Stochastic k-TSP

Viswanath Nagarajan (U Michigan)

Joint work: Alina Ene (BU), Rishi Saket (IBM)

k-TSP

- Metric (V, d) with root r and target k

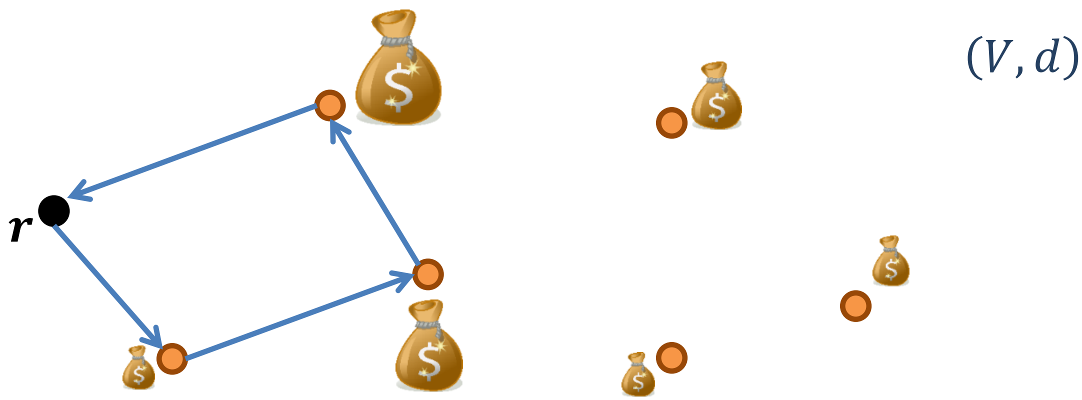
  2-approximation [Garg '05]

- Quota-TSP: vertices have non-uniform rewards

  5-approximation [Ausiello Leonardi Spaccamela '00]

- Orienteering: max-reward s.t. bound on tour length

  (2+ϵ)-approximation [Chekuri Korula Pal '12]



Stochastic Setting

- In practice data is often uncertain

  Many approaches: stochastic, robust, online models

- We consider a stochastic setting with random rewards

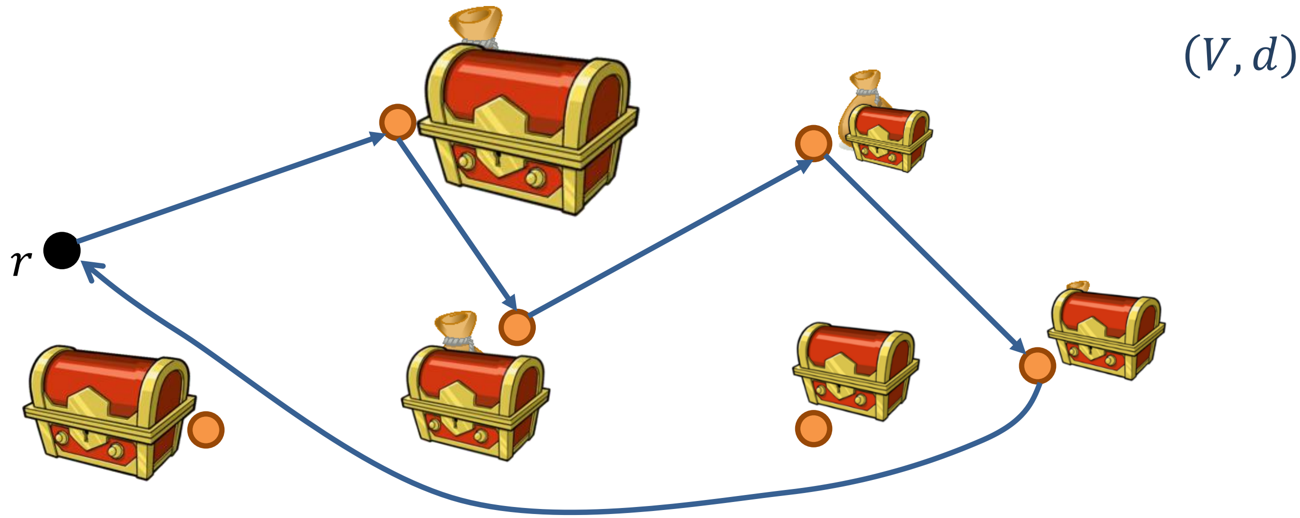
Possible outcomes :

