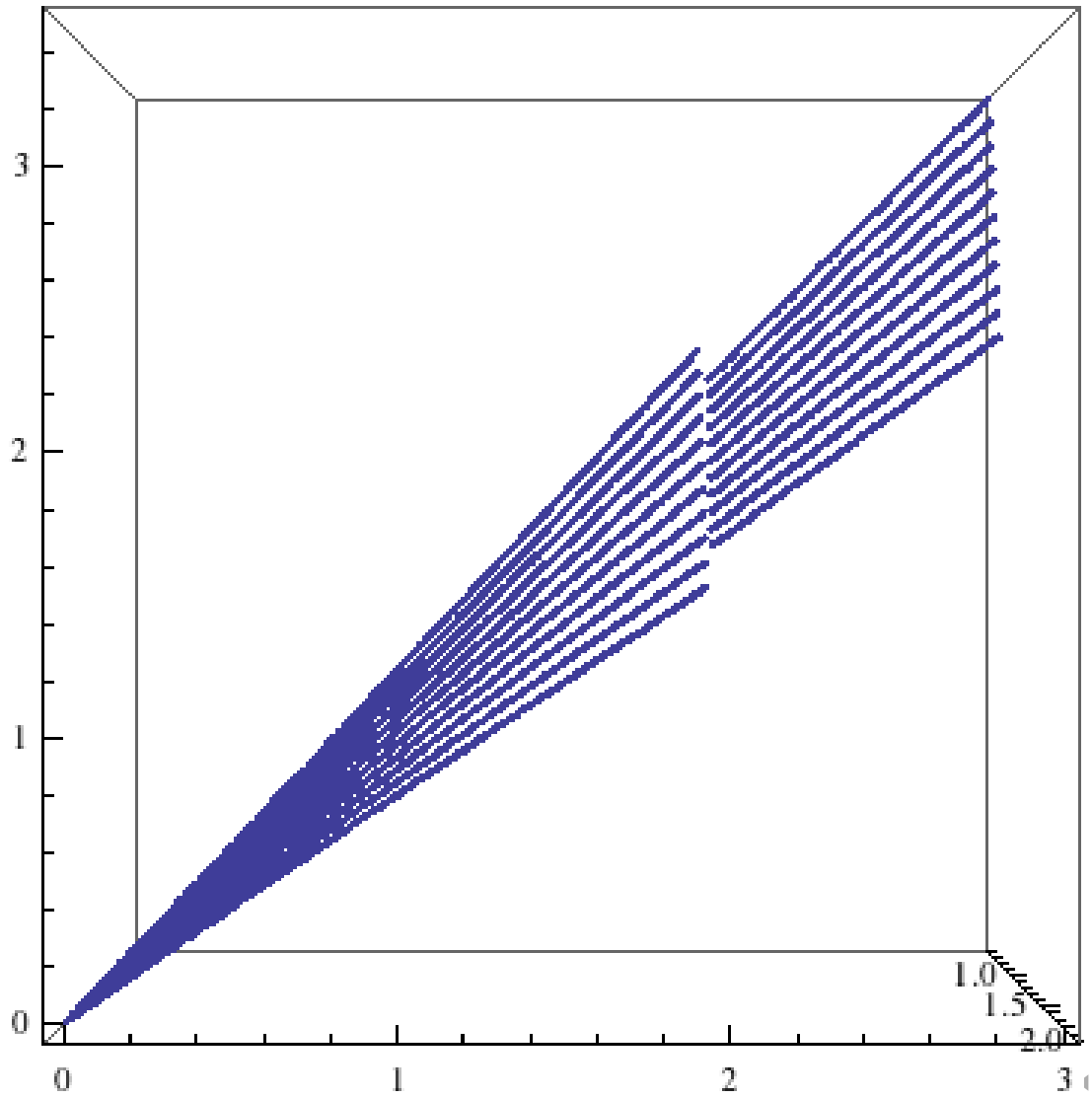
- The point maps back to itself

$$\begin{bmatrix} 1.134 \\ 1.13 \end{bmatrix} \sim \begin{bmatrix} 1.13390349074 \\ 1.30013888889 \end{bmatrix}$$

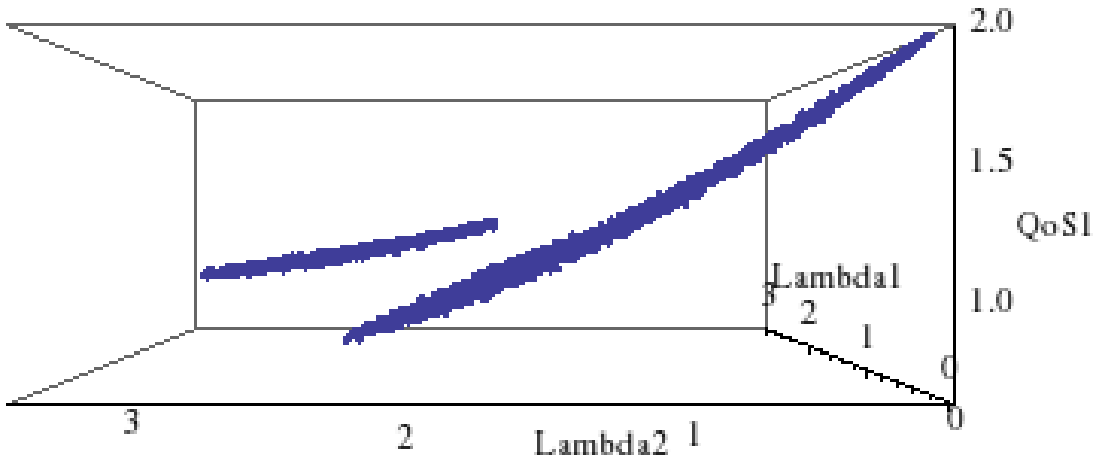
Computational Determination of Fixed Point

From the graphs plotted, via visual inspection it is seen that the QoS_1 curve intersects the plane between $\lambda_1 \in (1.0, 1.4)$ and $\lambda_2 \in (1.2, 1.6)$ and the QoS_2 curve intersects the plane between $\lambda_2 \in (1.2, 1.4)$ and $\lambda_1 \in (1.0, 1.6)$. We now compute the values of QoS_1 and QoS_2 within these regions.

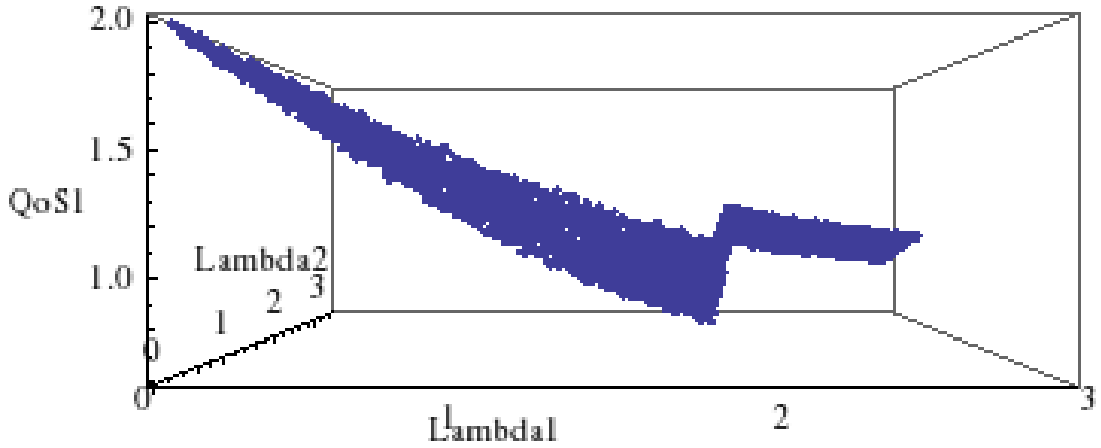
Top View QoS_1



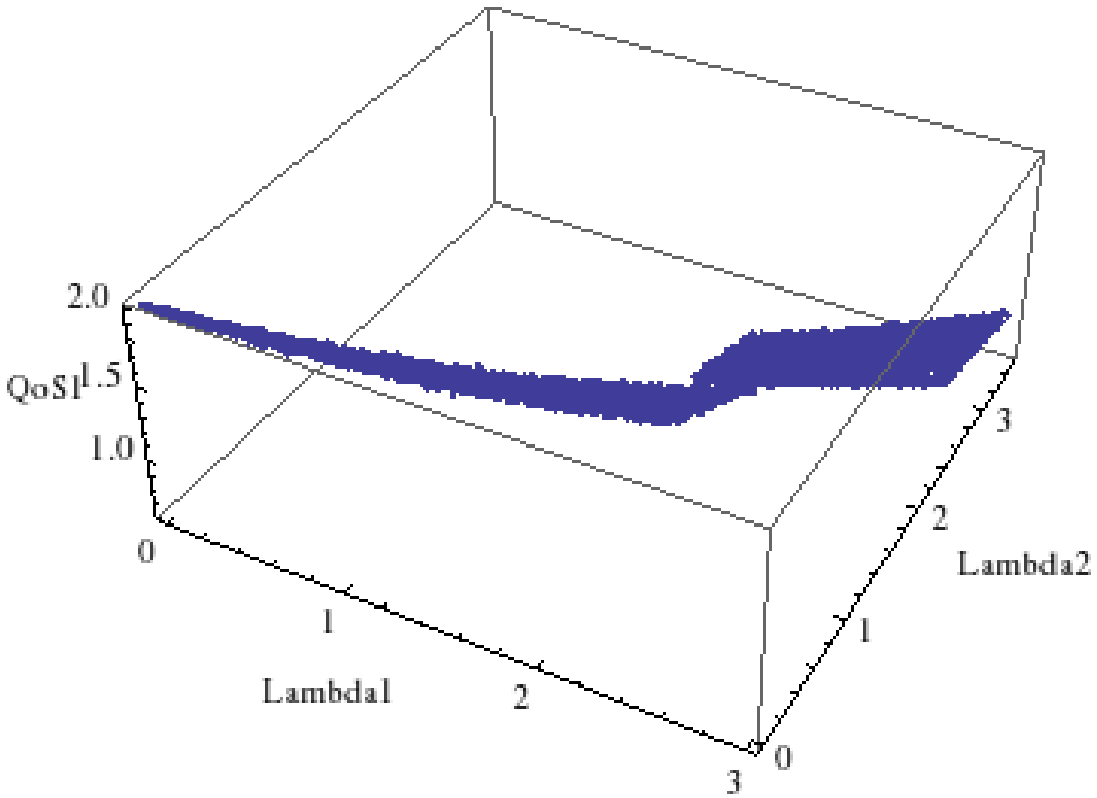
Side View QoS_1



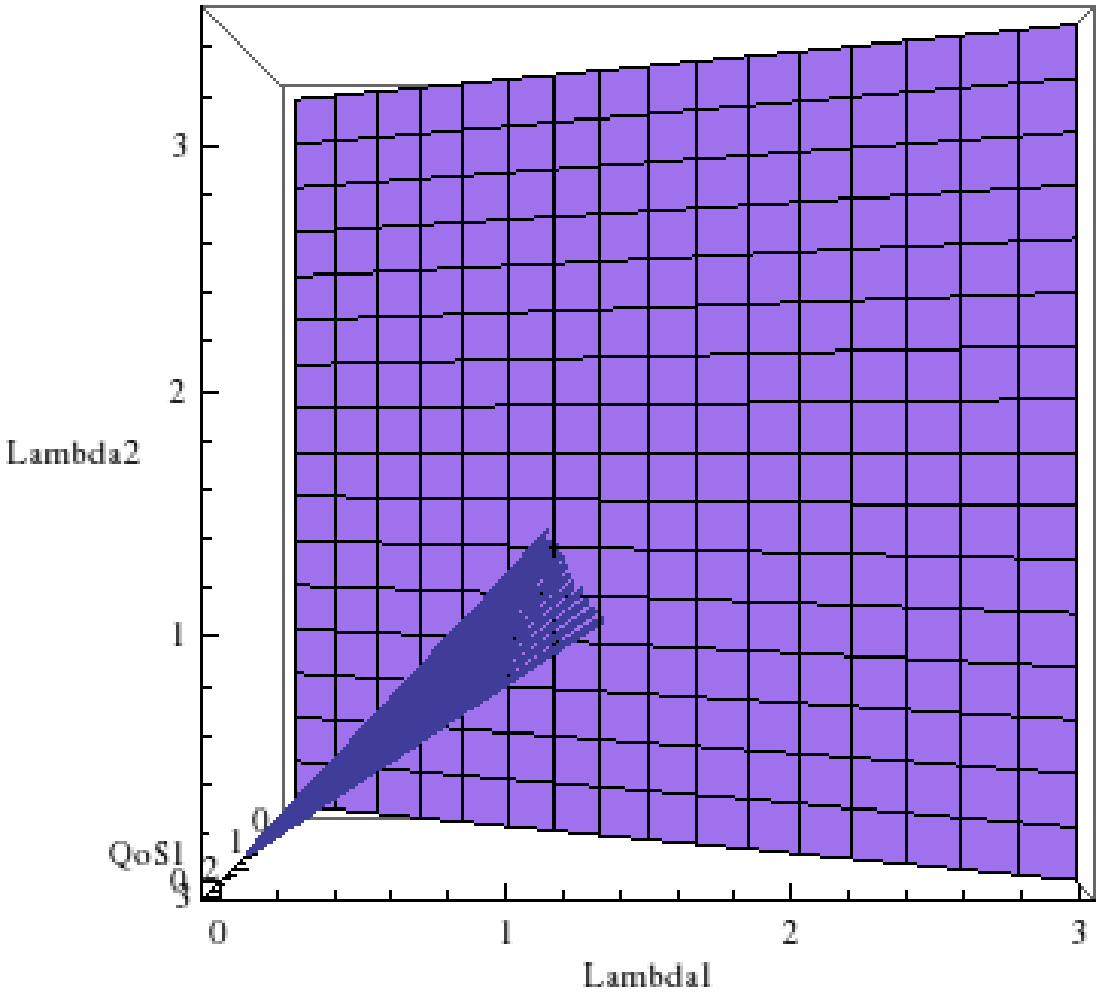
Front View QoS_1



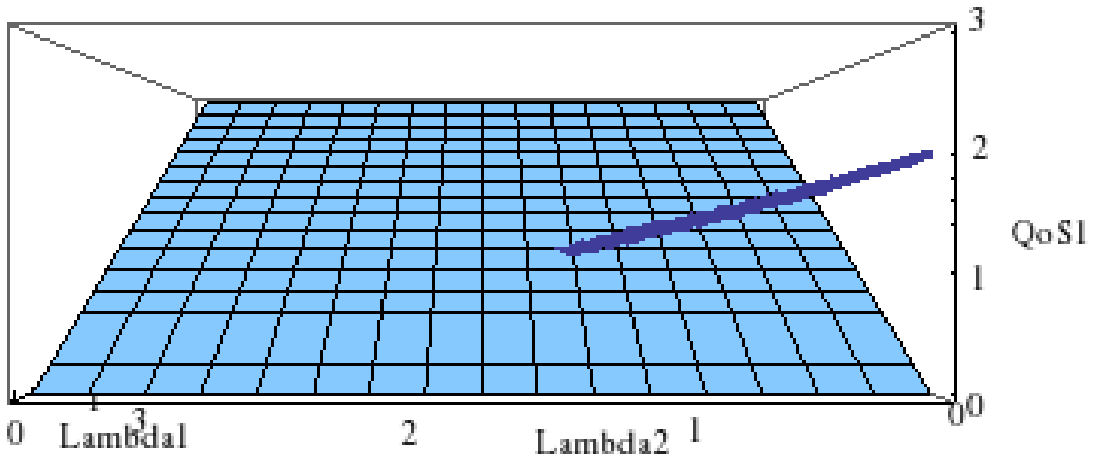
Angular View QoS_1



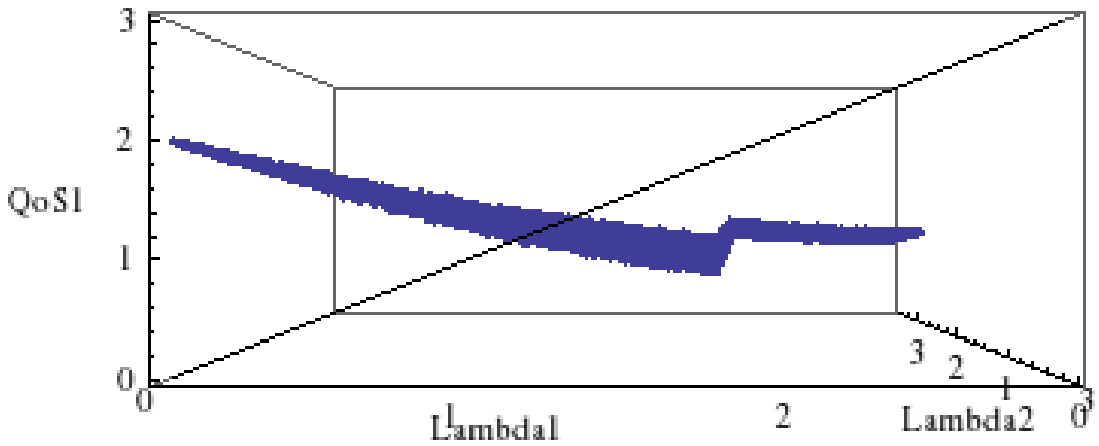
Top View QoS_1 Intersection with plane



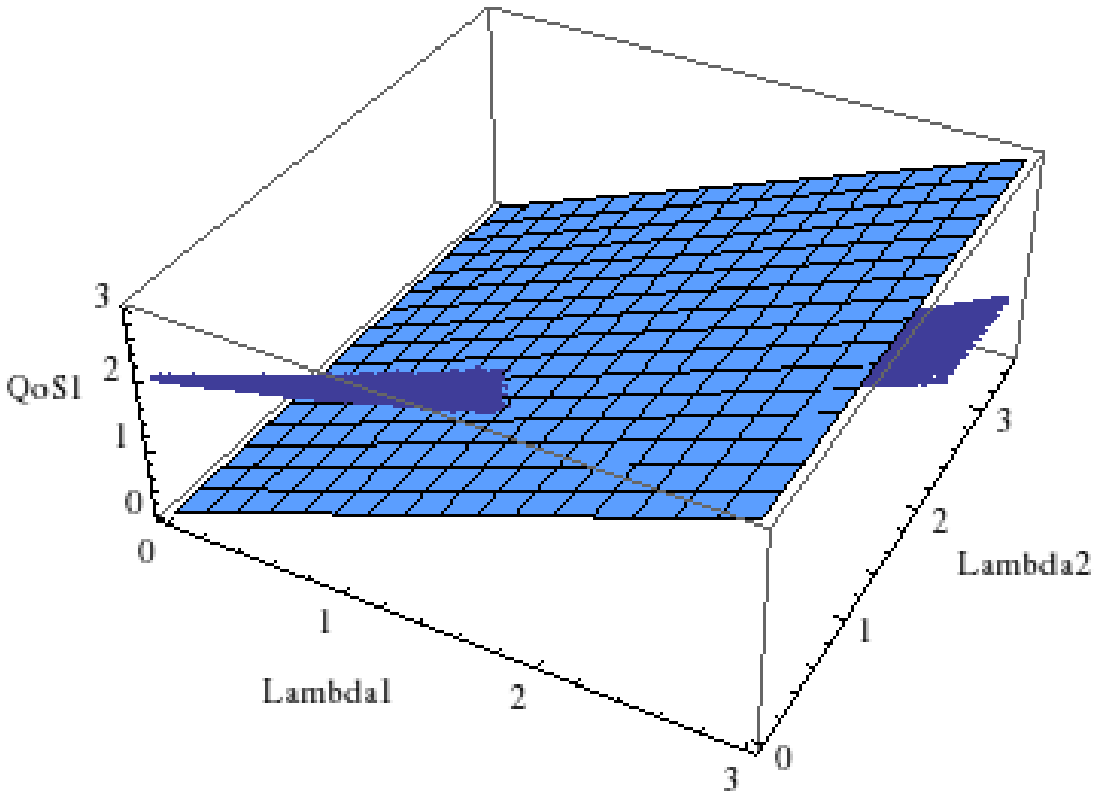
Side View QoS_1 Intersection with plane



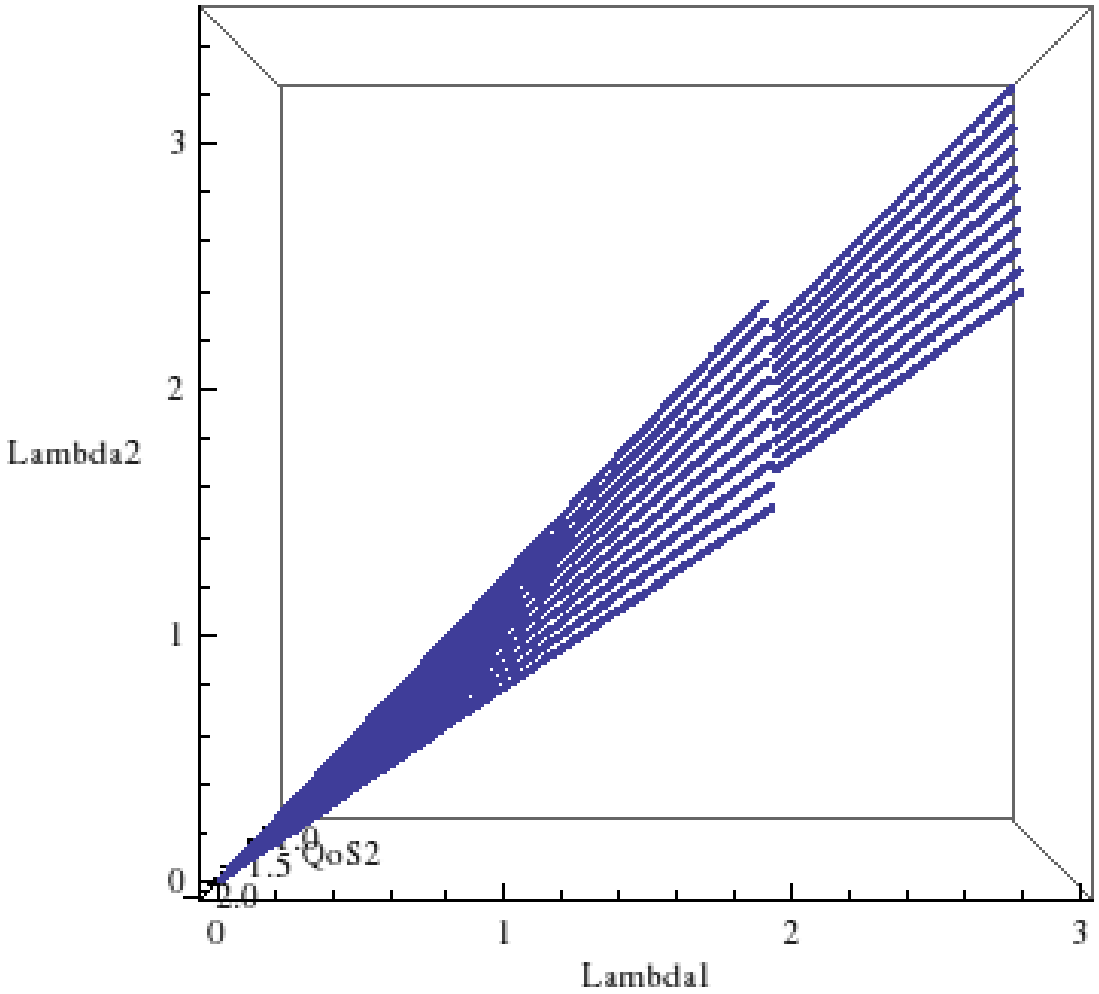
Front View QoS_1 Intersection with plane



