Enabling State Dependent Priority Service By Using Pricing Mechanisms That Encourage Users To Jump The Queue

Avi Giloni
Sy Syms School of Business, Yeshiva University

Phil Troy
Les Entreprises TROYWARE

Background

- Many many years ago
  - When organizations charged internal users for computing
  - There was a lot of interest in pricing mechanisms
  - To ensure that computing resources were effectively used
- Then personal computers came along
  - Everyone got their own
  - The idea of charging for computing lost its luster

Background

- But, organizations still provided lots of services, e.g. I.T.
  - Of a different nature
  - Usually involving support, rather than computing resources
  - And different users got different benefits from that support
  - And incurred different waiting costs while waiting for that support
- And thus there was still a need for controlling:
  - Who would get these services
  - In what order they would get them
- Because these resources are still expensive

Background

- Thus we consider organizational service facilities where:
  - Internal users bring jobs
  - Each group of users receives a specific gross benefit when their job is completed
  - Each group of users incurs waiting costs in a specific way

Our Goal

- Find a pricing mechanism that can be used to control who gets served and in what order
- Ideally, we would like that control mechanism to work for:
  - Complex processes
  - Arbitrary inter-arrival time distributions
  - Arbitrary processing time distributions

Related Research

- Naor (1969) The Regulation of Queue Size by Levying Tolls.
- Mendelson (1985) Pricing Computer Services: Queueing Effects.
- Mendelson & Whang (1990) Optimal Incentive-Compatible Pricing for the M/M/1 Queue.
- Afeche and Mendelson (2004) Pricing and Priority Auctions in Queueing Systems with a Generalized Delay Cost Structure.

Ideal Approach

- Combine
  - State dependent pricing - a pricing mechanism that changes as the number of customers or jobs being served or waiting for service, changes
- With
  - Multiple waiting lines (queues) each having a different priority

State Dependent Pricing

- Why we shouldn't use state dependent pricing:
  - It makes life difficult for users?
  - Prices will be harder to compute?
  - It just seems too complex?

State Dependent Pricing

- Why we should use state dependent pricing:
  - Individually optimal behavior may not be “socially” optimal
  - Prices can be used to align individual behavior to be “socially” optimal
  - Prices can vary with the number of jobs in the facility
  - Low prices can be used to encourage users to submit work when the facility is not busy
  - High prices can be used to discourage users from submitting work when the facility is busy

State Dependent Pricing

- To develop a preliminary model for state dependent pricing we assumed that:
  - User interarrival times are exponentially distributed
  - User service times are exponentially distributed
  - Users incur waiting costs at a non-negative rate

State Dependent Pricing For Facilities With First Come First Served Service

- We define:
  - K - the number of user groups
  - k - the user group number
  - λ_k - the arrival rate of group k users
  - b_k - the gross benefit that group k users receive for job processing
  - $w_{i,k}$- the expected waiting cost that group k users will incur if their job is accepted when there are already i users in the facility

State Dependent Pricing For Facilities With First Come First Served Service

- We also define:
  - $\beta_{i,k}$ - the expected net benefit that group k users will receive if their job is accepted when there are already i users in the facility
  - $\xi_{i,k}$ - the fraction of group k users whose jobs are accepted when there are already i users in the facility [0,1]
  - I - the maximum number of users allowed in the facility
  - μ - the rate at which a server processes jobs
  - μ_i - the rate at which the facility processes jobs when there are i users in the facility

State Dependent Pricing For Facilities With First Come First Served Service

- We observe that when there are i users already in the facility:
  - The expected user admission rate is $\Sigma_k \lambda_k \xi_{i,k}$
  - The expected rate of net benefit is $\Sigma_k \lambda_k \xi_{i,k} \ \beta_{i,k}$