- Techniques from deterministic case already suffice

  OR

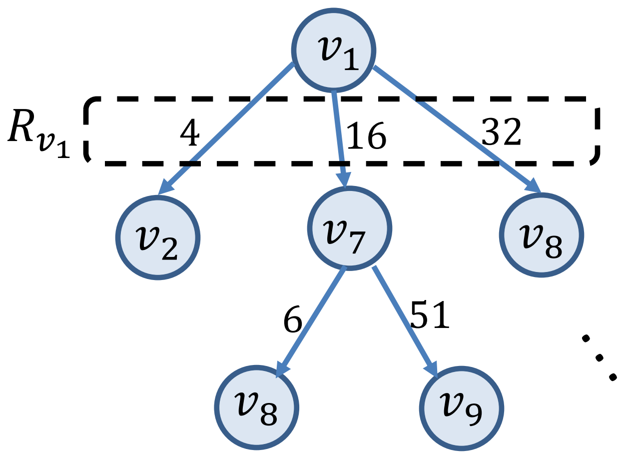
- Need new techniques to handle the stochastic case

Problem Definition

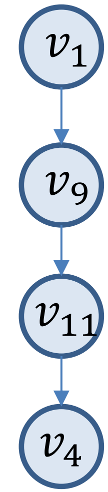


- Metric (V,d) with root r and target k
- Independent random variables $R_v \in \{0,1,\ldots k\}$ for rewards
- Instantiation R_v only known when v is visited
- Minimize expected length to achieve total reward $\geq$ k

Representing Solutions



Adaptive policy



Non- adaptive policy

Solution policy: adaptive vs non-adaptive

- adaptive: next step depends on observed rewards.
- non-adaptive: does not depend on observed rewards.

Adaptivity Gap: worst case gap between these policies.

Our Results

- $O(\log k)$-approximate adaptive algorithm
- $O(\log^2 k)$-approximate non-adaptive algorithm

  Also bounds adaptivity gap
- Adaptivity gap at least $e \approx 2.71$

  Even with single random reward and star metric
- Extension to submodular rewards (larger poly-log approximation)

  Uses submod-max adaptivity gap [Gupta N. Singla '17]

Talk Outline

- Related work
- Adaptive algorithm
- Non-adaptive algorithm
- Extension to submodular rewards

Related Work: Maximization

- Stochastic knapsack [Dean Goemans Vondrak '04]...

  Ad Gap $\leq$ 4

  adaptive 2-approx. [Bhalgat Goel Khanna '11]
- Stochastic matching [Chen Immorlica Karlin Mahdian Rudra '09]...

  Ad Gap $\leq$ 3.23 [Baveja Chavan Nikiforov Srinivasan Xu '18]

  adaptive 2-approx. for unweighted [Adamcyzk '10]
- Stochastic orienteering [Gupta Krishnaswamy N. Ravi '12] [Bansal N. '14]

  $\Omega(\text{loglog } B)^{1/2} \leq$ Ad Gap $\leq$ O(loglog B)
- Stochastic submodular-max [Gupta N. Singla '16 '17]

  Ad Gap $\leq$ 3

Related Work: Minimization

- Stochastic knapsack-cover [Deshpande Hellerstein Kletenik '14]

  Adaptive 2-approx.

- Stochastic covering IPs [Goemans Vondrak '06]

  d $\leq$ Ad. Gap $\leq$ d^2

- Stochastic submodular-cover [Im N. Zwaan '12]

  Adaptive (log $1/\epsilon$)-approx.

  Correlated setting [Navidi Kambadur N. '17]

  We use similar analysis here

Adaptive Algorithm

Initial Approach

Use orienteering in an iterative fashion

Assume an exact orienteering algorithm

<u>Algorithm for Deterministic k-TSP</u>

For i=0,1,2... solve Orienteering with length 2^i

until total reward $\geq$ k

O(1) approx.