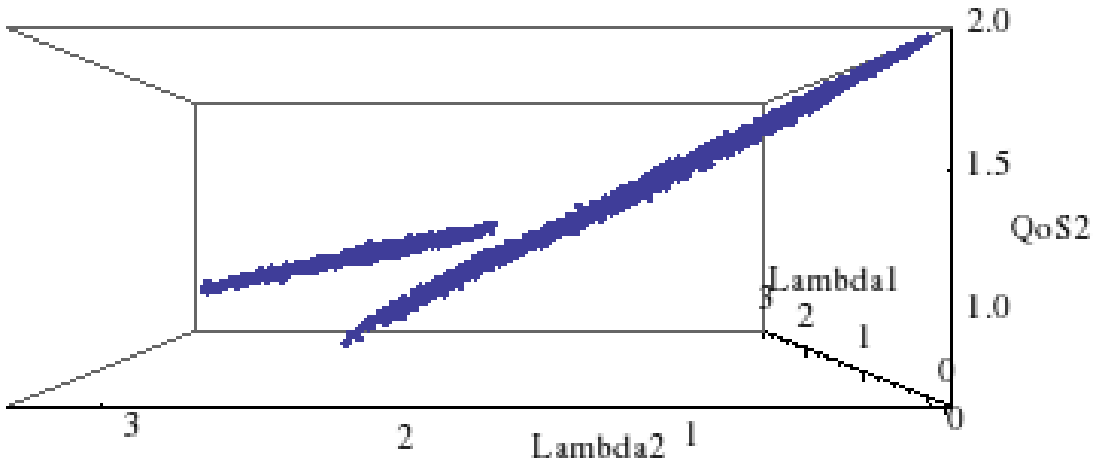
Angular View QoS_1 Intersection with plane



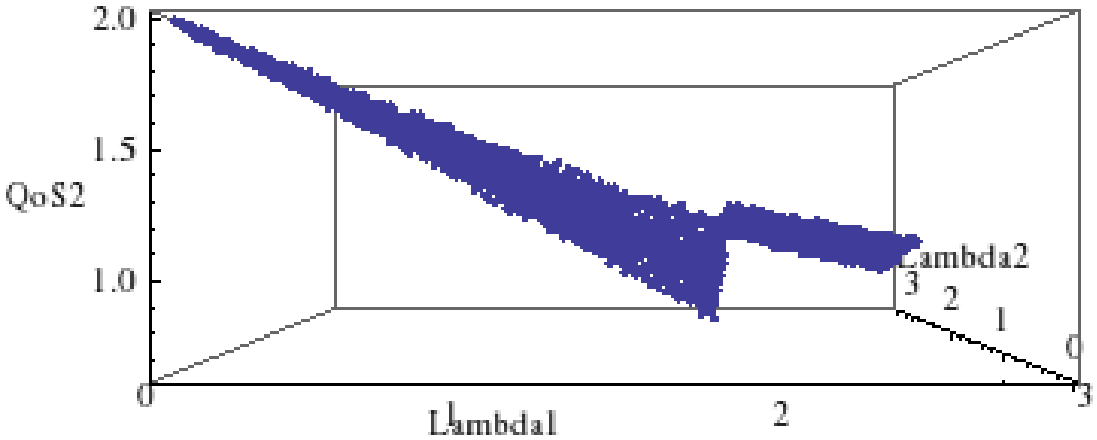
Top View QoS_2



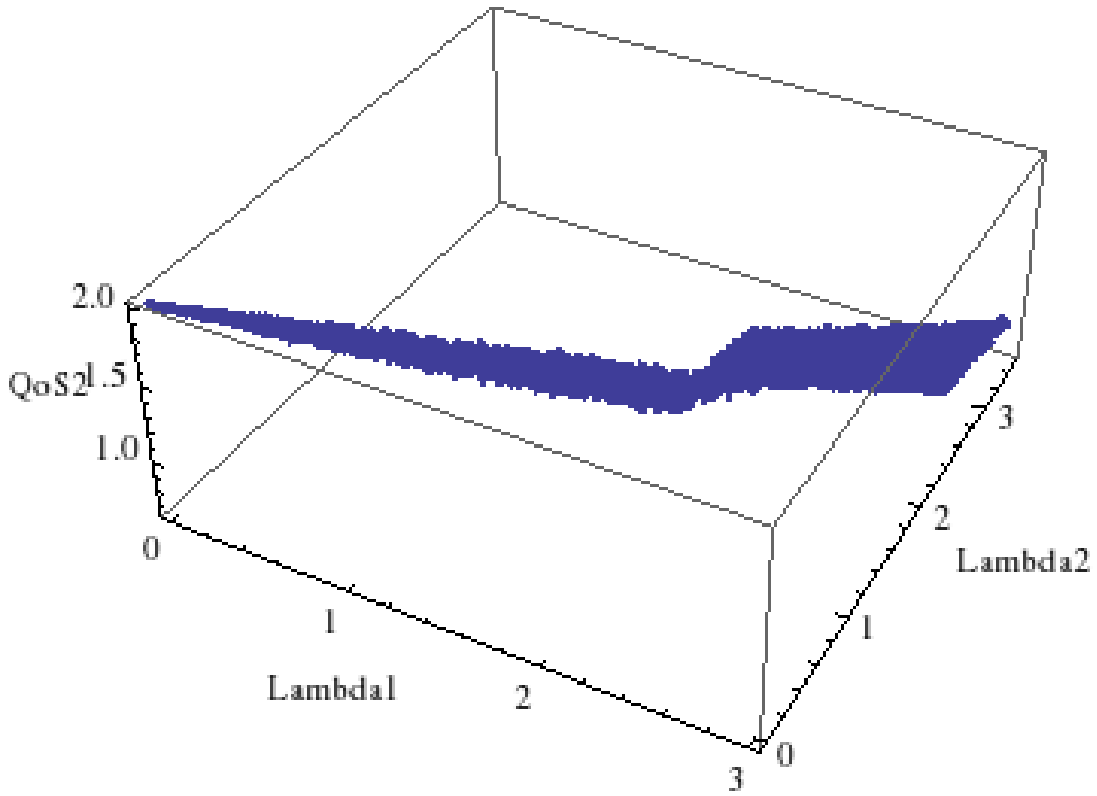
Side View QoS_2



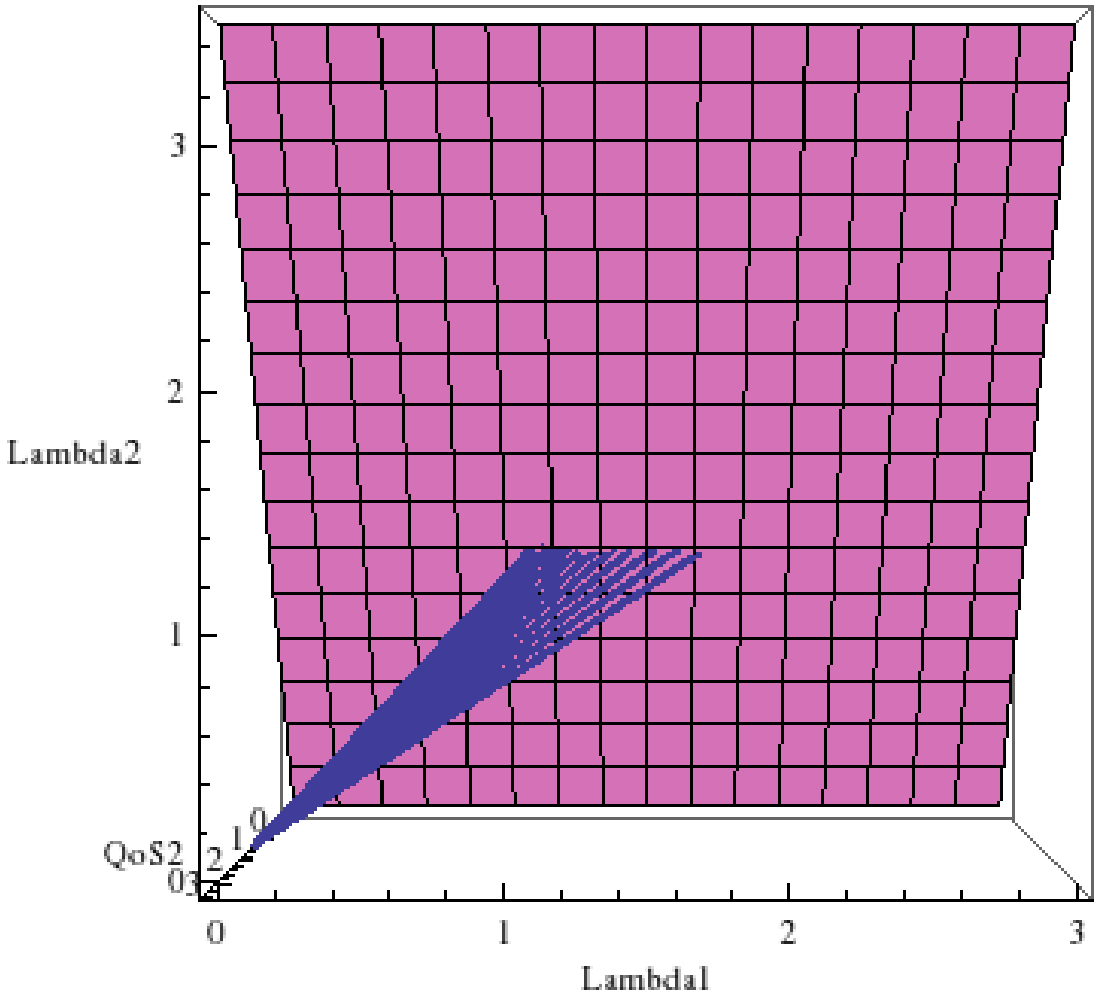
Front View QoS_2



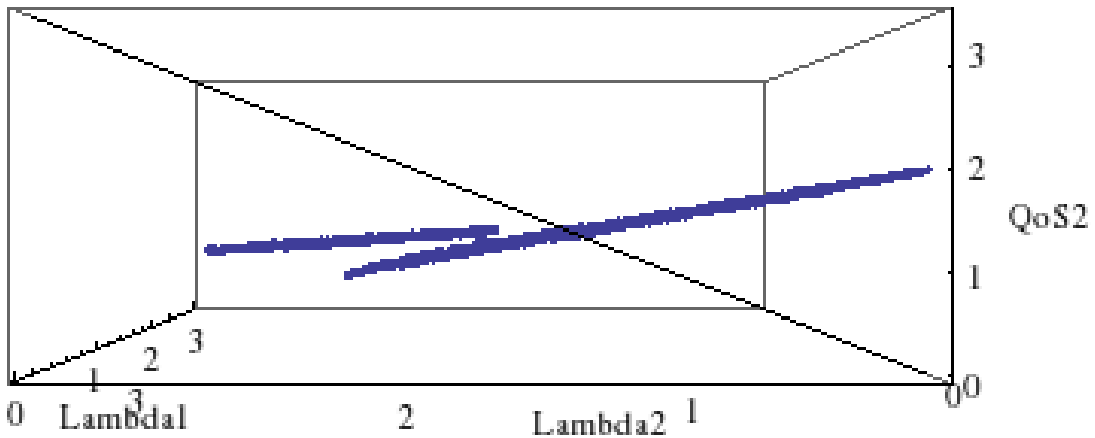
Angular View QoS_2



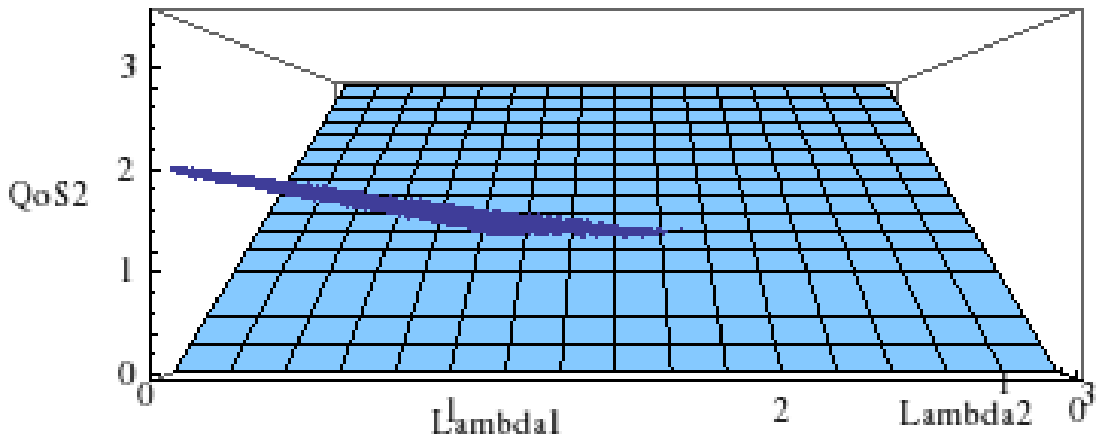
Top View QoS_2 Intersection with plane



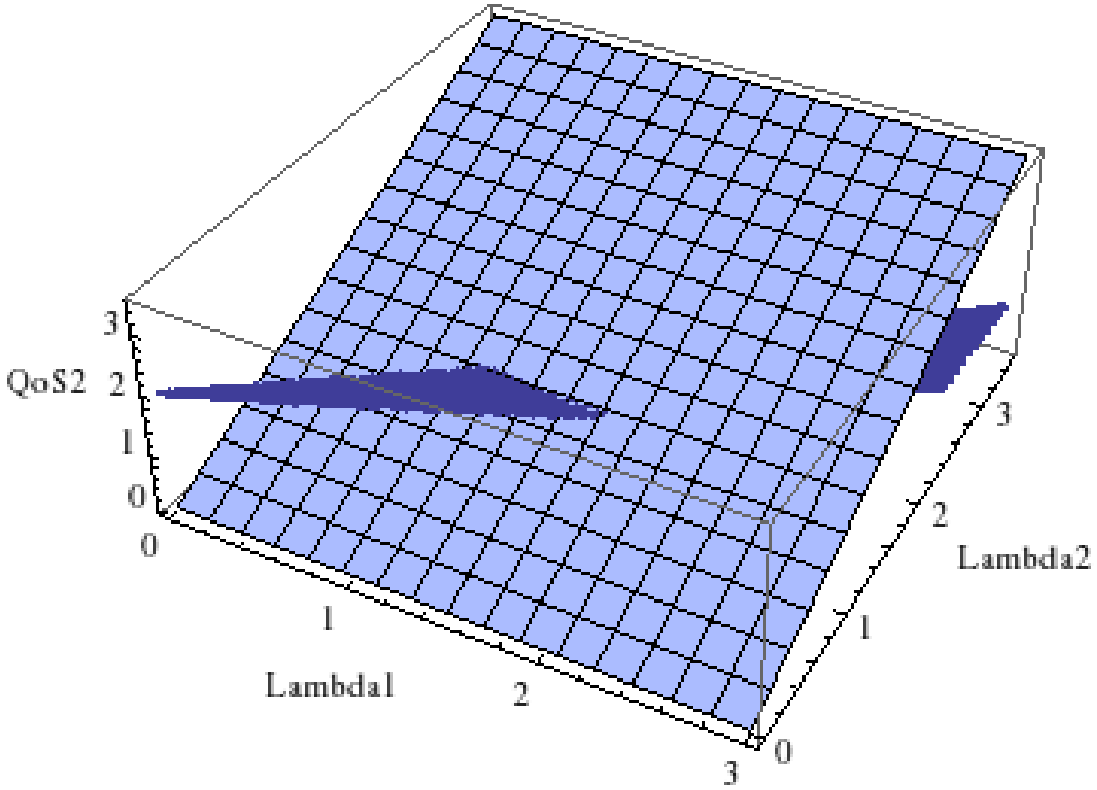
Side View QoS_2 Intersection with plane



Front View QoS_2 Intersection with plane



Angular View QoS_2 Intersection with plane



Determining Λ_1 and Λ_2

- Λ_1 is the set the set of points where $\lambda_1 = QoS_1(\lambda_1, *)$.
- Similarly, Λ_2 is the set the set of points where $\lambda_2 = QoS_2(*, \lambda_2)$.
- The intersection of these two sets will give us the fixed points.
- Computationally, with a tolerance of < 0.01 we get the following set of points as the intersection of Λ_1 and Λ_2.

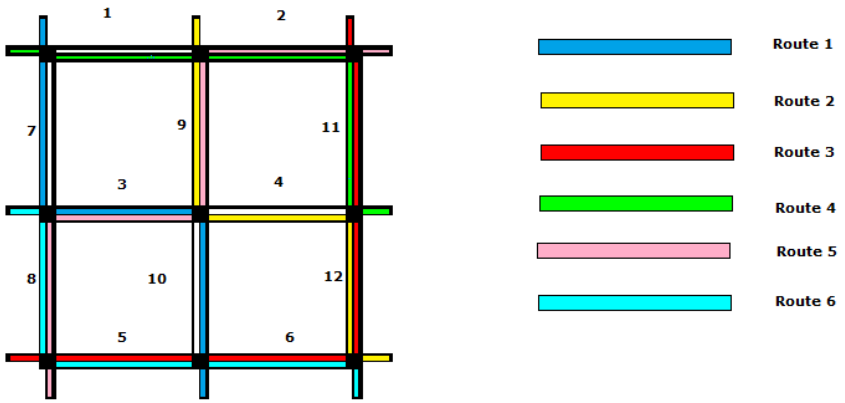
Common Points

λ_1	λ_2
1.13	1.2995
1.14	1.311
1.15	1.29375

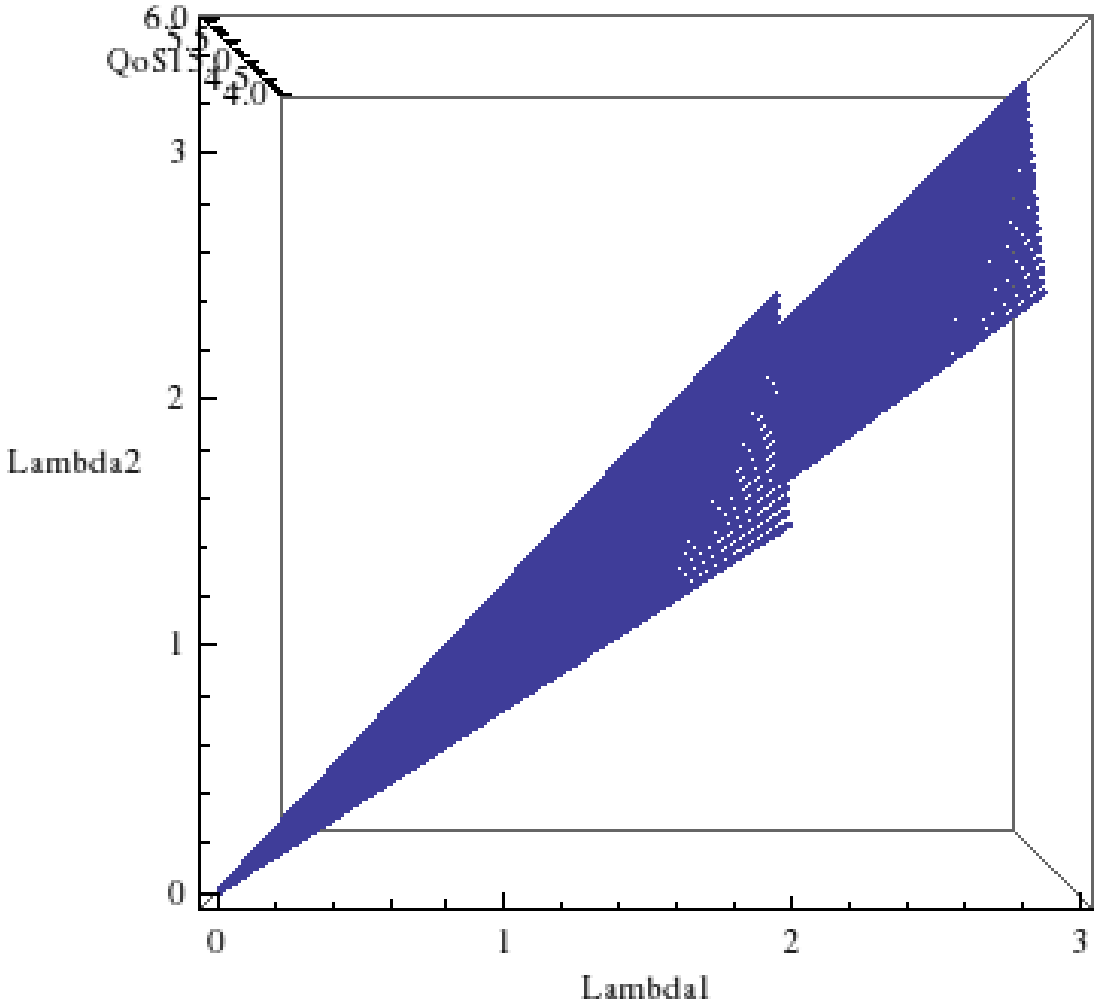
- We then decrease the threshold to < 0.005.The only common point obtained as the intersection of Λ_1 and Λ_2 is $(1.131, 1.30065)$ which is same as the theoretical fixed point.

Large Networks

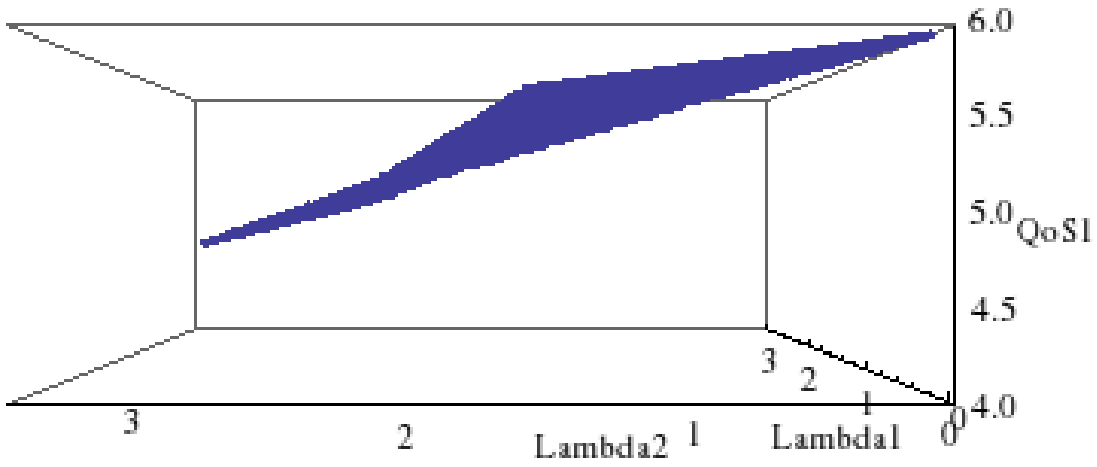
- We consider a network of multiple users having different O-D pairs, and multiple links.
- In these networks, the change in allocation of a given customer will affect the amount which the firms have to offer.
- The network consists of 6 routes, 12 links and having 3 firms on each link.