State Dependent Pricing For Facilities With First Come First Served Service

- We also observe that the problem of determining the optimal prices can be formulated and solved
  - As a non-discounted continuous time policy iteration problem
  - With the following value determination equation
    $\gamma = \Sigma_k \lambda_k \xi_{i,k} (\beta_{i,k} - \blacktriangledown v_i) + \mu_i \blacktriangledown v_{i-1}$
  - Where:
    - γ - is the rate at which the facility generates net benefit
    - v_i - is the relative value of there being i jobs in the facility
    - $\blacktriangledown v_i$- is the opportunity cost of admitting a user when there are already i customers in the facility, i.e. $v_i - v_{i+1}$

State Dependent Pricing For Facilities With First Come First Served Service

- When formulated this way, the optimal policy
  - Explicitly specifies the state dependent opportunity costs
  - Admits users whose net benefit is greater than or equal to these opportunity costs
  - Can be implemented by charging tolls equal to these state dependent opportunity costs
  - Will tend to keep the facility busy
  - Will tend to keep the queue from becoming large
  - Can easily be computed via policy iteration since there are only I tri-diagonal value determination equations

State Dependent Pricing For Facilities With First Come First Served Service

- Unfortunately, this may not work so well when:
  - There are two groups of users
  - Group 1 has high gross benefits and waiting costs and arrives at a low rate
  - Group 2 has low gross benefits and waiting costs and arrives at a high rate
- This can result in 2 possibilities:
  - We charge higher tolls and preclude group 2 users
  - We charge lower tolls and get less net benefit from group 1 users
- What we would ideally like is to process group 1 users before group 2 users
- This suggests the use of . . .

Priority Queues

- The idea behind priority queues is that:
  - There can be several queues
  - Jobs in lower numbered queues are processed in first come first served order before jobs in higher numbered queues
  - Users that incur waiting costs at the highest rate wait the shortest amount of time
- (Note that we are not considering the use of pre-emption)

Priority Queues & State Dependent Pricing

- For reference purposes only, the new notation is:
  - **i** - a vector containing the number of customers in each queue and the number of customers currently being served
  - When the facility was in state **i**
    - a(**i**,q) - the state after a customer is accepted to queue q
    - d(**i**) - the state after serving a user is completed
    - $\beta_{\mathbf{i},k,q}$ - the net benefit of group k users accepted to queue q
    - $\xi_{\mathbf{i},k,q}$ - the fraction of group k users accepted to queue q

Priority Queues & State Dependent Pricing

- For reference purposes only, the new value determination equations are:
  - $\gamma = \Sigma_k \lambda_k \xi_{\mathbf{i},k,q} (\beta_{i,k,q} - \blacktriangledown v_{i,\mathbf{a}(\mathbf{i},q)}) + \mu_i \blacktriangledown v_{i,\mathbf{d}(\mathbf{i})}$
  - Where:
    - γ - is the rate at which the facility generates net benefit
    - $v_{\mathbf{i}}$ - is the relative value of the facility being in state **i**
    - $\blacktriangledown v_{i1,i2}$ - is the opportunity cost of making a transition from state **i1** to state **i2**, i.e. $v_{\mathbf{i1}} - v_{\mathbf{i2}}$

Priority Queues & State Dependent Pricing

- The optimal policy when adding priority queues:
  - Explicitly determines the state dependent opportunity costs
  - Admits customers to the queue that maximizes the positive difference between their net benefit and these opportunity costs
  - Can be implemented via tolls
  - Will likely have higher tolls for higher priority queues than for lower priority queues
  - Will tend to keep the facility busy

Priority Queues & State Dependent Pricing

- Limitation (1) - How to determine expected waiting costs
- This affects decision as to which queue to join
- Consider a situation in which:
  - A user with moderate waiting costs arrives to the facility
  - There are very few jobs in the first or second queue
  - If user joins first queue, most likely higher toll
  - If user joins second queue:
    - Lower tolls
    - Length of wait is a function of the queue subsequent users join, which is a function of which queue this user joins, . . .

