Resource Optimization for P.C. Members: A Subreview Article

Michael A. Bender[1] Samuel McCauley [1] *Bertrand Simon*[2]
Shikha Singh[1] Frédéric Vivien[2]

1: Stony Brook University, NY, USA.
2: INRIA, ENS Lyon and Univ. Lyon, FR.

8th International Conference on Fun with Algorithms

June 2016

P.C.: need to review dozens of papers in one month

Do it all yourself:

- ☺ Know what to expect
- ☹ Give up social life and face the consequences: alcohol, divorce, depression. . .

Rely on others:

- ☺ Save your social life
- ☹ Maybe they don't want to

P.C.: need to review dozens of papers in one month

Do it all yourself:

- ☺ Know what to expect
- ☹ Give up social life and face the consequences: alcohol, divorce, depression. . .

Rely on others:

- ☺ Save your social life
- ☹ Maybe they don't want to

P.C.: need to review dozens of papers in one month

Do it all yourself:

- ☺ Know what to expect
- ☹ Give up social life and face the consequences: alcohol, divorce, depression. . .

Rely on others:

- ☺ Save your social life
- ☹ Maybe they don't want to

A bit more formally

Two ways to get a review

- expensive: do it yourself
- cheap but uncertain: ask for a subreviewer

Objectives:

- Provide a model to describe this problem
- Design strategies which minimize the expected cost

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability p_j
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent $(R_1, R_2, \ldots, R_k)$
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability p_j
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent $(R_1, R_2, \ldots, R_k)$
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability p_j
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent ($R_1, R_2, \ldots, R_k$)
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability $p_{i,j}$ at round i
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent (R_1, R_2, ..., R_k)
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability $p_{i,j}$ at round i
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent $(R_1, R_2, \ldots, R_k)$
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed

Model

N papers to review

- Attempt to recruit a subreviewer costs 1
- Reviewing yourself paper j costs $C_j \ggg 1$

Review requests

- Usually rejected: probability $p_{i,j}$ at round i
- $p_{i,j}$ is monotonically increasing in i

Gather requests in rounds

- k rounds where several requests can be sent $(R_1, R_2, \ldots, R_k)$
- 1 round to review the remaining papers yourself

Realistic (?) assumptions

- Reviews are accepted/rejected quickly (one round)
- Promised reviews are completed in time

Outline

Outline

1. Optimal strategy for unbounded requests
2. Optimal strategy for bounded requests
3. Unknown rejection probability
4. Conclusion

First case: one request round

Framework

- Linearity of expectation: study papers independently
  $\Longrightarrow$ Focus on one paper (rejection prob. p, self-reviewing cost C)
- $R \in \mathbb{R}^+$: number of requests sent (noninteger allowed)
- Objective: find R minimizing the expected cost

Expected cost: $E(R) = \dots$

Optimal strategy

- if $C \leq \frac{1}{\ln 1/p}$: send $R^* = 0$ requests for a cost of C
- else: send $R^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$ requests, expected cost $E^* = R^* + \frac{1}{\ln 1/p}$

First case: one request round

Framework

- Linearity of expectation: study papers independently
  $\Longrightarrow$ Focus on one paper (rejection prob. p, self-reviewing cost C)
- $R \in \mathbb{R}^+$: number of requests sent (noninteger allowed)
- Objective: find R minimizing the expected cost

Expected cost: $E(R) = R + \ldots$

Optimal strategy

- if $C \leq \frac{1}{\ln 1/p}$: send $R^* = 0$ requests for a cost of C
- else: send $R^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$ requests, expected cost $E^* = R^* + \frac{1}{\ln 1/p}$

First case: one request round

Framework

- Linearity of expectation: study papers independently
  $\Longrightarrow$ Focus on one paper (rejection prob. p, self-reviewing cost C)
- $R \in \mathbb{R}^+$: number of requests sent (noninteger allowed)
- Objective: find R minimizing the expected cost