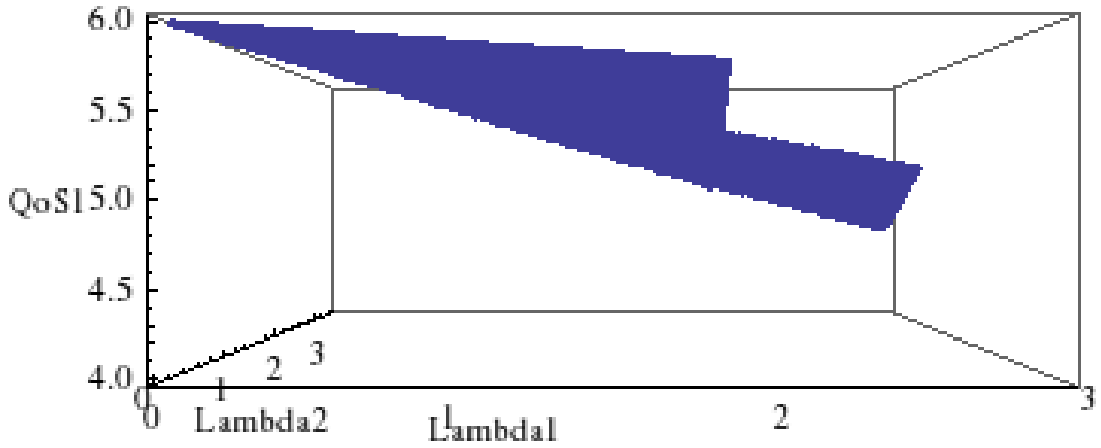
Top View QoS_1 Curve for a Large Network



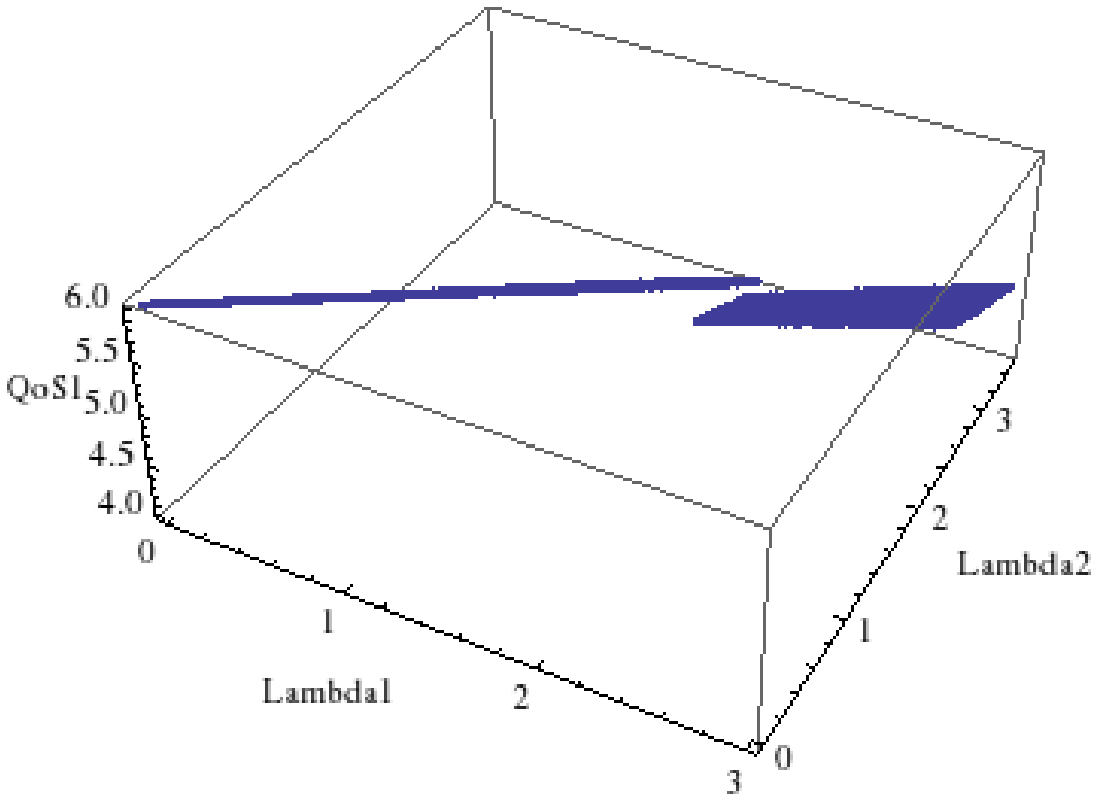
Side View QoS_1 Curve for a Large Network



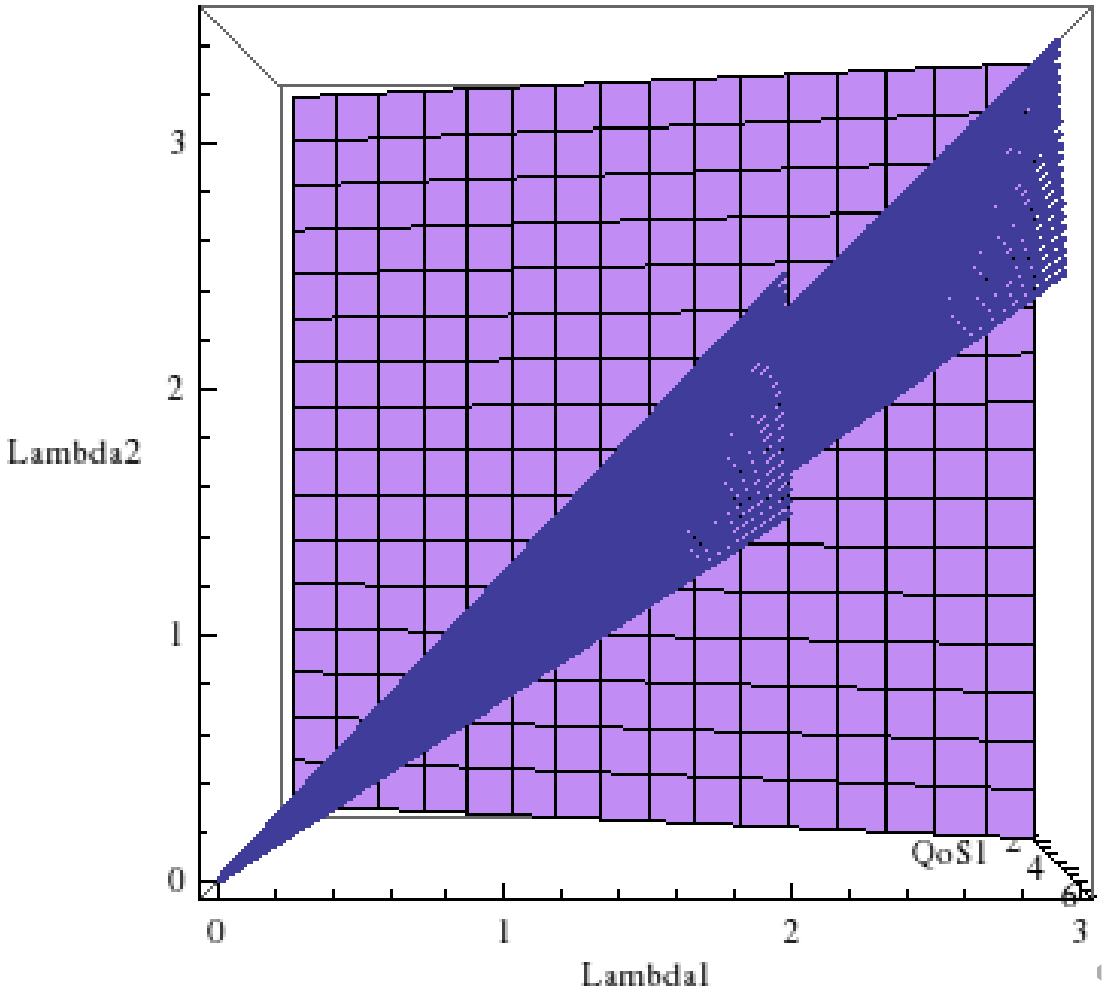
Front View QoS_1 Curve for a Large Network



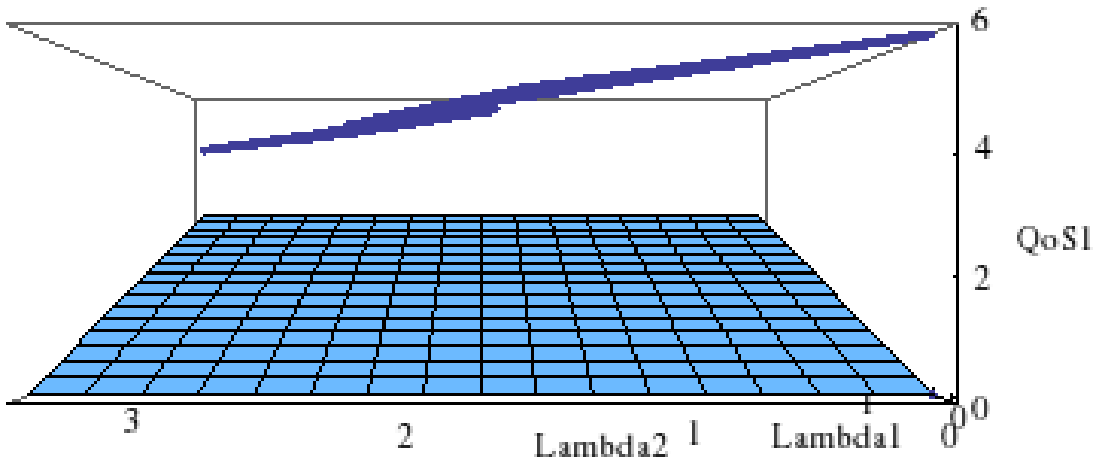
Angular View QoS_1 Curve for a Large Network



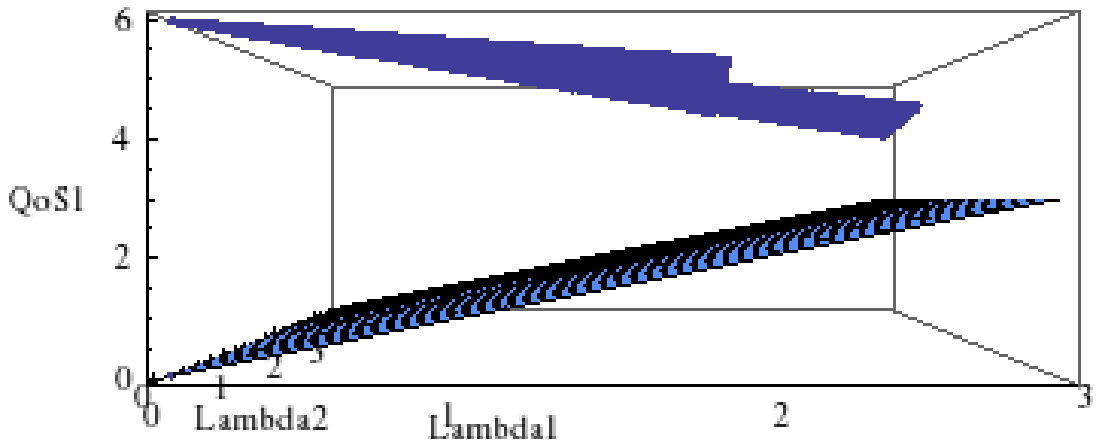
Top View QoS_1 Curve for a Large Network Intersection with plane



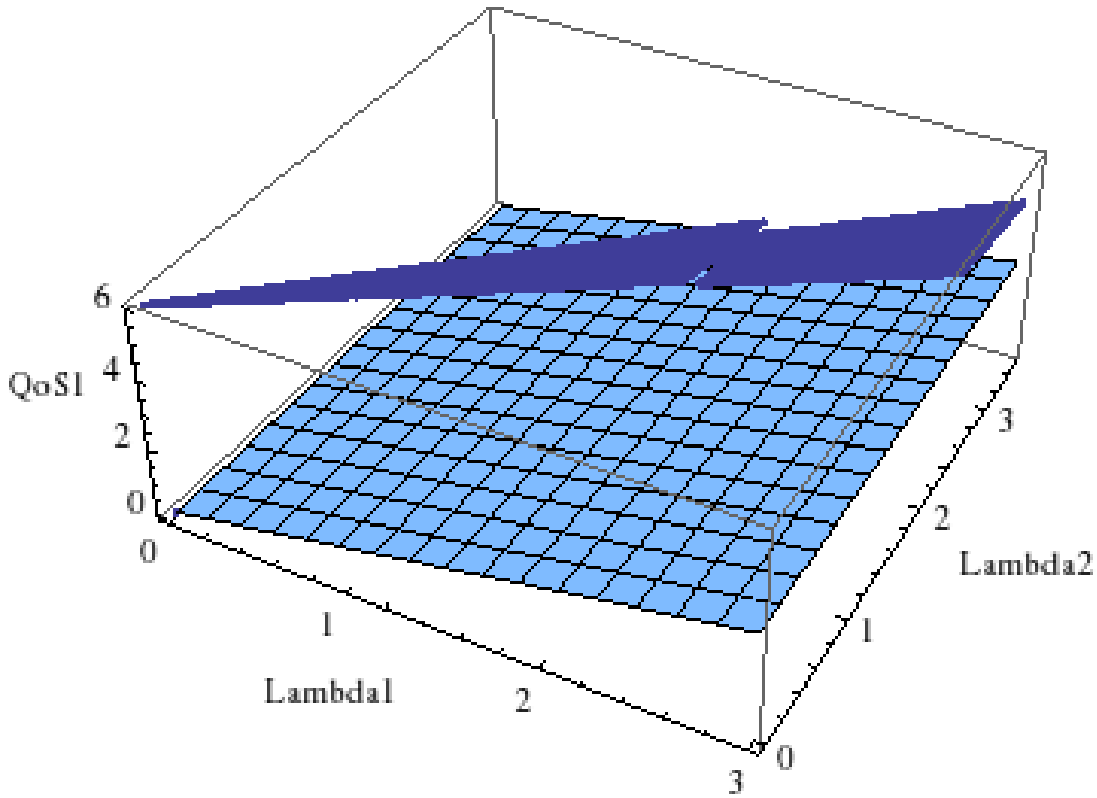
Side View QoS_1 Curve for a Large Network Intersection with plane



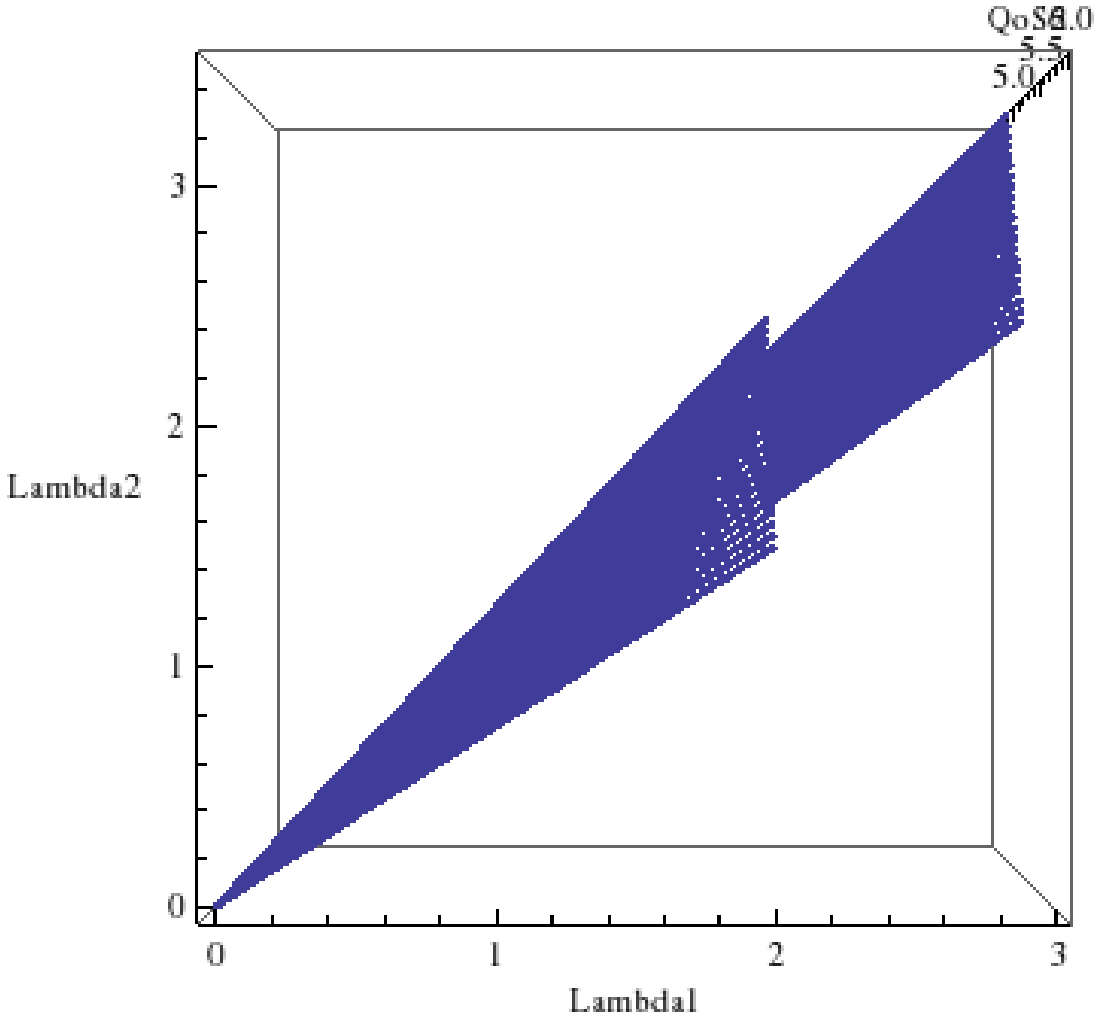
Front View QoS_1 Curve for a Large Network Intersection with plane



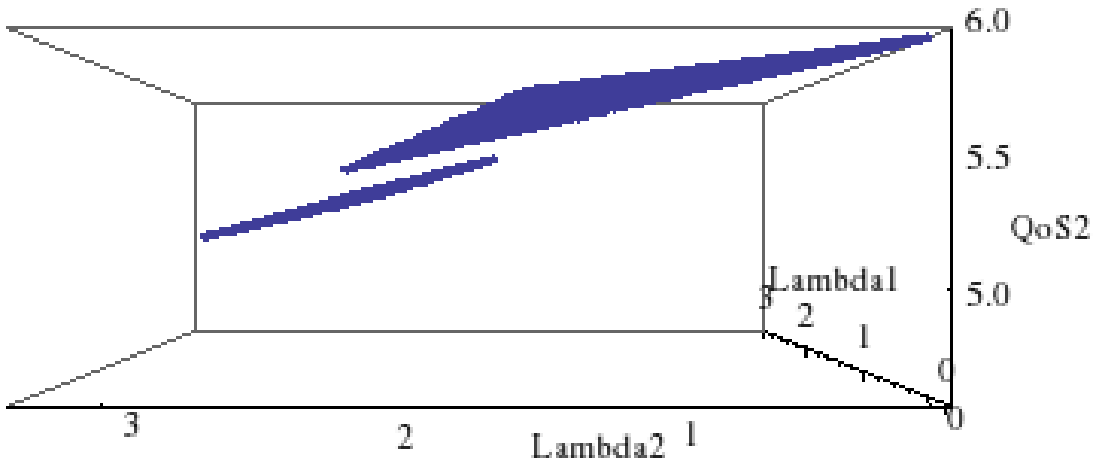
Angular View QoS_1 Curve for a Large Network Intersection with plane



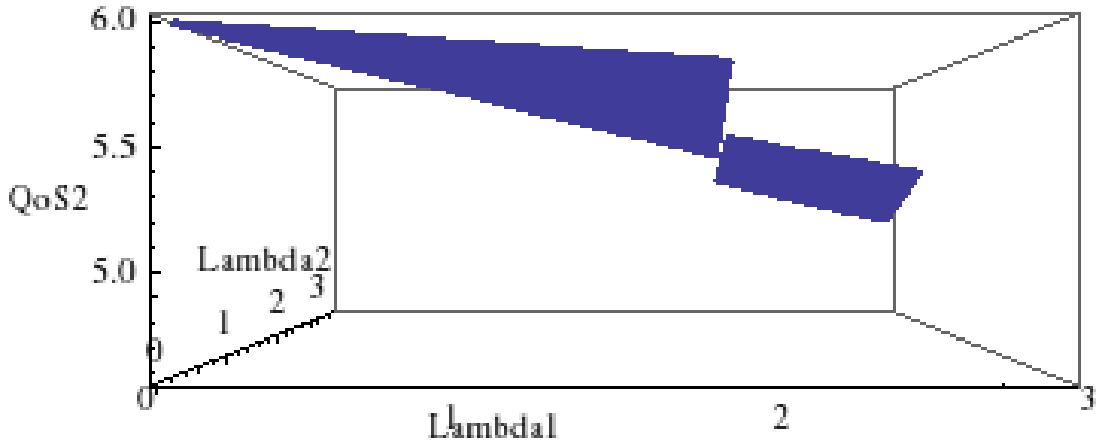
Top View QoS_2 Curve for a Large Network



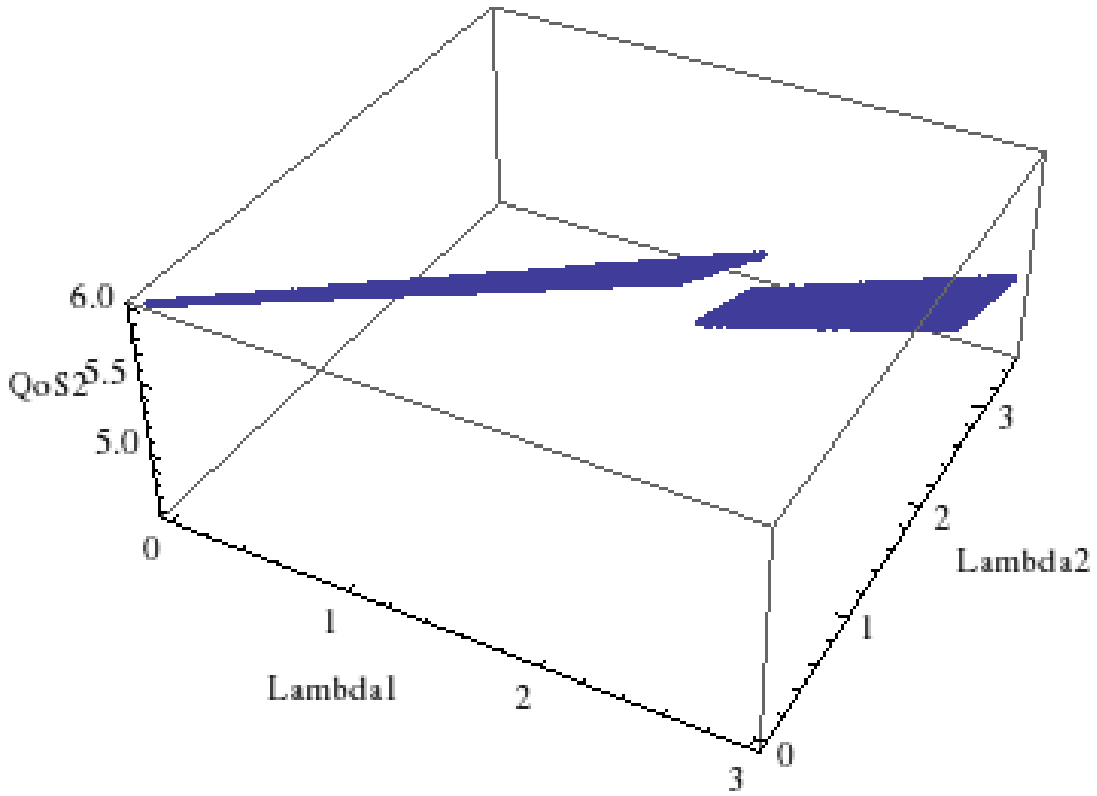
Side View QoS_2 Curve for a Large Network



