Some Optimization Problems in Dynamic Spectrum Access

Marceau Coupechoux, Hany Kamal, Philippe Godlewski

TELECOM ParisTech (INFRES/RMS) and CNRS LTCI

16 June 2010

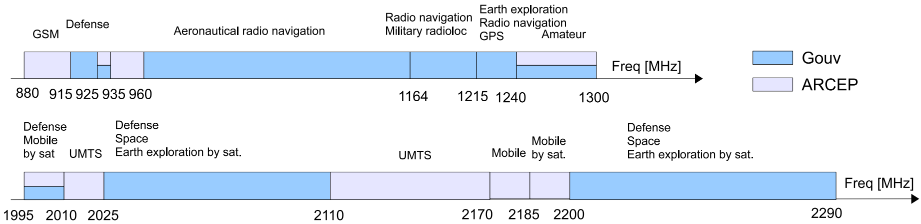
The Frequency Spectrum

The Frequency spectrum is a common good characterized by:

- A strong regulation
- High occupancy variations
- Possible congestions

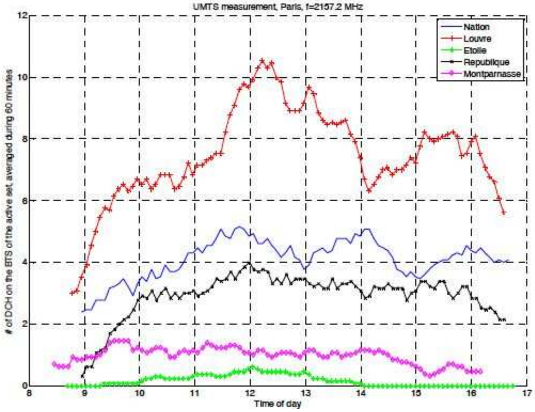
A Strong Regulation

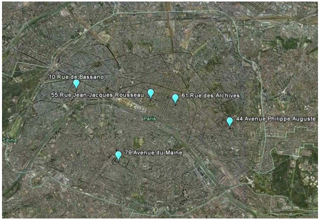
- The spectrum is divided into small parts
- The spectrum is not technology agnostic
- see ANFR frequency table [TNRBF]



High Occupancy Variations

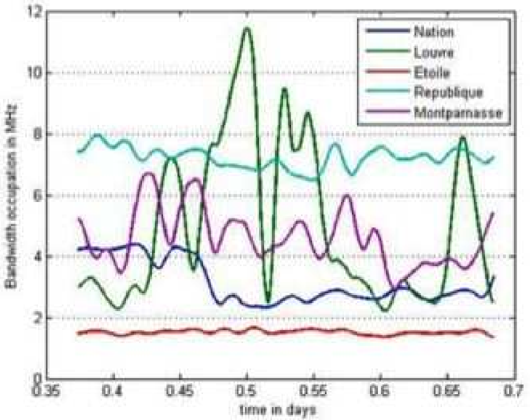
- UMTS Measurements (2.1GHz) [urc]
- 5 locations in Paris
- High spatio-temporal variations of the traffic





Possible Congestions

- PMR Measurements (450-470 MHz) [urc]
- Still high spatio-temporal variations
- Up to 94% of spectral occupancy



Technological Trends

Radio is becoming flexible [Buddhikot07, Filin08]

- Software Define Radio
- Cognitive Radio / Dynamic Spectrum Access

Technologies are multi-carrier [3gpp]

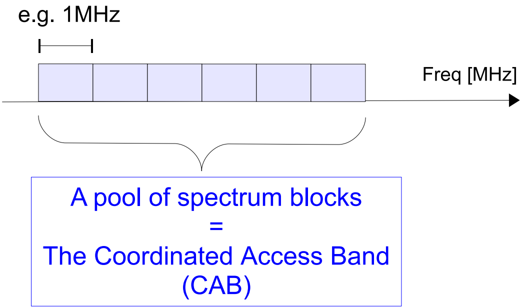
- OFDMA based standards (LTE, WiMAX)
- Carrier aggregation (HSPA, LTE Advanced)

Regulators are changing the rules [ofcom07]

- Spectrum is becoming technology agnostic (UMTS 900)
- Spectrum can be reused by secondary users (IEEE 802.22)

Study Framework

- We focus on a **mobile operator**
- Operating **one or several technologies**, e.g. LTE, HSPA, WiFi, etc
- Able to **lease** spectrum frequency blocks to the regulator [Buddhikot05]
- Willing to **optimize the spectrum usage** in some sense