Expected cost: $E(R) = R + p^R \ldots$

Optimal strategy

- if $C \leq \frac{1}{\ln 1/p}$: send $R^* = 0$ requests for a cost of C
- else: send $R^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$ requests, expected cost $E^* = R^* + \frac{1}{\ln 1/p}$

First case: one request round

Framework

- Linearity of expectation: study papers independently
  $\Longrightarrow$ Focus on one paper (rejection prob. p, self-reviewing cost C)
- $R \in \mathbb{R}^+$: number of requests sent (noninteger allowed)
- Objective: find R minimizing the expected cost

Expected cost: $E(R) = R + p^R C$

Optimal strategy

- if $C \leq \frac{1}{\ln 1/p}$: send $R^* = 0$ requests for a cost of C
- else: send $R^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$ requests, expected cost $E^* = R^* + \frac{1}{\ln 1/p}$

First case: one request round

Framework

- Linearity of expectation: study papers independently
  $\Longrightarrow$ Focus on one paper (rejection prob. p, self-reviewing cost C)
- $R \in \mathbb{R}^+$: number of requests sent (noninteger allowed)
- Objective: find R minimizing the expected cost

Expected cost: $E(R) = R + p^R C$

Optimal strategy

- if $C \leq \frac{1}{\ln 1/p}$: send $R^* = 0$ requests for a cost of C
- else: send $R^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$ requests, expected cost $E^* = R^* + \frac{1}{\ln 1/p}$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R_{k-1}, R_k) = R_1 + p^{R_1} \left(\qquad \qquad \qquad \qquad \right)$$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R_{k-1}, R_k) = R_1 + p^{R_1}\left(R_2 + \cdots + p^{R_{k-2}}\left(\qquad \right)\right)$$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R_{k-1}, R_k) = R_1 + p^{R_1}\left(R_2 + \cdots + p^{R_{k-2}}\left(R_{k-1} + p^{R_{k-1}}\left(\qquad \right)\right)\right)$$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R_{k-1}, R_k) = R_1 + p^{R_1}\left(R_2 + \cdots + p^{R_{k-2}}\left(R_{k-1} + p^{R_{k-1}}\left(R_k + p^{R_k} C\right)\right)\right)$$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R_{k-1}, R_k^*) = R_1 + p^{R_1}\left(R_2 + \cdots + p^{R_{k-2}}\left(R_{k-1} + p^{R_{k-1}}\left(E_k^*\right)\right)\right)$$

Optimal Strategy

- $R_k^* = \ln_{1/p}\left(C \ln \frac{1}{p}\right)$
- $E_k^* = R_k^* + \frac{1}{\ln 1/p}$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1, ..., R^*_{k-1}, R^*_k) = R_1 + p^{R_1}\left(R_2 + \cdots + p^{R_{k-2}}\left(E^*_{k-1}\right)\right)$$

Optimal Strategy

- $R^*_{k-1} = \ln_{1/p}\left(E^*_k \ln \frac{1}{p}\right) = \ln_{1/p}\left(1 + \ln(C \ln \frac{1}{p})\right)$
- $E^*_{k-1} = R^*_{k-1} + \frac{1}{\ln 1/p}$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$$E(R_1^*, ..., R_{k-1}^*, R_k^*) = E_1^*$$

Optimal Strategy

- $R_1^* = \ln_{1/p}\left(E_2^* \ln \frac{1}{p}\right) = \ln_{1/p}\left(1 + \ln\left(1 + \ldots \ln\left(1 + \ln\left(C \ln \frac{1}{p}\right)\right)\ldots\right)\right)$
- $E_1^* = R_1^* + \frac{1}{\ln 1/p}$

Generalisation to k rounds

Note: if $C \leq \frac{1}{\ln 1/p}$ then self-review immediately

Expression of the expected cost

$E(R_1^*, ..., R_{k-1}^*, R_k^*) = E_1^*$

Optimal Strategy

- $R_1^* = \ln_{1/p}\left(E_2^* \ln \frac{1}{p}\right) = \ln_{1/p}\left(1 + \ln\left(1 + \ldots \ln\left(1 + \ln\left(C \ln \frac{1}{p}\right)\right)\ldots\right)\right)$
- $E_1^* = R_1^* + \frac{1}{\ln 1/p}$

Do you look at the deadline before accepting a review?

