Budget Allocation for Sequential Customer Engagement

Craig Boutilier, Google Research,
Mountain View

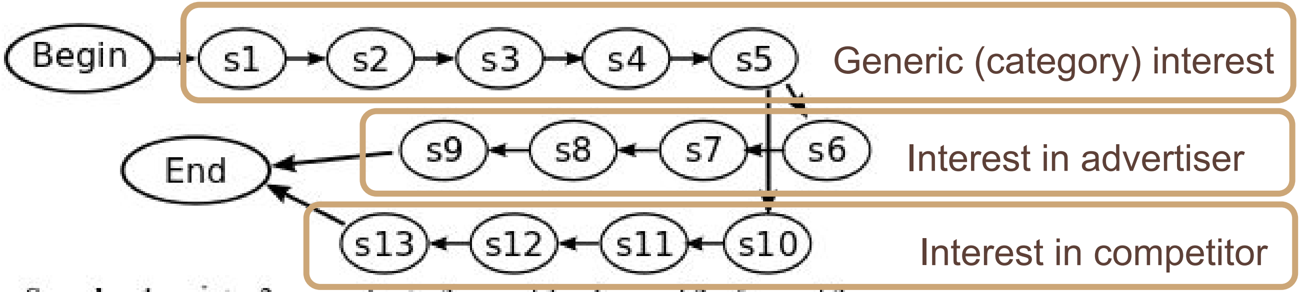
(joint work with Tyler Lu)

We're hiring:
https://sites.google.com/site/icmlconf2016/careers

Google™

Sequential Models of Customer Engagement

- Sequential models of marketing, advertising increasingly common
  - Archak, et al. (WWW-10)
  - Silver, et al. (ICML-13)
  - Theocarous et al. (NIPS-15), ...
  - Long-term value impact: Hohnhold, O'Brien, Tang (KDD-15)



Search: s1: unint; s2: general int; s3: search1, s4: search2, s5:search3
Advertiser: s6: interest1; s7: interest2; s8: interest3, s9: conversion
Compt'r: s10: interest1; s11: interest2; s12: interest3, s13: conversion

Sequential Models of Customer Engagement

- New focus at Google on RL, MDP models
  - sequential engagement optimization: ads, recommendations, notifications, ...
  - RL, MDP (POMDP?) techniques beginning to scale

Sequential Models of Customer Engagement

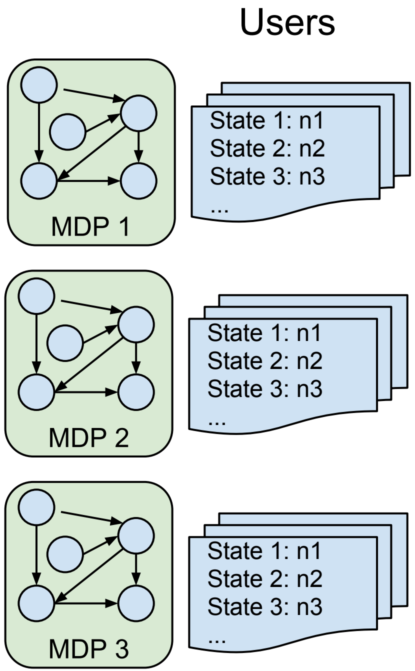
- New focus at Google on RL, MDP models
  - sequential engagement optimization: ads, recommendations, notifications, ...
  - RL, MDP (POMDP?) techniques beginning to scale
- But multiple wrinkles emerge in practical deployment
  - Budget, resource, attentional constraints
  - Incentive, contract design
  - Multiple objectives (preference assessment/elicitation)

This Work

- ❏ **Focus:** handling budget constraints in large MDPs
- ❏ **Motivation:** advertising budget allocation for large advertiser
- ❏ **Aim 1:** find “sweet spot” in spend (value/spend trade off)
- ❏ **Aim 2:** allocate budget across large customer population

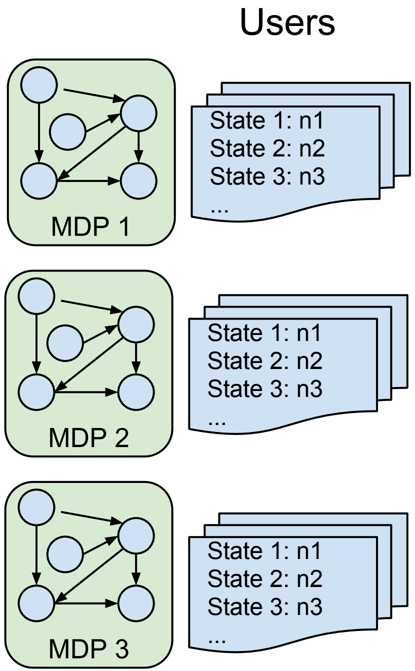
Basic Setup

- Set of *m* MDPs (each corresp. to a "user type")
  - States *S*, actions *A*, trans *P(s,a,s')*, reward *R(s)*, cost *C(s,a)*
  - Small MDPs, solvable by DP, LP, etc.
- Collection of *U* users
  - User *i* is in state *s[i]* of MDP *M[i]*
  - Assume state is fully observable



Basic Setup

- Set of *m* MDPs (each corresp. to a “user type”)
  - States *S*, actions *A*, trans *P(s,a,s')*, reward *R(s)*, cost *C(s,a)*
  - Small MDPs, solvable by DP, LP, etc.
- Collection of *U* users
  - User *i* is in state *s[i]* of MDP *M[i]*
  - Assume state is fully observable
- Advertiser has maximum budget *B*
- **What is optimal use of budget?**
  - Policy mapping ***joint*** state to ***joint*** action
  - Expected spend less than *B*



Potential Methods for Solving MDP

- ❑ Fixed budget (per cust.), solve constrained MDP **(Archak, et al. WINE-12)**
  - ❑ **Plus:** nice algorithms for CMDPs under mild assumptions
  - ❑ **Minus:** no tradeoff between budget/value, no coordination across customers