Outlines

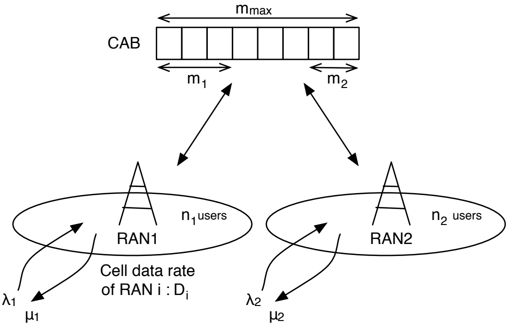
- Time DSA
  - Optimal Policies
  - A Simple Heuristic
  - Q-Learning Approaches
- Space-time DSA
  - Tabu Search on a Cell Cluster
  - Dynamic Scenario
  - Infinite Network
- Conclusion

Outlines

- **Time DSA**
  - **Optimal Policies**
  - **A Simple Heuristic**
  - **Q-Learning Approaches**
- Space-time DSA
  - Tabu Search on a Cell Cluster
  - Dynamic Scenario
  - Infinite Network
- Conclusion

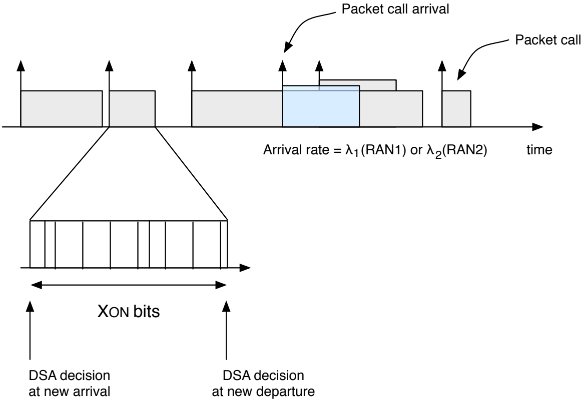
System Model

- Cell-by-cell DSA
- Two Radio Access Networks (RAN) operated by one operator
- A **CAB** (= a pool) made of frequency blocks
- **Cell capacity** is proportional to the leased bandwidth
- **Spectrum cost** is also proportional to the leased bandwidth



System Model

- **ON/OFF elastic traffic**
- Poisson arrivals (λ_1, λ_2) for packet calls
- Exp. volume of data to be downloaded (avg X_{ON})
- Service rate: $\mu_i = \frac{m_i D_i}{X_{ON}}$



- Cell nominal data rate (for one block): D_i
- Max. number of users per cell: (n_1^{max}, n_2^{max})
- Fair throughput scheduling

System Model

- **Reward** = Revenues − Costs
- Revenues = Sum of customer satisfactions [Enderle03]

$$\phi_i(n_i, m_i) = K_u(1 - \exp(-\mu_i / n_i \mu_{com}))$$

$$g_1(s) = n_1\phi_1(n_1, m_1) + n_2\phi_2(n_2, m_2)$$

- Spectrum cost is increasing with CAB occupancy

$$g_2(s) = K_B(m_1 + m_2)\exp\left(-\frac{m_{max} - m_1 - m_2}{m_{com}}\right)$$

- **Reward:**

$$g(s) = g_1(s) - g_2(s)$$

Note: K_u [euros], K_B [euros/MHz], μ_{com} [1/s], m_{com} are constants.

A DSA Policy

- A DSA policy dynamically assigns spectrum blocks to every RAN
- Trade-off :
  More spectrum $\Longrightarrow$ Higher spectrum cost
  More spectrum $\Longrightarrow$ Higher throughput and more revenues
- Chosen approaches: SMDP, heuristics, Q-learning

SMDP Formulation

- State space: $s = (n_1, m_1, n_2, m_2)$
  with constraints $n_1 \leq n_1^{max}$, $n_2 \leq n_2^{max}$ and $m_1 + m_2 \leq m_{max}$
- Reward function: $g(s)$
- Action space: $a = (a_1, a_2)$, $a_i \in \{0, -1, +1\}$

