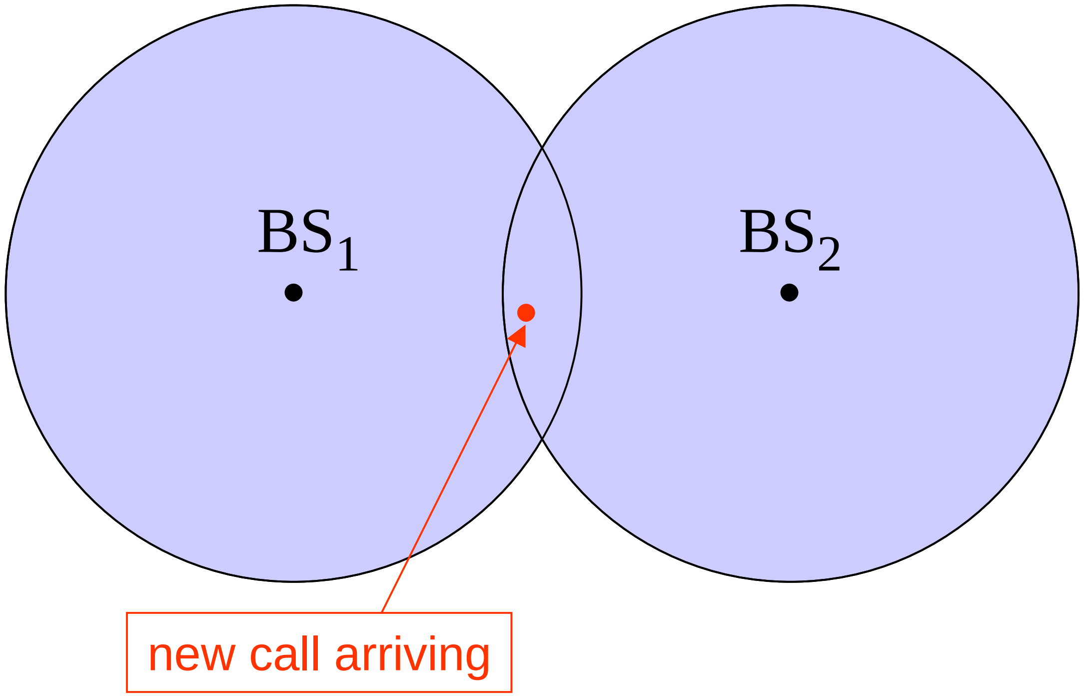
Load Balancing in Cellular Networks Using First Policy Iteration

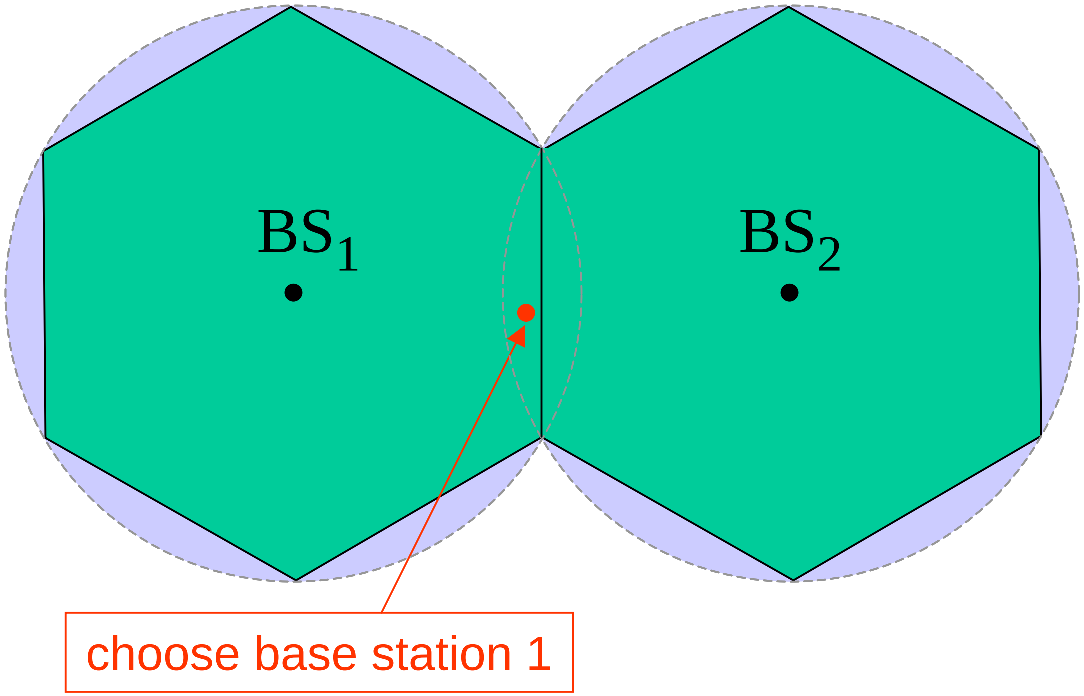
Johan van Leeuwaarden, Samuli Aalto & Jorma Virtamo
Networking Laboratory
Helsinki University of Technology