Potential Methods for Solving MDP

- Fixed budget (per cust.), solve constrained MDP **(Archak, et al. WINE-12)**
  - **Plus:** nice algorithms for CMDPs under mild assumptions
  - **Minus:** no tradeoff between budget/value, no coordination across customers
- Joint, constrained MDP (cross-product of individual MDPs)
  - **Plus:** optimal model, full recourse
  - **Minus:** dimensionality of state/action spaces make it intractable

Potential Methods for Solving MDP

- Fixed budget (per cust.), solve constrained MDP **(Archak, et al. WINE-12)**
  - **Plus:** nice algorithms for CMDPs under mild assumptions
  - **Minus:** no tradeoff between budget/value, no coordination across customers
- Joint, constrained MDP (cross-product of individual MDPs)
  - **Plus:** optimal model, full recourse
  - **Minus:** dimensionality of state/action spaces make it intractable
- We exploit **weakly coupled nature of MDP** **(Meuleau, et al. AAAI-98)**
  - No interaction except through budget constraints

Decomposition of a Weakly-coupled MDP

- Offline: solve budgeted MDPs
  - ** Solve each distinct MDP (user type); get VF $V(s,b)$ and policy $\pi(s,b)$
  - Notice value is a function of state ***and available budget*** b

Decomposition of a Weakly-coupled MDP

- Offline: solve budgeted MDPs
  - ** Solve each distinct MDP (user type); get VF $V(s,b)$ and policy $\pi(s,b)$
  - Notice value is a function of state ***and available budget*** *b*
- Online: allocate budget to maximize return
  - Observe state of each user *s[i]*
  - ** Optimally allocate budget *B*, with *b*[i]* to user *i*
  - Implement optimal budget-aware policy

Decomposition of a Weakly-coupled MDP

- Offline: solve budgeted MDPs
  - ** Solve each distinct MDP (user type); get VF $V(s,b)$ and policy $\pi(s,b)$
  - Notice value is a function of state ***and available budget*** b
- Online: allocate budget to maximize return
  - Observe state of each user $s[i]$
  - ** Optimally allocate budget B, with $b^*[i]$ to user i
  - Implement optimal budget-aware policy
- Optional: repeated budget allocation
  - Take action $\pi(s[i],b^*[i])$, with cost $c[i]$
  - Repeat (re-allocate all unused budget)

Outline

- Brief review of ***constrained MDPs (CMDPs)***
- Introduce ***budgeted MDPs (BMDPs)***
  - Like a CMDP, but without a fixed budget
  - DP solution method/approximation that exploits PWLC value function
- Distributed ***budget allocation***
  - Formulate as a multi-item, multiple-choice knapsack problem
  - Linear program induces a simple (and optimal) greedy allocation
- Some empirical (prototype) results

Constrained MDPs

- Usual elements of an MDP, but distinguish rewards, costs
  - Optimize value subject to an *expected budget constraint B*
  - Optimal (stationary) policy usually stochastic, non-uniformly optimal
  - Solvable by LP, DP methods

$$V^{\pi}(i) = r_i^{\pi(i)} + \gamma \sum_{j \in S} p_{ij}^{\pi(i)} V^{\pi}(j).$$

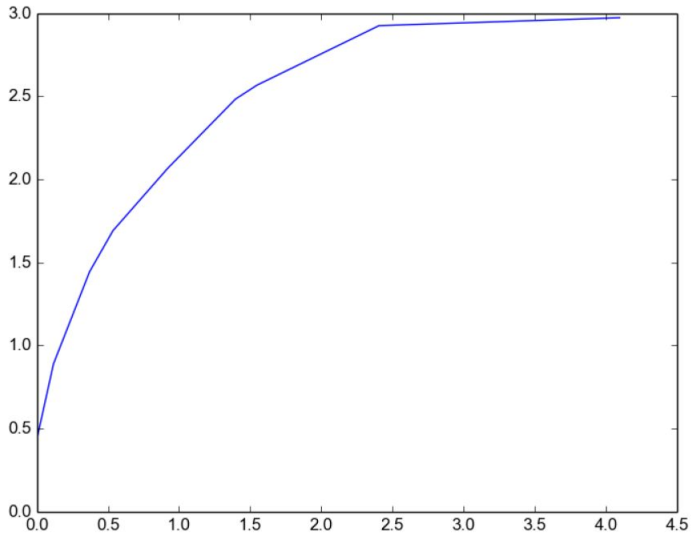
$$C^{\pi}(i) = c_i^{\pi(i)} + \gamma \sum_{j \in S} p_{ij}^{\pi(i)} C^{\pi}(j).$$

$$\operatorname*{argmax}_{\pi} \ \alpha_i V^{\pi}(i) \ \text{ s.t. } \ \alpha_i C^{\pi}(i) \leq B.$$

Budgeted MDPs

- CMDP's *fixed* budget doesn't support:
  - Budget/value tradeoffs in MDP
  - Budget tradeoffs across different MDPs

Budgeted MDPs



- CMDP's *fixed* budget doesn't support:
  - Budget/value tradeoffs in MDP
  - Budget tradeoffs across different MDPs
- ***Budgeted MDPs***
  - Want optimal VF *V(s,b)* of MDP given state *and budget*
  - A variety of uses (value/spend tradeoffs, online allocation)
  - Aim: find structure in continuous dimension *b*

Structure in BMDP Value Functions

- **Result 1:** For all s, VF is concave, non-decreasing in budget

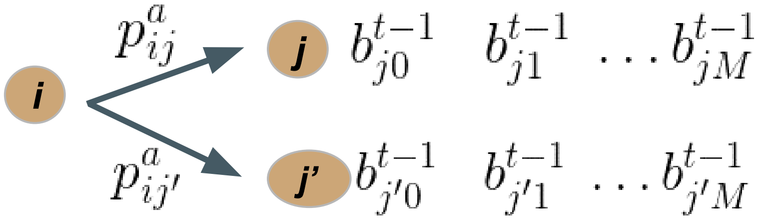
Structure in BMDP Value Functions

- **Result 1:** For all s, VF is concave, non-decreasing in budget
- **Result 2** (finite-horizon): VF is piecewise linear, concave (PWLC)
  - Finite number of useful (deterministic) budget levels
  - Randomized policies achieve “interpolation” between points
  - Simple dynamic program finds finite representation (i.e., PWL segments)
  - Complexity: representation can grow exponentially $O((|A|^d)^t)$
  - Simple pruning gives excellent approximations with few PWL segments

