Locating Mobile Servers under Stochastic Demands and Congestion

Oded Berman
Dmitry Krass
Seokjin Kim

Joseph L. Rotman School of Management
University of Toronto

Outline

- Problem Formulation
  - Assumptions & the Problem
- Overlapping Regions
  - Multi-class Multi-server Systems with Restricted Customer-Server Matchings
    - Caldentey & Kaplan (2002, Submitted to QS), Foley & McDonald (2001, AAP)
- 3 Location Models in the Literature
  - Feasible for the Problem?
    - ReVelle & Hogan (1989, TS), Ball & Lin (1993, OR), Marianov & ReVelle (1994, EJOR)
- 2 New Location Models
- Evaluation of Models & Conclusion

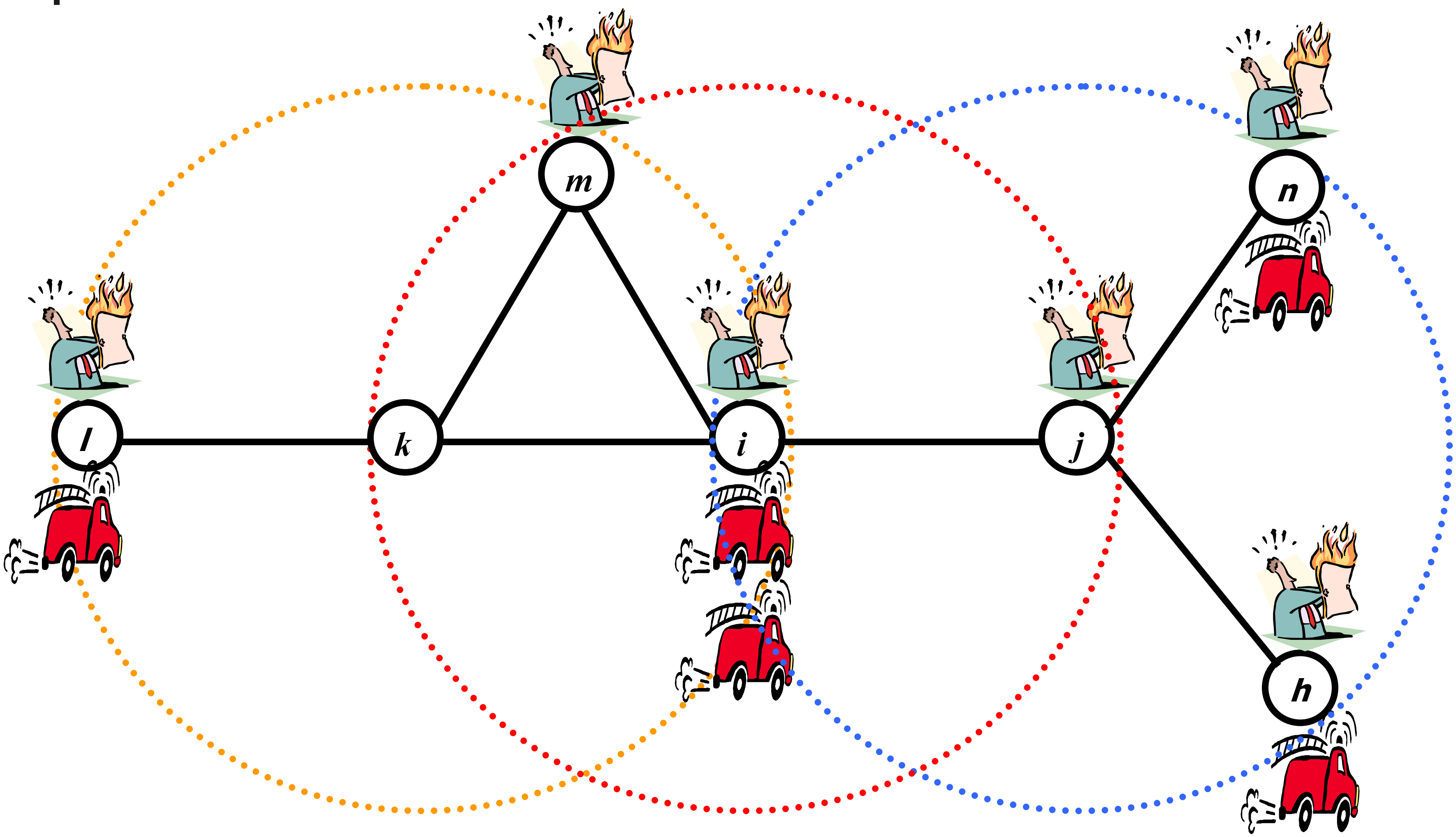
Problem Formulation[1]

- Assumptions
  - Discrete undirected network $G=(N,L)$
  - Poisson demands w/ λ_i at node $i,\ i \in N$
  - (Customer Assignment Policy) A customer is assigned to the closest available server within a pre-specified distance (radius). Or she is lost.
    - Ties are broken randomly
  - Each mobile server completes service in an exponential total service time w/ rate μ
    - total service time = travel time to node and back to home facility + on-scene service time
- Fact
  - Location exists at nodes.

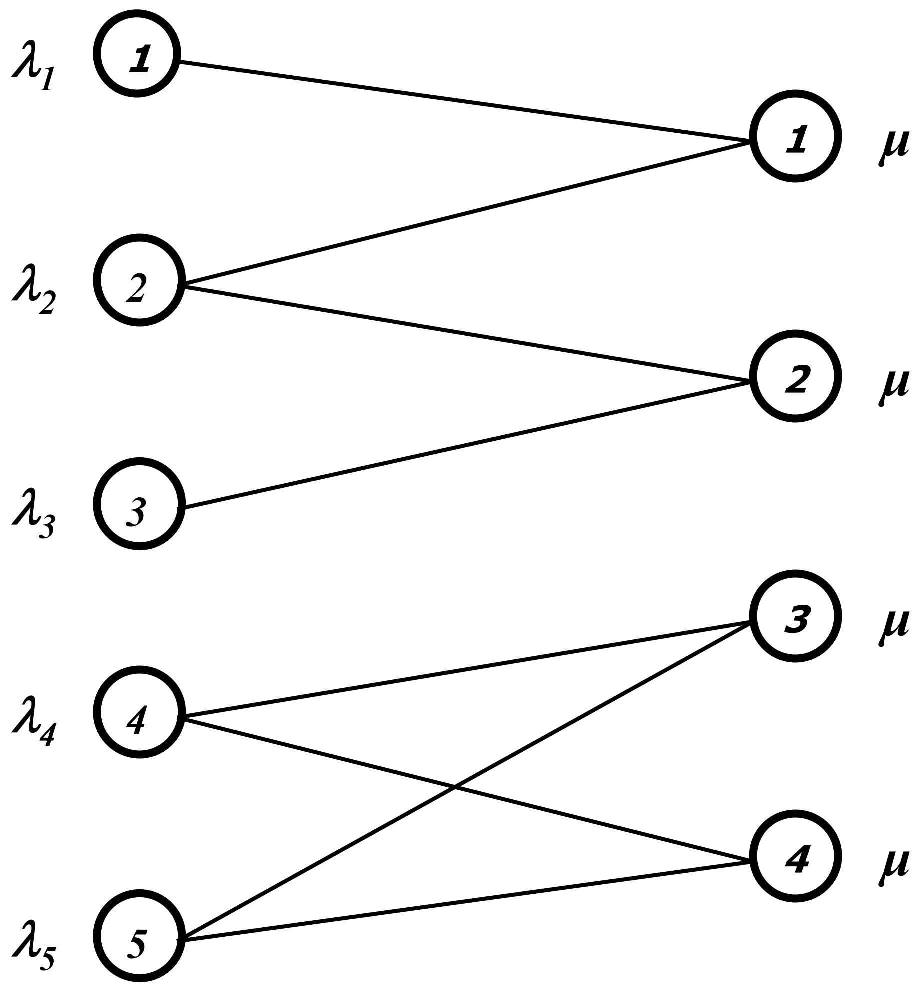
Problem Formulation[2]