Table: List of possible actions

Action	a vector	action index
Band1 constant and Band2 constant	$(0, 0)$	1
Band1 constant and Band2 increases	$(0, +1)$	2
Band1 constant and Band2 decreases	$(0, -1)$	3
Band1 increases and Band2 constant	$(+1, 0)$	4
Band1 increases and Band2 increases	$(+1, +1)$	5
Band1 increases and Band2 decreases	$(+1, -1)$	6
Band1 decreases and Band2 constant	$(-1, 0)$	7
Band1 decreases and Band2 increases	$(-1, +1)$	8
Band1 decreases and Band2 decreases	$(-1, -1)$	9

SMDP Transition Probabilities

- DSA decisions are taken at each new event (arrival or departure)
- $p_{s,s'}(a)$ is the proba. to go from s to s' if a is chosen
- Let $1/\nu_s(a)$ be the expected time until next decision epoch:

$$\begin{aligned} \nu_s(a) &= \mathbb{1}_{\{n_1 < n_1^{max}\}}\lambda_1 + \mathbb{1}_{\{n_2 < n_2^{max}\}}\lambda_2 \\ &\quad + \mathbb{1}_{\{n_1 > 0\}}\mu_1 + \mathbb{1}_{\{n_2 > 0\}}\mu_2 . \end{aligned}$$

- Transition probabilities are given by:

$$p_{s,s'}(a) = \begin{cases} \lambda_i/\nu_s(a) & \text{if } (n'_i = n_i + 1) \\ & \text{and } (\forall j\ m'_j = m_j + a_j), \\ \mu_i/\nu_s(a) & \text{if } (n'_i = n_i - 1) \\ & \text{and } (\forall j\ m'_j = m_j + a_j), \\ 0 & \text{otherwise.} \end{cases}$$

SMDP Uniformization

- Continuous Time Markov chain $\longrightarrow$ Equivalent Dicrete Time
- A small transition step $1/\nu$ $(\forall s, a,\ \nu_s(a) \leq \nu)$
- Transition probabilities are modified [Bertsekas07]:

$$\tilde{p}_{s,s'}(a) = \begin{cases} p_{s,s'}(a)\nu_s(a)/\nu & \text{if } s \neq s', \\ 1 - \sum_{s' \neq s} \tilde{p}_{s,s'}(a) & \text{otherwise.} \end{cases}$$

- Recall: a DSA policy R associates to each system state s an action $R(s)$ in the action space of s

SMDP Policy Iteration

Algorithm 1 Policy Iteration

1: **Initialization**: Let R be an arbitrary stationary policy.

2: **Value-determination**: For the current policy R, we solve the system of linear equations whose unknowns are the variables $\{J_R, h_R(s)\}$: $h_R(1) = 0$ and

$$h_R(s) = g(s) - J_R + \sum_{s' \in S} \tilde{p}_{s,s'}(R(s)) h_R(s').$$

3: **Policy improvement**: For each $s \in S$, we find:

$$R'(s) = arg \max_{a \in A(s)} \left\{ g(s) - J_R \quad + \sum_{s' \in S} \tilde{p}_{s,s'}(a) h_R(s') \right\}.$$

4: **Convergence test**: If $R' = R$, the algorithm is stopped, otherwise, we go to step 2 with $R := R'$.

SMDP Strengths and Weaknesses

Strengths:

- Provides centralized optimal policies
- Upper bounds on system performance
- Takes into account RAN loads (λ_1, λ_2), number of active users (n_1, n_2), dynamic of the system

Weaknesses:

- Dependent on system parameters (no threshold policy)
- Not usable in real-time
- Or requires massive storage of data

Heuristic DSA

Heuristic DSA principles:

- We neglect (n_1, n_2) variations and focus on (λ_1, λ_2)
- We assume that (m_1, m_2) is fixed for given (λ_1, λ_2)

- Each RAN acts as a M/M/1/n_i^{max}
- **Average reward**:

$$g_H(\lambda_1, \lambda_2, m_1, m_2) = \sum_{i=1}^{2} \sum_{n_i=0}^{n_i^{max}} \pi_{n_i}(\lambda_i) n_i \phi_i(n_i, m_i) - g_2(m_1, m_2)$$

where the $\pi_{n_i}(\lambda_i)$, $i \in \{1, 2\}$, $n_i \in \{0, ..., n_i^{max}\}$ are the steady state probabilities of a M/M/1/n_i^{max}