samuli.aalto@hut.fi

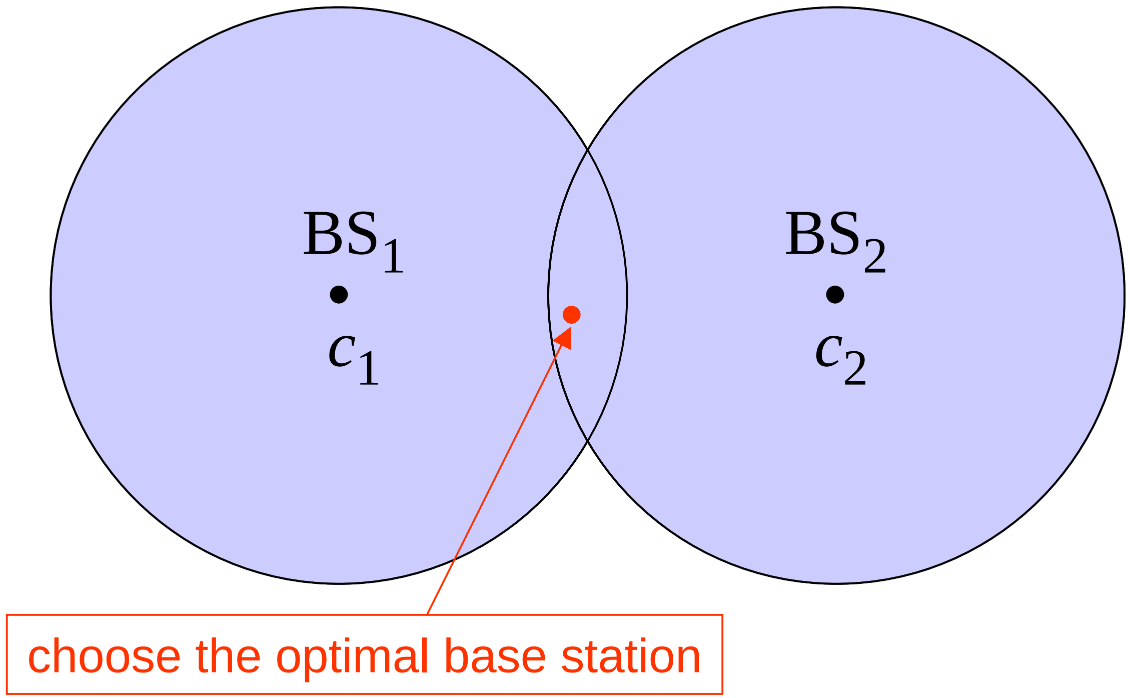
Problem formulation:
Routing of new calls in the overlapping area of two BS's