- The Problem is
  - to find a location $x = (x_1, .., x_J)$ minimizing the number of mobile servers to be located on the site set X, while keeping the availability $A_i(x)$ of node i at least the required level α, $i \in N$
  - where $A_i(x) = Prob($a customer at node i finds at least one available server upon calling for service, for a given location $x)$.
- Background
  - Applied for locating emergency service vehicles
  - A required availability α is enforced.
  - Hard to find $A_i(x)$ analytically even for a given location x. Simulation is an alternative.

Overlapping Regions



Bipartite Graph Representation



Problem Formulation[3]

- Notations
  - N = Set of nodes
  - X = Set of facility sites
  - $x = (x_1, .., x_J)$ = Location vector (Location)
  - x_j = Number of servers at facility j
  - $F(x)$ = Set of facilities
  - N_p = Set of nodes within the radius of point $p \in G$
  - N_i = Region $i \in N$
  - X_p = Set of facility sites within the radius of point p
  - $F_p(x)$ = Set of facilities within the radius of point p
  - $k_i(x)$ = Number of servers located in X_i
  - $\lambda(S)$ = Rate at which demands originate from $S \subseteq N$
  - $B(S)$ = Union of all the nodes in N_n, $n \in S$
    = Neighborhood of S

Problem Formulation[4]

- Mathematical programming formulation

$$(P) \qquad \min \sum_{j \in X} x_j$$

$$s.t. \quad A_i(x) \geq \alpha, \quad i \in N$$

$$x_j = 0, 1, \ldots$$

  - What most researchers have done

$$st \quad \left(A_i(x) \approx \right) \ A_i \geq \alpha, \quad i \in N$$

  - What we are trying to do

$$st \quad \left(A_i(x) \geq \right) \ A_i \geq \alpha, \quad i \in N$$

Local Regions

- Region-i $\quad N_i,\ i \in N$
- Region-i demand rate $\quad \lambda(N_i) = \sum_{n \in N_i} \lambda_n$
- Neighborhood of region i $\quad B(N_i) = \bigcup_{n \in N_i} N_n$
- Given a location x, is an $M/M/k_i(x)/k_i(x)$ system with demand rate $\lambda(N_i)$ embedded in N_i?
  - Not necessarily.
  - Yes, if all the demands originating from N_i are served only by servers at node i. (only for complete bipartite graphs)

M/M/k/k Systems

- Availability of the *M/M/k/k* system with demand rate d

$$A(d,k) = \left[\sum_{n=0}^{k-1} \frac{\left(\frac{d}{\mu}\right)^n}{n!} \right] \left[\sum_{n=0}^{k} \frac{\left(\frac{d}{\mu}\right)^n}{n!} \right]^{-1}$$