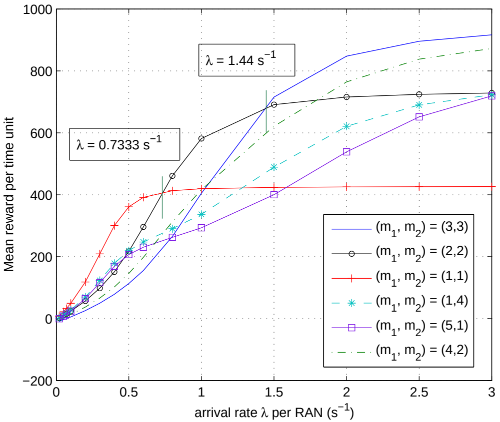
Heuristic DSA

Algorithm 2 Heuristic DSA

1: Estimate arrival rates λ_1 and λ_2.
2: **for all** (m_1, m_2) **do**
3: Compute the average reward g_H.
4: **end for**
5: Allocate bandwidth according to the tuple (m_1, m_2) that maximizes the average reward g_H.

Heuristic DSA

- Example: $\lambda_1 = \lambda_2 = \lambda$
- 'Link adaptation'-like curves provide allocations and thresholds



Q-Learning based DSA

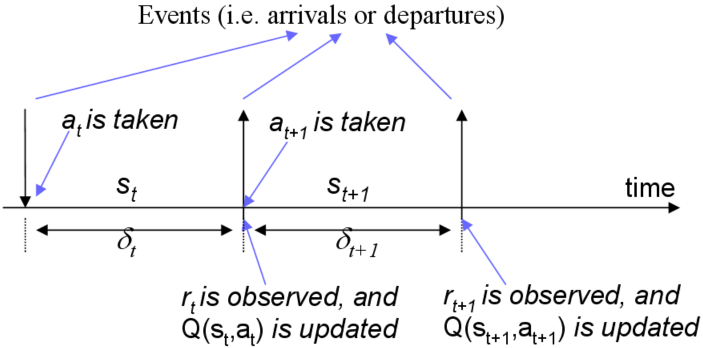
- System param. λ_1, λ_2, μ_1, μ_2 and X_{ON} are still needed

- QL is used to optimize discrete discounted-reward problems [Watkins89]
- [Tadepalli98] and [Abounadi01] have proposed RL algos for the average cost problem
- [Gosavi04] has proposed an algo. for average cost and continuous-time problems

QL Model

Gosavi's Q function update:

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\ r_t - \alpha\ \rho\ \delta_t + \alpha\ arg \max_{a \in A(s)} \{Q(s_{t+1}, a)\}$$



QL Model

- *Gosavi*'s algo. differ from value-iteration by substracting an estimate ρ of the average reward per time-unit
- ρ is estimated using a second learning factor:

$$C \leftarrow (1-\beta)C + \beta\ r_t$$

$$T \leftarrow (1-\beta)T + \beta\ \delta_t$$

$$\rho = C/T$$

QL Algorithm

Algorithm 3 Q-learning based DSA

Initialize the following parameters:

- ...
- the number of times Q is exploited: $k = 0$
- the number of visits to the state-action pair (s, a): $N_v(s, a) = 0$

repeat

  Exploration: with proba. p, a_t chosen at random

  Exploitation: w/ $1 - p$, choose action a_t that maximizes $Q(s_t, a)$

  Update $\alpha = 1/(1 + N_v(s, a))$ and $\beta = 1/(1 + k)$.

  Update $Q(s_t, a_t)$ and ρ

  $k \leftarrow k + 1$, $N_v(s_t, a_t) \leftarrow N_v(s_t, a_t) + 1$.

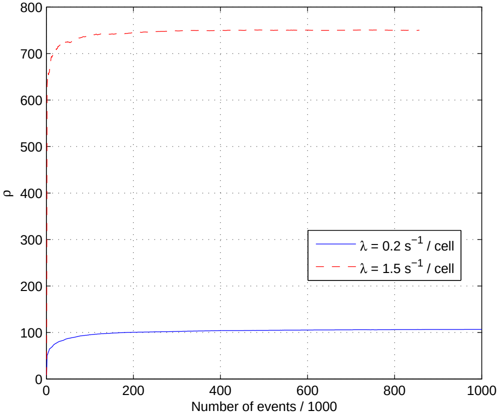
  $s_t \leftarrow s_{t+1}$.