Signal based routing



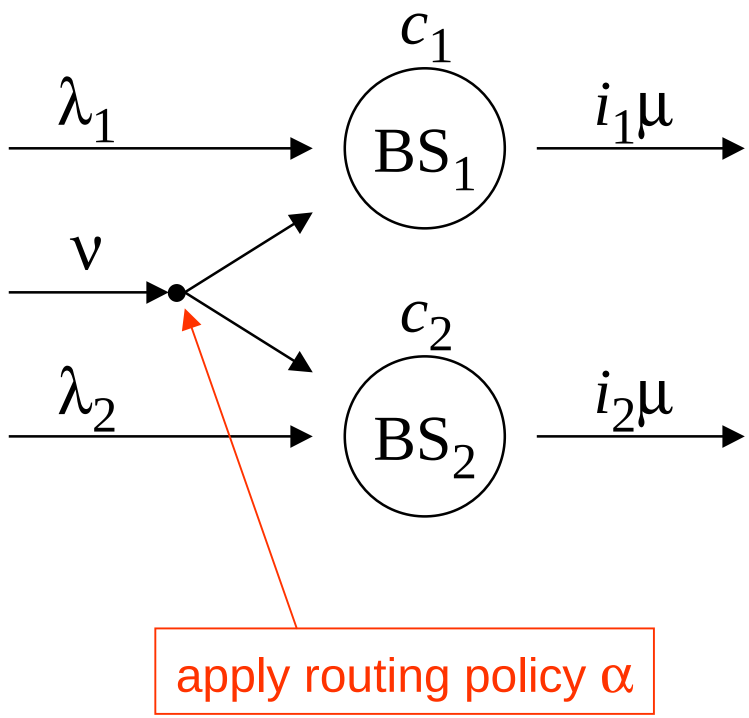
Load based routing



Model

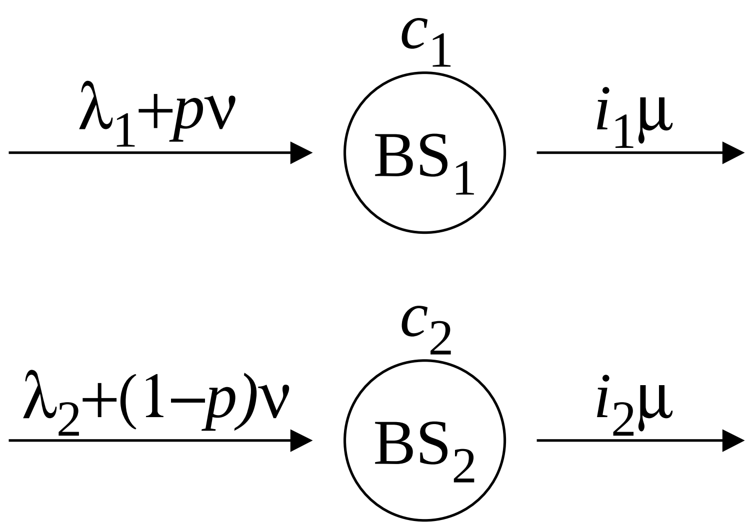
- Assumptions:
  - new calls arrive according to Poisson processes with rates λ_1, λ_2 and ν
  - $\lambda_1 \geq \lambda_2$
  - connection holding times exponentially distributed with mean $1/\mu = 1$
  - no mobility modelling

    $\Longrightarrow$ no handovers
- Notation:
  - c_k = capacity of BS_k
  - i_k = state of BS_k



Static (state-independent) routing policies

- **Randomized Routing**, RR(p)
  - arriving call in the overlapping area is routed to
    - BS_1 with probability p
    - BS_2 with probability $1 - p$
  - as a result, there are two independent Erlang loss systems with parameters (c_1, $\lambda_1 + p\nu$) and (c_2, $\lambda_2 + (1-p)\nu$)



Optimal Randomized Routing (ORR)

- For RR(p), the blocking probability $B(p)$ is clearly given by

$$B(p) = \frac{(\lambda_1 + p\nu)\,\mathrm{Erl}(c_1, \lambda_1 + p\nu) + (\lambda_2 + (1-p)\nu)\,\mathrm{Erl}(c_2, \lambda_2 + (1-p)\nu)}{\lambda_1 + \lambda_2 + \nu}$$

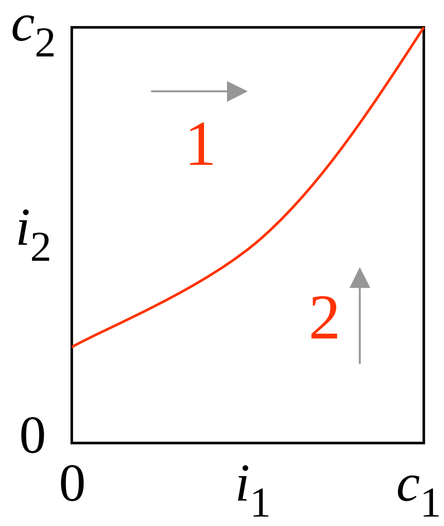
- The blocking probability is minimized by some p^*
- RR(p^*) is called the **Optimal Randomized Routing** (ORR)
- For the case $c_1 = c_2$, there is an explicit solution:

$$c_1 = c_2 \;\Rightarrow\; p^* = \begin{cases} \frac{1}{2}\left(1 - \frac{\lambda_1 - \lambda_2}{\nu}\right), & \text{if } \nu \geq \lambda_1 - \lambda_2 \\ 0, & \text{if } \nu < \lambda_1 - \lambda_2 \end{cases}$$

  – Idea: Balance the loads (as far as possible)