<u>Attempt for Stochastic k-TSP</u>

For i=0,1,2... solve Orienteering with

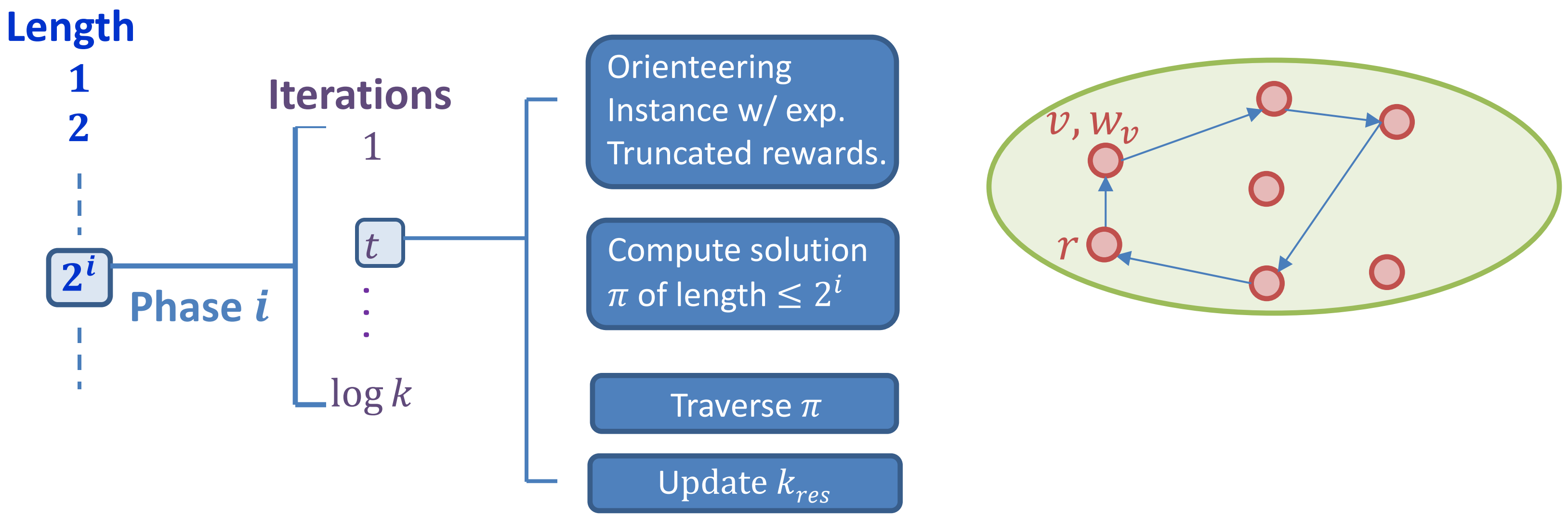
Length bound 2^i

Expected *truncated* rewards $w_v = E[\min(R_v , k_{res})]$

until total instantiated reward $\geq$ k

Does not work!

Algorithm



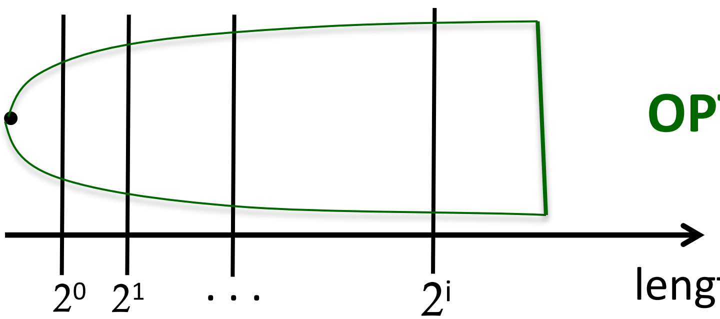
For each 2^i length solve $\log k$ iterations of Orienteering.

- Also allows using $O(1)$-approx. for Orienteering.

Thm: $O(\log k)$ approx. for stochastic k-TSP.