Priority Queues & State Dependent Pricing

- Limitation (2) - Dealing with non-linear waiting costs
- Consider a situation in which:
  - The rate at which a user incurs waiting costs decreases in time
  - The user initially joins queue with highest priority
  - The rate at which the user incurs waiting cost decreases
  - The user should switch to a lower queue at this point in time
  - This mechanism does not allow for this

Priority Queues & State Dependent Pricing

- Limitation (3) - Variability in actual net benefits
- In first come first served policy, net benefits variability:
  - User's service time
  - Prior customers' service times
- In priority queue policy, net benefits variability for users in secondary queues:
  - User's service time
  - Prior users' service times
  - Service times of subsequent users that join higher priority queues

Priority Queues & State Dependent Pricing

- Limitation (4) - How solve policy iteration equations
- First come first serve policy:
  - Number of equations O(I)
  - Tri-diagonal
- Priority queue policy:
  - Number of equations $O(I1 \cdot I2 \cdot I3 \cdots)$
  - No longer tri-diagonal

Allowing Users To Jump The Queue

- The Idea
  - Only have one queue
  - Allow users to move around within that queue
  - Users that benefit from move pay users that are disadvantaged by move
  - Treat non-linear waiting cost functions as piece-wise linear waiting cost functions

Allowing Users To Jump The Queue

- For reference purposes, new notation:
  - **i** - a vector containing the number of customers in the queue having each (piece-wise linear) waiting cost function, ordered from highest to lowest
  - When the facility is in state **i**
    - $a(\mathbf{i},k)$ - the state after a group k customer is accepted
    - $d(\mathbf{i})$ **-** the state after serving a customer is completed
    - $\beta_{\mathbf{i},k}$ - the net benefit of a group k customers is accepted, after compensating other customers for being moved
    - $\xi_{\mathbf{i},k}$ - the fraction of group k customers accepted

Allowing Users To Jump The Queue

- For reference purposes only, the new value determination equations are:
  - $\gamma = \Sigma_k \lambda_k \xi_{\mathbf{i},k} (\beta_{i,k} - \blacktriangledown v_{i,\mathbf{a}(\mathbf{i},k)}) + \mu_i \blacktriangledown v_{i,\mathbf{d}(\mathbf{i})}$
  - Where:
    - γ - is the rate at which the facility generates net benefit
    - $v_{\mathbf{i}}$ - is the relative value of the facility being in state **i**
    - $\blacktriangledown v_{i1,i2}$ - is the opportunity cost of making a transition from state **i1** to state **i2**, i.e. $v_{\mathbf{i1}} - v_{\mathbf{i2}}$

Allowing Users To Jump The Queue

- We observe that this pricing mechanism is very similar to the priority queue pricing mechanism in that it:
  - Explicitly determines the state dependent opportunity costs
  - Admits customers to the appropriate position in the queue if their expected net benefit exceeds these opportunity costs
  - Can be implemented via tolls
  - Will tend to keep the facility busy
  - Will allow accumulation of customers with lower waiting costs for processing when other customers are not in the facility

Allowing Users To Jump The Queue

- But:
  - This approach addresses the limitation (1) of determining expected net benefit
  - Because it can be done in same manner as for the first come first served policy

Allowing Users To Jump The Queue

- This approach also addresses limitation of not being able to handle non-linear waiting costs
  - When rate at which waiting cost changes, customers may jump the queue
  - Users with the highest waiting are always at the front of the queue

Allowing Users To Jump The Queue

- Furthermore, this approach addresses variability in net benefits
- Variability is minimized because customers compensate or are compensated for being moved

Allowing Users To Jump The Queue

- Finally, this approach partially addresses the computational complexity of solving policy iteration equations
- The complexity still exists, but:
  - The equations are fairly sparse
  - They appear amenable to value iteration
  - It seems likely that solution will be relatively insensitive
  - It might be possible to solve in a Just In Time manner
  - Initial values can be approximated

Our Goal

- Find a pricing mechanism that could be used to control who gets served and in what order
- Ideally, we wanted that pricing mechanism to work for:
  - Complex processes
  - Arbitrary processing time distributions
  - Complex processes

Our Results

- Found a plausible pricing mechanism for controlling who gets served and in what order

Need For Future Work

- Computation
- Extension to general service time distributions
- Extension to more complicated processes