Monotonically increasing rejection probabilities

Generalization of the previous study: $p_1 \leq p_2 \leq \cdots \leq p_k$

For simplicity: assume $C > \frac{1}{\ln 1/p_k}$ so all $R_i > 0$

Number of requests per round in the optimal solution

- $R_k = \ln_{\frac{1}{p_k}} \left(C \ln \frac{1}{p_k} \right)$
- for $1 \leq i < k$: $R_i = \ln_{\frac{1}{p_i}} \left(E^*_{i+1} \ln \frac{1}{p_i} \right)$ where $E^*_i = R_i + \frac{1}{\ln \frac{1}{p_i}}$
- $R_1 = \ln_{\frac{1}{p_1}} \left(\frac{\ln p_1}{\ln p_2} \left(1 + \ln \left(\frac{\ln p_2}{\ln p_3} \left(1 + \ldots \ln \left(\frac{\ln p_{k-1}}{\ln p_k} \left(1 + \ln \left(C \ln \frac{1}{p_k} \right) \right) \right) \ldots \right) \right) \right) \right)$

Monotonically increasing rejection probabilities

Generalization of the previous study: $p_1 \leq p_2 \leq \cdots \leq p_k$

For simplicity: assume $C > \frac{1}{\ln 1/p_k}$ so all $R_i > 0$

Number of requests per round in the optimal solution

- $R_k = \ln_{\frac{1}{p_k}} \left(C \ln \frac{1}{p_k} \right)$
- for $1 \leq i < k$: $R_i = \ln_{\frac{1}{p_i}} \left(E^*_{i+1} \ln \frac{1}{p_i} \right)$ where $E^*_i = R_i + \frac{1}{\ln \frac{1}{p_i}}$
- $R_1 = \ln_{\frac{1}{p_1}} \left(\frac{\ln p_1}{\ln p_2} \left(1 + \ln \left(\frac{\ln p_2}{\ln p_3} \left(1 + \ldots \ln \left(\frac{\ln p_{k-1}}{\ln p_k} \left(1 + \ln \left(C \ln \frac{1}{p_k} \right) \right) \right) \ldots \right) \right) \right) \right)$

Monotonically increasing rejection probabilities

Generalization of the previous study: $p_1 \le p_2 \le \cdots \le p_k$

For simplicity: assume $C > \frac{1}{\ln 1/p_k}$ so all $R_i > 0$

Number of requests per round in the optimal solution

- $R_k = \ln_{\frac{1}{p_k}} \left(C \ln \frac{1}{p_k} \right)$
- for $1 \le i < k$: $R_i = \ln_{\frac{1}{p_i}} \left(E^*_{i+1} \ln \frac{1}{p_i} \right)$ where $E^*_i = R_i + \frac{1}{\ln \frac{1}{p_i}}$
- $R_1 = \ln_{\frac{1}{p_1}} \left(\frac{\ln p_1}{\ln p_2} \left(1 + \ln \left(\frac{\ln p_2}{\ln p_3} \left(1 + \ldots \ln \left(\frac{\ln p_{k-1}}{\ln p_k} \left(1 + \ln \left(C \ln \frac{1}{p_k} \right) \right) \right) \ldots \right) \right) \right) \right)$