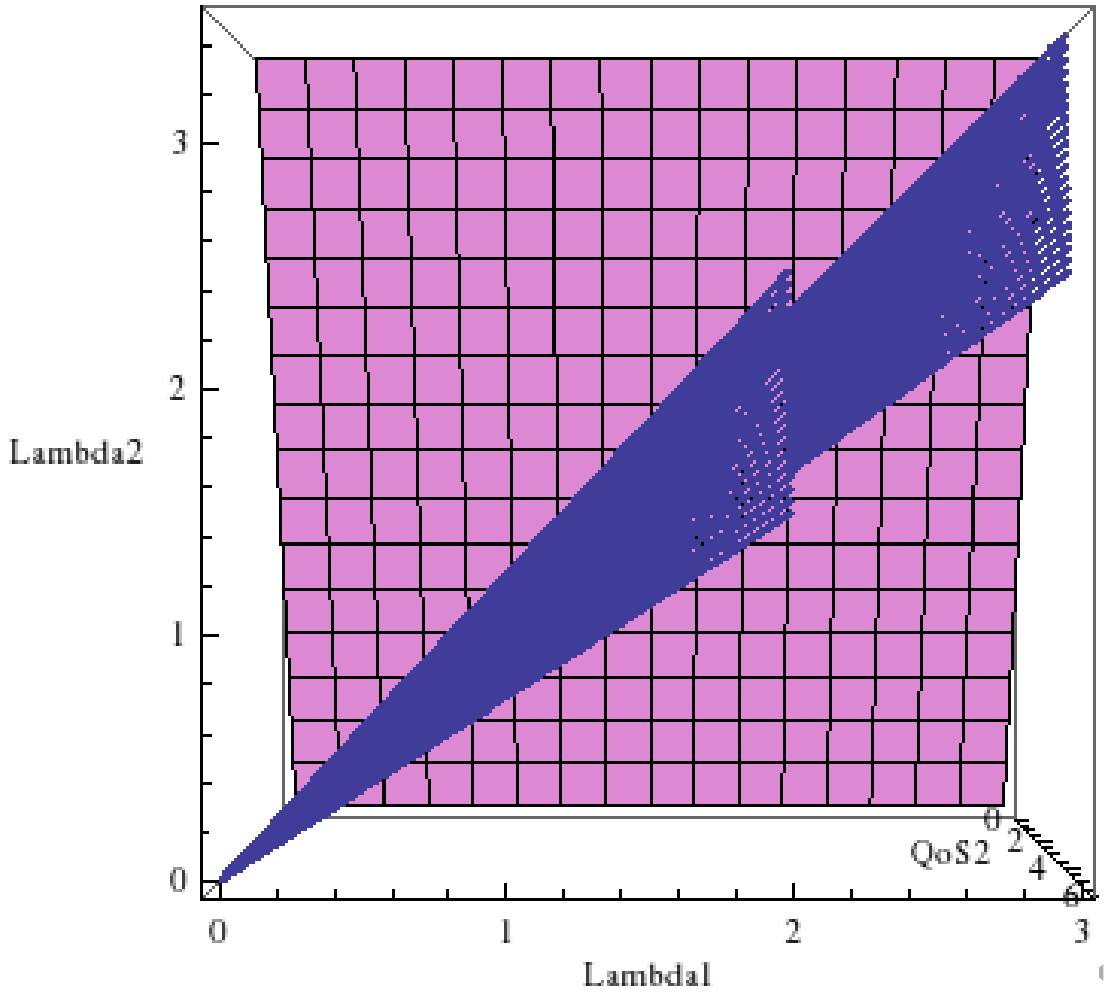
Front View QoS_2 Curve for a Large Network



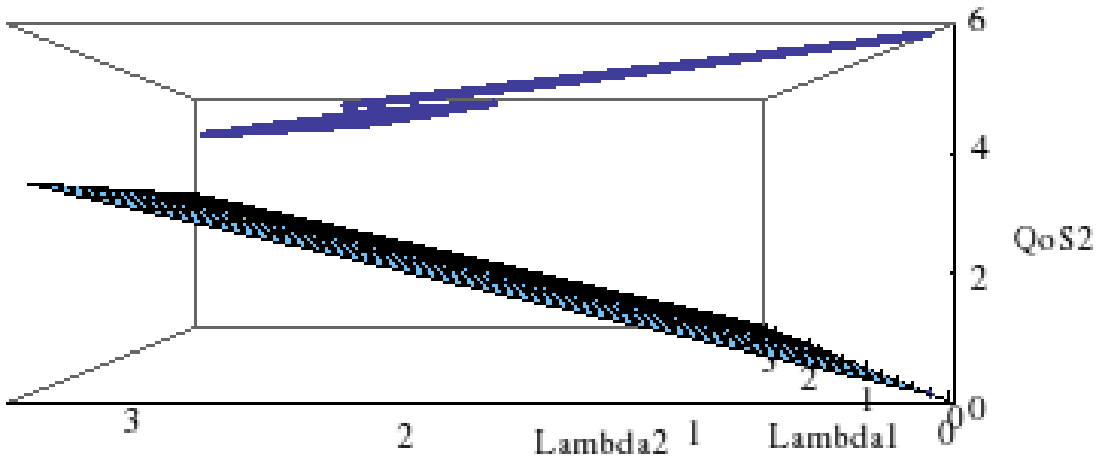
Angular View QoS_2 Curve for a Large Network



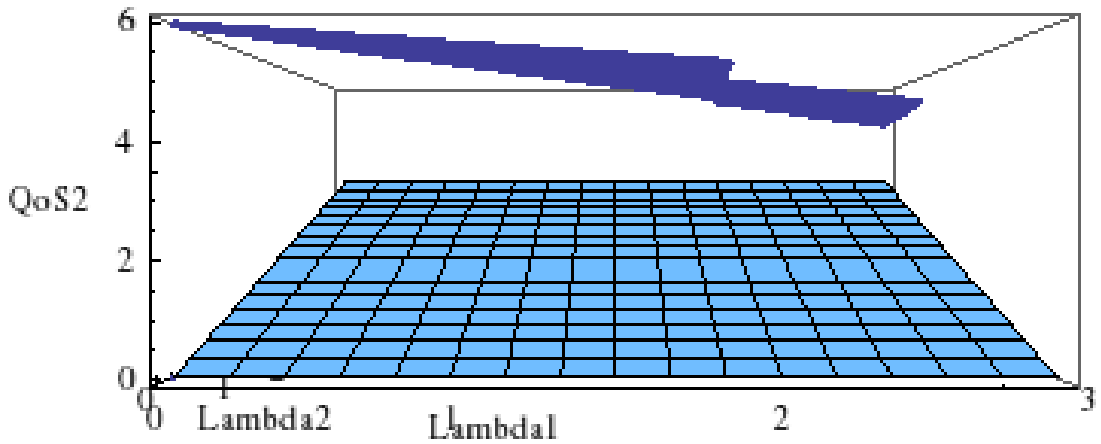
Top View QoS_2 Curve for a Large Network Intersection with plane



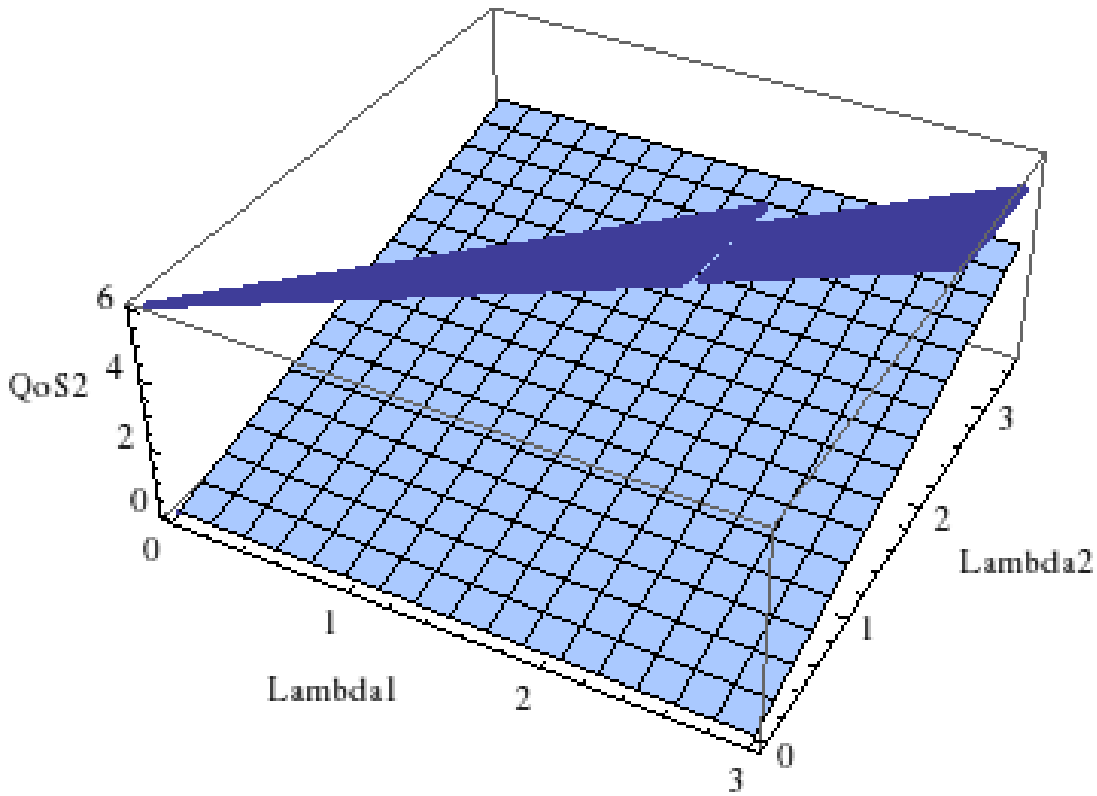
Side View QoS_2 Curve for a Large Network Intersection with plane



Front View QoS_2 Curve for a Large Network Intersection with plane



Angular View QoS_2 Curve for a Large Network Intersection with plane



- We see that the QoS Curves go above the equilibrium planes. If we project back these points onto the feasible region, we get the equilibrium to be

Equilibrium Point

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3.5 \end{bmatrix} \tag{17}$$

- In order to generate bids for a large networks for values of λ_1 and $\lambda_2 \in [0, 6)$, if we generate λ_1 upto 6 we will have 25 possible bids for a single customer.
- This will increase the time taken to pre-solve as the number of bid combinations increases polynomially. Thus, the new pre-solve will take 137.811 times more than the previous supports.
- There is a need to determine a new method for generating large bids without increasing the pre-solve time by a large amount.

Effect of Incentive Compatibility on Equilibrium

- The customer who is matched with the most expensive firm which is allocating resource is not incentive compatible.
- The customer is capable of increasing his utility by bidding incorrectly.
- The customers actions not only increases his utility but changes the allocation and utility of other customers.
- This change in system wide allocation can change the equilibrium of the system.

Effect of Incentive compatibility on single O-D pair networks I

How the player who is not incentive compatible is determined?

- In a single O-D pair network, we first determine the link on which the most expensive firm is offering a resource.
- We then determine the customers who are on the link.
- The customer who has bid the highest and who has been allocated a resource is considered as the customer who is not incentive compatible. We then change the price he bids and determine the allocations.
- The bid which gives him the most utility is taken as the bid that the customer will play.

Effect of Incentive compatibility on single O-D pair networks II

- We then use this new allocation to determine the new equilibrium of the system.
- Corresponding to a given bid pair there will be a unique bid pair that would be played instead.
- Thus each new bid pair has the same probability as the bid pair it is derived from.
- Thus, the same method where we categorise the bids and find their summations is applicable to determine the QoS curve.

Change in Equilibrium in systems with 2 Interacting Parameters

- In this section, the effect of the top customer not being incentive compatible on the equilibrium of the two parameter system is studied.
- As seen with the single parameter incentive compatibility, we determine the top player in the same way.
- Because of the way we are determining the the incentive compatible bid combination, it will have the same probability as that of the bid combination it is determined from. Thus, we can categorically divide them and find the summations to determine the following equations

QoS_1 and QoS_2 equation for smaller supports

$$QoS_1 = 2 - \frac{61}{288}\lambda_1^2 - \frac{1}{9}\lambda_1 + \frac{59}{108}\lambda_1\lambda_2 - \frac{2}{3}\lambda_2 - \frac{11}{54}\lambda_2^2 \quad (18)$$

$$QoS_2 = 2 - \frac{97}{288}\lambda_1^2 - \frac{1}{6}\lambda_1 + \frac{7}{9}\lambda_1\lambda_2 - \frac{1}{3}\lambda_2 - \frac{7}{18}\lambda_2^2 \quad (19)$$

QoS_1 and QoS_2 equation for larger supports

$$QoS_1 = -\frac{2191\lambda_1^2}{27000} + \frac{\lambda_1(519\lambda_2 - 340)}{2250} - \frac{11\lambda_2^2}{150} - \frac{2\lambda_2}{5} + 2 \quad (20)$$

$$QoS_2 = -\frac{217\lambda_1^2}{1800} + \frac{4}{75}\lambda_1(6\lambda_2 - 5) - \frac{7\lambda_2^2}{50} - \frac{\lambda_2}{5} + 2 \quad (21)$$

The equilibrium point determined when the top player is not incentive compatible

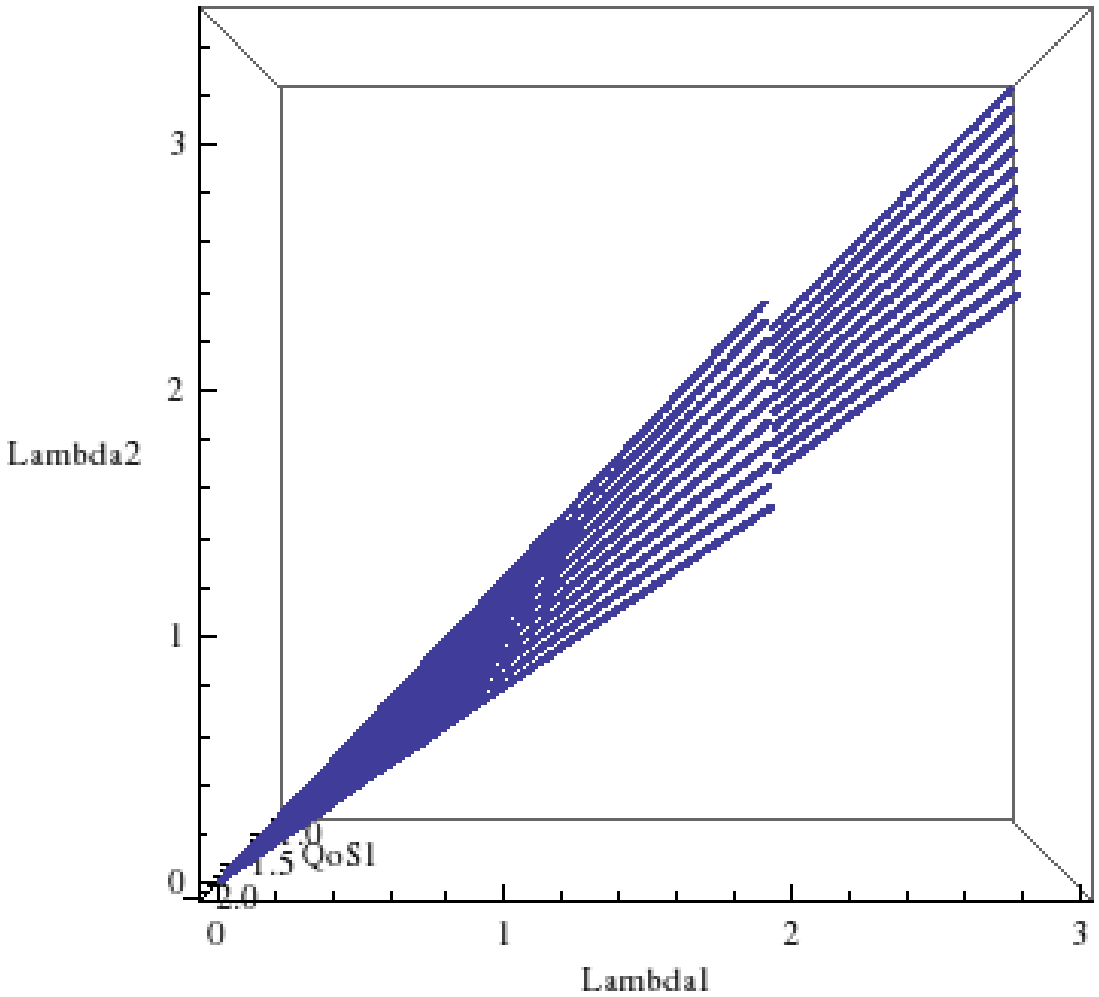
$$\begin{bmatrix} \lambda_{eq1IC} \\ \lambda_{eq2IC} \end{bmatrix} = \begin{bmatrix} 1.14137 \\ 1.38823 \end{bmatrix} \tag{22}$$

The equilibrium point obtained earlier is

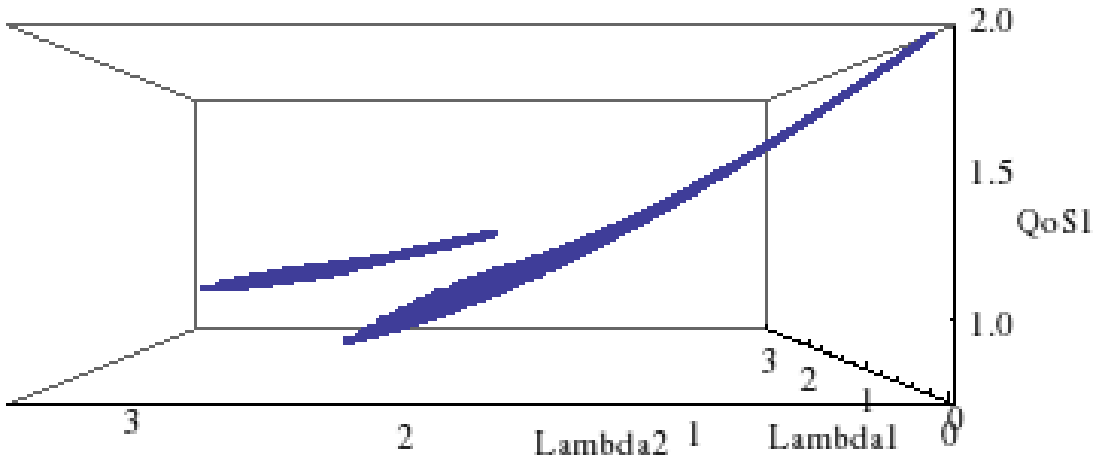
$$\begin{bmatrix} \lambda_{eq} \\ \lambda_{eq} \end{bmatrix} = \begin{bmatrix} 1.13 \\ 1.3 \end{bmatrix} \tag{23}$$

Clearly, the shift indicates an increase in the demand at equilibrium as well as increase in mean price that is bid.