BMDPs: Finite deterministic useful budgets

$V_D^t(i, b)$ has finitely many useful budget levels *b* (for any *i*, *t*)

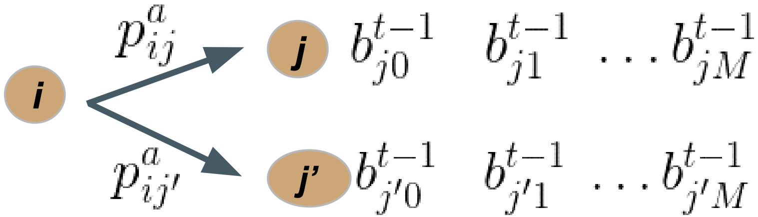
- "Next budget used" $\sigma : S_i^a \rightarrow [M]$



BMDPs: Finite deterministic useful budgets

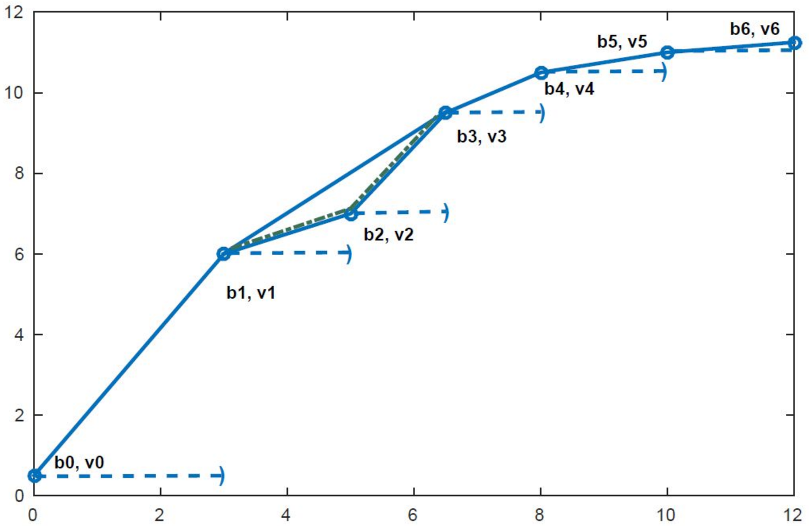
$V_D^t(i, b)$ has finitely many useful budget levels *b* (for any *i*, *t*)

- "Next budget used" $\sigma : S_i^a \to [M]$



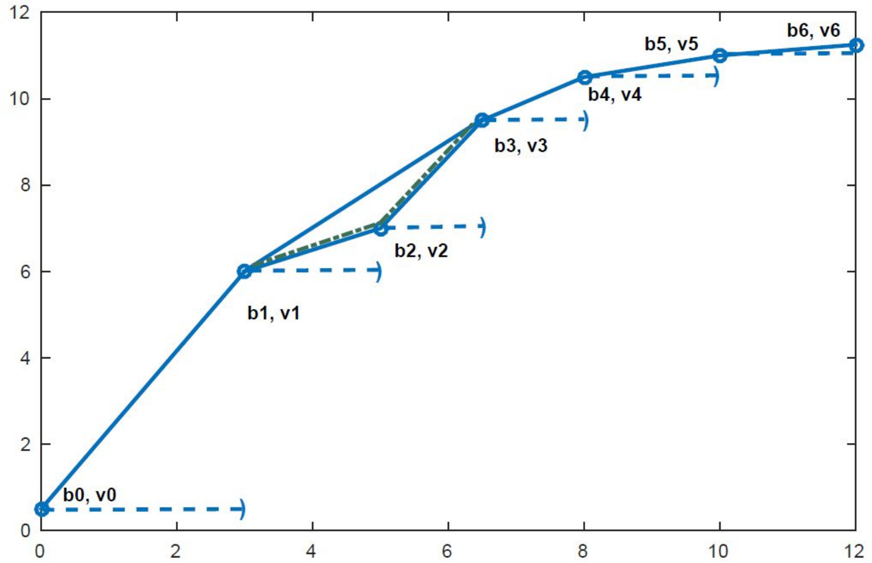
- Has cost: $$c_i^a + \sum_{j \in S_i^a} p_{ij}^a b_{\sigma(j)}^{j,t-1}$$

- Has value: $v_k^{i,t} = r_i^a + \gamma \sum_j p_{ij}^a v_{\sigma(j)}^{j,t-1}$



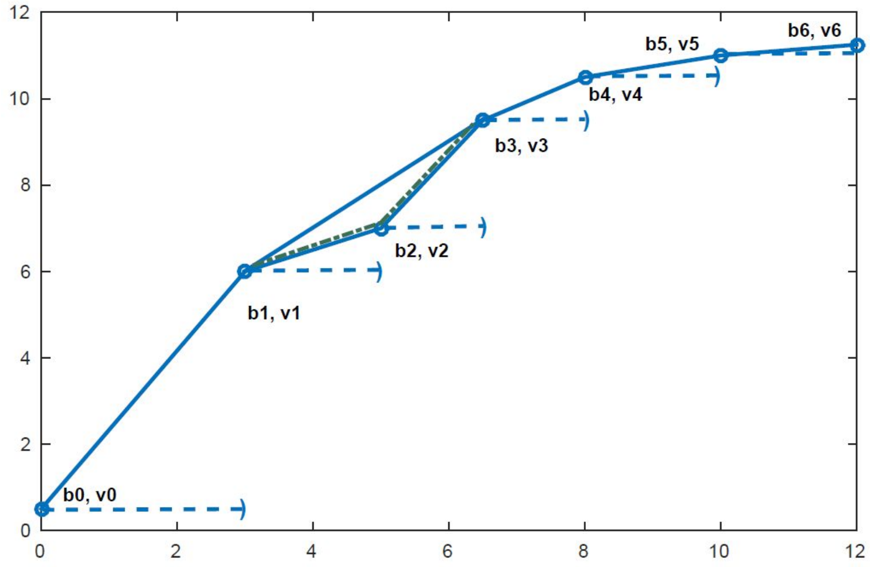
Budgeted MDPs: PWLC with Randomization

- Take union over actions, prune dominated budgets
  - Gives natural DP algorithm