Dynamic (state-dependent) routing policies

- **Dynamic routing policy** α:
  - when in state (i_1, i_2),
    arriving call in the overlapping area is routed to station $\alpha(i_1, i_2)$
- Policy α is **greedy** if
  - it chooses the other station whenever one is full, i.e.,
    - $\alpha(c_1, i_2) = 2$
    - $\alpha(i_1, c_2) = 1$
- Policy α is a **switch-over strategy** if
  - there is a non-decreasing switch curve $s(i_1)$ such that
    - $\alpha(i_1, i_2) = 1$, if $i_2 \geq s(i_1)$
    - $\alpha(i_1, i_2) = 2$, if $i_2 < s(i_1)$



Optimal dynamic policy (1)

- In principle, the optimal dynamic routing policy can be determined e.g. by the **policy iteration algorithm** (developed in the theory of **Markov Decision Processes**):
  - fix the **immediate cost rate** for each state
  - choose a **basic policy** α
  - determine the **relative costs of states** for the basic policy from the **Howard equations**
  - when iterating, the decision of the **iterated policy** α' made in state (i_1, i_2) minimizes the relative costs of the post-decision state (j_1, j_2) for the basic policy α
  - by this way, we get a new, better policy α' (with smaller average cost) used as the basic policy for the next iteration
  - an optimal policy is found as soon as the policy does not change anymore in this iteration

Optimal dynamic policy (2)

- Target: **minimize blocking probability** $\Rightarrow$ immediate cost rate $r(i_1, i_2)$ is as follows:

$$r(i_1,i_2) = \begin{cases} \lambda_1 + \lambda_2 + \nu, & \text{for } i_1 = c_1,\ i_2 = c_2 \\ \lambda_1, & \text{for } i_1 = c_1,\ i_2 < c_2 \\ \lambda_2, & \text{for } i_1 < c_1,\ i_2 = c_2 \end{cases}$$

- Howard equations for relative costs $v_\alpha(i_1, i_2)$:

$$r(i_1,i_2) - r_\alpha + \sum_{(j_1,j_2)} q_\alpha((i_1,i_2),(j_1,j_2))v_\alpha(j_1,j_2) = 0$$

- Iterated policy α':

$$\alpha'(i_1,i_2) = \begin{cases} 1, & \text{if } v_\alpha(i_1+1,i_2) \le v_\alpha(i_1,i_2+1) \\ 2, & \text{if } v_\alpha(i_1+1,i_2) > v_\alpha(i_1,i_2+1) \end{cases}$$

Relative costs for static policies

- If the basic policy is RR(p), we have two independent subsystems. Thus,

$$v_{\alpha}(i_1, i_2) = v_1(i_1) + v_2(i_2)$$

- Iterated policy α' is therefore:

$$\alpha'(i_1, i_2) = \begin{cases} 1, & \text{if } v_1(i_1+1) - v_1(i_1) \le v_2(i_2+1) - v_2(i_2) \\ 2, & \text{if } v_1(i_1+1) - v_1(i_1) > v_2(i_2+1) - v_2(i_2) \end{cases}$$

- Moreover, these subsystems are Erlang loss systems for which the relative costs are easily found:

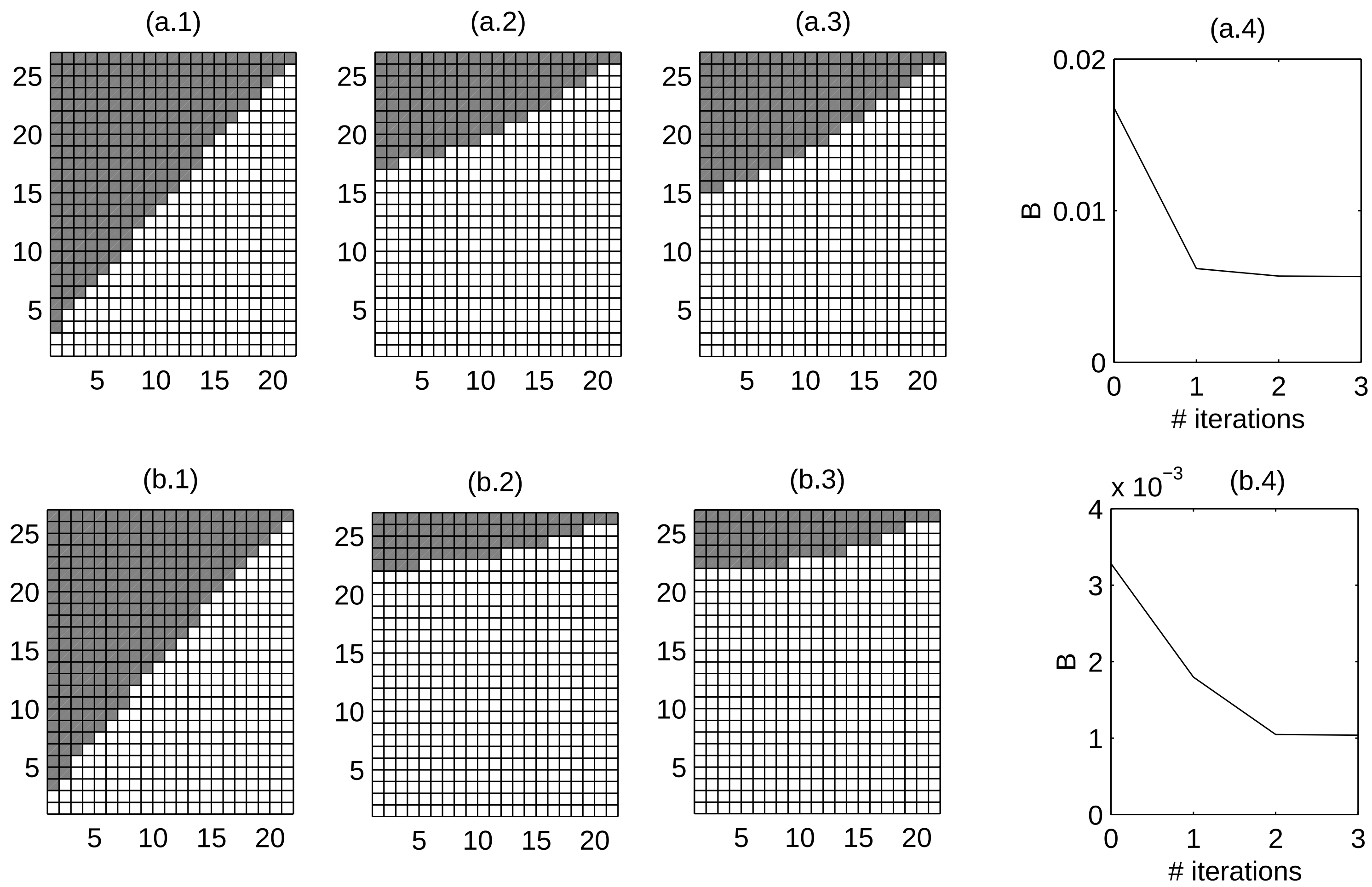
$$v(i+1) - v(i) = \frac{\mathrm{Erl}(c, \lambda)}{\mathrm{Erl}(i, \lambda)}$$

First policy iteration: policy FPI

- All iterated policies are dynamic
- The calculation of the relative costs for dynamic policies is (much) more demanding, albeit possible
  - linear equation system of $(c_1 + 1)(c_2 + 1)$ variables
- On the other hand, it is known that (typically) the first iteration step is the most significant
- Straightforward idea:
  - Use ORR as the basic policy
    - The two independent Erlang loss systems are $(c_1, \lambda_1 + p^* \nu)$ and $(c_2, \lambda_2 + (1-p^*)\nu)$
  - Iterate only once
- Call this **FPI**
  - It is easily seen to be a greedy switch-over strategy

FPI vs. Optimal dynamic policy

$\lambda_1 = \nu = 10,\ c_1 = 20,\ c_2 = 25,$ (a) $\lambda_2 = 10$, (b) $\lambda_2 = 5$