  $t \leftarrow t + 1$.

until End of the learning period

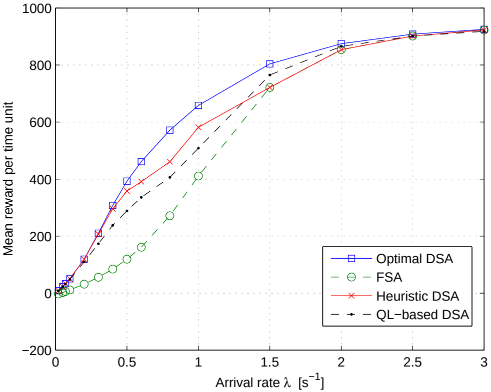
QL Convergence

- Example of convergence speed
  $\Longrightarrow$ agent will keep learning for 200 thousand events



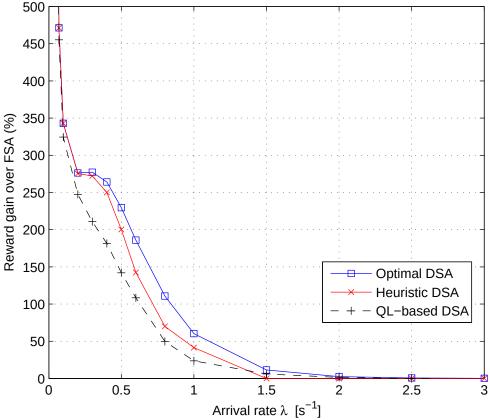
Performance Evaluation

- QL and Heuristic achieve similar performance
- But QL does not require the knowledge of system parameters
- As load increases, all algos converge to FSA



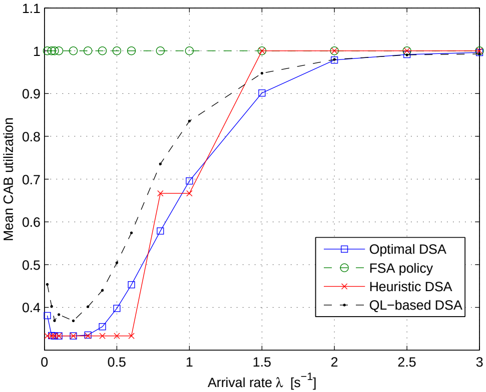
Performance Evaluation

- At low loads, proposed algos provide significant gains
- At very low loads, proposed algos are optimal



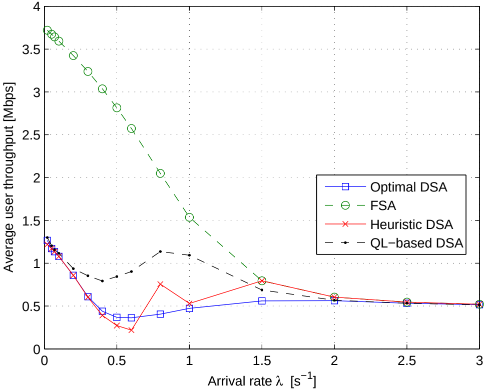
Performance Evaluation

- FSA allocates by definition half of the CAB to each RAN
- Results are explained by a better utilization of the spectrum



Performance Evaluation

- However, at the cost of a reduced user throughput !

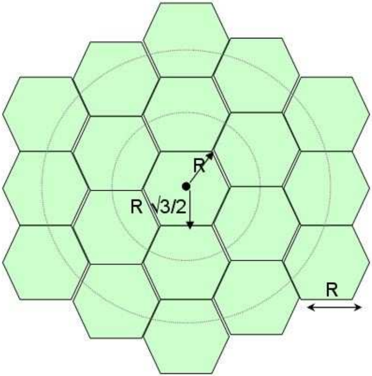


Outlines

- Time DSA
  - Optimal Policies
  - A Simple Heuristic
  - Q-Learning Approaches
- **Space-time DSA**
  - **Tabu Search on a Cell Cluster**
  - **Dynamic Scenario**
  - **Infinite Network**
- Conclusion

Network Model

- A single operator with a single RAT
- Leasing of the spectrum bands
- DSA at cell level
- Hexagonal network



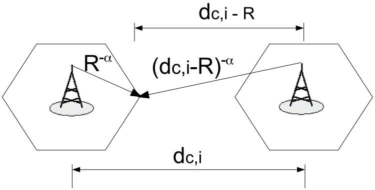
Network Model

- Carrier to Interference Ratio (CIR) and cell capacity

$$CIR_c^f = \frac{R^{-\alpha}}{\sum_{i=1}^{B_f} (d_{c,i} - R)^{-\alpha}}$$

$$C_c = \sum_{f=1}^{F_c} W_f \ log_2(1 + CIR_c^f)$$

- Fair throughput scheduling among users : $D_c = C_c / N_c$.



