Dynamic Rate Control Algorithms for HDR Throughput Optimization

Sem Borst, Phil Whiting

Mathematics Research Center, Bell Laboratories
Preliminaries

The HDR Concept

- Time divided into 1.67ms slots
- Pilot signals enable channel prediction
- Users scheduled one at a time within a cell

Our Approach

- Apply Max-Min Fair Rule using Throughput Targets: \( \max \min_{m} \frac{y_{m}}{\alpha_{m}} \)
- Utilise Weights - Shadow Costs
- Allow for statistical dependence between user rate declarations
- Weights determined via Stochastic Control
An Optimality Principle

\[ Y_m(n) = X_m(n)R_m(n), m = 1, \ldots, M \]
\[ y_m(N) = \frac{\mathbb{E} \sum_{n=1}^{N} Y_m(n)}{N} \]

where \( X_m(n) \) are binary 0-1 indicator variables. The objective is to maximise some \( H(y_1, \ldots, y_M) \) where \( y_m = \lim \inf y_m(N) \) where \( H \) is increasing.

Throughput Balancing

**Principle 0.1** If \( \exists w \geq 0 \) such that

\( i) \quad w^T y \) is maximal and
\n\( ii) \quad y_1 = \cdots = y_M \)

then \( y \) is optimal max-min fair.

---

**Figure 2**: Applying the Revenue Vector \( w \)
Existence of Optimal Revenue Vector $w^*$

This says we can find an optimal policy by solving a linear program using the stationary probabilities.

$R_{ij}$ be the rate that user $i$ would receive when the system is in state $j$ and $p_j$ the corresponding stationary probabilities.

**Lemma 0.1** Policy $\pi$ is optimal iff $x_{ij}^\pi$, $z^\pi$ are an optimal solution to the following linear program:

\[
\begin{align*}
\text{max} & \quad z \\
\text{sub} & \quad z \leq \sum_{j \in J} p_j R_{ij} x_{ij} \quad i = 1, \ldots, M \\
& \quad \sum_{i=1}^M x_{ij} \leq 1 \quad j \in J \\
& \quad x_{ij} \geq 0 \quad i = 1, \ldots, M, j \in J.
\end{align*}
\]

Given the above, this says that an optimal revenue vector exists.

**Theorem 0.1** If policy $\pi$ is optimal, then there exists a vector $w^* \geq 0$ such that

\[
x_{ij}^\pi \left[ w_{ij}^* R_{ij} - \max_{m=1, \ldots, M} w_{mj}^* R_{mj} \right] = 0,
\]

for all $i = 1, \ldots, M, j \in J$. 
Control Outline I

User selection: Revenue based

- \( m = \arg \max w_k R_k \)
- Price updates based on sample throughputs at current price

Increments: Determines the size of the updates:

- \( \sum_n \delta_n < \infty \)
- Resets – increments changed according to process behaviour
- Each user becomes max, each user becomes above average

Samples: Sets the sample size at the \( n \)th step

- \( K_n, \quad K_n \to \infty \)
- Samples continuously perturbed to avoid tie breaks
Control Outline II

Control Update: Determined recursively

\[ w(n + 1) = w(n) - \delta(n) \cdot v(w(n)) \]

- \(n\) is the \(n\)th measurement period. \(v\) is random.
- \(L(n) = n^\beta, \beta > 0\), number of samples in the \(n\)th period

Throughput Measurements: These are used to determine \(v\)

- \(w(n)\) is fixed during the sample period
- Samples continuously perturbed to avoid tie breaks
- \(X^n_m = \sum_{k=K(n)+1}^{K(n+1)} X_{m,k}\)
- \(X_{m,k}\) total throughput in slot \(k\) for user \(m\). \(X^n_m\) total user throughput

Resets: Reduce the step size

- \(\delta(n) = \delta_{k(n)}\) with \(\{\delta_k, k = 1, 2, \ldots\}\)
- E.g. \(\delta_k = a^{-k}, a > 1\delta_k = k^{-\alpha}, \alpha > 1.0\)
Assumptions

Large Deviations

**Assumption 0.1** (Large-Deviations Assumption)

Let $X_m^n(w)$ be the throughput per slot obtained by user $m$ in a sample period of length $n$ under price vector $w$.