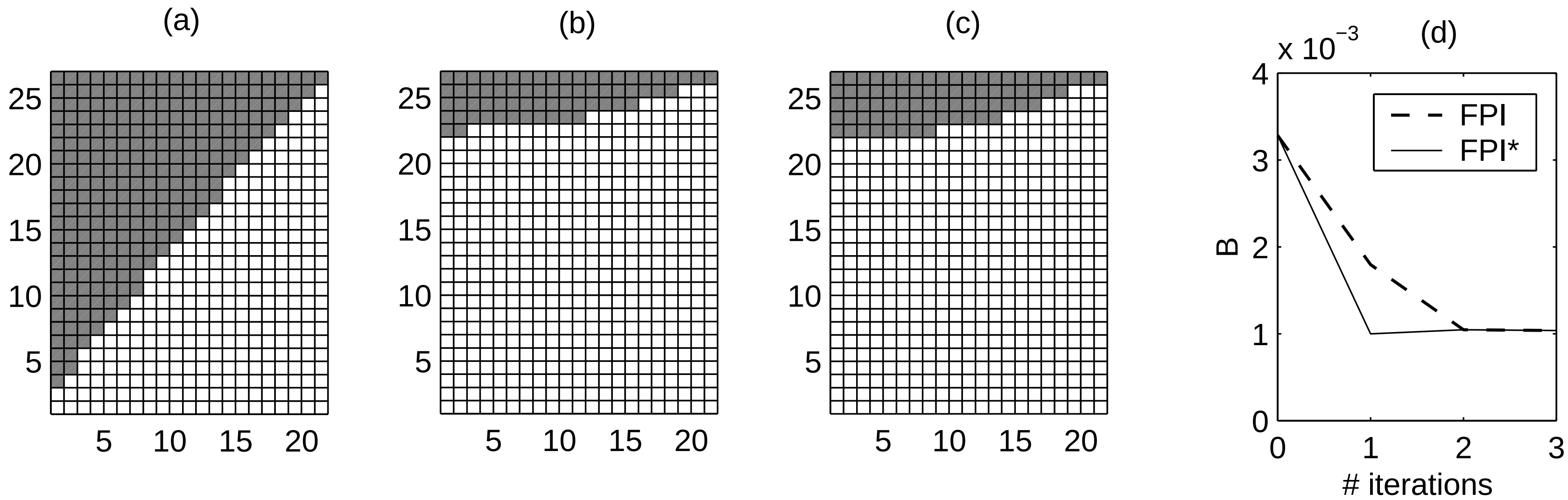


First policy iteration: policy FPI*

- But, is ORR an optimal basic policy (among static policies) minimizing the blocking probability of the iterated policy (after one step)?
- Johan's idea:
  - ORR tries to balance the loads, thus ignoring the impact of (possibly) different dedicated streams
  - What if we simply ignore the flexible arrival stream (with rate ν)?
    - This basic policy rejects all new calls in the overlapping area!
    - The two independent Erlang systems are (c_1, λ_1) and (c_2, λ_2)
  - Iterate only once
- Call this **FPI***
  - It is also easily seen to be a greedy switch-over strategy