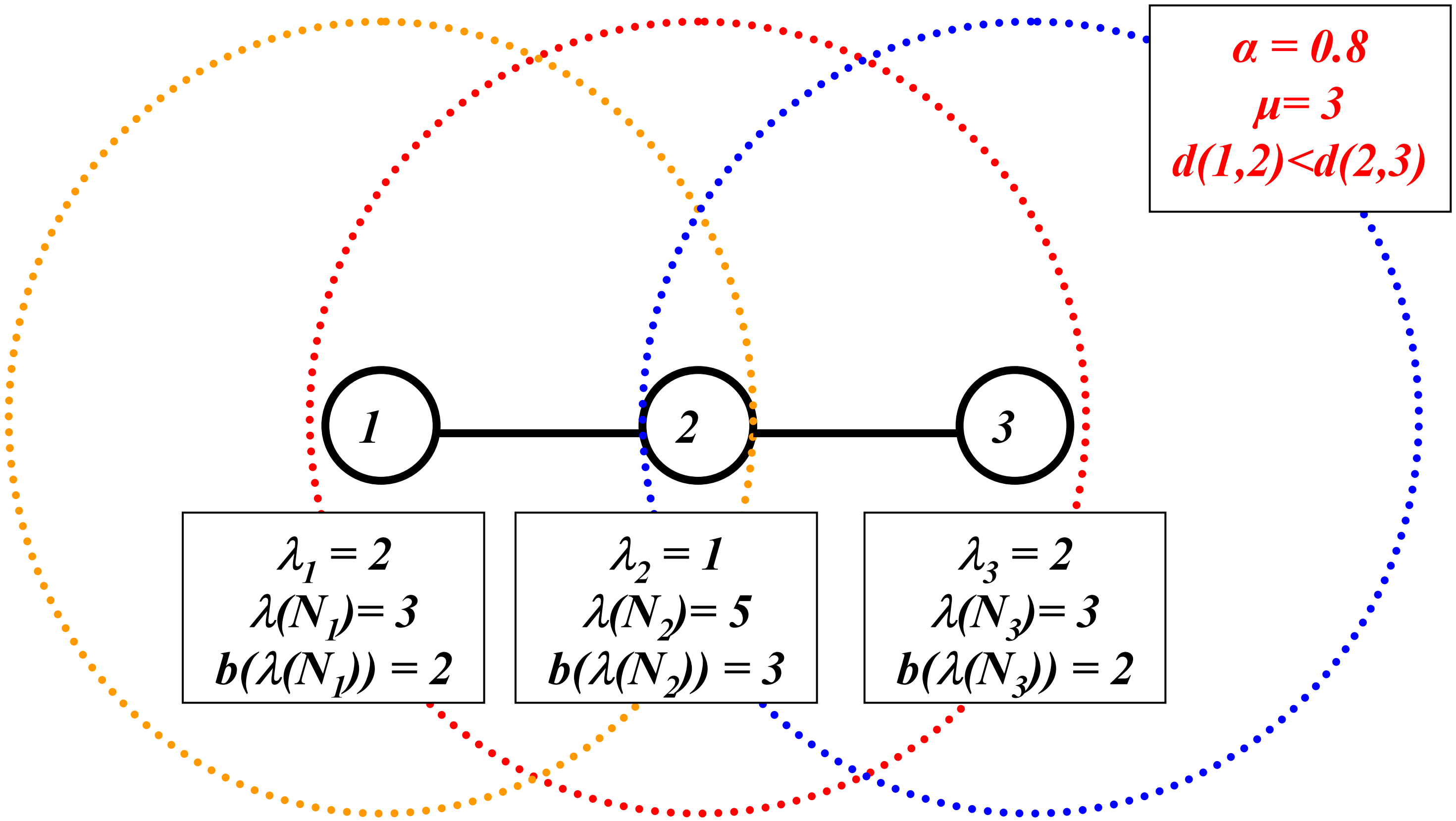
  - Decreasing in d; increasing in k

- Minimum number of servers for α–availability

$$b(d) = \min \{ k \geq 0 \mid A(d,k) \geq \alpha \}$$

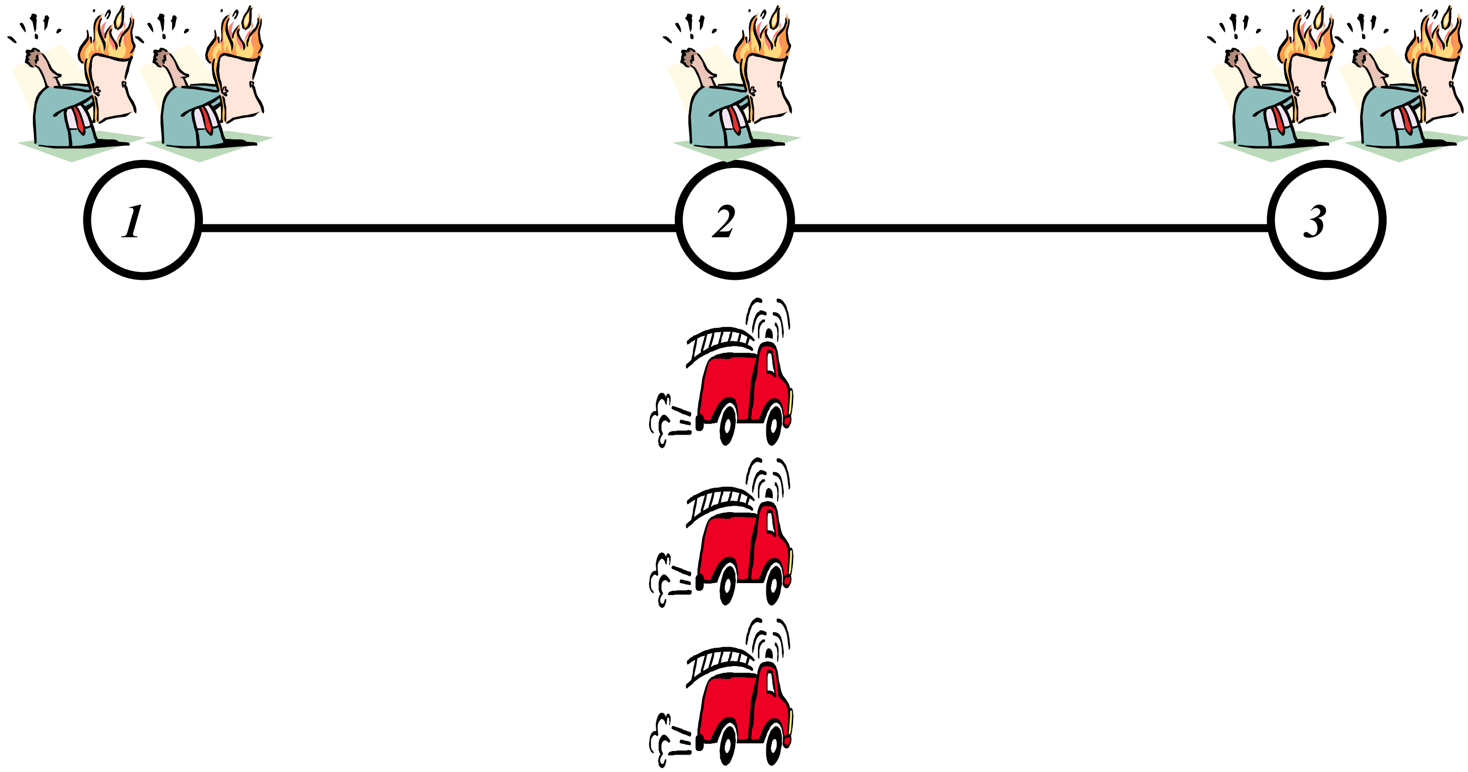
  - Non-decreasing in d; $A(d,b(d)) \geq \alpha$

Example: Path w/ Large Radius



- Centered location: $x^c = (x_1^c = 0,\ x_2^c = 3,\ x_3^c = 0)$
  - *M/M/3/3* with demand rate *5*

A Feasible Location for (P)



$$A_1(x^c) = A_2(x^c) = A_3(x^c) = 0.84 > \alpha = 0.8$$

ReVelle and Hogan's Model

- Region-specific approximations for availability
- Linear integer programming formulation

$$(RH) \qquad \min \sum_{j \in X} x_j$$
$$s.t. \sum_{j \in X_i} x_j \geq b_i, \;\; i \in N$$
$$x_j = 0,\, 1,\, \ldots$$

- where b_i is the smallest $k_i \geq 0$ s.t.

$$(A_i(x) \approx) \;\; A_i = 1 - \rho_i^{k_i} = 1 - \left(\frac{\lambda(N_i)}{k_i \mu}\right)^{k_i} \geq \alpha$$

Ball & Lin's Model

- Linear integer programming formulation

$$(BL)\quad \min \sum_{j\in X}\sum_{k=1}^{K} x_{jk} \qquad s.t.\quad x_{jk}=0,1$$

$$\sum_{j\in X_i}\sum_{k=1}^{K} -\log(P[D(j)\geq k])\,x_{jk} \geq -\log(1-\alpha),\quad i\in N$$