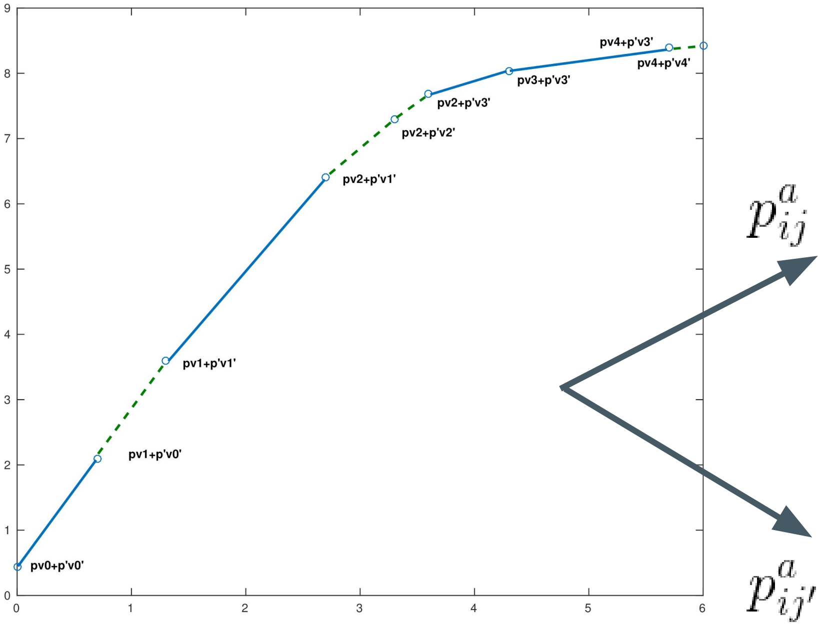
Budgeted MDPs: PWLC with Randomization

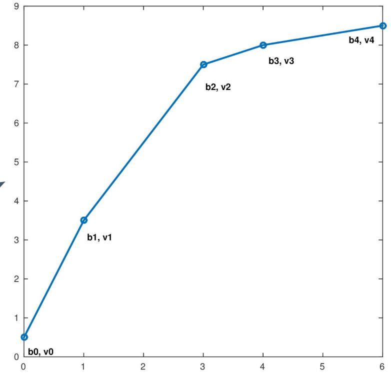
- Take union over actions, prune dominated budgets
  - Gives natural DP algorithm
- Randomized spends (actions) improve expected value
  - PWLC rep'n (convex hull) of deterministic VF
- A simple greedy approach gives Bellman backups of stochastic value functions

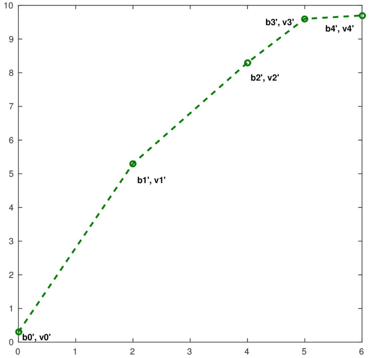


Budgeted MDPs: Intuition behind DP

Finding Q-values:







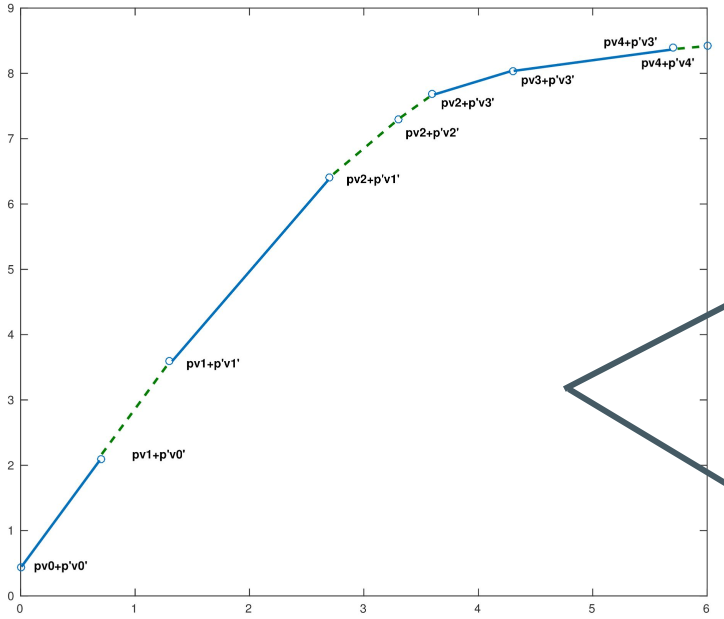
$$Q^t(i,a,b) = \max_{\mathbf{b} \in R_+^n} + \gamma \sum_{j \leq n} p_{ij}^a V^{t-1}(j, b_j)$$

$$\text{subj. to } c_i^a + \gamma \sum_{j \leq n} p_{ij}^a b_j \leq b$$

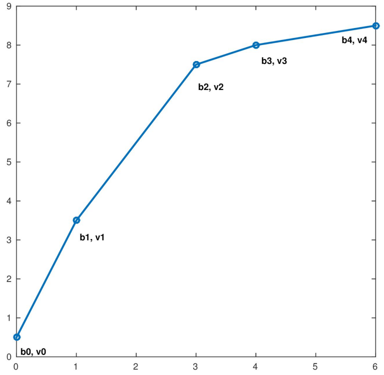
Budgeted MDPs: Intuition behind DP

Finding Q-values:

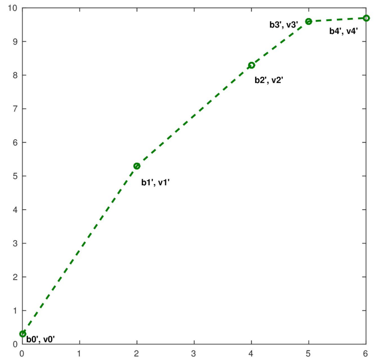
- ❑ Assign incremental budget to successor states in ***decr. order*** of slope of V(s), or “bang-per-buck”
- ❑ Weight by transition probability
- ❑ Ensures finitely many PWLC segments



p_{ij}^a



$p_{ij'}^a$



$$Q^t(i,a,b) = \max_{\mathbf{b}\in R_+^n} +\gamma \sum_{j\leq n} p_{ij}^a V^{t-1}(j,b_j)$$

$$\text{subj. to } c_i^a + \gamma \sum_{j\leq n} p_{ij}^a b_j \leq b$$

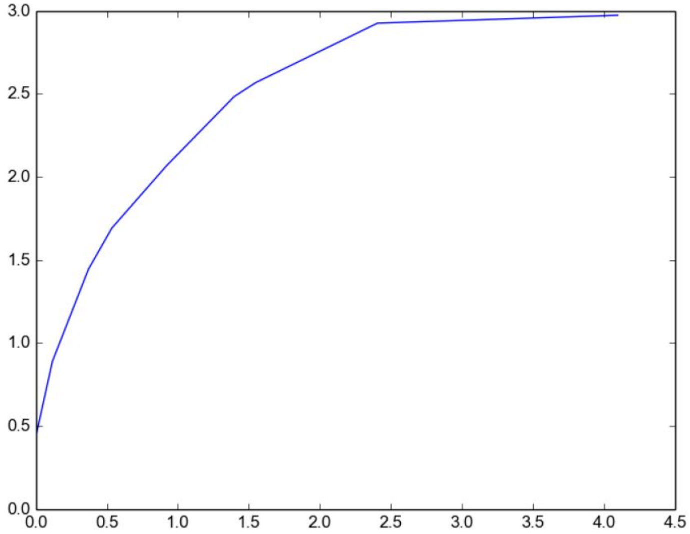
Budgeted MDPs: Intuition behind DP

Finding VF (stochastic policies):

$$V^t(i,b) = \max_{\mathbf{p}\in\Delta(a)} p_a Q^t(i,a,b_a)$$

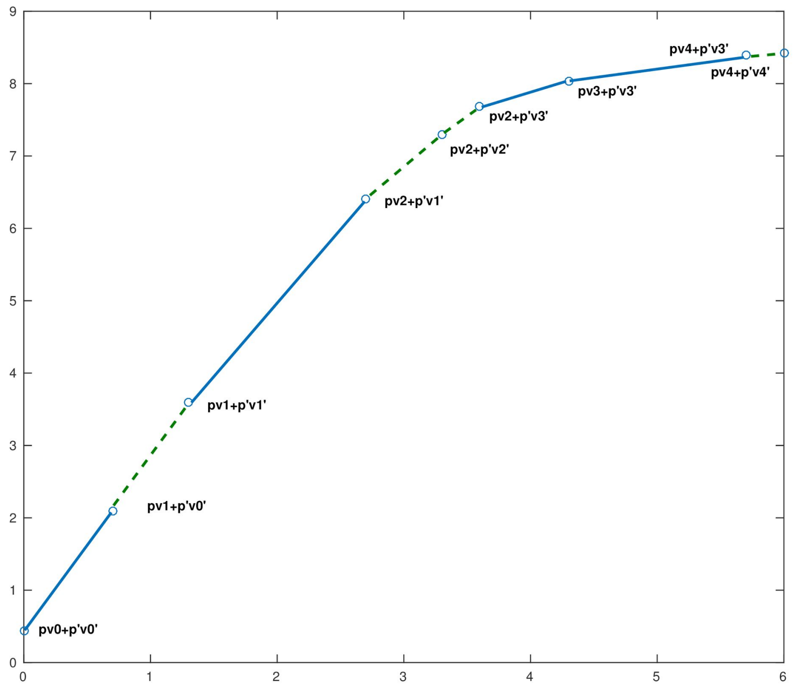
$$\text{s.t.} \sum_a p_a b_a \leq b$$

- ❑ Take union of all Q-functions, remove dominated points, obtain convex hull



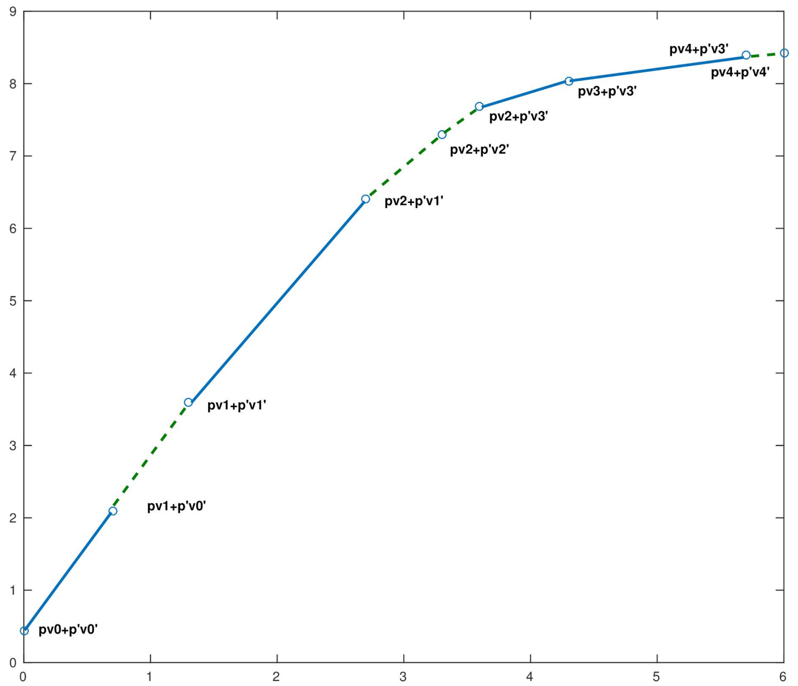
Approximation

- Simple pruning scheme for approx.
  - Budget gap between adjacent points small
  - Slopes of two adjacent segments close
  - Some combination (product of gap, delta)