FPI vs. FPI* vs. Optimal dynamic policy

$$\lambda_1 = \nu = 10,\ \ \lambda_2 = 5,\ \ c_1 = 20,\ \ c_2 = 25$$

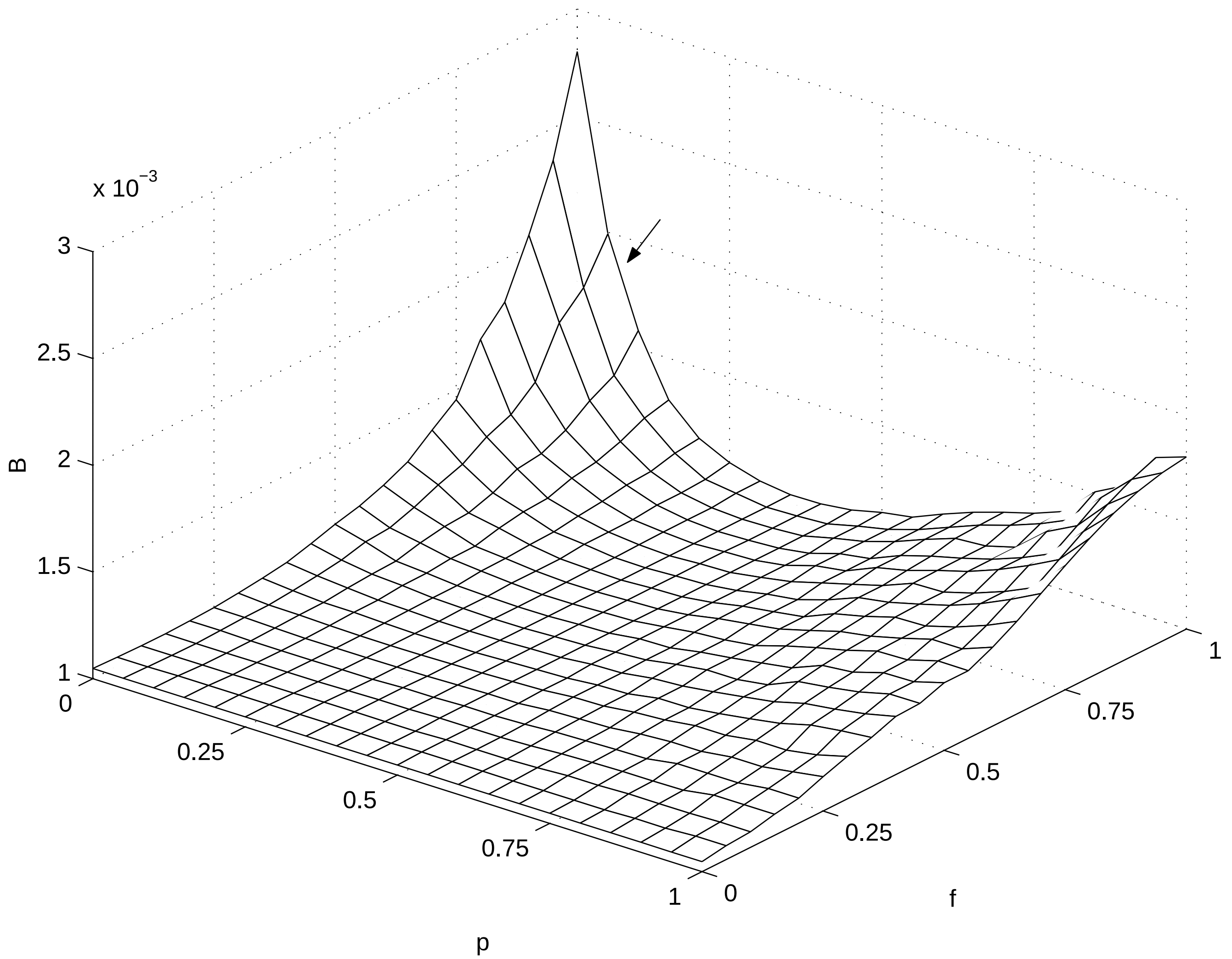


First policy iteration: basic policy optimization

- Consider a combined admission & routing policy RAR(f,p) that
  - accepts the new call arriving in the overlapping area with prob. f
  - routes an accepted call to station 1 with prob. p
- This is a static policy
  - the two independent Erlang loss systems are $(c_1, \lambda_1+pf\nu)$ and $(c_2, \lambda_2+(1-p)\,f\nu)$
- Note that
  - FPI is the iterated policy corresponding to basic policy RAR(1,p^*)
  - FPI* is the iterated policy corresponding to basic policy RAR(0)
- Parameters f and p can be optimized so that the blocking probability of the iterated policy (after one step) is minimized
  - the optimal value seems to be $f = 0$ leading to FPI*

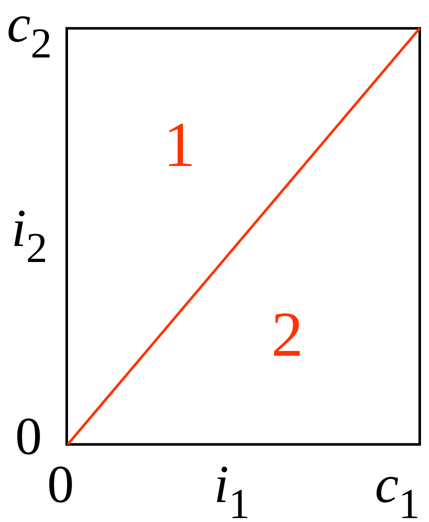
Optimal basic policy

$$\lambda_1 = \nu = 10,\ \ \lambda_2 = 5,\ \ c_1 = 20,\ \ c_2 = 25$$

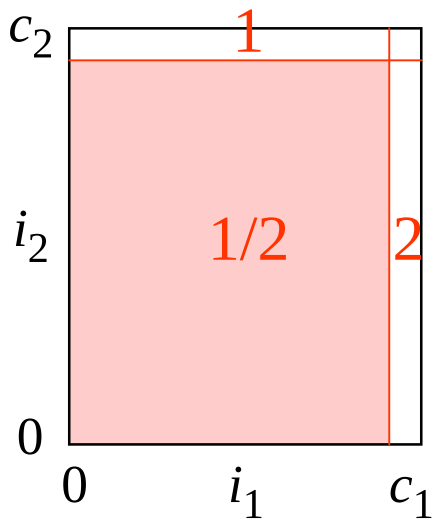


Greedy heuristic policies

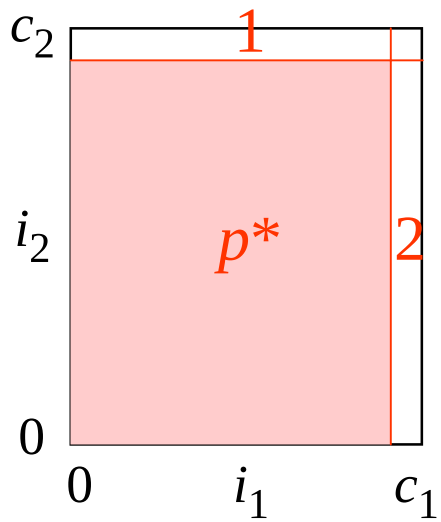
Least ratio routing
(LRR)