- Nonlinear availability constraints

$$1-A_i(x)\leq 1-A_i=\prod_{j\in X_i}\prod_{k=1}^{K} P[D(j)\geq k]^{x_{jk}}\leq 1-\alpha,\quad i\in N$$

  - $D(j)$ = Number of demands from N_j during $(0,U)$
    ~ Poisson with mean $U\lambda(N_j)$
  - U = Upper bound on exponential service times

Marianov & ReVelle's Model

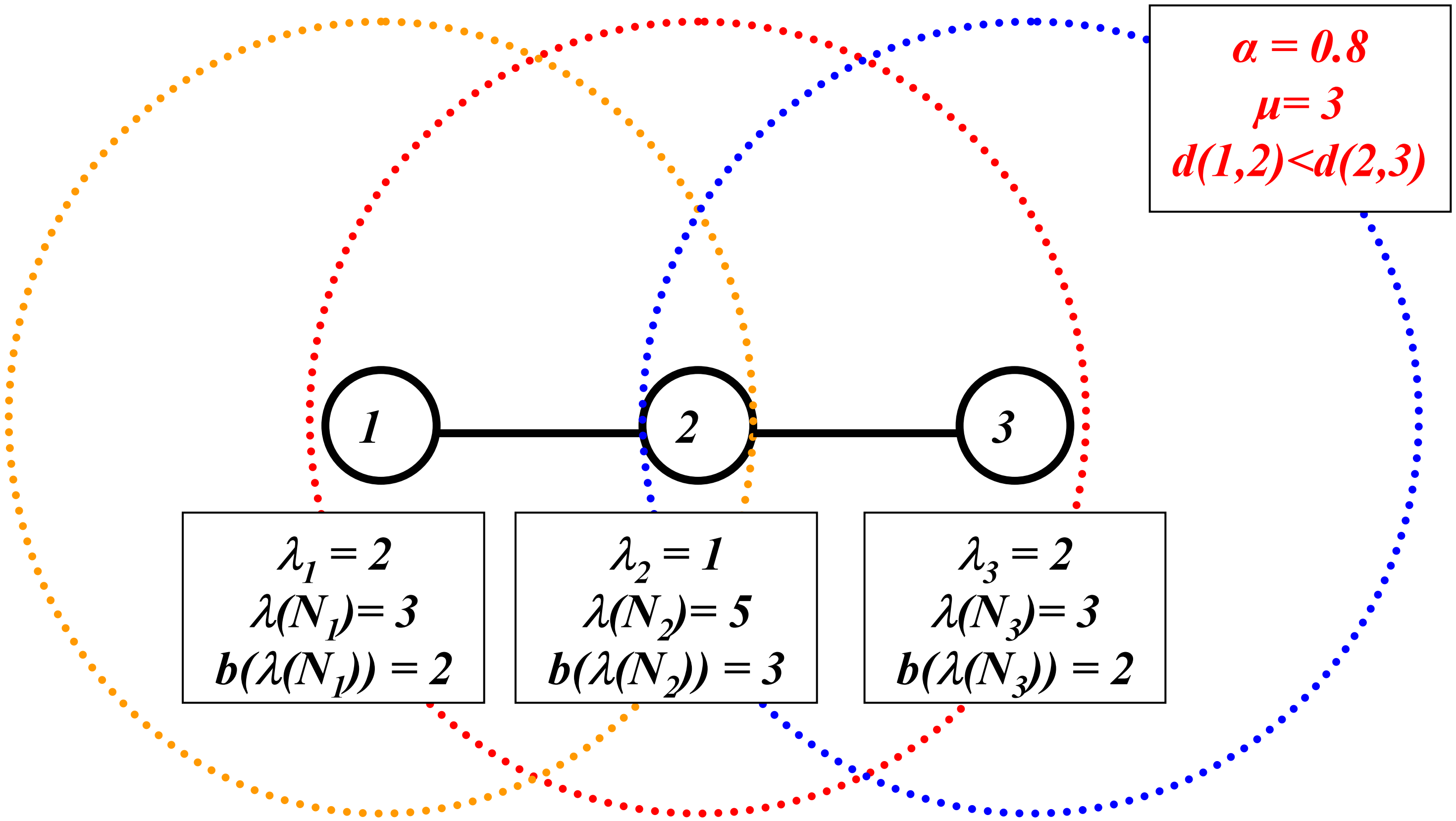
- Assumption:
  - For a given location x, N_i is an $M/M/k_i(x)/k_i(x)$ system with demand rate

$$\lambda(N_i) = \sum_{n \in N_i} \lambda_n$$

- Linear integer programming formulation

$$\begin{aligned} (MR) \qquad & \min \sum_{j \in X} x_j \\ & s.t. \sum_{j \in X_i} x_j \geq b(\lambda(N_i)), \ \ i \in N \\ & x_j = 0, 1, \ldots \end{aligned}$$

Example Revisited[1]



- Centered location: $x^c = (x_1^c = 0,\ x_2^c = 3,\ x_3^c = 0)$
  - *M/M/3/3* with demand rate *5*

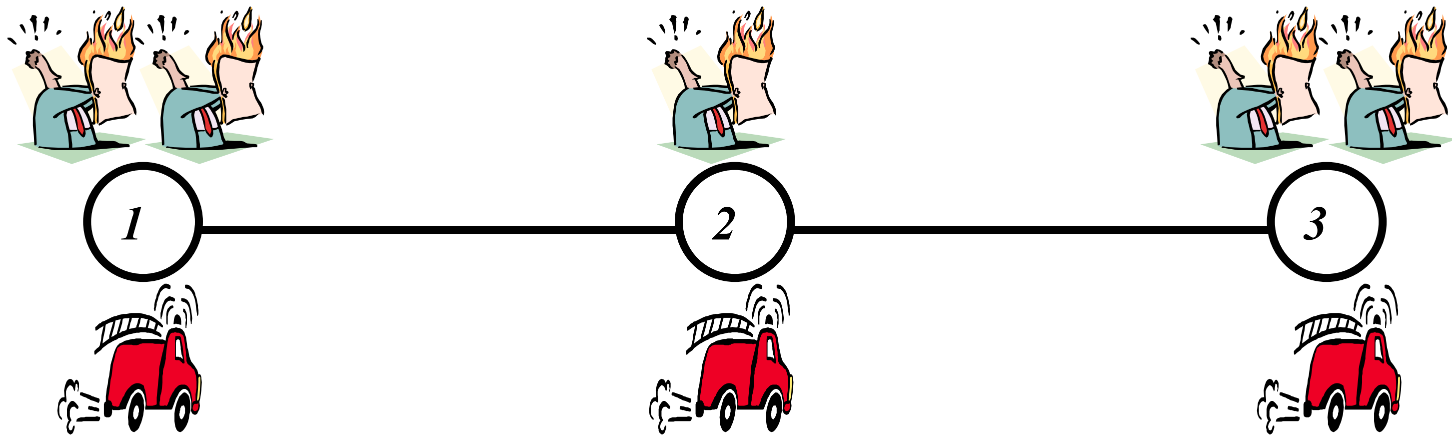
Example Revisited[2]

