Adaptive Modulation
with Smoothed Flow Utility

Ekine Akuiyibo  Stephen Boyd

Electrical Engineering Department, Stanford University

ITMANET 05/24/10
The problem

• choose transmit power(s) and flow rate(s) to optimally trade off average utility and power

• utilities are functions of *time-smoothed* flow rates

• with each flow we associate a
  – smoothing time scale
  – concave increasing utility function

• our model:
  – channel gains are random
  – no interference
Smoothed flow utility

- wireless link supports $n$ flows in period $t$
- $f_t \in \mathbb{R}^n$ is flow rate vector
- $s_t \in \mathbb{R}^n$ is smoothed flow rate vector: $s_{t+1} = \Theta s_t + (I - \Theta)f_t$
  - $\Theta = \text{diag}(\theta)$, $\theta_j \in [0, 1)$
  - $T_j = 1 / \log(1/\theta_j)$ is smoothing time for flow $j$
- $U : \mathbb{R}^n \to \mathbb{R}$: separable concave utility function
- smoothed flow utility is

$$
\bar{U} = \lim_{N \to \infty} \mathbb{E} \frac{1}{N} \sum_{\tau=0}^{N-1} U(s_{\tau})
$$
Channel model and average power

- capacity in period $t$ is (up to a constant) $\log(1 + g_t p_t)$
  - $p_t \geq 0$ is transmit power
  - $g_t$ is channel gain (up to constant)

- power required to support flow $f_t$: $p_t = \phi(1^T f_t, g_t) = (e^{1^T f_t} - 1)/g_t$

- average power is $\bar{P} = \lim_{N \to \infty} E \frac{1}{N} \sum_{\tau=0}^{N-1} p_\tau$

- $g_t$ IID exponential (for example)

- $f_t$ (and therefore $p_t$) can depend on $g_t$, but not $g_{t+1}, g_{t+2}, \ldots$
Optimal policy

- (state feedback) policy: \( f_t = \varphi(s_t, g_t) \)
- goal: choose policy \( \varphi \) to maximize \( \bar{U} - \lambda \bar{P} \)
- \( \lambda > 0 \) is used to trade off average utility and power
- a convex stochastic control problem
- optimal value is \( J^* \)
General ‘solution’ via dynamic programming

- optimal policy is

$$\varphi^*(z, g) = \arg\max_{w \geq 0} \left\{ V^*(\Theta z + (I - \Theta)w) - \lambda \phi(1^T w, g) \right\}$$

- $V^*$ is value function, (any) solution of Bellman equation

$$J^* + V^*(z) = \mathbb{E} \left\{ U(z) + \max_{w \geq 0} \left\{ V(\Theta z + (I - \Theta)w) - \lambda \phi(1^T w, g) \right\} \right\}$$
Value iteration

1. update unnormalized estimate of $V^*$:

$$\tilde{V}^{(k+1)}(z) := \mathbf{E}\left\{ U(\tilde{z}) + \max_{w \geq 0} \left\{ V^{(k)}(\Theta z + (I - \Theta)w) - \lambda \phi(1^T w, g) \right\} \right\}$$

2. normalize (and get new estimate of $J^*$):

$$J^{(k+1)}(z) := \tilde{V}^{(k+1)}(0); \quad V^{(k+1)}(z) := \tilde{V}^{(k+1)}(z) - J^{(k+1)}$$

- $V^{(k)} \to V^*$, $J^{(k)} \to J^*$
- iteration preserves concavity, monotonicity, so $V^*$ is concave, increasing
- can carry out numerically for $n$ very small (say, 1 or 2)
No transmit region

- from convex analysis, $\varphi^*(z, g) = 0$ if and only if

$$g \nabla V^*(\Theta z) \leq \left(\frac{\lambda}{1 - \theta_1}, \ldots, \frac{\lambda}{1 - \theta_n}\right)$$

(assuming here $V^*$ is differentiable)

- interpretation: don’t transmit if
  - channel is bad ($g$ small)
  - or, smoothed flows are large ($z$ large $\Rightarrow \nabla V^*(\Theta z)$ small)
Single-flow examples

- two examples:
  - light smoothing \((T = 1; \theta = 0.37)\)
  - heavy smoothing \((T = 50; \theta = 0.98)\)

- \(U(s) = s^{1/2}; g_t \sim \mathcal{E}(1)\)

- \(\lambda\)s chosen to yield \(\bar{U} = 0.8\)

- (optimal) average power is \(\bar{P} = 0.9\) for \(T = 1\); \(\bar{P} = 0.3\) for \(T = 50\)
  - smoothing allows \(3\times\) reduction in power
Optimal policies
Sample power trajectories
ADP policy for multiple flows

- for more than 3 flows (say), computing $V^*$ intractable
- approximate dynamic programming (ADP) policy:

$$
φ^{adp}(z, g) = \arg\max_{w \geq 0} \{V^{adp}(\Theta z + (I - \Theta)w) - \lambdaφ(1^T w, g)\}
$$

- $V^{adp}$ is an approximate or surrogate value function
- ADP can work surprisingly well, even when $V^{adp}$ is not a particularly good approximation of $V^*$
- some general methods for coming up with a surrogate:
  - use exact value function for simpler problem
  - learning (e.g., Q-learning) or optimization over a basis
Separable surrogate value function

- we propose surrogate value function
  \[ V^{\text{adp}}(z) = V_1^*(z_1) + \cdots + V_n^*(z_n) \]
  - \( V_j^* : \mathbb{R} \rightarrow \mathbb{R} \) is value function for \( j \)th flow alone
  - can evaluate \( \varphi^{\text{adp}}(z, g) \) very fast via waterfilling

- \( V^{\text{adp}} \) is separable, but policy \( \varphi^{\text{adp}} \) is not

- (optimizing over basis of separable surrogate value functions yields very little performance improvement)

- policy \( \varphi^{\text{adp}} \) seems to work well . . . but how suboptimal is it?
An upper bound on $J^*$

- we relax (i.e., ignore) causality requirement, i.e., we have complete knowledge of future channel gains

- for each channel gain realization, results in (large, but convex) multi-period optimization problem

- expected value of optimal cost (obtained by Monte Carlo simulation) is upper bound on $J^*$

- called prescient bound $J^{\text{pre}}$ (since it assumes future is known)
Numerical example

• two flows on a single link, with light \((T = 1)\) and heavy \((T = 50)\) smoothing

\[
U(s_1, s_2) = s_1^{1/2} + s_2^{1/2}; \quad \phi(f_t, g_t) = \lambda/g_t(e^{1T}f_t - 1)
\]

• \(g_t \sim \mathcal{E}(1), \lambda = 1\)

• we run 1000 realizations, each of length \(N = 1000\)

• \(J_{\text{adp}} = -13.9; J_{\text{pre}} = -13.8\)

• so \(J_{\text{adp}}\) is at most 0.1-suboptimal
Final observations

• time smoothing has great affect on
  – optimal policy
  – average power needed

• rough interpretation of optimal policy:
  – with smoothing, wait for good channel, unless desperate
  – and so, save power
  – more smoothing $\Rightarrow$ more opportunistic, less power

• multi-flow ADP policy
  – surrogate is sum of single-flow value functions
  – performance is nearly optimal, as shown by upper bound on $J^*$