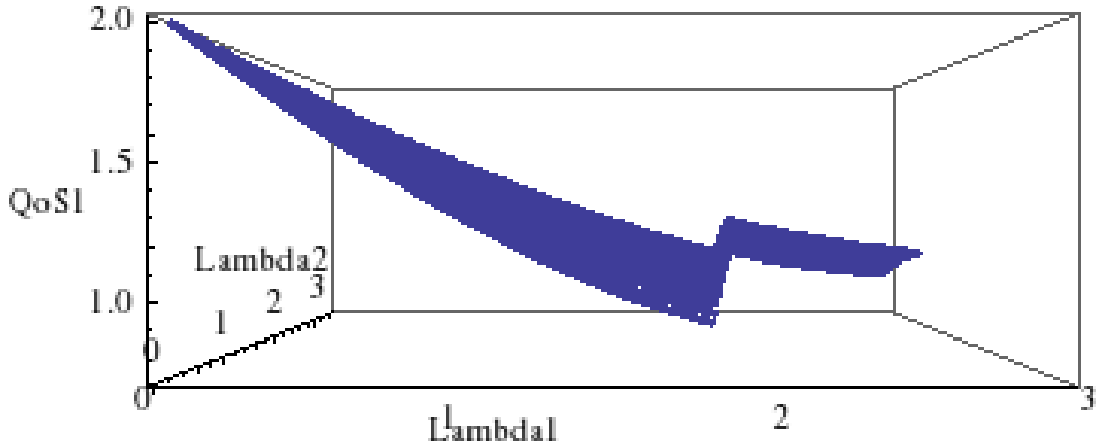
Top View QoS_1



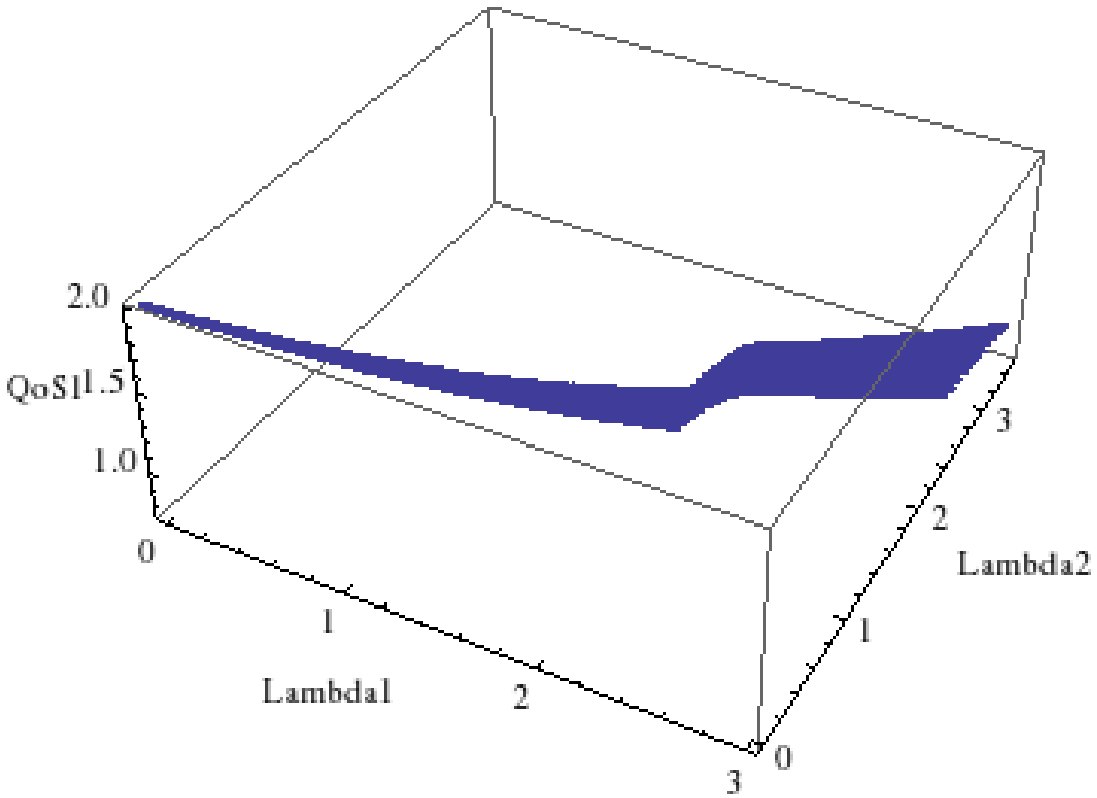
Side View QoS_1



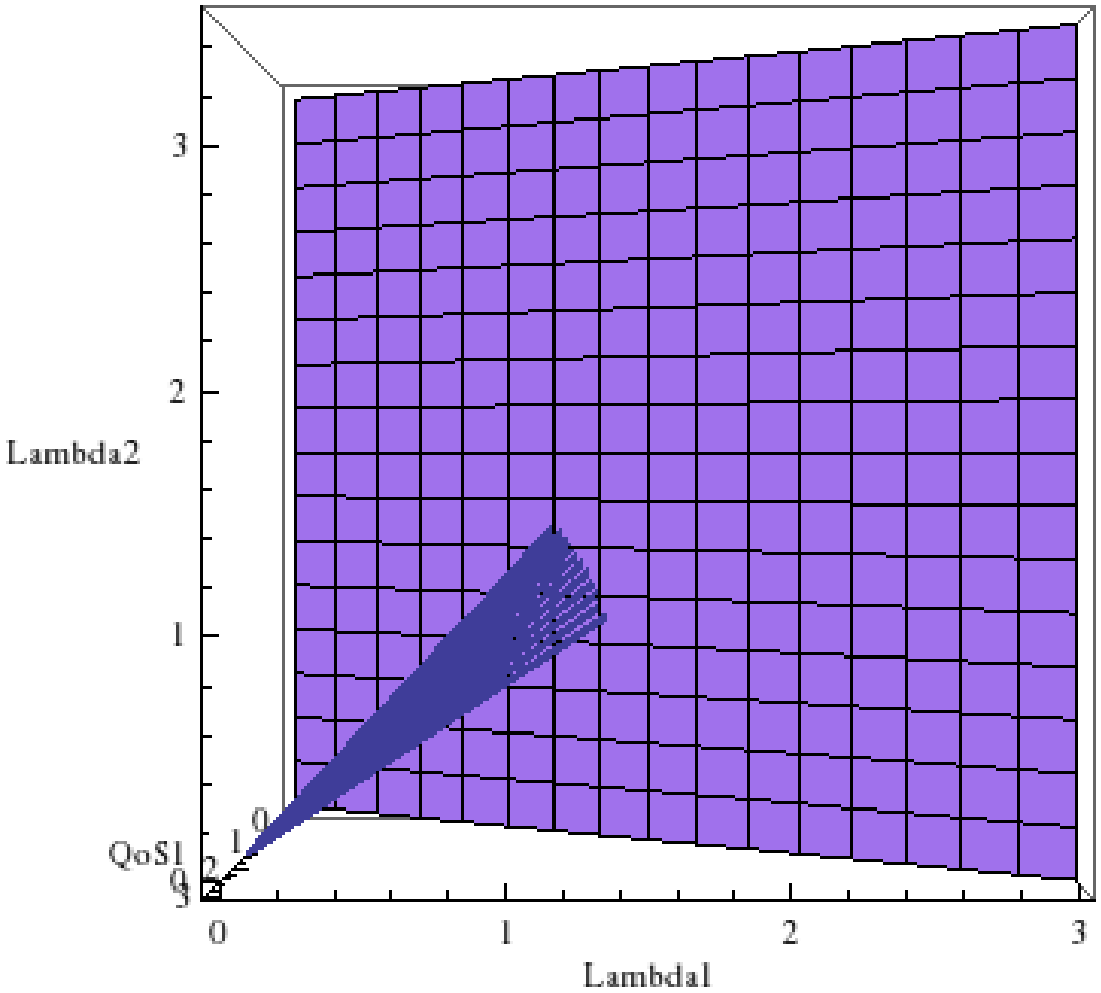
Front View QoS_1



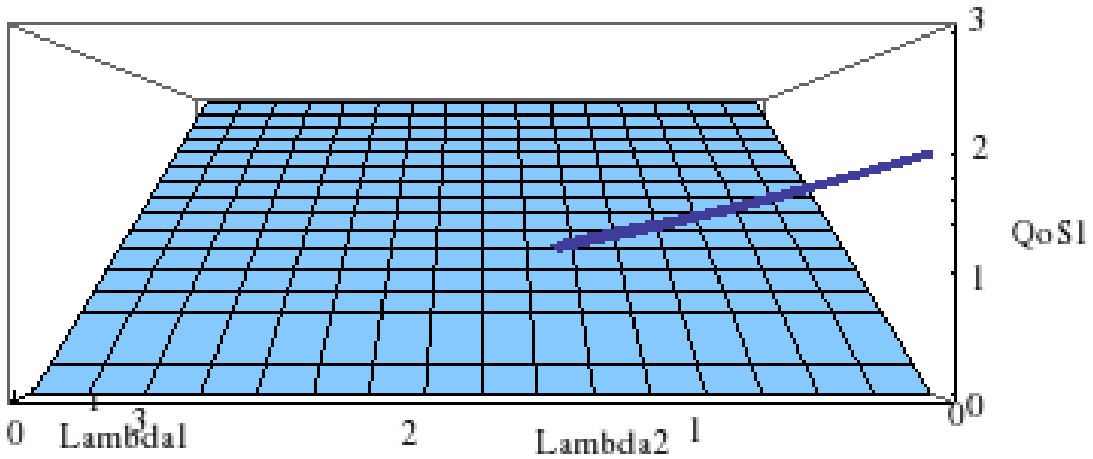
Angular View QoS_1



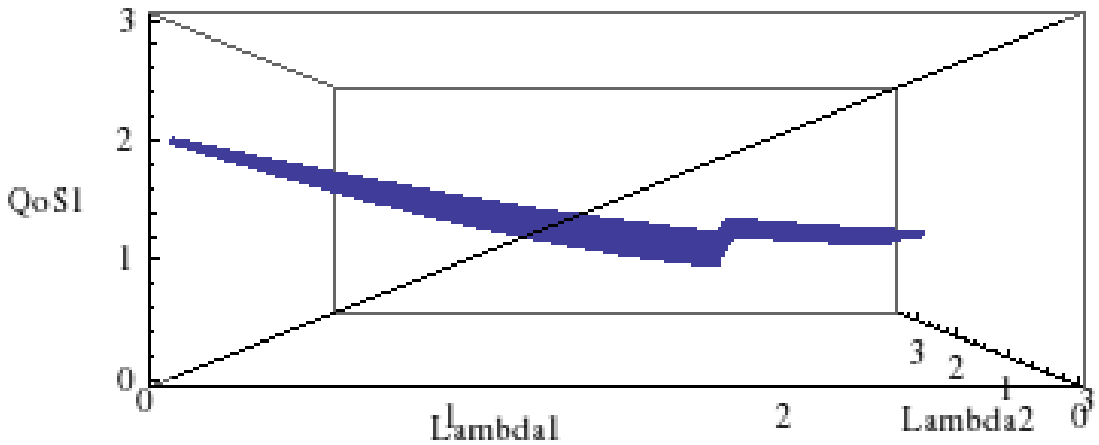
Top View QoS_1 Intersection with plane



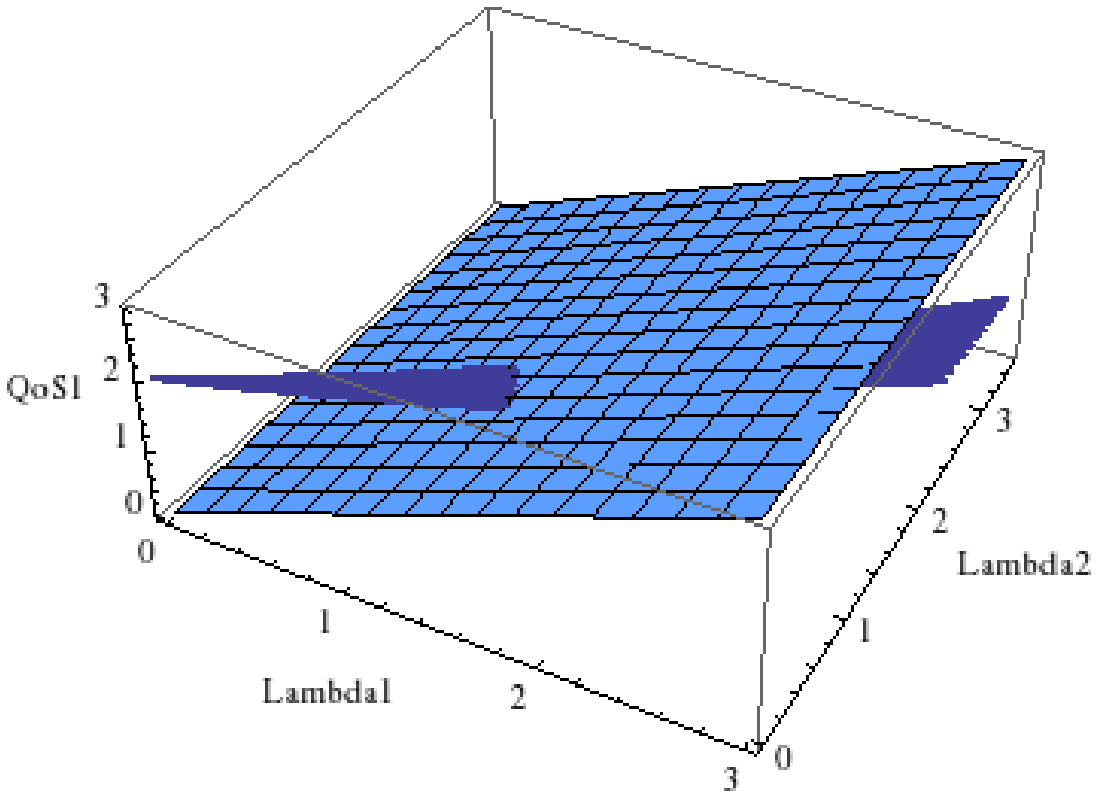
Side View QoS_1 Intersection with plane



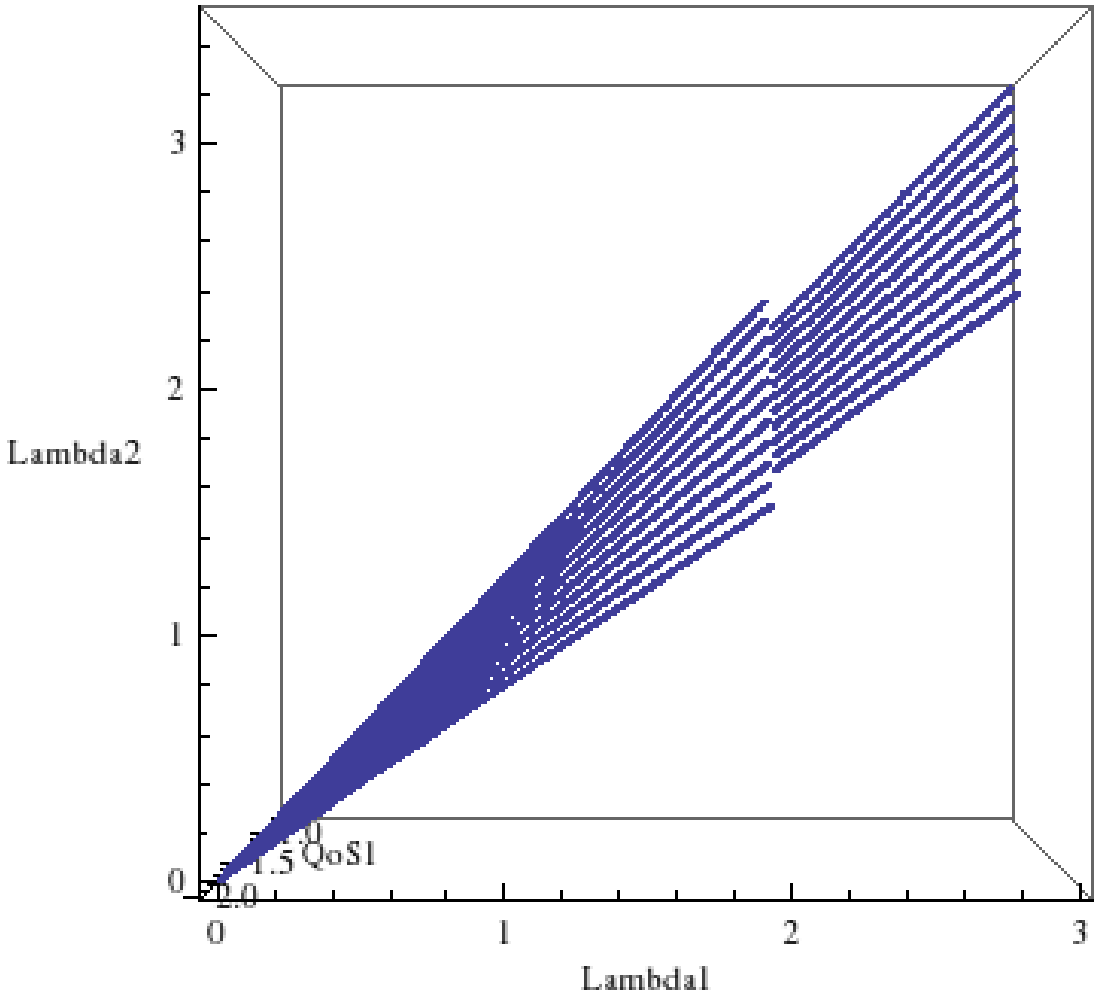
Front View QoS_1 Intersection with plane



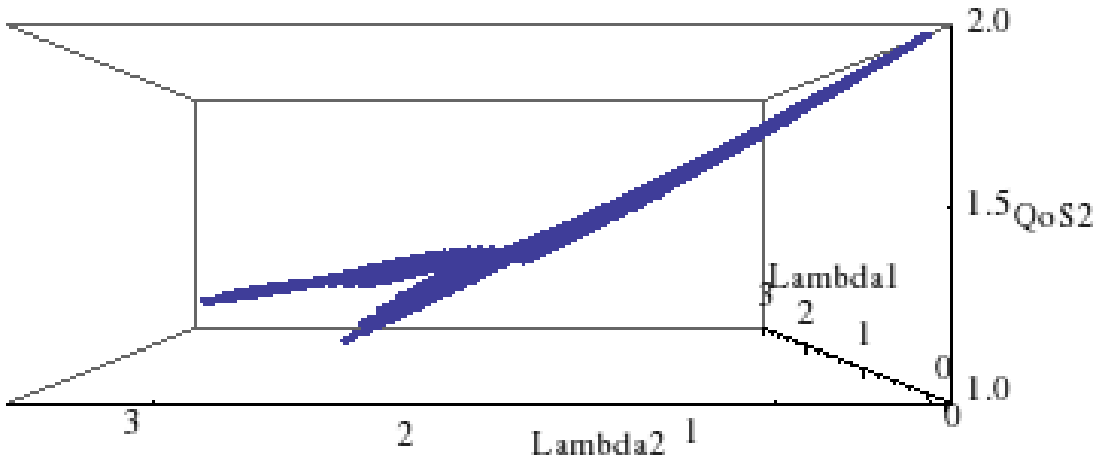
Angular View QoS_1 Intersection with plane