- Marianov & ReVelle's model

$$
\begin{aligned}
\min \quad & x_1 + x_2 + x_3 \\
s.t. \quad & x_1 + x_2 \geq 2 \\
& x_1 + x_2 + x_3 \geq 3 \\
& x_2 + x_3 \geq 2 \\
& x_1, x_2, x_3 = 0, 1, \ldots
\end{aligned}
$$

- Solution: $x^m = (x_1^m = 1,\ x_2^m = 1,\ x_3^m = 1)$
- Compare with: $x^c = (x_1^c = 0,\ x_2^c = 3,\ x_3^c = 0)$

An Infeasible Location for (P)



$$A_1(x^m) = 0.76 < \alpha = 0.8,$$

$$A_2(x^m) = 0.88 > \alpha = 0.8,$$

$$A_3(x^m) = 0.77 < \alpha = 0.8.$$

Model BKK1

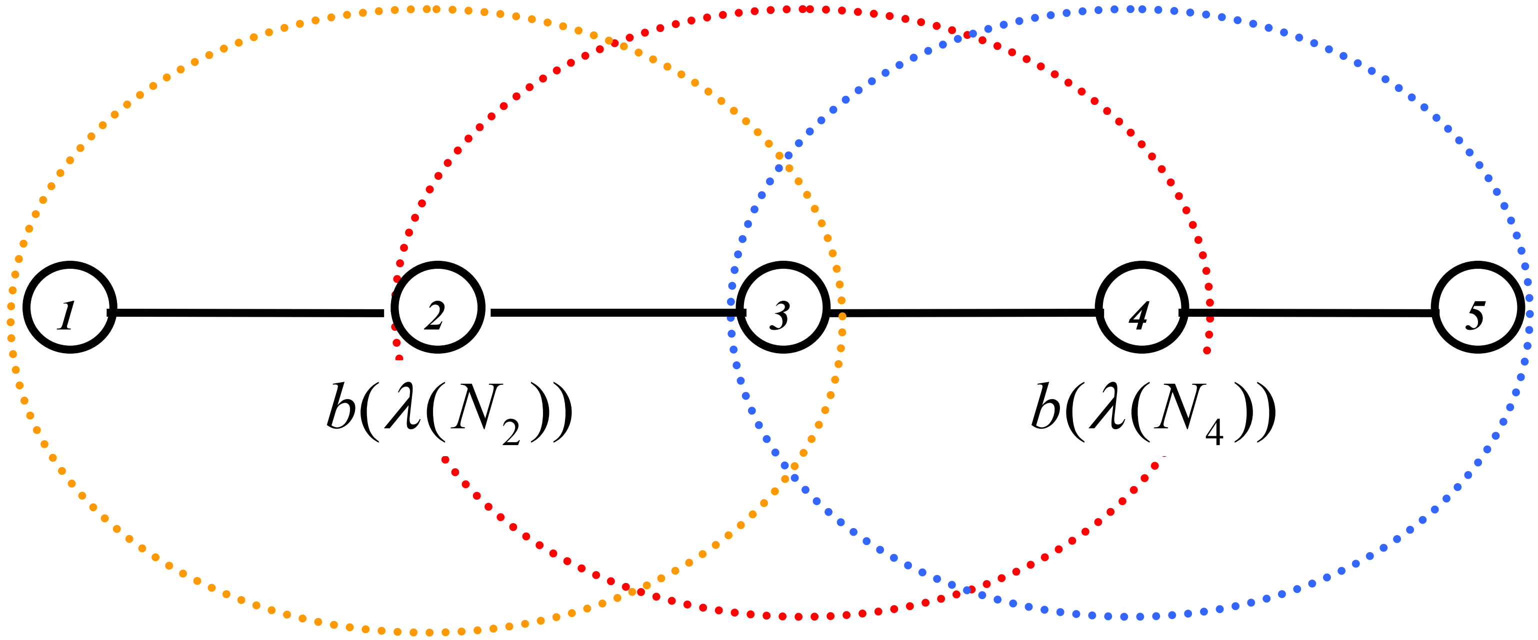
- Facility-specific lower bounds for availability
- Place “all or nothing” at facility j.
- Weighted set covering problem formulation

$$
\begin{aligned}
(BKK1) \qquad & \min \sum_{j \in X} b(\lambda(N_j)) y_j \\
& s.t. \sum_{j \in X_i} y_j \geq 1 \quad i \in N \\
& y_j = 0, 1 \quad j \in X
\end{aligned}
$$

- where $y_j = 1$, if a facility is located at facility j and $y_j = 0$, otherwise