Monotonically increasing rejection probabilities

Generalization of the previous study: $p_1 \leq p_2 \leq \cdots \leq p_k$

For simplicity: assume $C > \frac{1}{\ln 1/p_k}$ so all $R_i > 0$

Number of requests per round in the optimal solution

- $R_k = \ln_{\frac{1}{p_k}} \left(C \ln \frac{1}{p_k} \right)$
- for $1 \leq i < k$: $R_i = \ln_{\frac{1}{p_i}} \left(E^*_{i+1} \ln \frac{1}{p_i} \right)$ where $E^*_i = R_i + \frac{1}{\ln \frac{1}{p_i}}$
- $R_1 = \ln_{\frac{1}{p_1}} \left(\frac{\ln p_1}{\ln p_2} \left(1 + \ln \left(\frac{\ln p_2}{\ln p_3} \left(1 + \ldots \ln \left(\frac{\ln p_{k-1}}{\ln p_k} \left(1 + \ln \left(C \ln \frac{1}{p_k} \right) \right) \right) \ldots \right) \right) \right) \right)$

OK, we optimized a function, but can you send 3.14159265 emails?

Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**
$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}} (E \ln \frac{1}{p_i}) \right\rfloor$
if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$
$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of the last round in function of R_2

Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

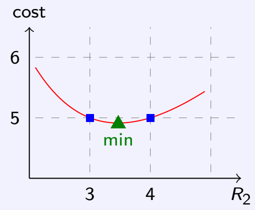
$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**

$R_2 \leftarrow \lfloor 3.47 \rfloor$

if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$

$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of the last round in function of R_2



Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 0$, $R_2 = 3$
- cost: $E = 16$

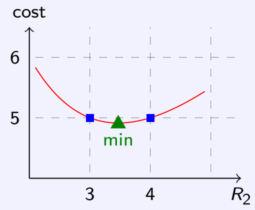
Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**
$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}} (E \ln \frac{1}{p_i}) \right\rfloor$
if $2 > 2$ **then** $R_i \leftarrow R_i + 1$
$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of the last round in function of R_2



Instance: $p_1 = \frac{1}{4},\ p_2 = \frac{1}{2},\ C = 16$

- current solution: $R_1 = 0,\ R_2 = 3$
- cost: $E = 16$

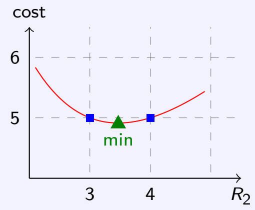
Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**
$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}} (E \ln \frac{1}{p_i}) \right\rfloor$
if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$
$E \leftarrow 3 + 2$

Expected cost of the last round in function of R_2



Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 0$, $R_2 = 3$
- cost: $E = 5$

Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

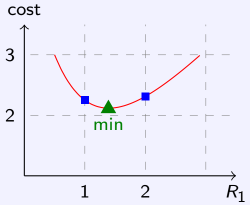
$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**

$R_1 \leftarrow \lfloor 1.39 \rfloor$

if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$

$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of all rounds in function of R_1



Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 1$, $R_2 = 3$
- cost: $E = 5$

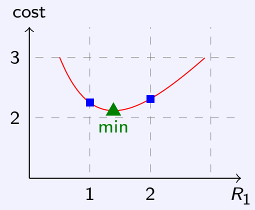
Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**
$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}} \left(E \ln \frac{1}{p_i} \right) \right\rfloor$
if $1.25 > 1.31$ **then** $R_i \leftarrow R_i + 1$
$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of all rounds in function of R_1



Instance: $p_1 = \frac{1}{4},\ p_2 = \frac{1}{2},\ C = 16$

- current solution: $R_1 = 1,\ R_2 = 3$
- cost: $E = 5$

Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

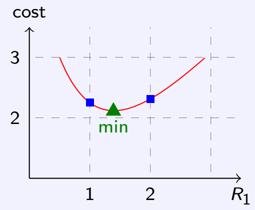
$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**

$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}}\left(E \ln \frac{1}{p_i}\right) \right\rfloor$

if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$

$E \leftarrow 1 + 1.25$

Expected cost of all rounds in function of R_1



Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 1$, $R_2 = 3$
- cost: $E = 2.25$