Given a price vector $w \in \mathcal{W}$ and $\xi > 0$, there exist a $\zeta$-neighborhood $N^\zeta(w)$ of $w$ and numbers $D^\zeta_m(w) > 0$ such that

$$\mathbb{P}\{ | X_m^n(w') - \Xi_m(w) | > \xi \} \leq e^{-D^\zeta_m(w)n}$$

for all $w' \in N^\zeta(w)$, $m = 1, \ldots, M$.

Boundary Conditions

**Assumption 0.2** There exists a positive constant $\delta^* > 0$ such that for all price vectors $w \in \mathcal{W}_v$, for any ‘right direction’ $v(w)$, and for any $\delta \in (0, \delta^*)$,

$$w + \delta v(w) \in \mathcal{W}_v.$$

$T$ Function

**Assumption 0.3** There exist positive constants $\delta^* > 0$, $\eta > 0$ such that for all price vectors $w \notin \Gamma_\epsilon$, for any ‘right direction’ $v(w)$, and for any $\delta \in (0, \delta^*)$,

$$T(w + \delta v(w)) \leq T(w) - \delta \eta.$$
Choices for the Lyapunov Function $T$

The first is Max - Min Expected Throughput

$$T(w) = \Xi_{\text{max}}(w) - \Xi_{\text{min}}(w).$$

This can be shown to be a Lyapunov function with the move to average algorithm.

The second choice that we consider is Expected Revenue

$$T(w) = \sum_{m=1}^{M} w_m \Xi_m(w).$$

To see this is a Lyapunov function consider $i, j$ with

$$\Xi_i(w') < \Xi_j(w')$$

and $w' = w + \delta (e_i - e_j)$.

Then

$$T(w') = \sum_{m=1}^{M} w'_m \Xi_m(w')$$

$$= \sum_{m=1}^{M} w_m \Xi_m(w') + \delta (\Xi_i(w') - \Xi_j(w'))$$

$$\leq \sum_{m=1}^{M} w_m \Xi_m(w) + \delta (\Xi_i(w') - \Xi_j(w'))$$

$$= T(w) + \delta (\Xi_i(w') - \Xi_j(w'))$$
Two Algorithms

Move to Average
The update direction \( v(w) \) is determined using the below average set \( \Omega^- \) and the above average set \( \Omega^+ \).

\[
\begin{align*}
v_i(w) &= \frac{w_i}{\sum_{m \in \Omega^-} w_m}, \quad i \in \Omega^- \\
v_j(w) &= \frac{-w_j}{\sum_{k \in \Omega^+} w_k}, \quad j \in \Omega^-
\end{align*}
\]

Update Extreme
Increment the minimum user and decrement the maximum user:

\[
\begin{align*}
i(1) &= \arg \min_{m=1,\ldots,M} O_m(k) \\
i(2) &= \arg \max_{m=1,\ldots,M} O_m(k)
\end{align*}
\]
(R_m', R_k') \text{ revenue } w = w_m R_m + w_k R_k \\
(R'_j, 0) \text{ revenue } w = w R'_j \quad \text{B switched off} \\
(0, R'_n) \text{ revenue } w = w R'_n \quad \text{A switched off} 

Figure 4: Fast Power Control

Two Cells using On-Off Power Control

A Coordination Example

- Power Control Decisions via the Prices
- Predictions necessary for each state
Numerical Results

\( w \) for Two Users, Independent Exponential \( w \) For two Users

Figure 5: Normalized expected throughput \( \Xi_i(w) \) as function of \( w \).

Figure 6: Price trajectories for 2 users vs. \( w^* \) (non-geometric step sizes).
Rayleigh Fading $w$ for Eight Users

Figure 7: Sample Rayleigh fading

Figure 8: Price trajectories for 8 users over 15000 slots vs. $w^*$ (Move-to-Average algorithm).
Eight Users Continued

Figure 9: Price trajectories for 8 users with 15,000 slots vs. $w^*$ (Update-Extreme algorithm).
Figure 10: Cycled control: lowered SNR, user 3 (Move-to-Average algorithm with $\delta_k = k^{-2}$).
Conclusions

- Achievable rates determined by using revenue vector plus throughput balancing.
- This principle applies generally not just for one-at-a-time scheduling.
- Impractical to estimate optimal weight from empirical channel statistics.
- Wide range of stochastic approximation algorithms can be used to determine revenue vector for given targets using throughput balancing.
- Algorithms may be used for admission control and coordinated operation.
- Proportional fair is a revenue based algorithm with revenue proportional to the reciprocal throughput.
- The proportional fair algorithm converges to a unique fixed point.
- Fixed point lies on the boundary and maximise sum log throughputs.