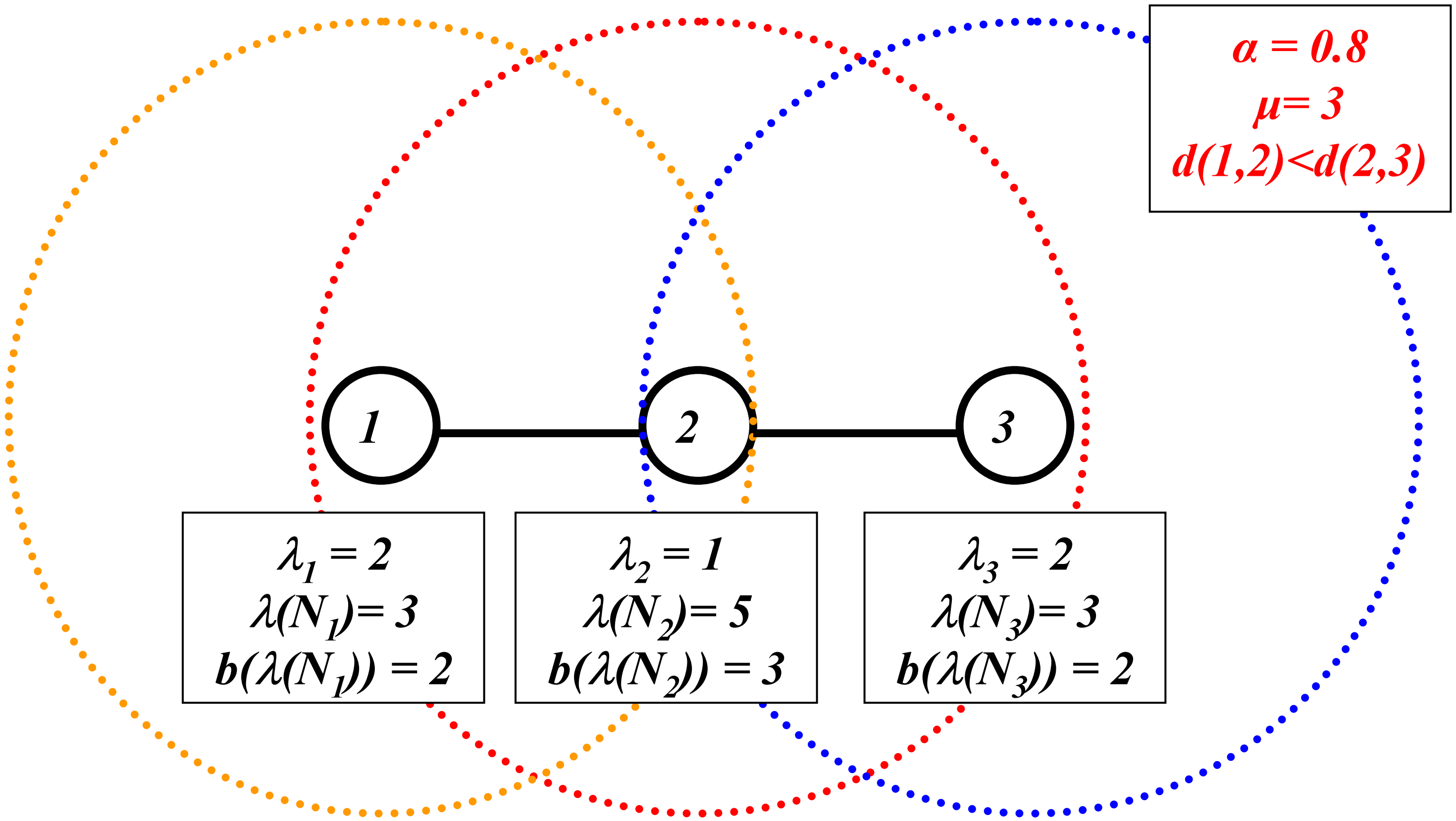
Why is BKK1 Feasible for (P)?

- Decoupling the underlying system into *M/M/k/k* systems



- Insensitive to customer assignment policies and queue capacities

Example Revisited[3]



- Centered location: $x^c = (x_1^c = 0,\ x_2^c = 3,\ x_3^c = 0)$
  - *M/M/3/3* with demand rate *5*

Example Revisited[4]

- BKK1 for the example

$$\begin{aligned} \min \quad & 2y_1 + 3y_2 + 2y_3 \\ s.t. \quad & y_1 + y_2 \geq 1 \\ & y_1 + y_2 + y_3 \geq 1 \\ & y_2 + y_3 \geq 1 \\ & y_1, y_2, y_3 = 0, 1 \end{aligned}$$

- Optimal solution: $y^b = (y_1^b = 0,\ y_2^b = 1,\ y_3^b = 0)$
- Server allocation: $x^b = (x_1^b = 0,\ x_2^b = 3,\ x_3^b = 0) = x^c$
- BKK1 is safe!

$$A_1(x^b) = A_2(x^b) = A_3(x^b) = 0.84 > \alpha = 0.8$$

Model BKK2[1]

- Node-specific lower bounds for availability
- Goal programming approach
- (1st Step) Solve weighted set covering problem for facility location

$$
\begin{aligned}
(WSCP) \qquad & \min \sum_{j \in X} y_j \, b\Big(\sum_{n \in N_j} \lambda(N_n)\Big) \\
& s.t. \sum_{j \in X_i} y_j \geq 1 \quad i \in N \\
& y_j = 0, 1 \quad j \in X
\end{aligned}
$$

- Let y^* be optimal for *(WSCP)*.

Model BKK2[2]

- (2nd Step) Solve integer program to allocate servers

$$
\begin{aligned}
(BKK\,2) \qquad & \min \sum_{j \in X} x_j \\
& s.t. \;\; x_j \le M y_j^* \qquad j \in X \\
& \sum_{j \in X_i} x_j \ge b(\lambda(B(F_i(y^*)))) \qquad i \in N \\
& x_j = 0,\, 1,\, \ldots \qquad j \in X
\end{aligned}
$$

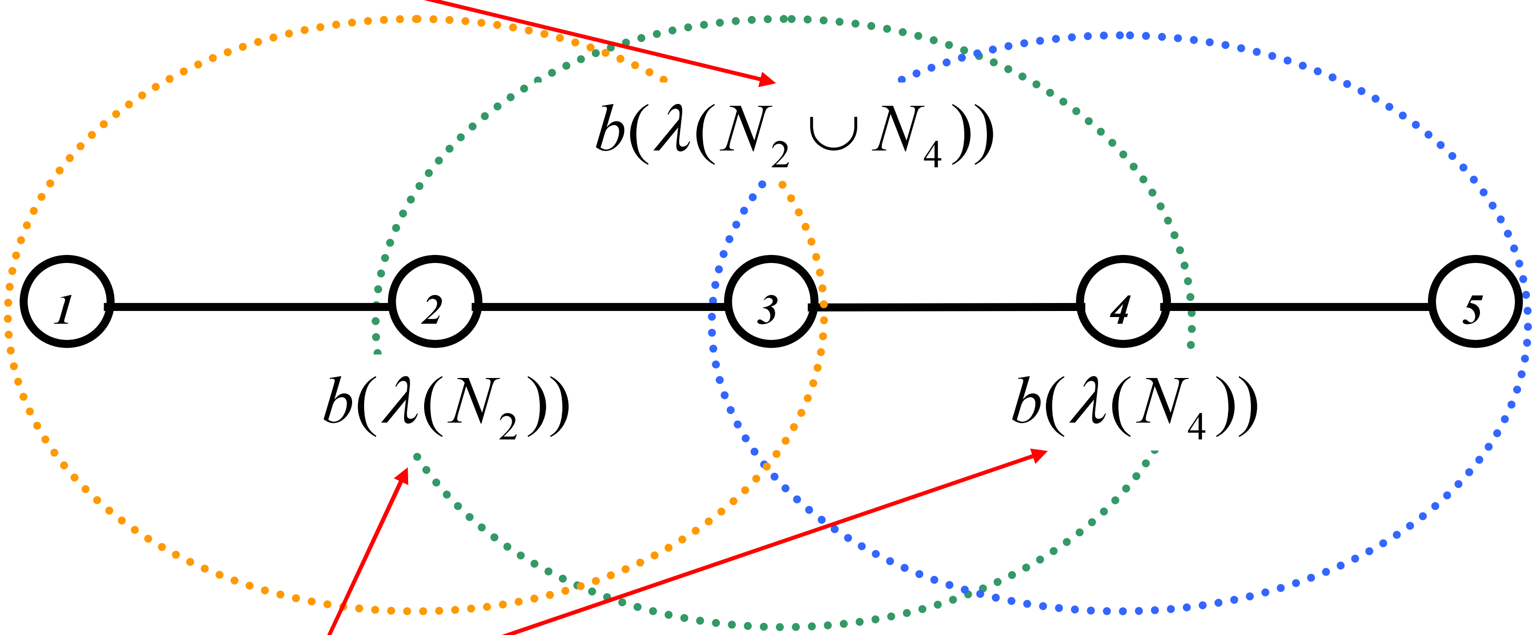
  - where M is a big number
- Fact
  - Extends readily to capacitated-facility cases