Network Model

- Reward $=$ Revenues $-$ Spectrum Cost
- Revenues $=$ Sum of customer satisfactions

$$\phi_c(D_c) = K_u(1 - \exp(-D_c/D_{com}))$$

- Spectrum cost $\propto$ Leased spectrum bandwidth

$$K_B \ W_f \ F$$

- Reward:

$$g = \sum_{c=1}^{B} N_c \phi_c(D_c) - K_B \ W_f \ F$$

Note: K_u [euros], K_B [euros/MHz] and D_{com} [bps] are constants.

Network Model

- A DSA policy assigns spectrum blocks to every cell in the RAT
- Trade-off :
  More spectrum $\Longrightarrow$ Higher spectrum cost
  More spectrum $\Longrightarrow$ Higher throughput and more revenues
- Chosen approach: Tabu search [Glover89] [urc2]

Tabu Search Approach Illustrated

- **A solution:** s is a boolean matrix of size $F_{max} \times B$, $s_{f,c} = 1$ if frequency f is assigned to cell c

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

  In this example, $F_{max} = 3$, $B = 5$ and $F = 2$.
- **A move:** m is a boolean matrix of size $F_{max} \times B$, *one* or *two* elements of m are non-zero, i.e., we allow to:
  - remove an assigned frequency to a cell
  - add a new frequency to a cell
  - replace a used frequency by an unused frequency
- **A neighbor:** $s' = s \oplus m$ for some m in the set of possible moves
- **Attribute of s:** $g(s)$

Tabu Search Approach Illustrated

Algorithm 4 TS algorithm for DSA

Initialization: an initial solution s_{init} is found.
$s \leftarrow s_{init}$
$g_{max} \leftarrow g(s_{init})$
while Nb. of iterations $\leq$ *MAXITER* **do**
    Neighborhood formation: all *possible* neighbors of the initial solution s are created, except those who are listed as tabu.
    Neighbor selection: s' not in Tabu List and that maximizes $g(s')$
    Tabu list update: the reward $g(s')$ corresponding to the selected solution s' is added to the Tabu List.
    Max. reward update:
    if $g(s') > g_{max}$, **then** $g_{max} \leftarrow g(s')$ **end if**
end while

Tabu Search Approach Illustrated

Notes:

- Tabu List size is not a real issue
- Solutions with the same reward are equivalent for the algorithm
- Initialization:
  - Total number of frequencies to be used by the operator is unknown
  - Solution set is divided in search spaces $\{1, ..., F_{max}\}$
  - Random solutions are generated in every search space
  - The best solution ever seen is s_{init}
- Total number of neighbors is:

$$F_{max}\ B - B_{s0} + \sum_{c=1}^{B} F_c\ (F_{max} - F_c)$$

i.e., generating all possible neighbors is very feasible.

Performance in Case of Heterogeneous Traffic

- We study 8 'hot spots' scenarios
- Spatial heterogeneity is increasing
- The last scenario is the homogeneous one

Table: Studied users distributions and corresponding standard deviations σ

central cell	middle-circle cells	outer-circle cells	σ
33	2	1	7.28
27	3	1	5.88
21	4	1	4.58
15	5	1	3.46
15	3	2	2.94
9	6	1	2.76
9	4	2	1.73
3	3	3	0

Performance in Case of Heterogeneous Traffic

- We compare FSA and DSA

Fixed Spectrum Access (FSA)

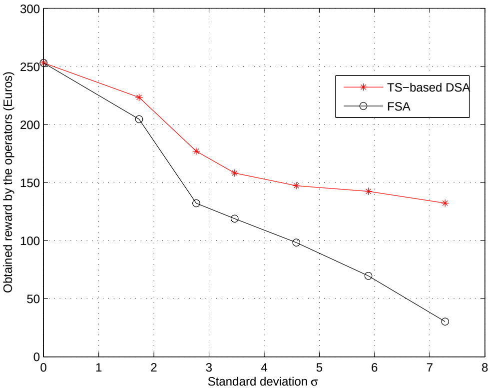
- TS is launched on the homogeneous case
- Frequency allocation is kept constant for all scenarios

Dynamic Spectrum Access (DSA)

- TS is launched for each scenario
- There is one frequency allocation per scenario