Approximation

- Simple pruning scheme for approx.
  - Budget gap between adjacent points small
  - Slopes of two adjacent segments close
  - Some combination (product of gap, delta)
- Integrate pruning directly into convex hull algorithm
- Error bounds derivable (*computable*)
- Hybrid scheme seems to work best
  - Aggressive pruning early
  - Cautious pruning later
  - Exploit contraction properties of MDP

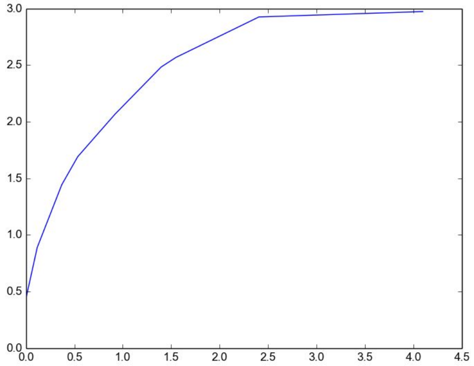


Policy Implementation and Spend Variance

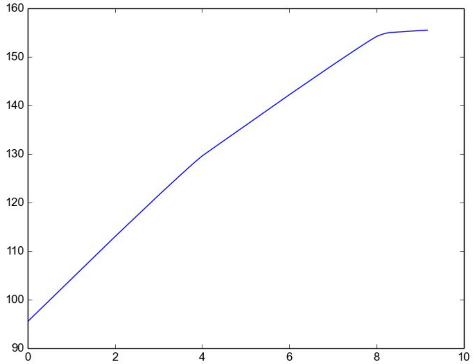
- Policy execution somewhat subtle
- Must track (final) budget mapping (from each state
  - Must implement spend “assumed” at next reached state
  - Essentially “solves” CMDP for all budget levels
- Variance in actual spend may be of interest
  - Recall we satisfy budget in expectation only
  - Variance can be computed exactly during DP algorithm (expectation of variance over sequence of multinomials)

Budgeted MDPs: Some illustrative results

- Synthetic 15-state MDP (search/sales funnel)
  - States reflect interest in general, advertiser, competitor(s)
  - 5 actions (ad intensity) with varying costs
- Optimal VF (horizon 50):



	No pruning	Mild	Aggressive	Mild then No
Segments	3066 (0–5075)	18.3 (0–47)	10.4 (0–26)	480.8 (0–877)
Max. Err.	—	4.84 (26.61)	4.84 (26.61)	0.21 (58.77)
Max. Rel. Err.	—	40.9% (4.24)	48.7% (1.54)	2.3% (0.55)
CPU Time (s.)	1055.4	17.54	10.36	28.67



Budgeted MDPs: Some illustrative results

- “MDP” derived from advertiser data
  - 3.6M “touchpoint” trajectories (28 distinct events)
  - VOMC model/mixture learned
  - 452K states / 1470 states; hypothesized actions, synthetic costs
  - Unsatisfying models: *not too controllable (*opt. policies mostly by no-ops)

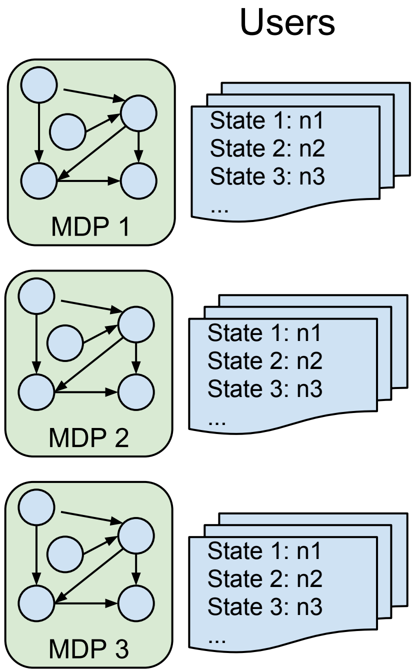
Budgeted MDPs: Some illustrative results

- "MDP" derived from advertiser data
  - 3.6M "touchpoint" trajectories (28 distinct events)
  - VOMC model/mixture learned
  - 452K states / 1470 states; hypothesized actions, synthetic costs
  - Unsatisfying models: *not too controllable* (opt. policies mostly by no-ops)
- Large model (aggr. prun.): 11.67 segs/state; 1168s/iteration

	No pruning (1469)	Mild (1469)	Aggr. (1469)	Mild then No (1469)
Segments	251.5 (74–359)	234.2 (77–342)	25.6 (5–39)	76.84 (18–321)
Max. Err.	—	5.13 (171.56)	28.88 (169.33)	3.94 (167.61)
Max. Rel. Err.	—	2.99% (171.56)	12.32% (169.33)	2.35% (167.61)
CPU Time (s.)	19918.9	10672.5	1451.8	2390.0

Online Budget Allocation

- Collection of *U* users each with her own MDP
  - For simplicty, assume a single MDP
  - But each user *i* is in state *s[i]* of MDP *M[i]*
  - State of joint MDP: $|S|$-vector of user counts
- Advertiser has maximum budget *B*
- **What is optimal use of budget?**



Online Budget Allocation

- Optimal VFs, policies for user-level BMDPs used to allocate budget
  - Motivated by Meuleau et al. (1998) weakly coupled model
- Online *budget allocation problem (BAP):*

$$\max_{b[i]:i \leq C} \sum_{i \leq C} V(s[i], b[i]) \quad s.t. \quad \sum_{i \leq C} b[i] \leq B$$