BKK1 and BKK2

- Optimal set covering solution

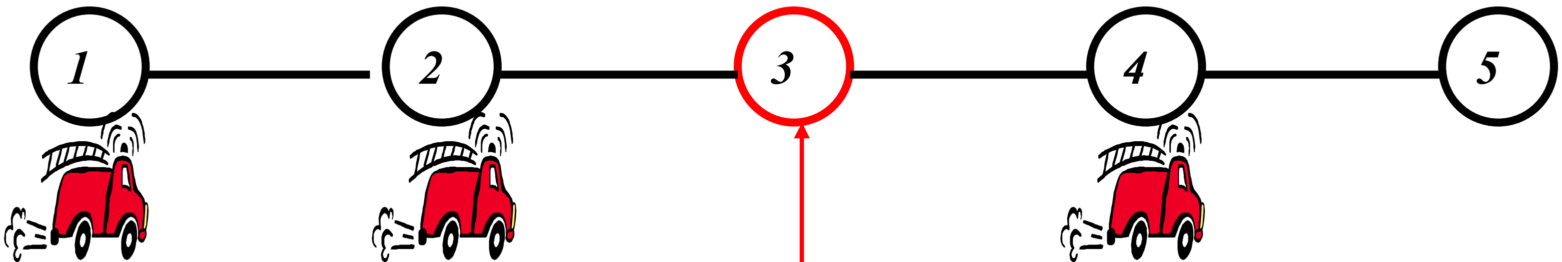
$$y^* = (y_1^* = 0,\ y_2^* = 1,\ y_3^* = 0,\ y_4^* = 1,\ y_5^* = 0)$$