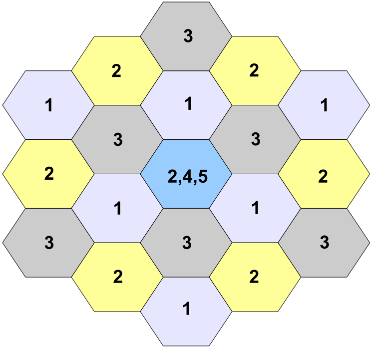
Performance in Case of Heterogeneous Traffic

- For $\sigma = 0$, both methods achieve the same reward
- Advantage of DSA is increasing with heterogeneity
- Reward$\times$3 in the most heterogeneous case



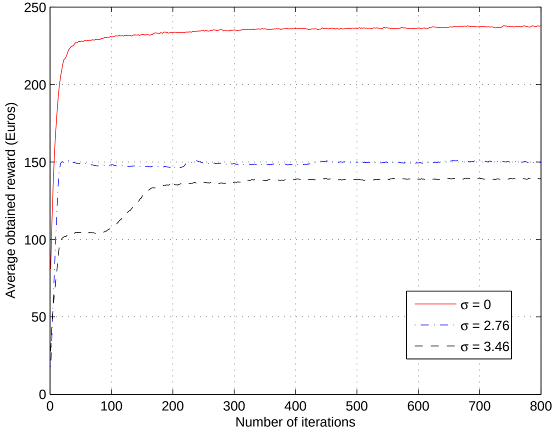
Performance in Case of Heterogeneous Traffic

- Obtained spectrum assignment using TS DSA for $\sigma = 7.28$
- 3 frequencies for the central cell
- Regular allocation for outer cells $\approx$ reuse 3



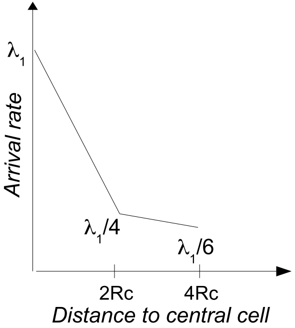
Convergence Speed

- Around 200 or 300 iterations provide very good results
- Is it possible to use TS in real time ?



Dynamic scenario

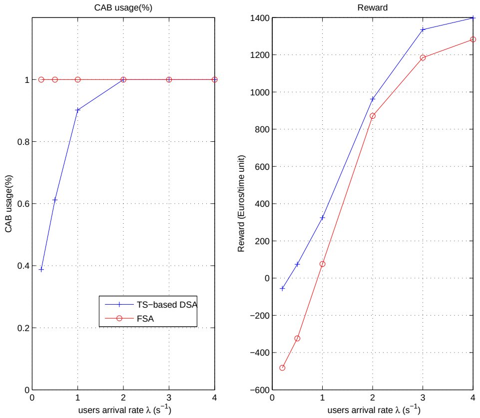
- Assumed traffic : ON/OFF (X_{ON} bits, λ s^{-1})
- Monte carlo simulations
- Arrival rate is decreasing with the distance to the central cell
- Average arrival rate : λ



- TS is launched for 300 iterations at the very beginning
- At each event (arrival or departure), TS is launched for 10 iterations
- Initial solution is the allocation at the time TS is launched

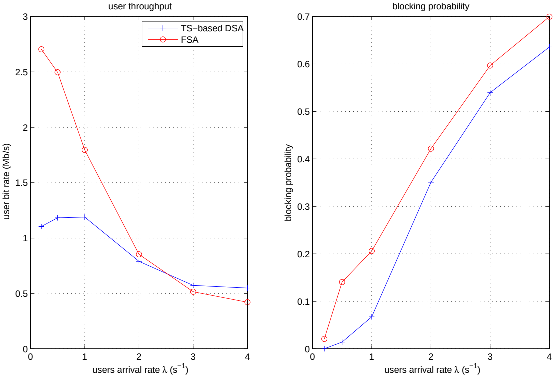
Dynamic scenario

- For low loads, only part of the spectrum is used
- At $\lambda = 1\ \mathrm{s}^{-1}$, reward is $\times$ 3
- At $\lambda = 4\ \mathrm{s}^{-1}$, gain is $+13\%$



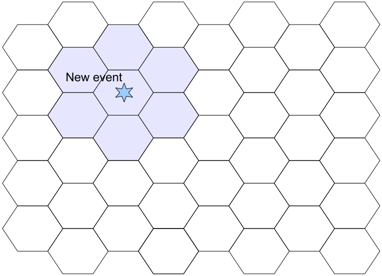
Dynamic scenario

- Throughput is proportional to the bandwidth
  $\Longrightarrow$ user throughput is less with DSA
- Radio resources are used where needed
  $\Longrightarrow$ blocking probability is lowered



Further Work: Infinite Network

- How to extend to an infinite network ?