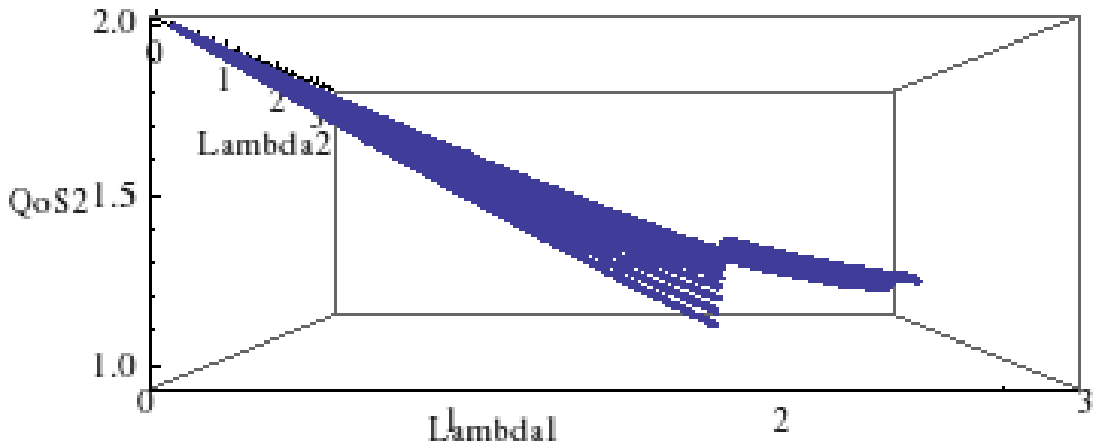
Top View QoS_2



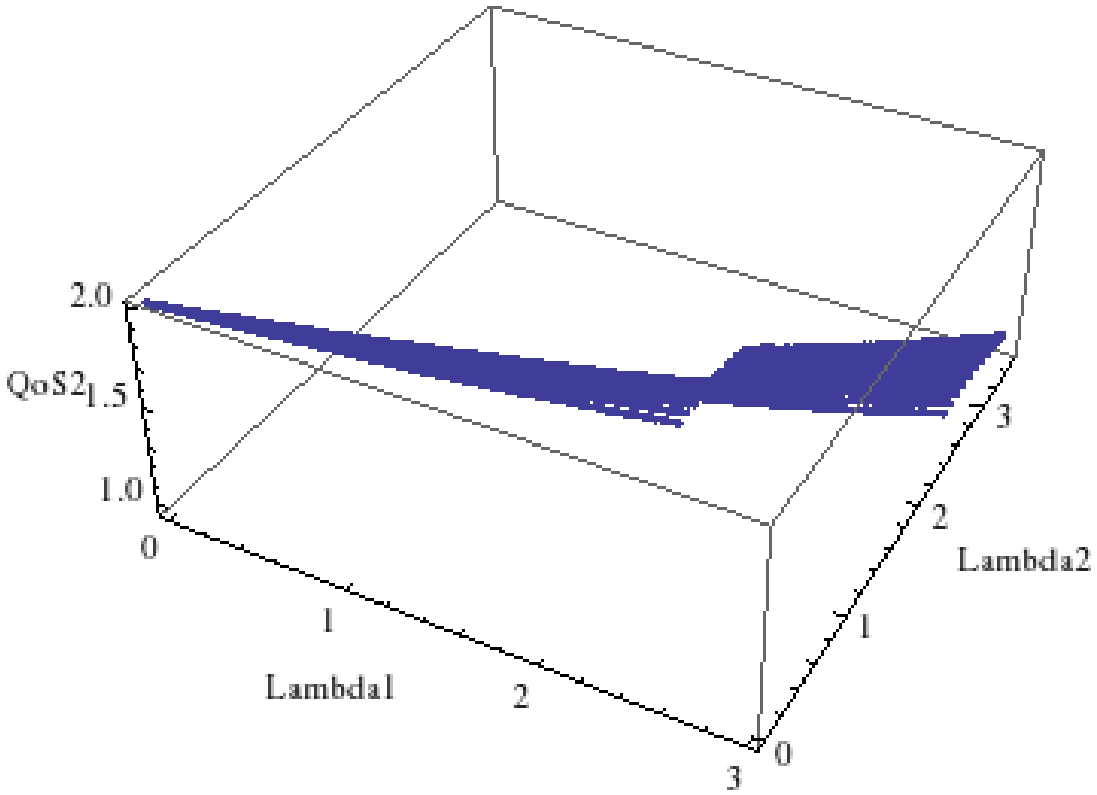
Side View QoS_2



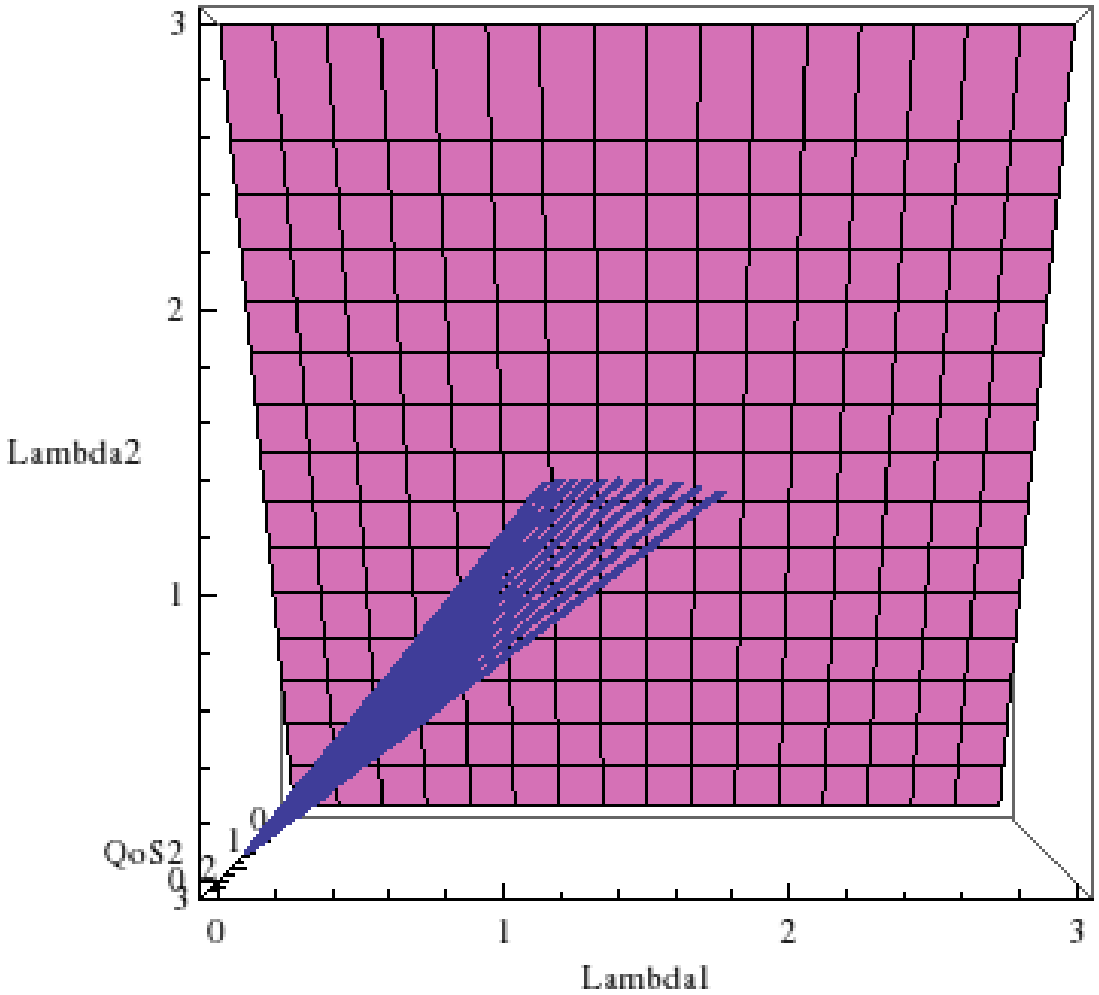
Front View QoS_2



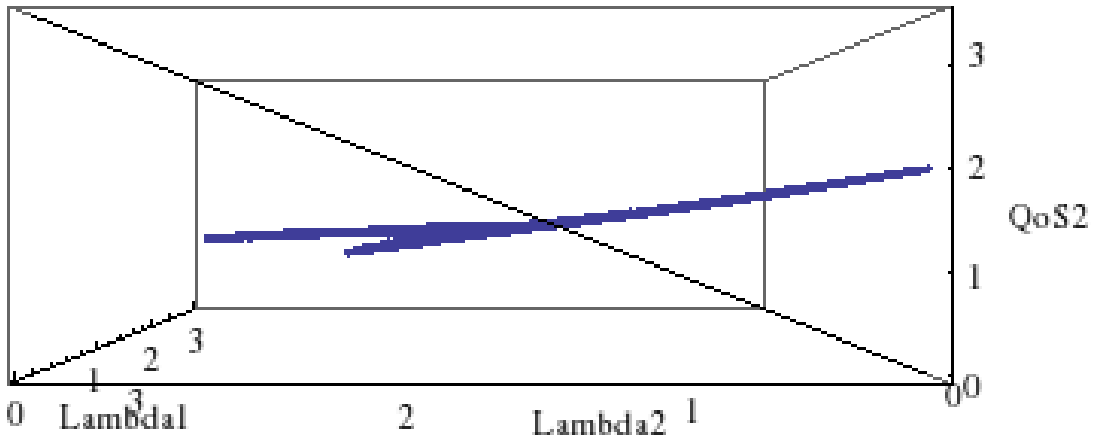
Angular View QoS_2



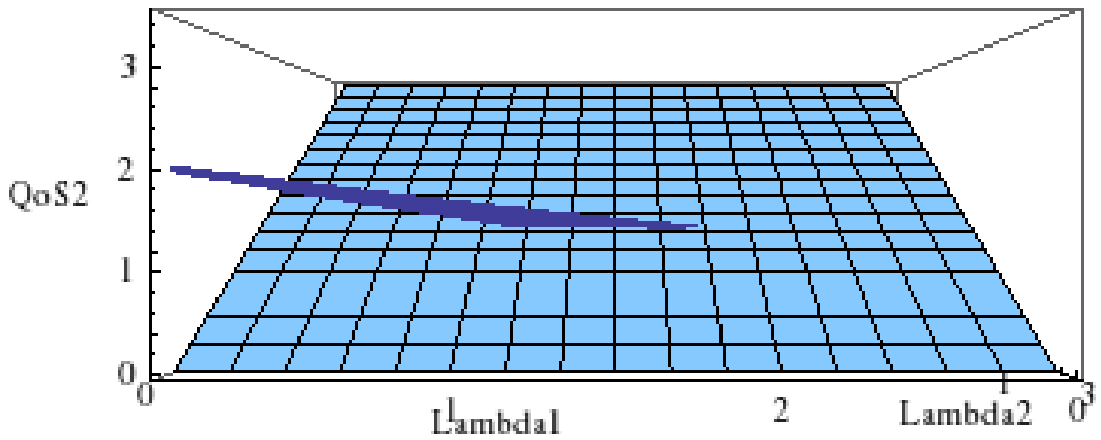
Top View QoS_2 Intersection with plane



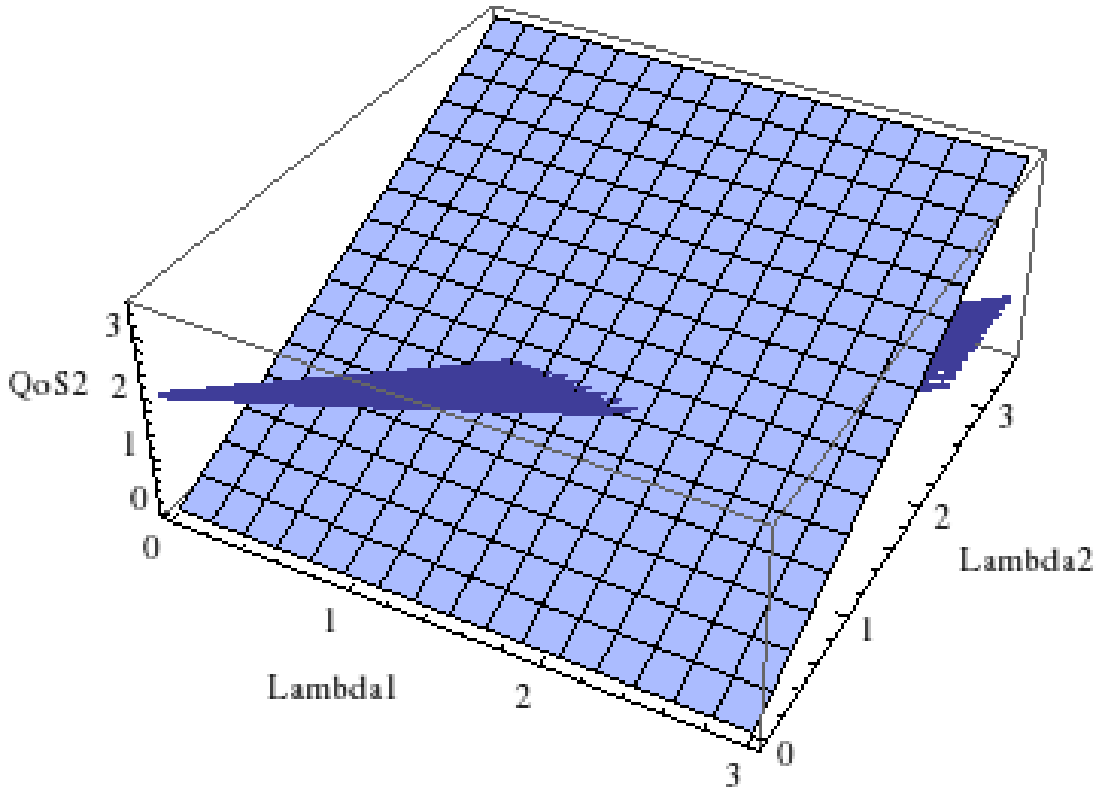
Side View QoS_2 Intersection with plane



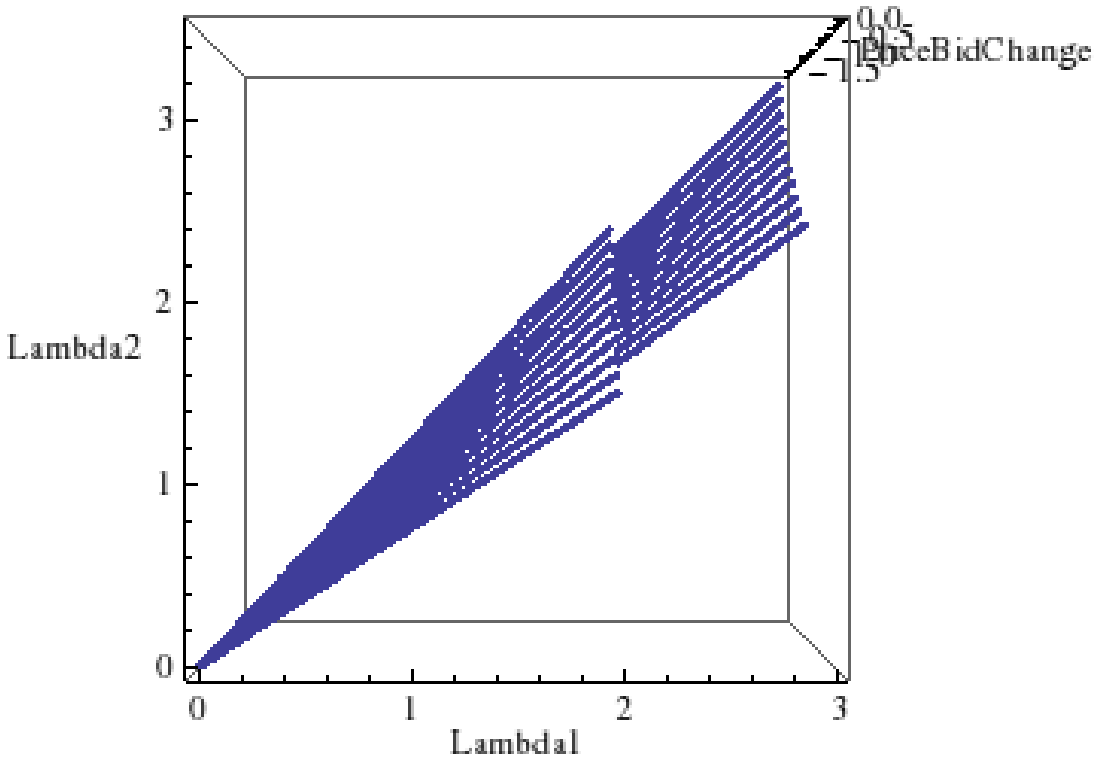
Front View QoS_2 Intersection with plane



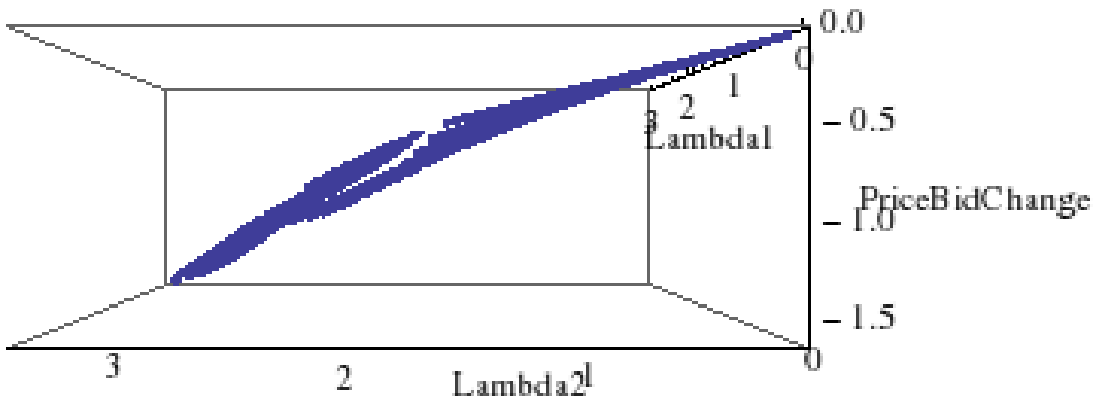
Angular View QoS_2 Intersection with plane