- BKK2

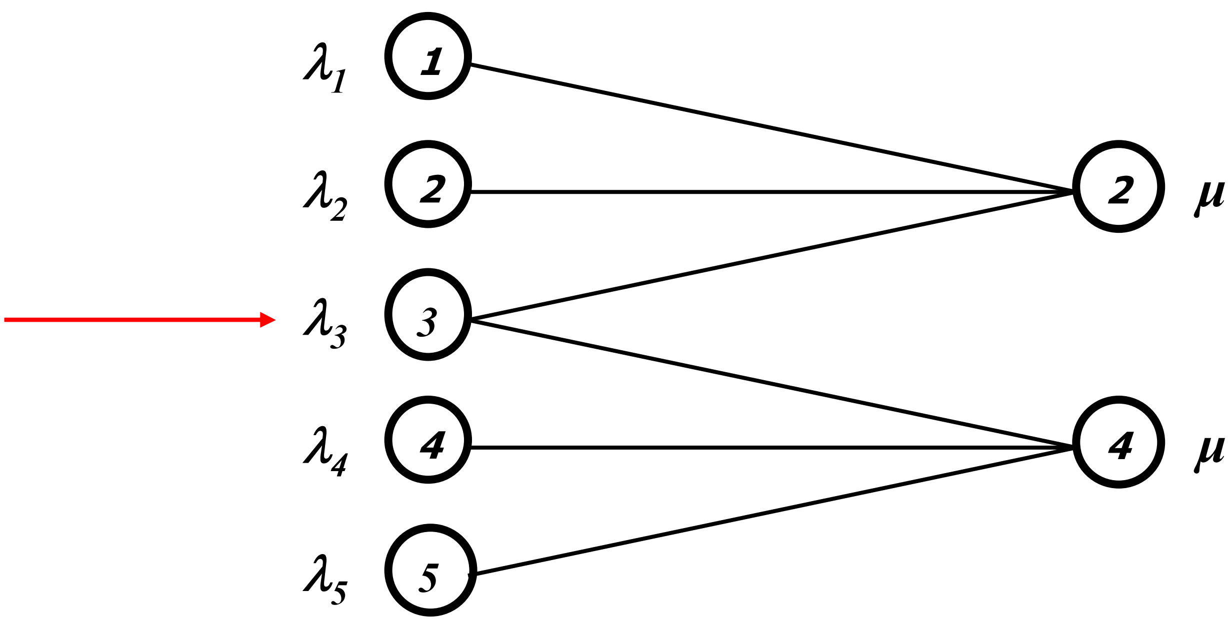


- BKK1

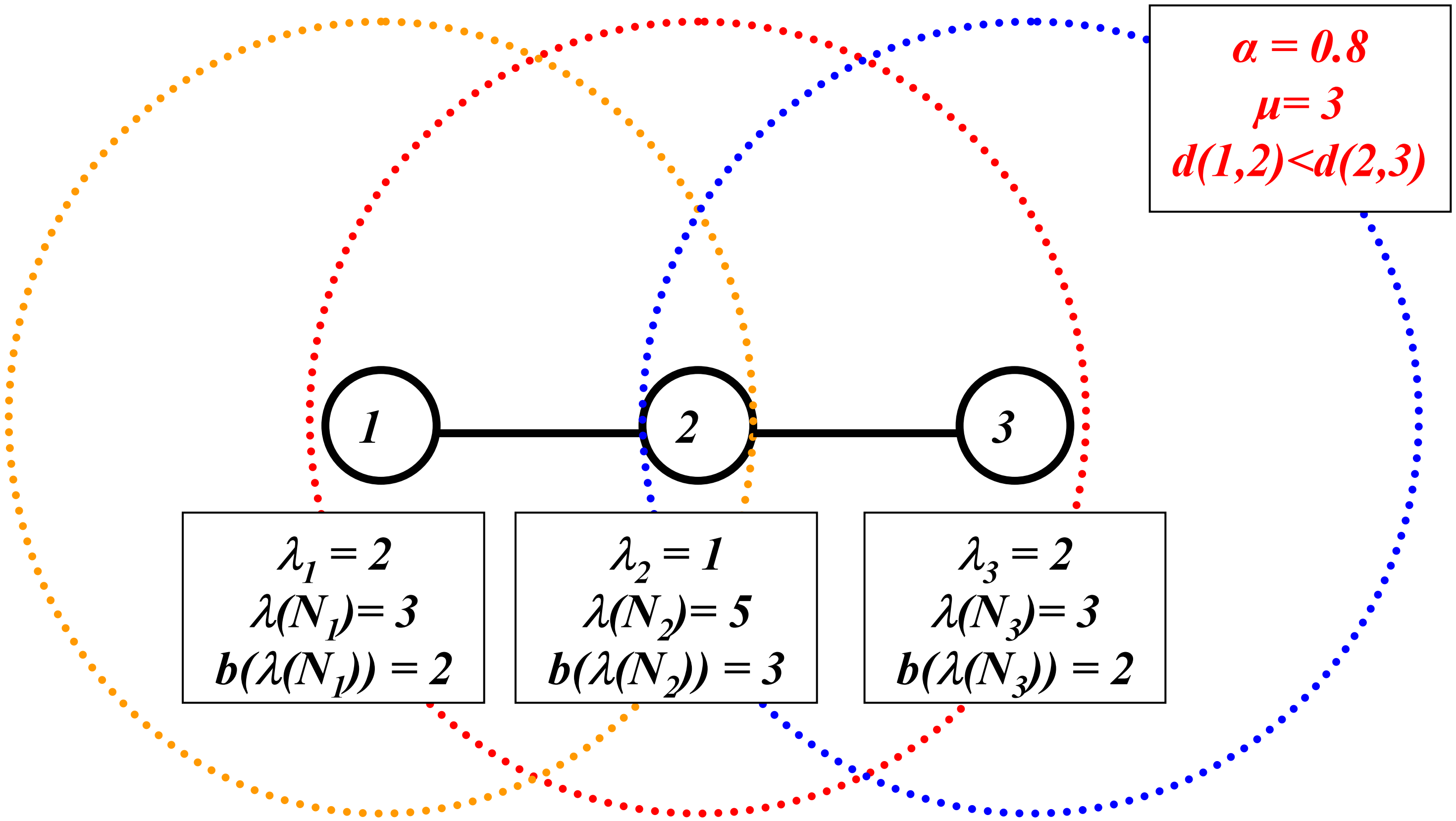
Why is BKK2 Feasible for (P)?



$$M / M / 2 / 2 \;\; with \; \sum_{i=1}^{5} \lambda_i$$



Example Revisited[5]



- Centered location: $x^c = (x_1^c = 0,\ x_2^c = 3,\ x_3^c = 0)$
  - *M/M/3/3* with demand rate *5*

Example Revisited[6]

- Set covering problem for facility location

$$\begin{aligned} \min \quad & y_1 + y_2 + y_3 \\ s.t. \quad & y_1 + y_2 \geq 1 \\ & y_1 + y_2 + y_3 \geq 1 \\ & y_2 + y_3 \geq 1 \\ & y_1, y_2, y_3 = 0, 1 \end{aligned}$$

- Optimal solution: $y^s = (y_1^s = 0,\ y_2^s = 1,\ y_3^s = 0)$
- Server allocation: $x^s = (x_1^s = 0,\ x_2^s = 3,\ x_3^s = 0) = x^c$
- BKK2 is safe!

$$A_1(x^s) = A_2(x^s) = A_3(x^s) = 0.84 > \alpha = 0.8$$

Simulation Experiments

- Data sets for 108 instances
  - $|N| = 20, 30, 50, \quad \alpha = 0.8, 0.85, 0.9, 0.95$
  - $\lambda_i \sim Uniform(1,10), \quad \mu = 24, 36, 48$
  - Link lengths $\sim Uniform(1,50)$
  - Radii = ($0.4, 0.6, 0.8$) (Average Shortest Distance)
- For each instance,
  - 5 Location models (with 2 choices of U for Ball & Lin's model) are formulated.
  - Simulated on locations (CPLEX8.1 solutions) given by the models

Computational Results[1]

LOCATION MODELS	ReVelle & Hogan	Ball & Lin w/ 50%	Ball & Lin w/ 75%	Marianov & ReVelle	BKK1	BKK2

- Fraction of infeasible nodes

AVERAGE	*0.1131*	*0.1984*	*0.0000*	*0.1003*	*0.0000*	*0.0000*
MAXIMUM	*0.5500*	*0.7500*	*0.0000*	*0.3333*	*0.0000*	*0.0000*

- Minimum deviations from α

AVERAGE	*-0.0702*	*-0.0252*	*0.0664*	*-0.0738*	*0.0239*	*0.0237*
MINIMUM	*-0.4104*	*-0.0713*	*0.0223*	*-0.3629*	*0.0000*	*0.0000*

- (Computational time) / $|N|$

AVERAGE	*0.0006*	*0.0070*	*0.0228*	*0.0003*	*0.0002*	*0.0005*
MAXIMUM	*0.0277*	*0.2057*	*0.7064*	*0.0098*	*0.0025*	*0.0055*

Computational Results^2

LOCATION MODELS	ReVelle & Hogan	Ball & Lin w/ 50%	Ball & Lin w/ 75%	Marianov & ReVelle	BKK1	BKK2

- (Number of servers) / $|N|$

AVERAGE	*0.6165*	*0.5840*	*0.7956*	*0.6152*	*0.6836*	*0.7264*
MINIMUM	*0.3000*	*0.3000*	*0.4400*	*0.3000*	*0.3600*	*0.4200*
MAXIMUM	*1.1000*	*1.0500*	*1.3667*	*1.1500*	*1.2333*	*1.2333*

- (Number of facilities) / $|N|$

AVERAGE	*0.3834*	*0.3185*	*0.3247*	*0.3857*	*0.2795*	*0.3080*
MINIMUM	*0.2000*	*0.1200*	*0.1400*	*0.2200*	*0.1200*	*0.1400*
MAXIMUM	*0.6500*	*0.5500*	*0.5500*	*0.7500*	*0.5500*	*0.5500*

Prospective Research

- Need better lower bounds
  - Bigger underestimates for availability to locate fewer servers
- Relax Markovian assumption on total service times
  - Total service time = travel time to a node and back to home facility + on-scene service time.
  - BKK1 with zero-capacity queues
- Need stability for infinite capacity queues
  - Place queues at facilities for BKK1
- Prioritize customers
  - Customer assignment policy