Online Budget Allocation

- Optimal VFs, policies for user-level BMDPs used to allocate budget
  - Motivated by Meuleau et al. (1998) weakly coupled model
- Online *budget allocation problem (BAP):*

$$\max_{b[i]:i\leq C} \sum_{i\leq C} V(s[i], b[i]) \;\; s.t. \;\; \sum_{i\leq C} b[i] \leq B$$

- Solution is optimal assuming *"expected budget" commitment*
  - Not truly optimal: no recourse ***across*** users
  - Equivalent to: allocate budget; once fixed, "solve" CMDP, implement policy
  - Alternative (later): *dynamic budget reallocation (DBRA)*

Solving the Budget Allocation Problem

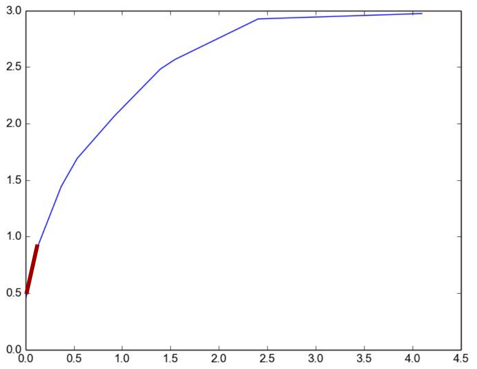
- Multi-item version of *multiple-choice knapsack (MCKP)*
  - Sinha, Zoltners OR79 analyze MCKP as MIP
  - LP relaxation solvable with greedy alg. using “bang-per-buck” metric

Solving the Budget Allocation Problem

- Multi-item version of *multiple-choice knapsack (MCKP)*
  - Sinha, Zoltners OR79 analyze MCKP as MIP
  - LP relaxation solvable with greedy alg. using “bang-per-buck” metric
- Assigning *discrete useful budgets* (UBAP) to users is an MCKP
  - LP relaxation of UBAP is exactly our BAP
  - Greedy method solves BAP (LP relaxation of UBAP) optimally

$$BpB_{jk} = \frac{V(j, \beta_{jk}) - V(j, \beta_{jk-1})}{\beta_{jk} - \beta_{jk-1}}.$$

Bang-per-buck for (user in) state j already allocated useful budget β_{jk-1}

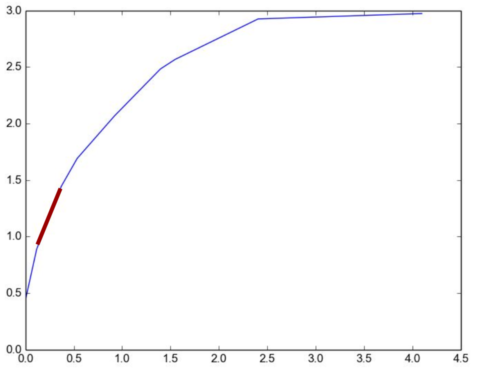


Solving the Budget Allocation Problem

- Multi-item version of *multiple-choice knapsack (MCKP)*
  - Sinha, Zoltners OR79 analyze MCKP as MIP
  - LP relaxation solvable with greedy alg. using “bang-per-buck” metric
- Assigning *discrete useful budgets* (UBAP) to users is an MCKP
  - LP relaxation of UBAP is exactly our BAP
  - Greedy method solves BAP (LP relaxation of UBAP) optimally

$$BpB_{jk} = \frac{V(j, \beta_{jk}) - V(j, \beta_{jk-1})}{\beta_{jk} - \beta_{jk-1}}.$$

Bang-per-buck for (user in) state j already allocated useful budget β_{jk-1}

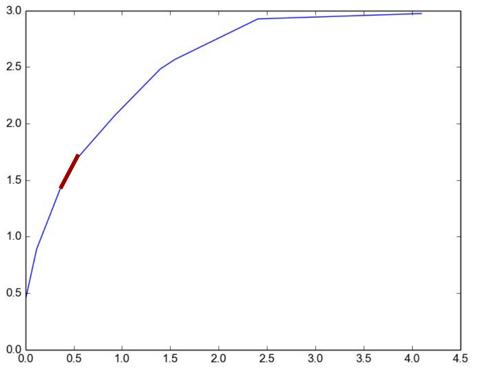


Solving the Budget Allocation Problem

- Multi-item version of *multiple-choice knapsack (MCKP)*
  - Sinha, Zoltners OR79 analyze MCKP as MIP
  - LP relaxation solvable with greedy alg. using “bang-per-buck” metric
- Assigning *discrete useful budgets* (UBAP) to users is an MCKP
  - LP relaxation of UBAP is exactly our BAP
  - Greedy method solves BAP (LP relaxation of UBAP) optimally

$$BpB_{jk} = \frac{V(j, \beta_{jk}) - V(j, \beta_{jk-1})}{\beta_{jk} - \beta_{jk-1}}.$$

Bang-per-buck for (user in) state j already allocated useful budget β_{jk-1}

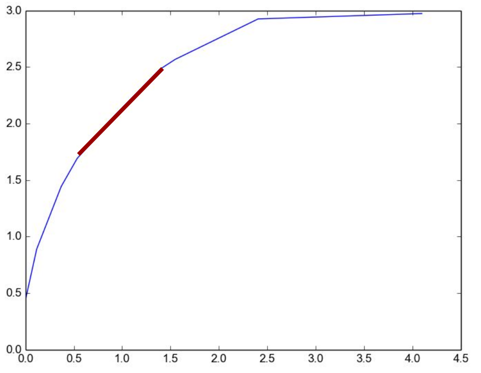


Solving the Budget Allocation Problem

- Multi-item version of *multiple-choice knapsack (MCKP)*
  - Sinha, Zoltners OR79 analyze MCKP as MIP
  - LP relaxation solvable with greedy alg. using “bang-per-buck” metric
- Assigning *discrete useful budgets* (UBAP) to users is an MCKP
  - LP relaxation of UBAP is exactly our BAP
  - Greedy method solves BAP (LP relaxation of UBAP) optimally