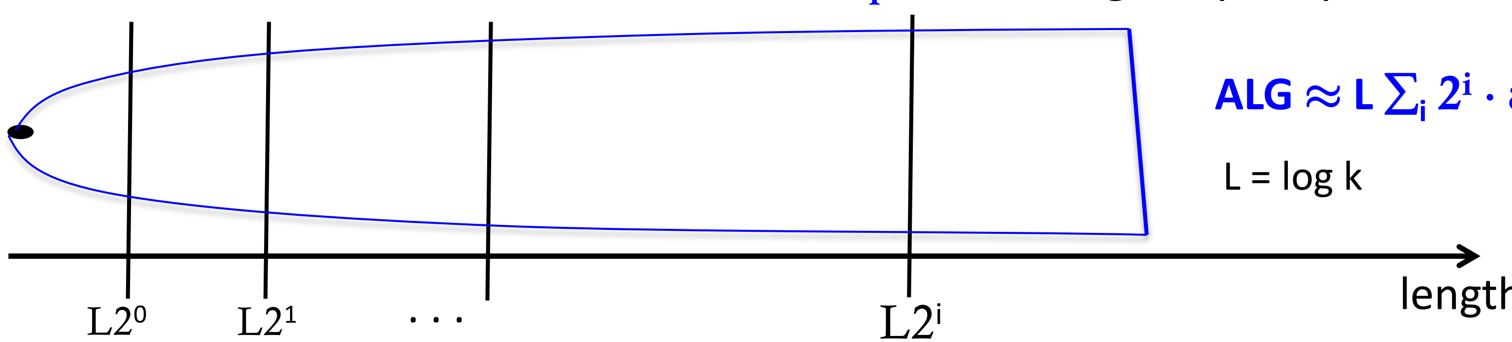
Analysis Outline

u_i = Pr[OPT costs > 2^i]



$\text{OPT} \approx \sum_i 2^i \cdot u_i$

a_i = Pr[ALG goes past phase i]



$\text{ALG} \approx L \sum_i 2^i \cdot a_i$

L = log k

Relate **a** to **u**

$a_i \leq 0.25 \cdot a_{i-1} + u_i$

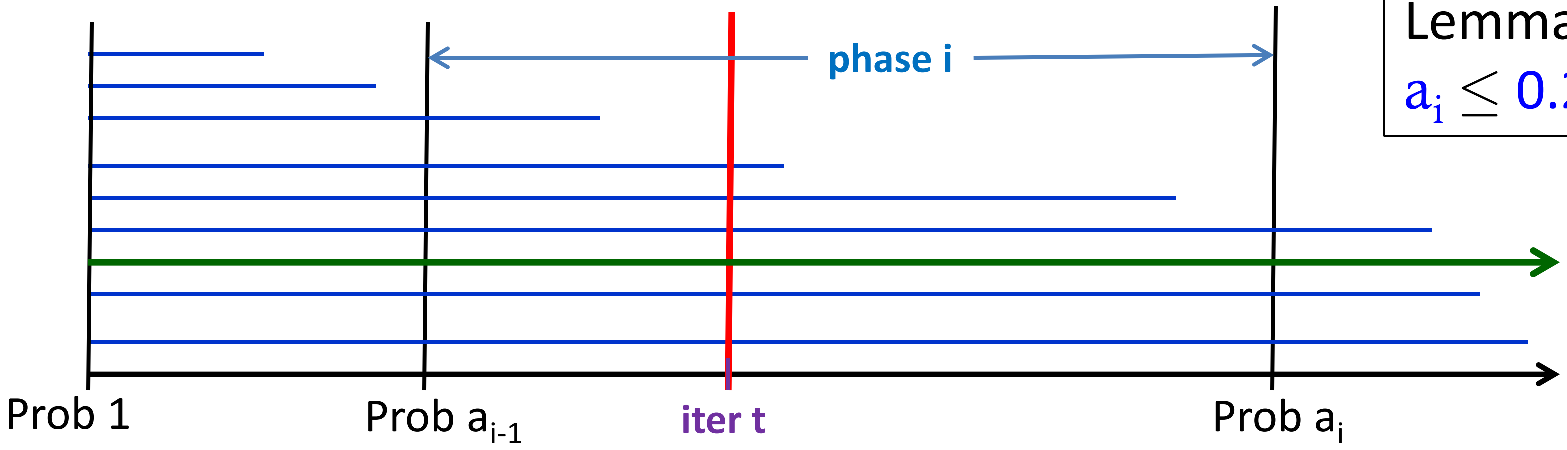
$\Downarrow$

$\text{ALG} \leq O(L) \cdot \text{OPT}$

Similar idea in min-latency TSP [Chaudhuri Godfrey Rao Talwar '03]