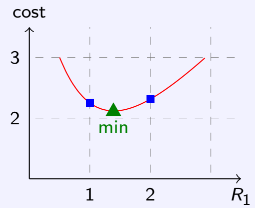
Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds
for i from k to 1 **do**
$R_i \leftarrow \left\lfloor \ln_{\frac{1}{p_i}} \left(E \ln \frac{1}{p_i}\right) \right\rfloor$
if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$
$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of all rounds in function of R_1



Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 1$, $R_2 = 3$
- cost: $E = 2.25$

Optimal strategy for integral allocation

Principle: - compute backwards the noninteger optimal solution
- round to the best integer

Algorithm BACKTRACK

$E \leftarrow C$ ▷ E: expected cost of last rounds

for ...

if $p_i^{R_i} E > 1 + p_i^{R_i+1} E$ **then** $R_i \leftarrow R_i + 1$

$E \leftarrow R_i + p_i^{R_i} E$

Expected cost of all rounds in function of R_1

Simple and linear-time algorithm $\Longrightarrow$ optimal integral solution

Instance: $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, $C = 16$

- current solution: $R_1 = 1$, $R_2 = 3$
- cost: $E = 2.25$

Outline

First case: constant probabilities and unbounded rounds

This section: allow only integral requests

Remark on optimal strategies

- Never send 2 requests in a round
- Determined by $k^* = \#(\text{rounds where you send 1 request})$

Lemma (Optimal Strategy)

- *if* $C < \frac{1}{1-p} : k^* = 0$, *review yourself*
- *if* $C > \frac{1}{1-p} : k^* = \infty$, *keep requesting*
- *if* $C = \frac{1}{1-p} : k^* \in \mathbb{N} \cup \{\infty\}$, *whatever*

Bounded rounds, constant probabilities

Framework

- Self-reviewing cost: C
- Budget of R requests
- Spread them among k rounds, with rejection prob. p

General algorithm

1. $R^* \leftarrow \#requests$ sent by the solution of BACKTRACK
2. if $R^* \leq R$: problem solved
3. else: find an optimal strategy which sends exactly R requests

Main idea: the way requests are spread does not depend on C

$$E(R_1, R_2, \ldots, R_k) = R_1 + p^{R_1} R_2 + p^{R_1+R_2} R_3 + \cdots + p^R C$$

$\Longrightarrow$ find another C for which one optimal strategy sends R requests

Bounded rounds, constant probabilities

Framework

- Self-reviewing cost: C
- Budget of R requests
- Spread them among k rounds, with rejection prob. p

General algorithm

1. $R^* \leftarrow \#requests$ sent by the solution of BACKTRACK
2. if $R^* \leq R$: problem solved
3. else: **find an optimal strategy which sends exactly R requests**

Main idea: the way requests are spread does not depend on C

$$E(R_1, R_2, \ldots, R_k) = R_1 + p^{R_1} R_2 + p^{R_1+R_2} R_3 + \cdots + p^R C$$

$\Longrightarrow$ find another C for which one optimal strategy sends R requests

Supporting lemmas

Lemma

If $R^* > R$, there exists an optimal strategy with R requests.

Lemma

When C increases, each R_i spans every integer in ascending order.

Remarks

- Continuous model: each R_i increases continuously with C
- Result: no rounding breaks this property

Lemma

There exists a value C_R for which an optimal strategy sends R requests.

Supporting lemmas

Lemma

If $R^* > R$, there exists an optimal strategy with R requests.

Lemma

When C increases, each R_i spans every integer in ascending order.

Remarks

- Continuous model: each R_i increases continuously with C
- Result: no rounding breaks this property

Lemma

There exists a value C_R for which an optimal strategy sends R requests.