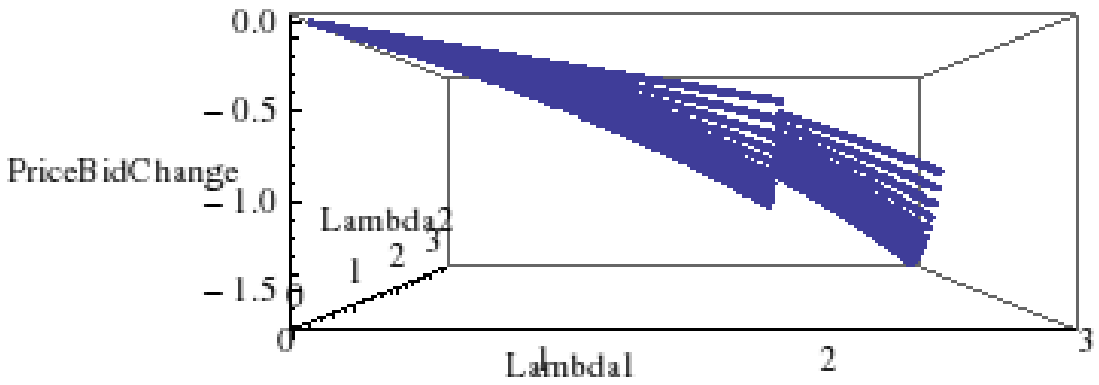
Top View Average Change in price



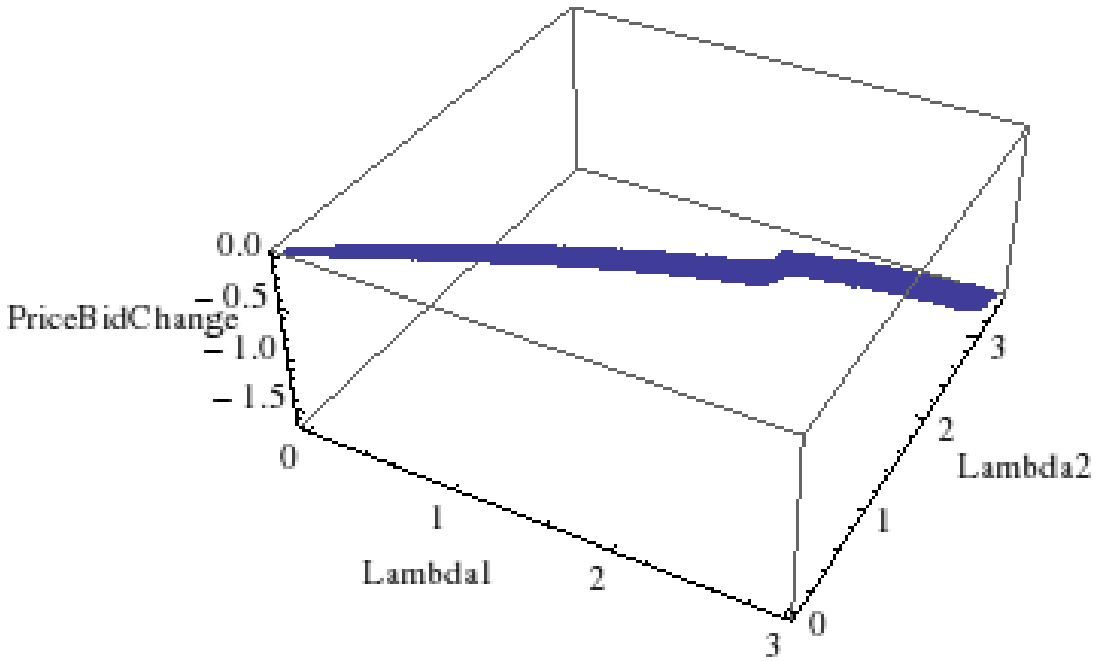
Side View Average Change in price



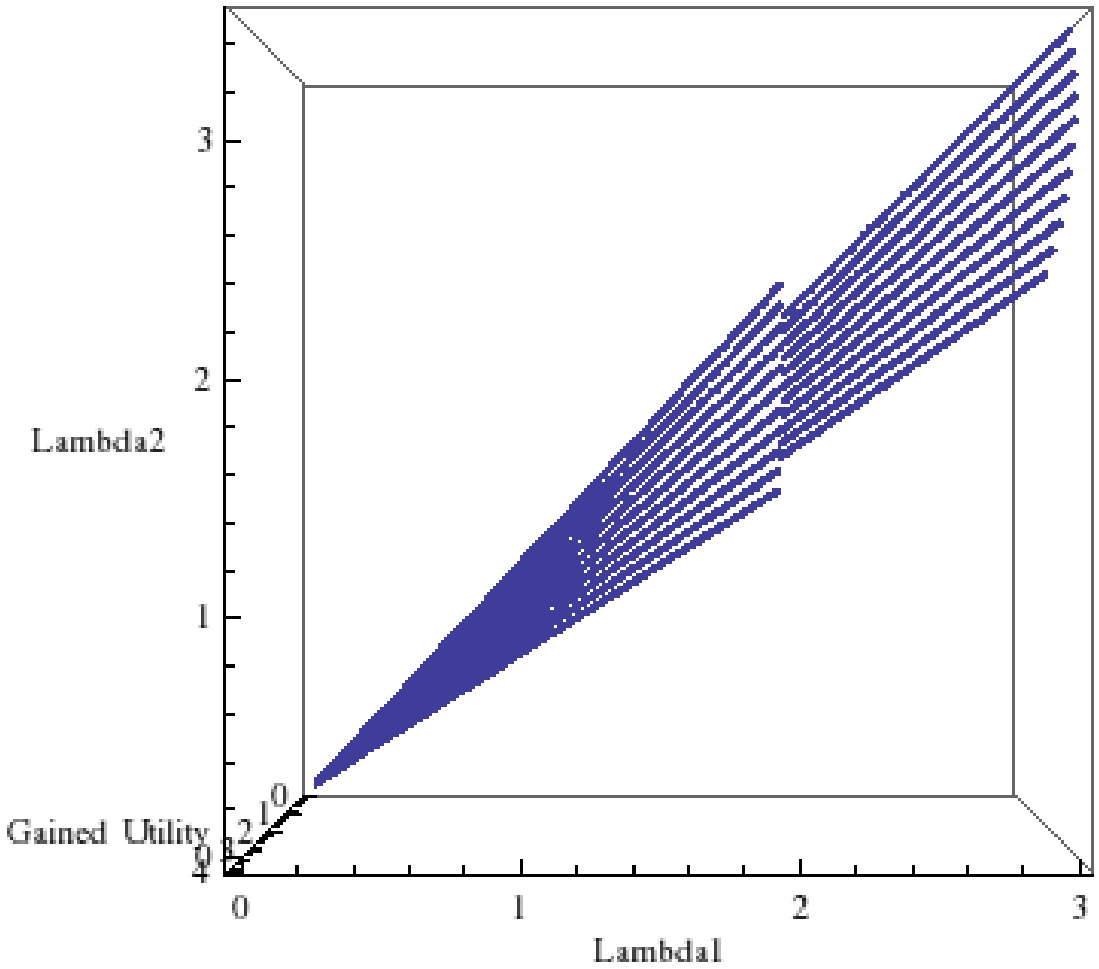
Front View Average Change in price



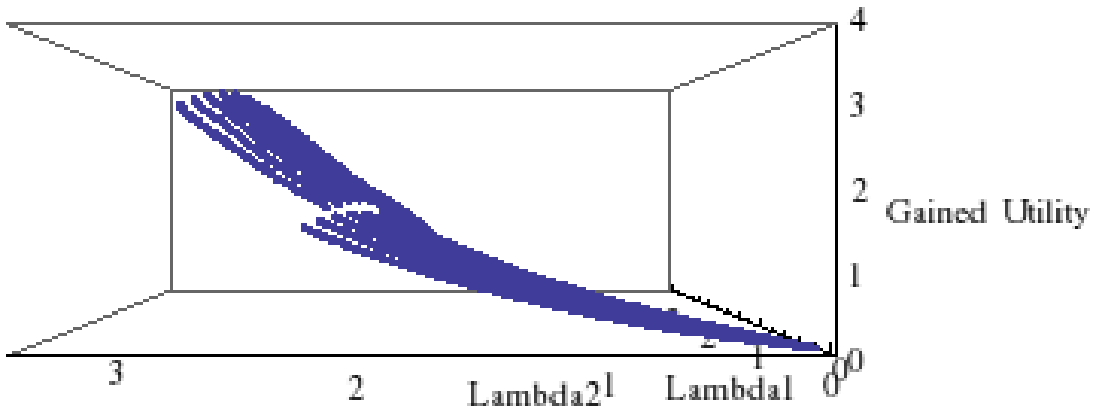
Angular View Average Change in price



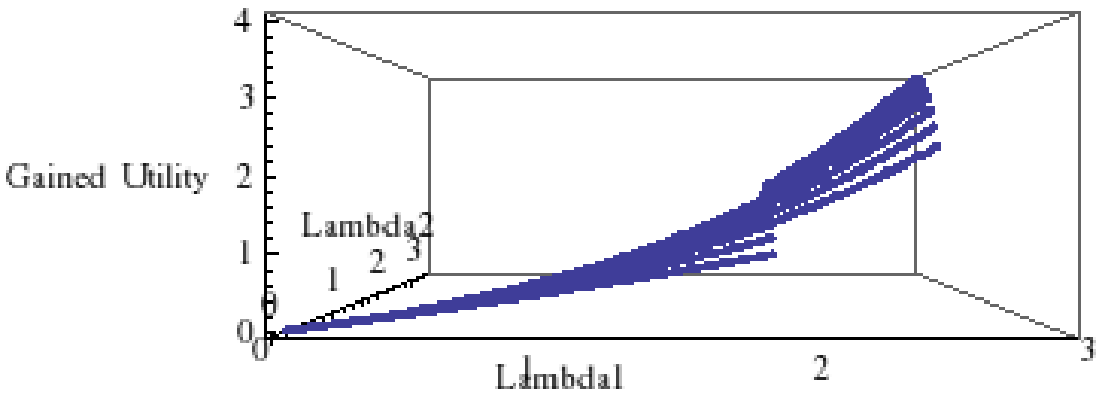
Top View Average Change in utility



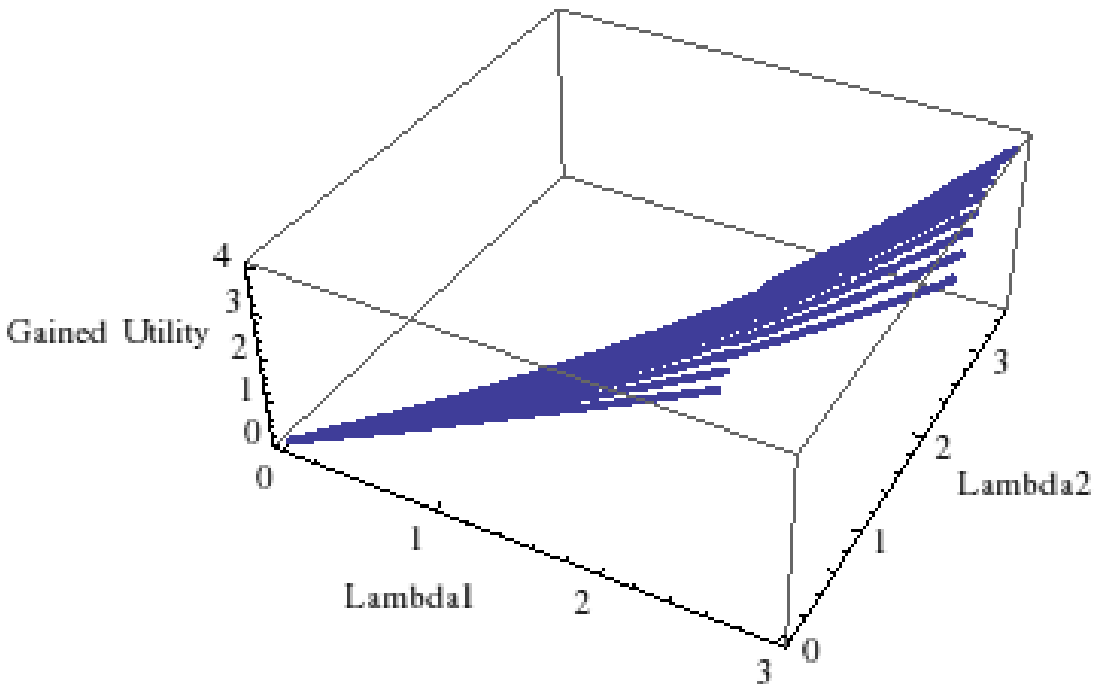
Side View Average Change in utility



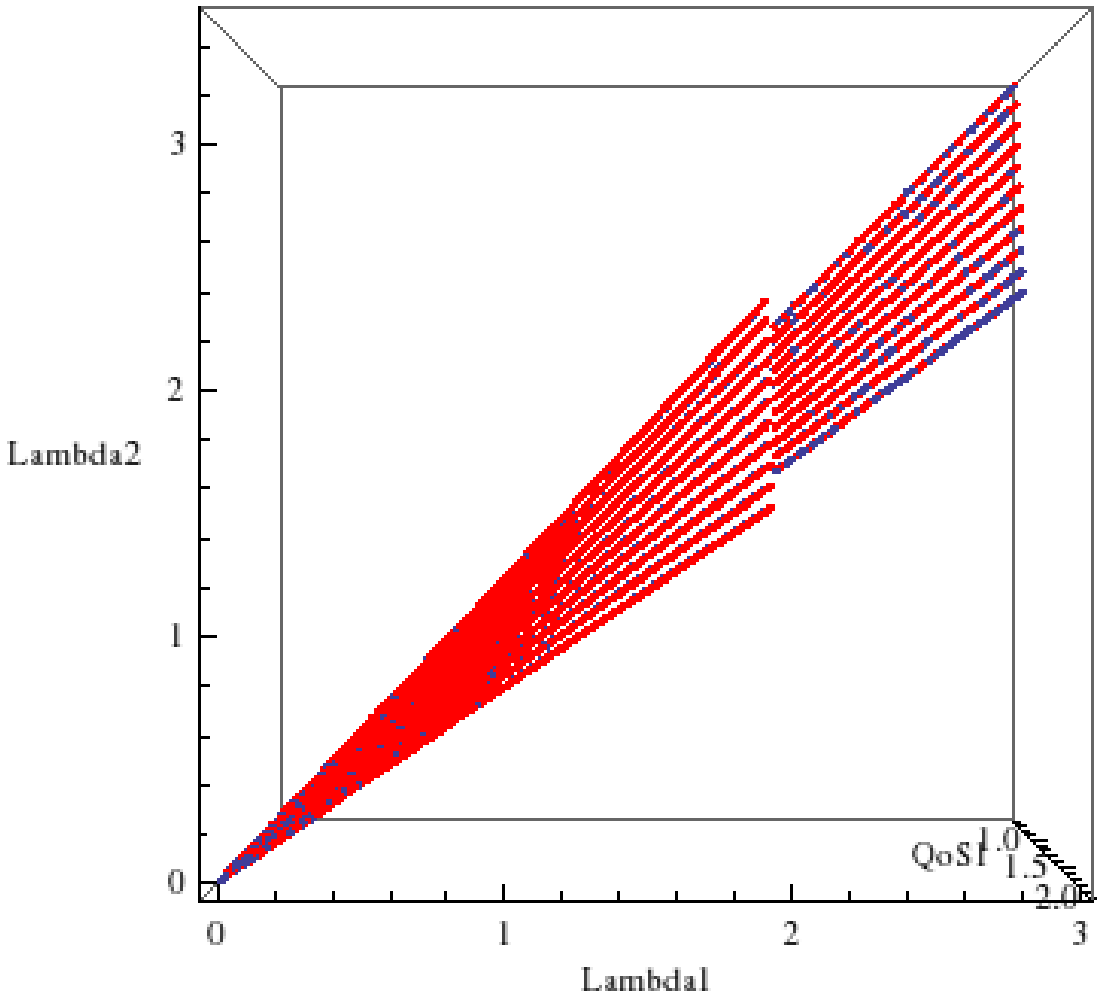
Front View Average Change in utility



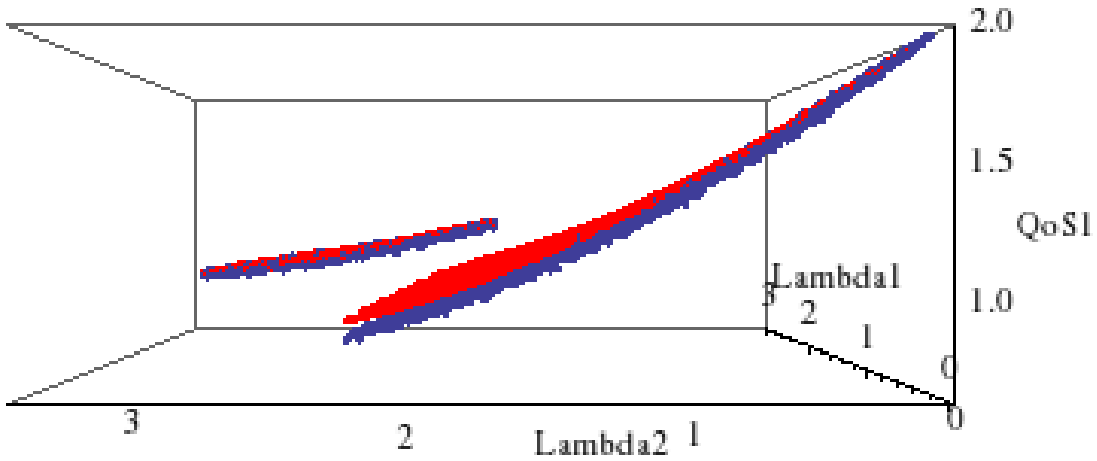
Angular View Average Change in utility



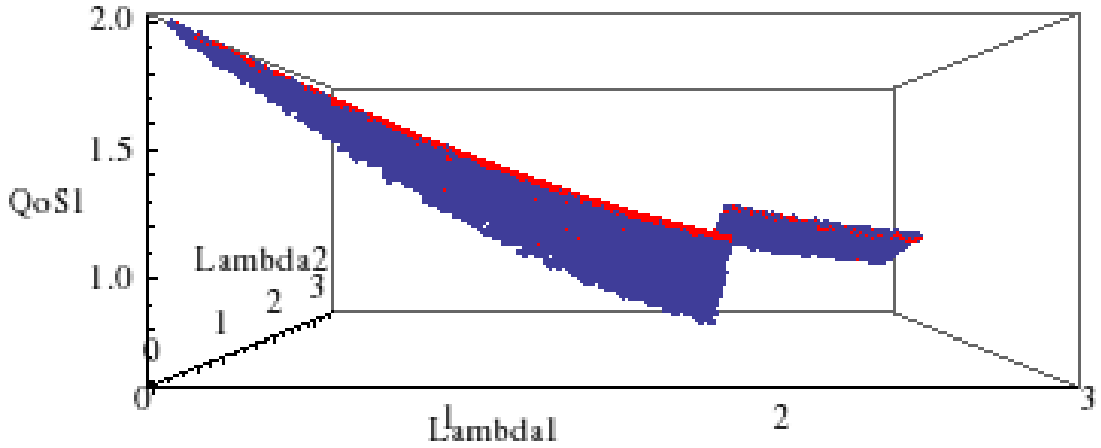
Top View Comparison QoS_1



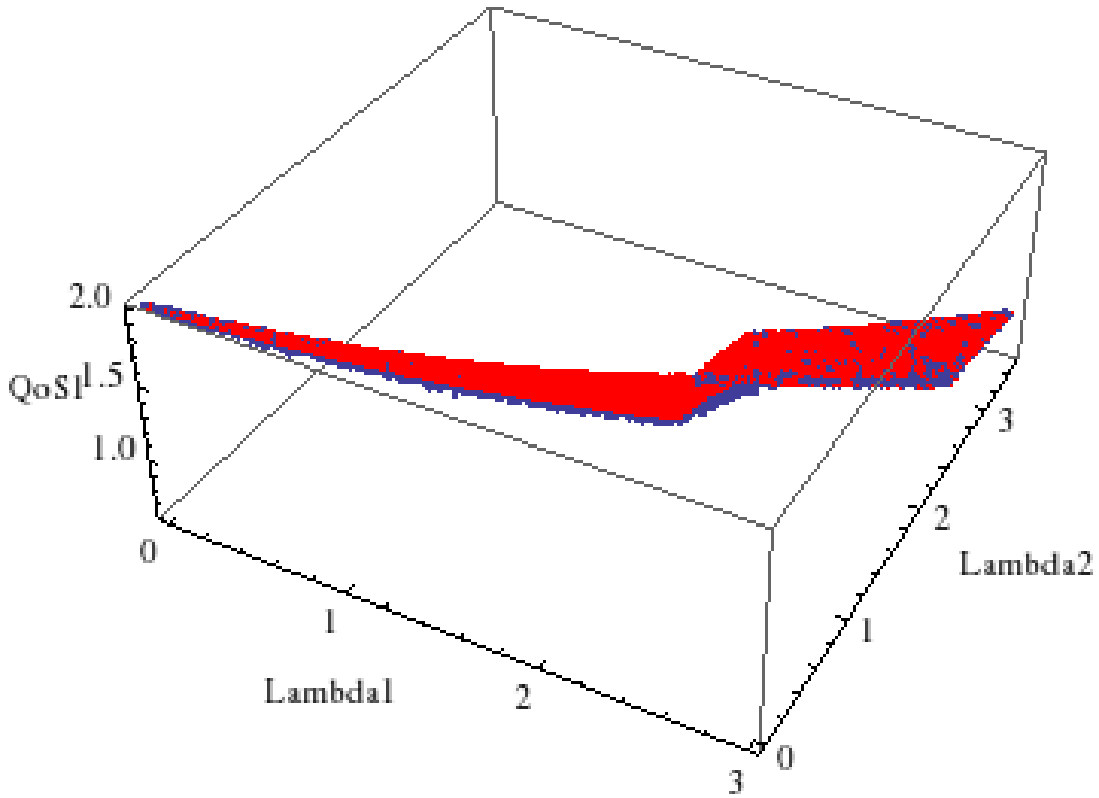
Side View Comparison QoS_1



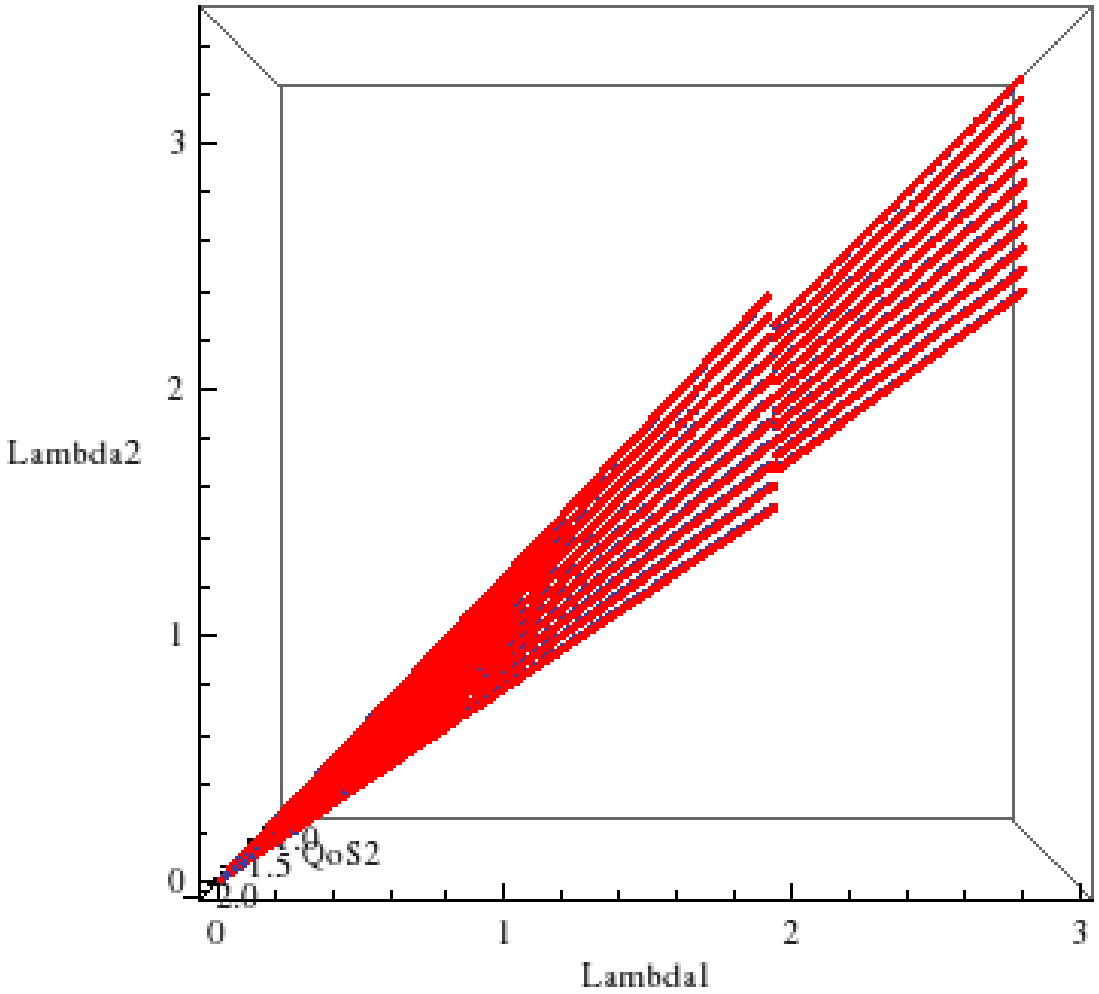
Front View Comparison QoS_1



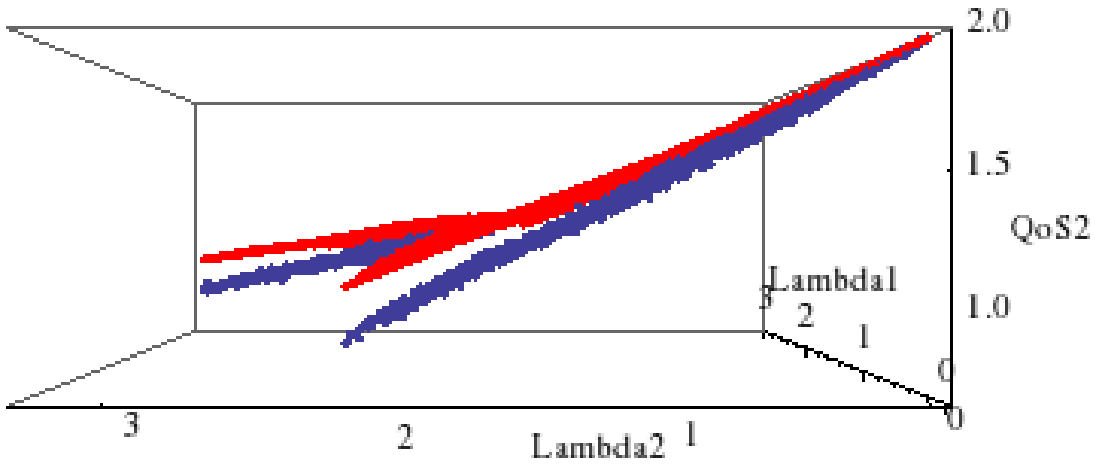
Angular View Comparison QoS_1



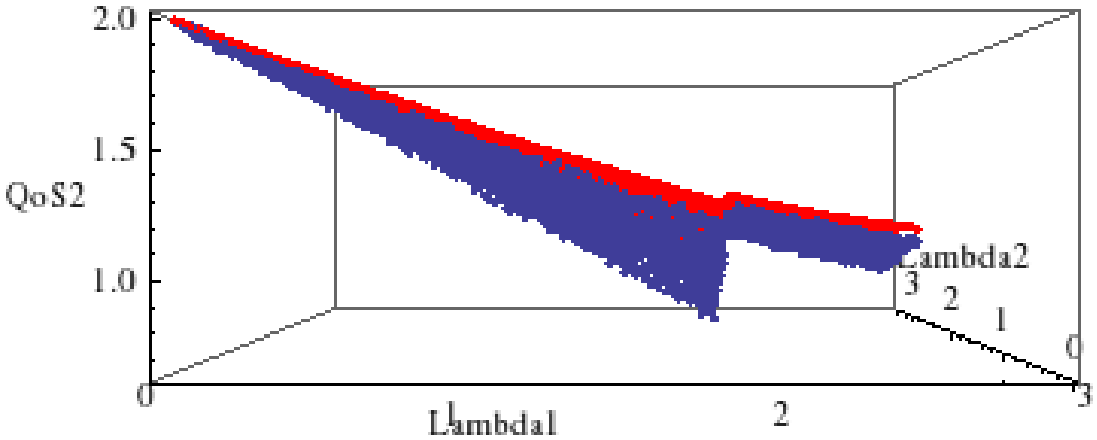
Top View Comparison QoS_2



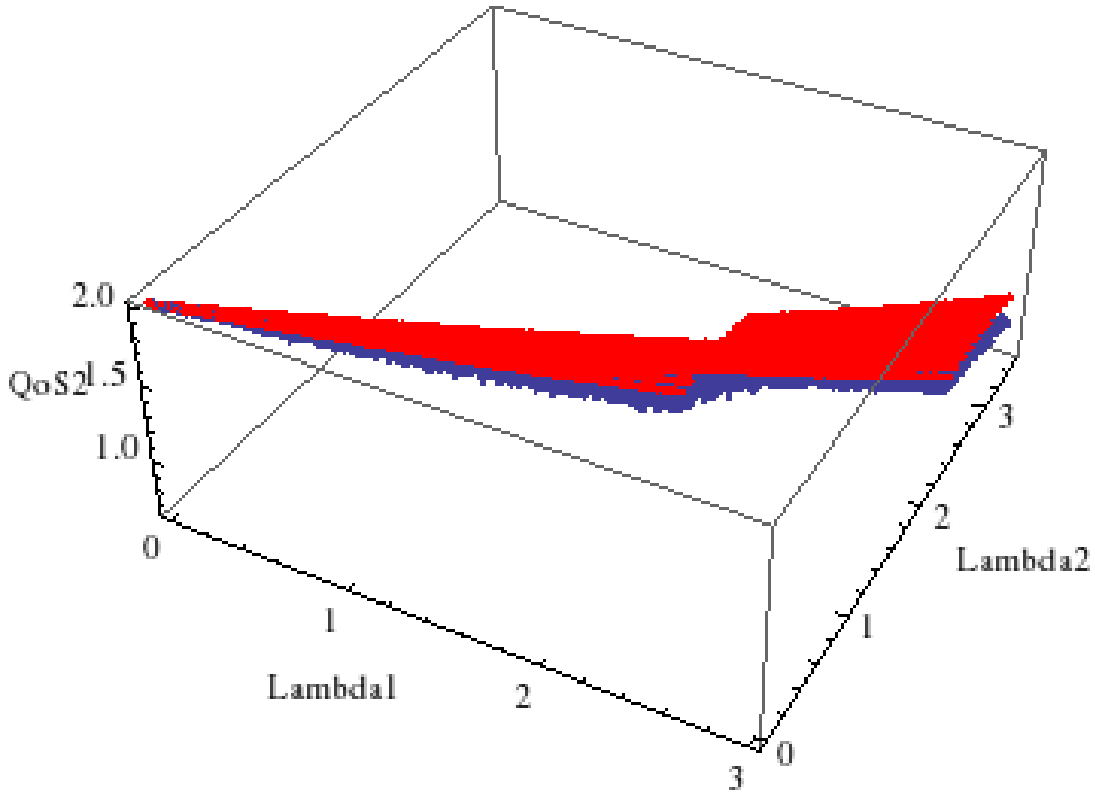
Side View Comparison QoS_2



Front View Comparison QoS_2



Angular View Comparison QoS_2



- Similar to the single parameter system, the equilibrium point has shifted from $(1.13, 1.3)$ to $(1.4137, 1.38823)$ which indicates an increase in the demand at equilibrium as well as increase in the mean price that is bid.
- It can be concluded that the top customer in a system not being incentive compatible will shift the equilibrium to a higher value both in terms of the mean demand as well as the mean price that is bid.
- *Hence, we will underestimate actual demand if we consider that all players are incentive compatible*.

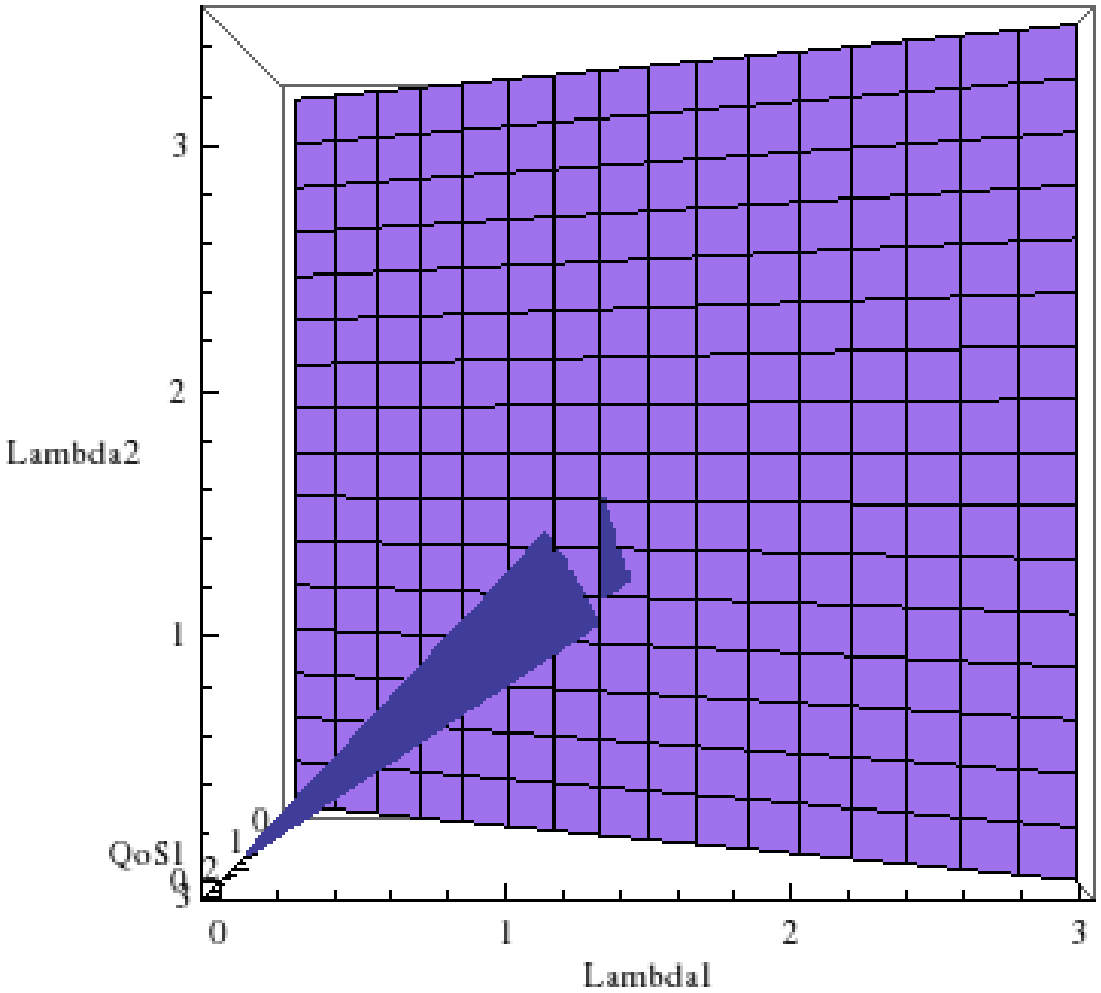
The reasons for the QoS curves to shift upwards in both cases is

- With top customers reducing their price bid, when not incentive compatible, they are getting the resource at a lower price, thus increasing their utility.
- This happens because the customer who is playing the lowest gets knocked out of the system. Thus there will be a need to increase demand from the lower customers part in order to get the resource.
- Similarly, the lowest customer must also increase his price bid so that they do not get knocked out of the system. This increases the equilibrium of the system in terms of mean price that is being bid (λ_2).

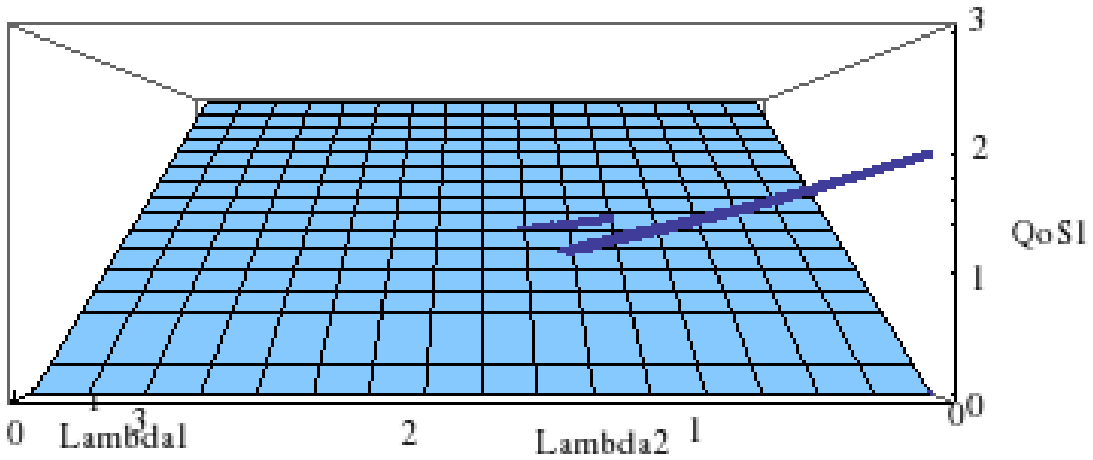
Equilibrium Sets

- The existence of an equilibrium points in the previous cases is guaranteed by the presence of a continuous and compact supports.
- The existence of an equilibrium point is seen when the Planes at 45 degrees to the x axis and y axis intersect the respective QoS curves.
- The behaviour of the system in the condition when the planes pass through the discontinuities is to be studied.
- The same problem instance as seen in the Single O-D pair with 2 parameters is studied.
- In this case where the planes pass through the discontinuity is when the change of support happens at $\lambda_1 = 1.33$.
- The system behaviour for this setting is studied. It is seen that the planes intersect both curves while passing through the discontinuities.