Also used in stochastic submod-cover [Im N. Zwaan '12]

Analysis (phase i)

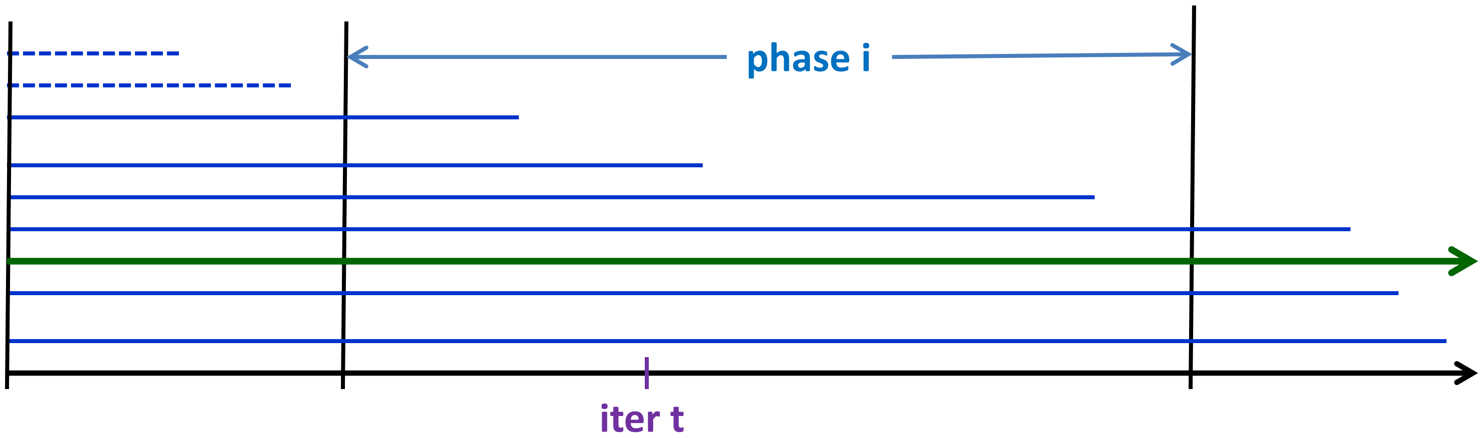


Lemma:
$a_i \leq 0.25 \cdot a_{i-1} + u_i$

- s = state of algorithm (outcomes of some rewards)
- H(t,i) = states at iteration t of phase i
- $\text{Gain}(s) = \dfrac{E[\min\{\Delta \text{ Reward}, k_{res}\}]}{k_{res}}$
- $G(t,i) = E_{s \leftarrow H(t,i)}[\text{Gain}(s)]$ and $G(i) = \sum_t G(t,i)$

1) Upper bound $G(i) \leq (\ln k)\, a_{i-1}$
2) Lower bound $G(i) \geq \Omega(\ln k) \cdot (a_i - u_i)$

Analysis (Upper Bound)