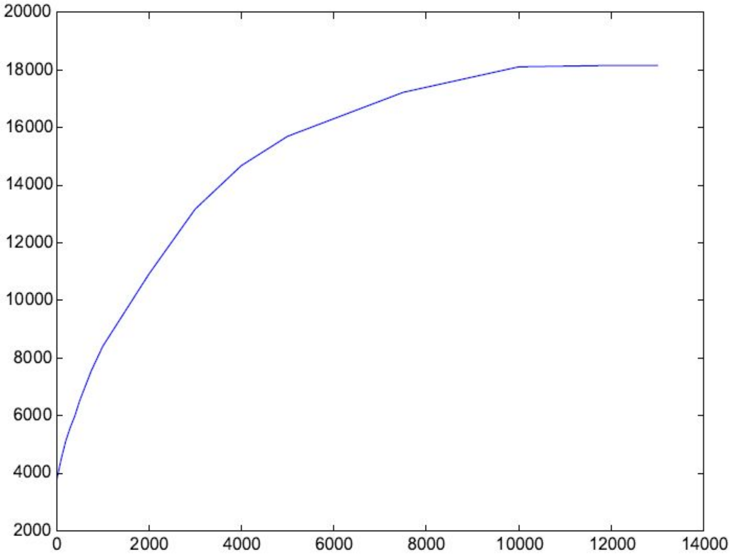
$$BpB_{jk} = \frac{V(j, \beta_{jk}) - V(j, \beta_{jk-1})}{\beta_{jk} - \beta_{jk-1}}.$$

Bang-per-buck for (user in) state j already allocated useful budget β_{jk-1}

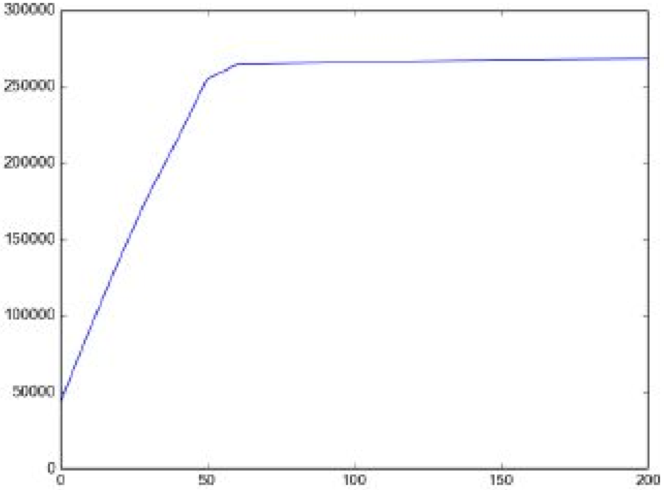


Online Allocation: Illustrative Results

- Fast GBA allows quick determination (ms.) of sweet spot in spend
  - Can directly plot budget-value trade-off curves



15-state synth. MDP, 1000 users



452K-state MDP, 1000 users

Alternative Methods

- **Greedy budget allocation (GBA)**
- **Dynamic budget reallocation (DBRA)** *(see Meuleau et al. (1998))*
  - Perform GBA at each stage, take *immediate* optimal action
  - Observe new state (or each user), re-allocate remaining budget using GBA
  - Allows for recourse, budget re-assignment; Reduces odds of overspending
- **Static user budget (SUB)**
  - Allocate *fixed* budget to each user using GBA at initial state
  - Ignore next-state:budget mapping, enact policy using *remaining* user budget
  - No overspending possible
- **Uniform budget allocation (UBA)**
  - Assign each user the same budget B/M; solve one CMDP per state (no BMDP)

Online Allocation: Illustrative Results

- 15-state synth. MDP, 1000 users (all at initial state)

Total Budget	BMDP Value	DBRA Value	SUB Value
1000	8209.9	8578.8 (830.5)	4106 (707)
2000	10,905	11,019 (964)	4429 (825)
5000	15,692	15,658 (1239)	5270 (830.5)
10,000	18,110	17,942 (—)	6329 (1159)

  - Variance in per-user spend high (e.g., last row: 28.7% of users oversp. >50%)
  - But average across population close to budget
  - DBRA: "guarantees" budget constraint, and can offer some recourse
  - Note: UBA and GBA identical if all users start at same state

Online Allocation: Illustrative Results

- 15-state synth. MDP, 1000 users *(spread over 12 non-term. states)*

Total Budget	GBA Value	UBA Value
1000	39818.6	36997.2
2000	44559.5	40311.8
5000	53177.7	47142.4
10,000	58356.8	53773.8

  - GBA exploits BMDP solution to make tradeoffs across users
  - UBA has no information to differentiate high-value vs. low-value states

Online Allocation: Illustrative Results

- ❏ 452K-state synth. MDP, 1000 users (across 50 initial states)

Budg.	BMDP Val.	DBRA Val.	SUB Val.	UBA Val.
15	113358	99236 (3060)	112879 (1451)	106373
25	157228	142047 (3060)	157442 (2589)	149175

  - ❏ Results more mixed since MDP not very “controllable” (quite random)
  - ❏ UBA (uniform allocation to all users, as if BMDP solution were not available at allocation time, but CMDP solution per-state is available)

Next Steps

- Deriving genuine MDP models from advertiser data
  - Reallocation helps very little with VOMC-MDP (due to hypothesized actions)
- Large MDPs *(feature-based states, actions)*
- Parameterized models, mixtures, ...
- The reinforcement learning setting (unknown model)
- Extensions:
  - Partial (including periodic) observability
  - Censored observations
  - Limited controllability

Applications to Social Choice

- Much of SC involves allocation of resources to population
  - E.g., how to best determine distribution of resources to different area of public policy (health care, education, infrastructure)
- Best use of allocated resources depends on “user-level” MDPs
  - Especially true in dynamic/sequential domains with constrained capacity, e.g., smart grid, constrained medical facilities, other public facilities/infrastructure
  - User’s preferences for particular policies highly variable
- Use of BMDPs can play a valuable role in assessing tradeoffs:
  - Allocation of resources across users within a policy domain
  - Allocation of resources across domains