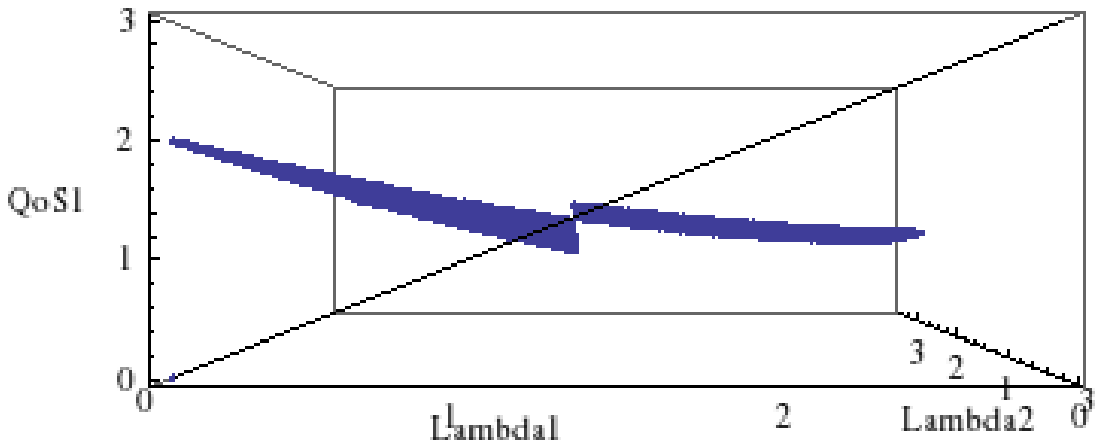
Top View QoS_1



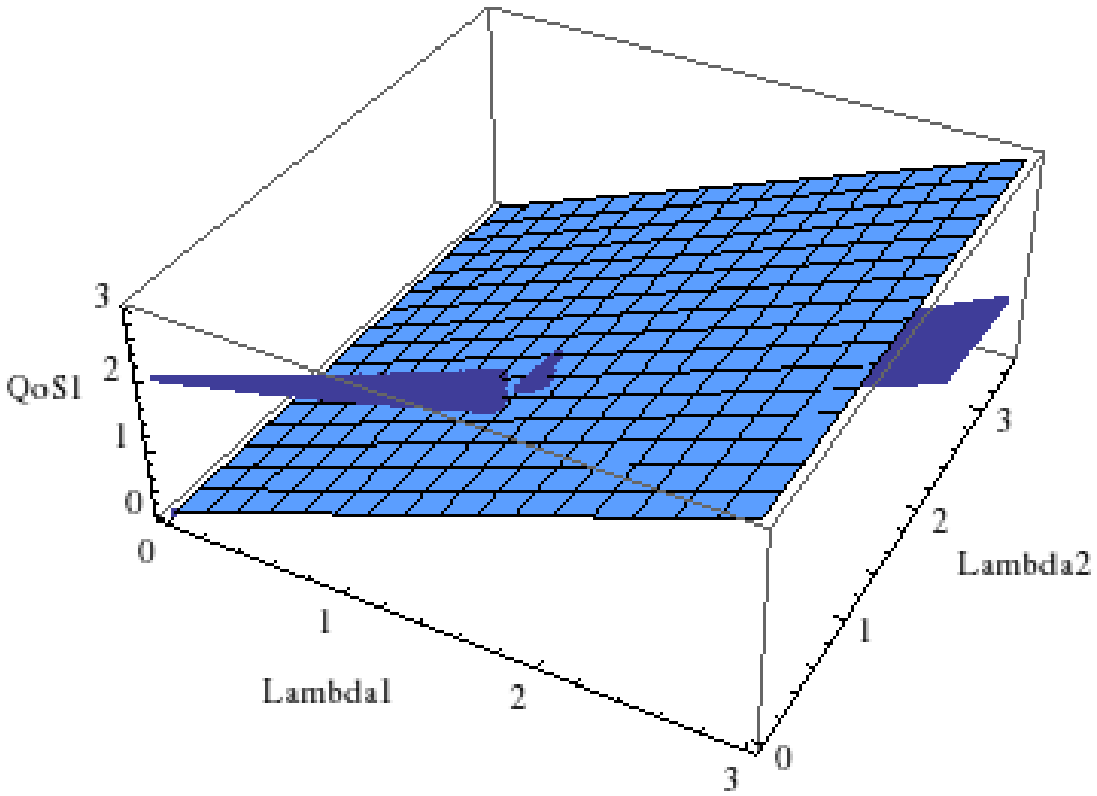
Side View QoS_1



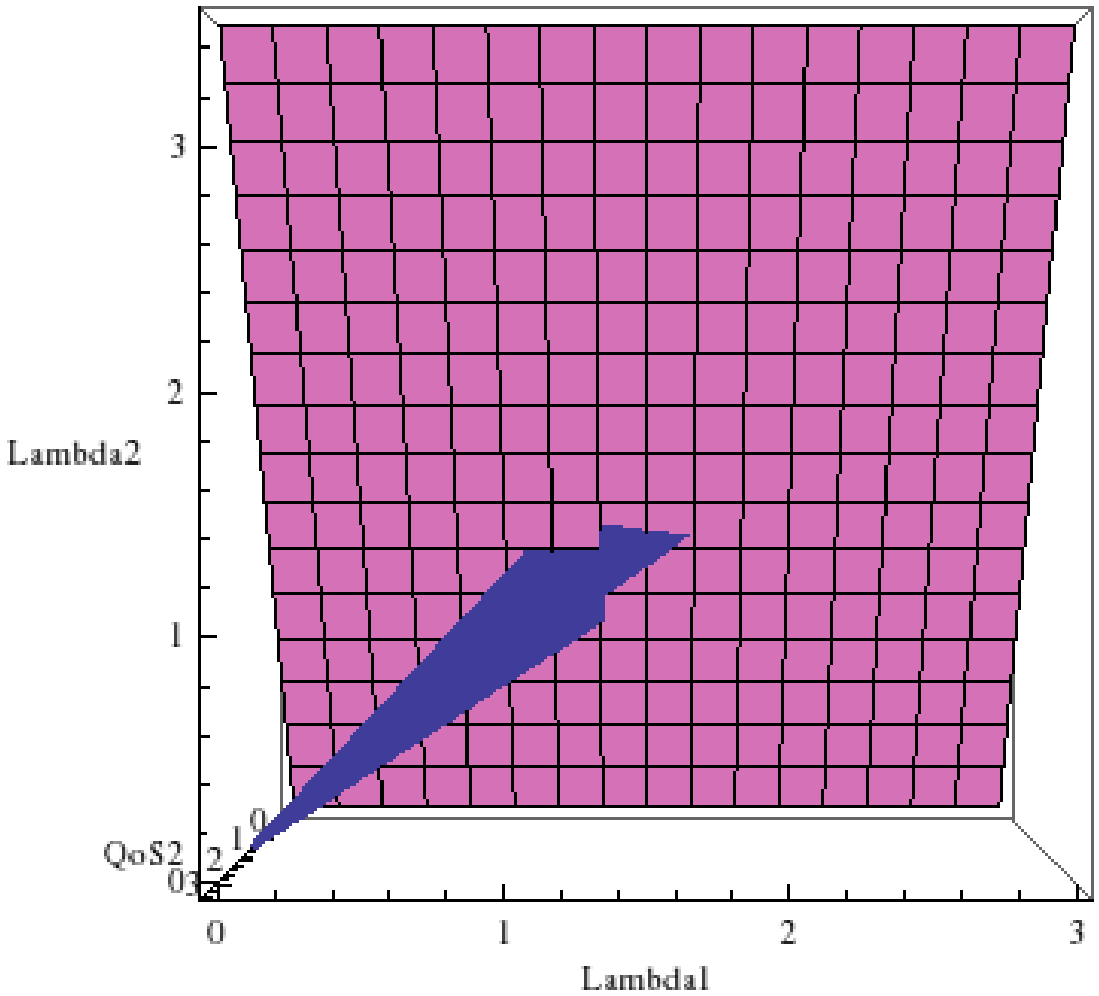
Front View QoS_1



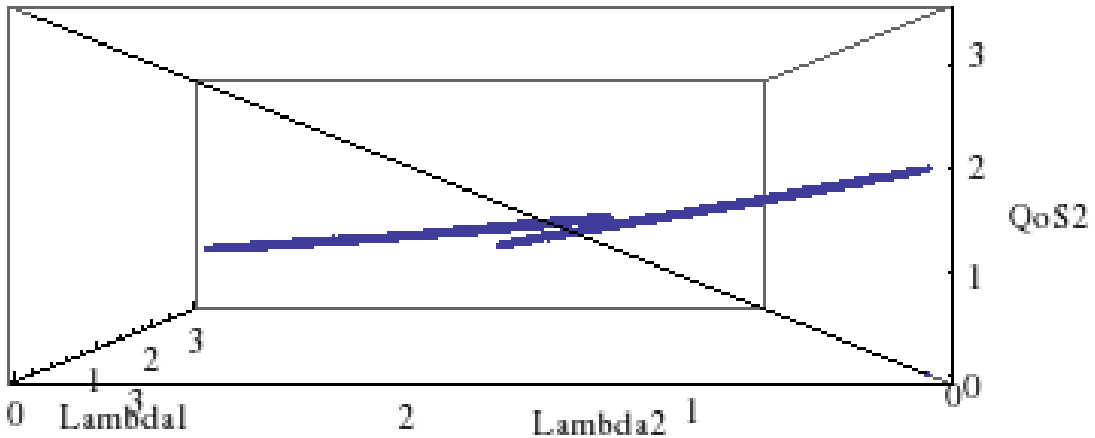
Angular View QoS_1



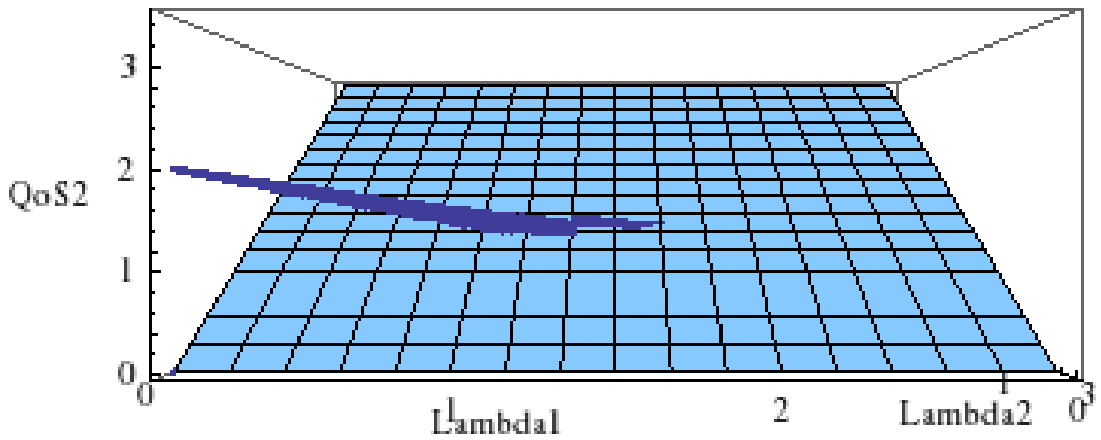
Top View QoS_2



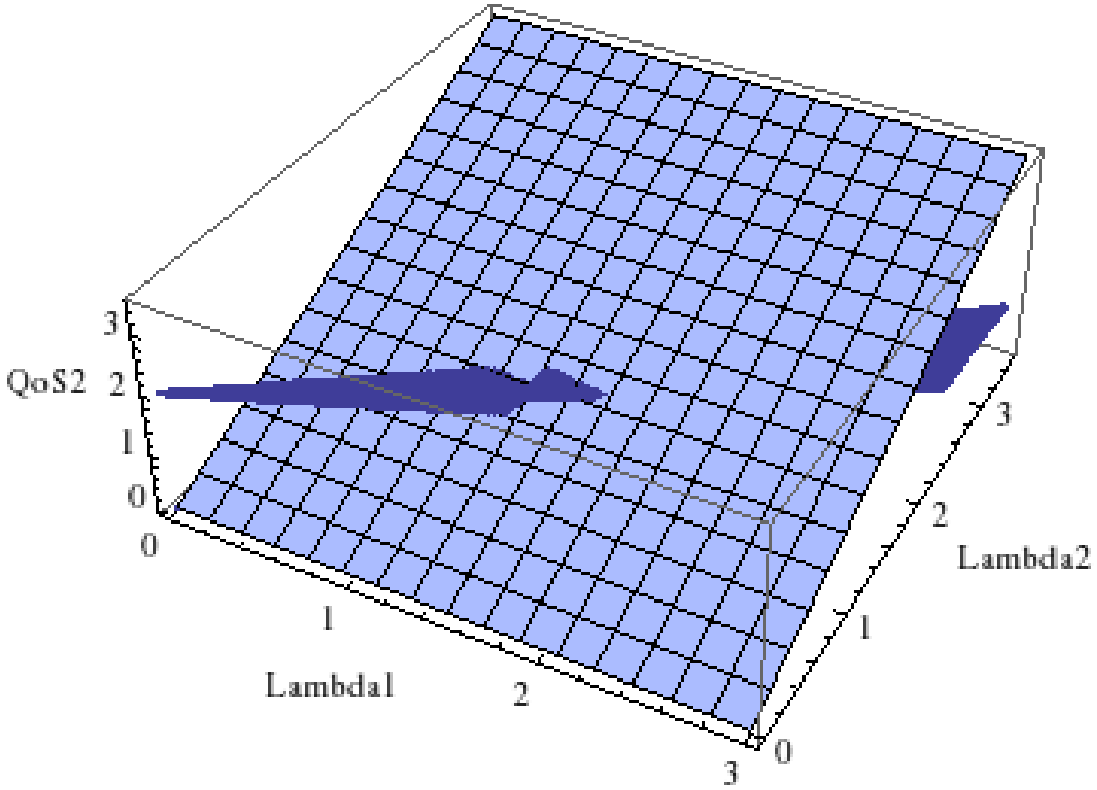
Side View QoS_2



Front View QoS_2



Angular View QoS_2



Since the equations of the curves are the same, we get the following points as equilibrium points

$$\lambda_{1eq} = 1.134 \quad \lambda_{2eq} = 1.3$$

and

$$\lambda_{1eq} = 1.358 \quad \lambda_{2eq} = 1.409$$

Starting at a point we determine the next point by $\lambda_{n+1} = f(\lambda_n)$.

Convergence to Equilibrium $(\lambda_1, \lambda_2) = (1.134, 1.3)$

Sl. No. (i)	λ_{1i}	λ_{2i}
0	3	2.7
1	1.06752	1.0304
2	1.28106	1.42446
3	1.07304	1.23291
4	1.16926	1.33562
5	1.11569	1.28117
6	1.14368	1.31017
7	1.12865	1.29474
8	1.13661	1.30295
9	1.13237	1.29858
10	1.13462	1.30091
11	1.13342	1.29967
12	1.13406	1.30033
13	1.13372	1.29998
14	1.1339	1.30017
15	1.13381	1.30007
16	1.13386	1.30012
17	1.13383	1.30009

Convergence to Equilibrium $(\lambda_1, \lambda_2) = (1.358, 1.409)$

Sl. No. (i)	λ_{1i}	λ_{2i}
0	1.2	0.9
1	1.34667	1.46
2	1.34383	1.40122
3	1.36122	1.4133
4	1.35674	1.40816
5	1.35848	1.40988
6	1.35789	1.40926
7	1.3581	1.40948
8	1.35803	1.4094
9	1.35805	1.40943
10	1.35804	1.40942
11	1.35805	1.40942
12	1.35805	1.40942

Observations

- In order to obtain equilibrium sets, we need points that pass through the discontinuities without intersecting with any of the curves.
- We see that as in the single O-D pair case where the QoS curve increase after the discontinuity. There will be two equilibrium points.
- We will now go on to study a case where the equilibrium drops after the discontinuity in order to understand the system behaviour.

Existence of Equilibrium Sets with New QoS Measures

$$QoS_1 = \sum_i \frac{x_i}{d_i} \tag{24}$$

and

$$QoS_2 = \sum_i \frac{\text{price paid}}{\text{maximum price payable}} \tag{25}$$

$$f_1(QoS_1) = 1 + QoS_1 \tag{26}$$

$$f_2(QoS_1) = 1 + QoS_2 \tag{27}$$

The following QoS curves are obtained in such a setting

$$f_1(QoS_1) = 0.0162x^2 - 0.185xy + 0.1667x + 0.037y^2 + 0.6667y + 1 \quad (28)$$

$$f_2(QoS_2) = 0.138889xy + 0.25x + 0.111111y^2 + 0.333333y + 1 \quad (29)$$

The equations of the QoS curves for higher supports are

$$f_1(QoS_1) = 0.173333x - 0.0906667yx + 0.00311111x^2 + 0.00533333y^2 + 0.4y + 1. \quad (30)$$

$$f_1(QoS_2) = 0.3x - 0.0866667xy - 0.00666667x^2 + 0.04y^2 + 0.2y + 1. \quad (31)$$

Equilibrium sets

Sl. No. (i)	λ_{1i}	λ_{2i}
0	3	2.8
1	1.94821	1.9856
2	2.1396	2.04971
3	1.82977	1.80927
4	2.07357	1.96445
5	1.80983	1.78763
6	2.06569	1.95405
7	1.80734	1.78498
8	2.06473	1.95278
9	1.80703	1.78465
10	2.06461	1.95262
11	1.807	1.78461
12	2.06459	1.9526
13	1.80699	1.78461
14	2.06459	1.9526

Thus, the existence of equilibrium sets is seen when the planes pass through the discontinuity completely.

Effect of Incentive Compatibility on Equilibrium Point to Equilibrium Set

Considering the case when the players are not incentive compatible and for the same functional equation seen previously, we get

$$f_1(QoS_1) = \frac{61}{288.0}\lambda_1^2 + \frac{1}{9}\lambda_1 - \frac{59.0}{108}\lambda_1\lambda_2 + \frac{2}{2}\lambda_2 + \frac{11}{54}\lambda_2^2 + 1 \quad (32)$$

$$f_1(QoS_2) = \frac{97}{288}\lambda_1^2 + \frac{1}{6}\lambda_1 - \frac{7}{9}\lambda_1\lambda_2 + \frac{1}{3}\lambda_2 + \frac{7}{18}\lambda_2^2 + 1 \quad (33)$$

The equations of QoS curves observed after the change of supports are

$$\tilde{f}_1(QoS_1) = \frac{2191.0}{27000}\lambda_1^2 - \frac{519.0}{2250}\lambda_1\lambda_2 + \frac{340.0}{2250}\lambda_1 + \frac{11.0}{150}\lambda_2^2 + 0.4\lambda_2 + 1 \quad (34)$$

$$\tilde{f}_2(QoS_2) = \frac{217.0}{1800}\lambda_1^2 - \frac{24.0}{75}\lambda_1\lambda_2 + \frac{20.0}{75}\lambda_1 + \frac{7.0}{50}\lambda_2^2 + \frac{1}{5}\lambda_2 + 1 \quad (35)$$

- Consider the situation where the change in support takes places at $\lambda_1 = 1.73$.
- For the case where all customers are incentive compatible the equilibrium obtained is at the point $(1.73847, 1.70105)$.
- Under the same conditions, when we determine equilibrium in the setting where the top customer is not incentive compatible the equilibrium obtained is a set $(1.72586, 1.65652)$ and $(1.92415, 1.68655)$ with the points toggling alternatively between each other.

Thus, it is seen that the non incentive compatibility of the customer shifts the equilibrium from a fixed point to an equilibrium set.

Effect of Incentive Compatibility on Equilibrium Set

- The change of support considered is at $\lambda_1 = 1.75$.
- Upon determining the equilibrium for the following system, an equilibrium set having a cycle is obtained which toggles between the following points in sequence.

$$(2.03482, 1.91560), (1.79798, 1.77492),$$

$$(1.75914, 1.72226), (1.74458, 1.70764)$$

- We now determine the equilibrium under the condition that the to customer is not incentive compatible with the same change of support.
- We again obtain an equilibrium set, but of two points that toggle alternatively. The points are $(1.72586, 1.65652)$ and $(1.92415, 1.68655)$.

Equilibrium Sets and Business Cycles

If we consider the toggling between two equilibrium points as a phase in the business cycle, we can interpret following :

- The change in the equilibrium sets is a shortening of the business cycle toggling between a phase of high demand to a period of low phase.
- This above happens in contrast to a cycle which has 4 phases of slowly decreasing demand which suddenly jumps to high demand at the end of 4 phases.

Computational issues in single O-D pair networks

- A simple iteration of λ_1 from 0 to 3 with step size of 0.01, for 20 values of λ_2 for each λ_1 run for 1000 iterations each, would require $300 \times 20 \times 1000$ runs of the optimization model, which is $60,00,000$ times.
- Hence, we need to exploit the problem structure.
- We address the above problem by pre solving for all the possible bid combinations and determine the allocations before hand.
- After that, the QoS of the system is computed at that point and stored in a list along with the bid combination.

Advantages of presolving in single O-D pair networks

- The first and the most important one is that it drastically reduces the time to run the simulation. In the single O-D pair case, the total number of optimization problems we have to run is 11^2
- Now within the Monte Carlo simulation , upon generating a random bid, we need to find its corresponding QoS values from the list created during pre-solving.
- Secondly, the pre-solving helps with determining the equations for the QoS curves as it exhaustively searches all possible bid combinations.
- Finally, it avoids discontinuities caused by a linear program having multiple optimal solutions.

Computational issues in large networks

- The importance of pre solving becomes all the more evident for large networks with multiple users. However, it does not completely solve the problem.
- The optimization problem takes longer to solve than previously with the single O-D pair case.
- The number of pre-solving examples are very large. The total number of times we have to pre-solve is $17,71,561$ which is still smaller than $60,00,000$ but large in itself.
- Searching through the list becomes extremely slow.
- The output file from pre-solving was 190 MB which makes it hard to load the data into Python.

Some possible solutions

- Better search by systematic presolving.
- Storing values in separate files is possible with systematic presolving.

Conclusions

- Change from the NUM framework to NMD to stop people from gaming the system.
- The applicability of linear programming relaxations depends only on the total unimodularity of the route link incidence matrix.
- Existence of Equilibrium in single parameter systems.
- Existence of Equilibrium in two parameter systems.
- Showing that Large Networks will operate at highest demand and price.
- Due to incentive compatibility we will always underestimate actual demand.
- Computational issues can be reduced through systematic presolving.

Interesting research avenues

- Developing a mechanism where all customers are incentive compatible.
- Develop a computationally efficient algorithm to determine strategic bids of top customers, instead of enumeration. This will help in studying larger and more complex networks.
- Equilibrium aspects in dynamic interaction of networks.

1. Frank P Kelly, Aman K Maulloo, and David KH Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research society, pages 237-252, 1998.
2. Frank Kelly and Elena Yudovina. Stochastic networks. Cambridge University Press, 2014.
3. G Tychogiorgos and KK Leung. Optimization-based resource allocation in communication networks. Computer Networks, 66:32-45, 2014.
4. N Hemachandra, K S N Rajesh, and Mohd. A Qavi. A model for equilibrium in some service-provider user-set interactions. Annals of Operations Research. 1572-9338, 1-21, 2015.
5. N Hemachandra, S Tripathi, and K Patil. Equilibrium sets in some $GI/M/1$ queues. Working paper.
6. Rahul Jain. Network market design part I: bandwidth markets. IEEE Communications Magazine, 50(11):78-83, 2012.
7. S K Sinha, N Rangaraj, and N Hemachandra. Pricing surplus server capacity for mean waiting time sensitive customers. European Journal of Operations Research, 205, 159-171, 2010.