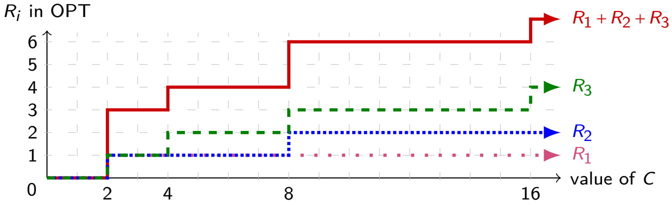
Some intuition

Recall on the algorithm

- If one C_R known $\Longrightarrow$ can compute OPT for C with R requests
- Difficulty: C_R can be unique so *hard* to find

Example: three request rounds, $p = 1/2$

- $C = 8$
  - optimal strategies: $R_1 = 1$, $R_2 \in \{1,2\}$, $R_3 \in \{2,3\}$
  - use 4, 5 or 6 requests
- No other value of C induces 5 requests:



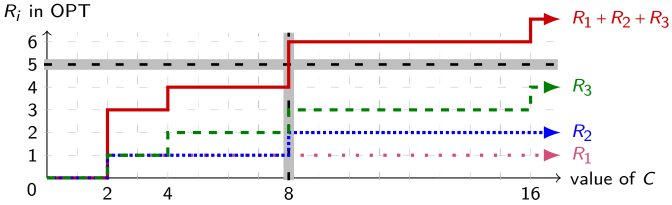
Some intuition

Recall on the algorithm

- If one C_R known $\Longrightarrow$ can compute OPT for C with R requests
- Difficulty: C_R can be unique so *hard* to find

Example: three request rounds, $p = 1/2$

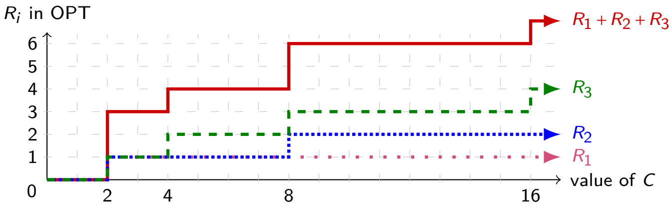
- $C = 8$
  - optimal strategies: $R_1 = 1$, $R_2 \in \{1,2\}$, $R_3 \in \{2,3\}$
  - use 4, 5 or 6 requests
- No other value of C induces 5 requests:



Computing a self-reviewing cost C_R from C

Algorithm ComputeCr

```
R ← #requests sent by Backtrack for a self-reviewing cost of C
[L, U] ← [1/(1−p), C]          ▷ we maintain C_R ∈ [L, U] via binary search
Reduce [L, U] until every round satisfies R_i(U) ∈ {R_i(L), R_i(L) + 1}
while ∑_i R_i(L) < R do
    Find analytically how much to increase L to send 1 more request
```



Outline

Lost regularity

Considered instance: 2 request rounds, $p_1 = 1/4$, $p_2 = 1/2$

Optimal distribution depends on C

- 2 requests
  - $C = 16$: $(R_1, R_2) = (2, 0)$
  - $C = \ \ 8$: $(R_1, R_2) = (1, 1)$

R_1 can decrease while budget increases

- $C = 16$
  - 2 requests: $(R_1, R_2) = (2, 0)$
  - 3 requests: $(R_1, R_2) = (1, 2)$

Results

A single paper

- Greedy algorithm: $(k+1)$-approximation
- Idea: assume you know E_{OPT}
  - at round i, send $R_i(E_{\text{OPT}}) = \left\lfloor E_{\text{OPT}} \middle/ \prod_{j<i} p_j^{R_j} \right\rfloor$ until budget reached
- "Guess" E_{OPT} by binary search
  - reduce an interval $[L, U]$ ensuring:

$$E(\{R_i(L)\}) \geq (k+1)L \qquad \& \qquad E(\{R_i(U)\}) \leq (k+1)U$$

  - FUN fact: E_{OPT} might be outside the final interval

N papers sharing a common budget

- Solved by dynamic programming in $O(NkR^2)$

Results

A single paper

- Greedy algorithm: $(k+1)$-approximation
- Idea: assume you know E_{OPT}
  - at round i, send $R_i(E_{\text{OPT}}) = \left\lfloor E_{\text{OPT}} \Big/ \prod_{j<i} p_j^{R_j} \right\rfloor$ until budget reached
- "Guess" E_{OPT} by binary search
  - reduce an interval $[L, U]$ ensuring:

$$E(\{R_i(L)\}) \geq (k+1)L \qquad \& \qquad E(\{R_i(U)\}) \leq (k+1)U$$

  - FUN fact: E_{OPT} might be outside the final interval

N papers sharing a common budget

- Solved by dynamic programming in $O(NkR^2)$

Results

A single paper

- Greedy algorithm: $(k+1)$-approximation
- Idea: assume you know E_{OPT}
  - at round i, send $R_i(E_{\mathrm{OPT}}) = \left\lfloor E_{\mathrm{OPT}} \Big/ \prod_{j<i} p_j^{R_j} \right\rfloor$ until budget reached
- "Guess" E_{OPT} by binary search
  - reduce an interval $[L, U]$ ensuring:

$$E(\{R_i(L)\}) \geq (k+1)L \qquad \& \qquad E(\{R_i(U)\}) \leq (k+1)U$$

  - FUN fact: E_{OPT} might be outside the final interval

N papers sharing a common budget

- Solved by dynamic programming in $O(NkR^2)$

Outline

Unknown but constant rejection probability

Unbounded rounds

- Send 1 request during C rounds
- 2-approximation ($\approx$ ski rental problem)

Two rounds

- send $\left\lceil \sqrt{\frac{C}{\ln C}} \right\rceil$ requests
- $4\left(\sqrt{\frac{C}{\ln C}}+2\right)$-approximation

Bounded rounds

- send $\left\lceil C^{\frac{i}{k+1}} \right\rceil$ requests at round i
- $(k+1)(C^{\frac{1}{k+1}}+1)$-approximation

Outline

1. Optimal strategy for unbounded requests
2. Optimal strategy for bounded requests
3. Unknown rejection probability
4. Conclusion

How to preserve your health and relationship if you have:

Many many friendly colleagues *[unbounded requests]*

- Splittable ones *[non-integral requests]*: exact formula
- Regular ones: optimal rounding algorithm

Limited colleagues *[bounded requests]*

- Insensible to deadlines *[constant rejection prob.]*: optimal algorithm
- Busy colleagues: $\approx k$-approx or expensive dynamic programming

'Unpredictable but stable' colleagues *[unknown prob.]*

- $\approx kC^{1/k}$-approximation

Moral

Build a proper set of colleagues and study them closely

How to preserve your health and relationship if you have:

Many many friendly colleagues *[unbounded requests]*

- Splittable ones *[non-integral requests]*: exact formula
- Regular ones: optimal rounding algorithm

Limited colleagues *[bounded requests]*

- Insensible to deadlines *[constant rejection prob.]*: optimal algorithm
- Busy colleagues: $\approx k$-approx or expensive dynamic programming

'Unpredictable but stable' colleagues *[unknown prob.]*

- $\approx kC^{1/k}$-approximation

Moral

Build a proper set of colleagues and study them closely

How to preserve your health and relationship if you have:

Many many friendly colleagues *[unbounded requests]*

- Splittable ones *[non-integral requests]*: exact formula
- Regular ones: optimal rounding algorithm

Limited colleagues *[bounded requests]*

- Insensible to deadlines *[constant rejection prob.]*: optimal algorithm
- Busy colleagues: $\approx k$-approx or expensive dynamic programming

'Unpredictable but stable' colleagues *[unknown prob.]*

- $\approx kC^{1/k}$-approximation

Moral

Build a proper set of colleagues and study them closely

How to preserve your health and relationship if you have:

Many many friendly colleagues *[unbounded requests]*

- Splittable ones *[non-integral requests]*: exact formula
- Regular ones: optimal rounding algorithm

Limited colleagues *[bounded requests]*

- Insensible to deadlines *[constant rejection prob.]*: optimal algorithm
- Busy colleagues: $\approx k$-approx or expensive dynamic programming

'Unpredictable but stable' colleagues *[unknown prob.]*

- $\approx kC^{1/k}$-approximation

Moral

Build a proper set of colleagues and study them closely