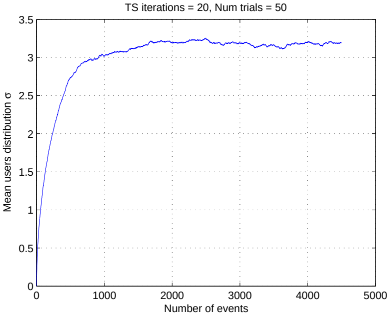


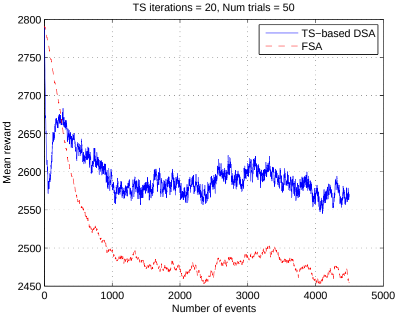
Local algorithm:

- TS is launched on a 19 cell cluster
- Centered where a new event occurs
- Other cell assignments unchanged

Further Work: Infinite Network

- As heterogeneity increases, FSA reward decreases
- +5% reward in favor of DSA





Model:

- Starting point: homogeneous traffic (5 users/cell)
- Arrivals and departure occurs with uniform distribution
- Max number of users/cell is 10

Conclusion

- Various mathematical tools have been tested for the resource allocation problem
- Significant gains can be achieved by considering time and spatial variations of the traffic
- Two frontiers: real-time implementation and infinite network
- New models: green reward, flat rate

References I

[Buddhikot07] M. Buddhikot, "Understanding Dynamic Spectrum Allocation: Models, Taxonomy and Challenges," in *Proc. IEEE DySPAN'07*, pp. 649-663, 2007.

[3gpp] www.3gpp.org

[Filin08] S. Filin et al., "Dynamic Spectrum Assignment and Access Scenarios, System Architecture, Functional Architecture and Procedures for IEEE P1900.4 Management System," in *Proc. IEEE CrownCom'08*, pp. 1-7, 2008.

[Buddhikot05] M. Buddhikot, P. Kolodzy, K. Ryan, J. Evans, and S. Miller, "DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," in *Proc. IEEE WoWMoM'05*, pp. 78-85, 2005.

References II

[ofcom07] Roke Manor, "A Study into Dynamic Spectrum Access - Final Report", Produced for: Ofcom. Against Contract: SES-2005-13 Report No: 72/06/R/353/U, March 2007.

[urc] SYSTEMATIC URC Project, "Field Measurements Processing", D2.2.3, March 2009.

[urc2] SYSTEMATIC URC Project, "Dynamic Spectrum Allocation Algorithms v2", D2.1.3, April 2009.

[Enderle03] N. Enderlé and X. Lagrange, "User Satisfaction Models and Scheduling Algorithms for Packet-Switched Services in UMTS," in *Proc. VTC'03*, vol. 3, pp. 1704-1709, 2003.

[Bertsekas07] D. P. Bertsekas, "Dynamic Programming and Optimal Control," third edition, Athena Scientific, 2007.

References III

[Tadepalli98] P. Tadepalli, and D. Ok, "Model-based average reward reinforcement learning," Elsevier, Artificial Intelligence, vol 100, issue 1-2, pp. 177 - 224, 1998.

[Abounadi01] J. Abounadi and D. Bertsekas, "Learning algorithms for Markov decision processes with average cost," SIAM Journal on Control and Optimization, vol 40, Issue 3, pp. 681-698, 2001.

[Watkins89] C.J. Watkins, "Learning from Delayed Rewards", Ph.D. Thesis, Kings College, Cambridge, England, 1989.

[Gosavi04] A. Gosavi, "Reinforcement learning for long-run average cost," Elsevier, European journal of operational research, Traffic and Transportation Systems Analysis, pp. 654-674, 2004.

[Glover89] F. Glover, "Tabu Search, Parti I", ORSA Journal on Computing, vol. 1, 1989.

References IV

[TNRBF] ANFR, Tableau National de Répartition des Bandes de Fréquences, www.anfr.fr