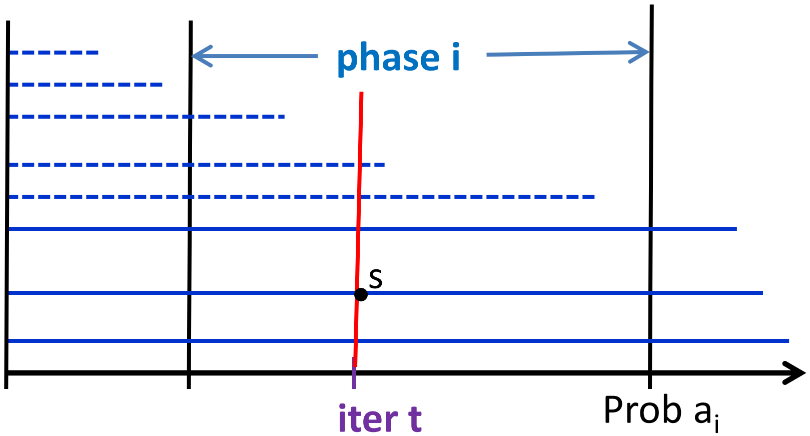


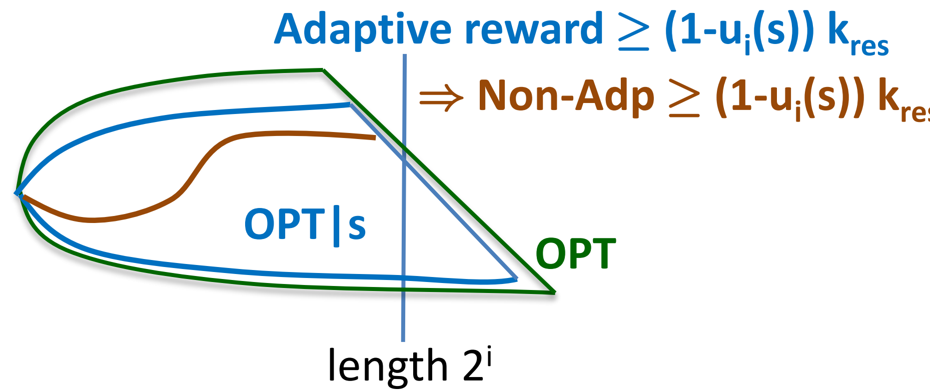
- Fix a decision path in ALG
- Contribution to G(i) = $\sum_t G(t,i) = \sum_t \frac{\Delta \text{Reward}_t}{k_{res}}$

$$\leq \frac{1}{k} + \frac{1}{k-1} + \ldots + \frac{1}{1} \leq \ln k$$

$\Rightarrow$ **G(i) $\leq$ (ln k) a_{i-1}**

Analysis (Lower Bound)





- Fix iteration t in phase i and state s
- Bound Gain(s) using orienteering instance J(s) — length bnd 2^i; reward $E[\min(R_v, k_{res})]$

A. optimum(J(s)) $\geq (1-u_i(s)) \cdot k_{res}$ where $u_i(s) = \Pr[\text{OPT} > 2^i \mid s]$

B. Gain(s) $\geq (1-1/e) \cdot \alpha \cdot \frac{\text{OPT(J(s))}}{k_{res}}$

α-approx. orienteering

$G(t,i) \geq (1-1/e)\cdot\alpha \cdot (1-u_i - (1-a_i)) = \Omega(1)\cdot(a_i - u_i)$

$\Rightarrow$ **$G(i) \geq \Omega(\log k) \cdot (a_i - u_i)$**

Non-Adaptive Algorithm

Adaptive To Non-Adaptive

Simulate the adaptive algorithm

- Possible orienteering instances in phase i iteration t

  Same bound 2^i on length

  Different truncation levels k_{res} (for det. rewards)
- Bucket the truncation into (log k) levels
- Run (log k) many orienteering instances at each (i,t)

Thm: $O(\log^2 k)$ approx. for non-adaptive stochastic k-TSP.

- Also upper bounds adaptivity gap

Don't know better result even w.r.t. non-adaptive OPT

Adaptivity Gap Lower Bound

Online bidding: given n, find *random* sequence $B=(b_1, b_2 \cdots)$ of [n]

Sequence S, target T costs **C(S,T)** = sum of bids in S until some bid $\geq$ T

$$\min_{B:\text{ dist. on seq}} \max_{T \in [n]} \frac{E_{S \leftarrow B}[C(S,T)]}{T} \geq e$$

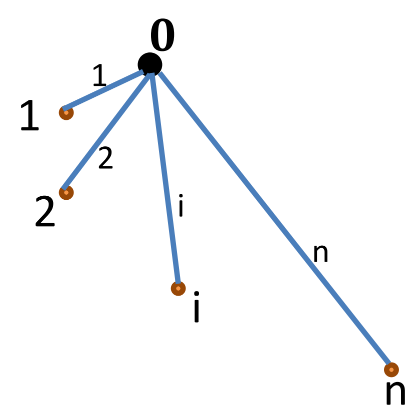
[Chrobak Kenyon Noga Young '08]