Overflow routing
(OFR)

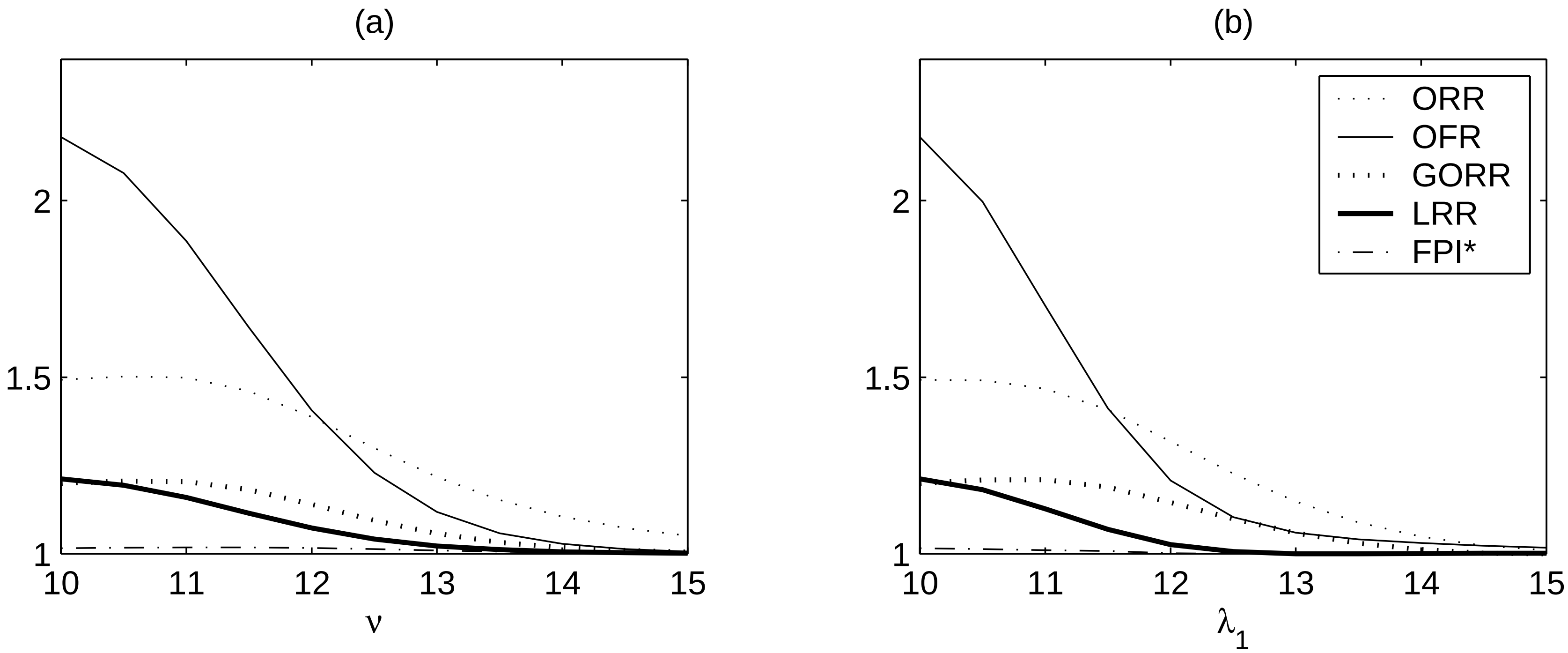


Greedy ORR
(GORR)



ORR vs. greedy heuristic policies vs. FPI*

$$\lambda_1 = \nu = \lambda_2 = 10,\ c_1 = 25,\ \ c_2 = 15$$



Open questions

- Rigorous proofs
  - Optimal dynamic policy is a greedy switch-over policy, isn't it?
  - FPI* is the optimal one-step-iterated policy based on static basic policies, isn't it?
- Sensitivity analysis
  - What if λ_1, λ_2 and ν are only approximately known?
  - Is FPI* still a good policy?
- Mobility modelling & handovers
  - geometric approach
  - stochastic approach

THE END