Stochastic k-TSP

- Target $k = 2^{n+1}$, rewards $R_i = 2^i$ for nodes $i \in [n]$
- Single random reward $R_0 = (k-2^i)$ w.p. p_i for $i \in [n]$

Choose prob p_i to maximize adaptivity gap

$$= \max_{p = \text{ dist. on } [n]} \min_{S:\text{ seq.}} \frac{E_{T \leftarrow p}[C(S,T)]}{E_{T \leftarrow p}[T]}$$



Submodular Reward Function

- Metric (V,d) with root r and target k
- Reward function $f : 2^V \rightarrow R_+$, monotone submodular
- Min length to collect reward at least k

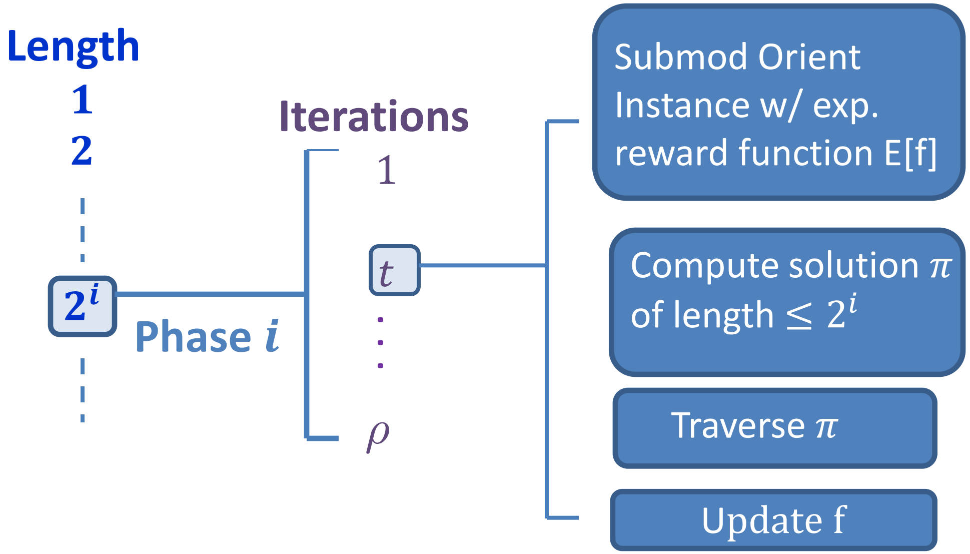
$O(\log^{3+\delta} n)$ approx. [Calinescu, Zelikovsky '05]

$\Omega(\log^{2-\delta} n)$ hard-to-approx. [Halperin Krauthgamer '03]

Stochastic setting: each vertex active w.p. p_i and minimize expected length so that f(active) $\geq$ k

- Generalizes stochastic k-TSP for Bernoulli random vars

Algorithm for Submodular Case



Expected function
$Ef(S) = E_{A \leftarrow \mathbf{p}}[f(S \cap A)]$

$\rho = \rho_{\text{orient}} \cdot \log 1/\epsilon$ iterations instead of log k

Theorem: Adaptive $O(\log^{2+\delta} n \cdot \log 1/\epsilon)$ approximation.

- Uses $\rho_{\text{orient}} = O(\log^{2+\delta} n)$ [Calinescu, Zelikovsky '05]
- Submodular-max adaptivity gap ≤ 3 [Gupta N. Singla '17]

Open Questions

- O(1)-approximation for stochastic k-TSP?

  For either adaptive or non-adaptive

- Adaptivity gap?

  Interesting even for covering knapsack (k-TSP on star metric)

  There is adaptive 2-approx. [Deshpande Hellerstein Kletenik ‘14]

  Can get non-adaptive O(1)-approx. via different approach

  Max-knapsack well understood [Dean Goemans Vondrak ‘04]...

- Other stochastic minimization problems?

Thank You!