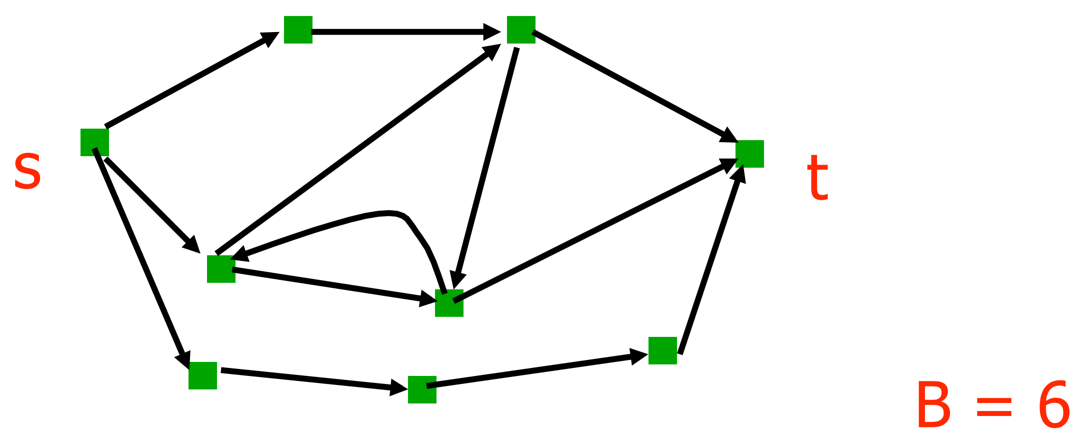
Orienteering and related problems: mini-survey and open problems

Chandra Chekuri
University of Illinois (UIUC)

Orienteering

Input: Graph (undir or dir) G, nodes s, t and budget B

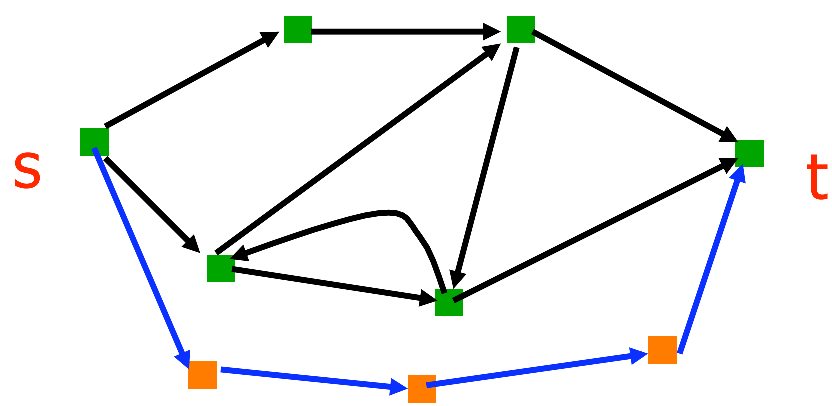
Goal: find $s \rightarrow t$ walk/path P of length $\leq$ B that maximizes number of nodes in P



Orienteering

Input: Graph (undir or dir) G, nodes s, t and budget B

Goal: find $s \rightarrow t$ walk/path P of length $\leq B$ that maximizes number of nodes in P

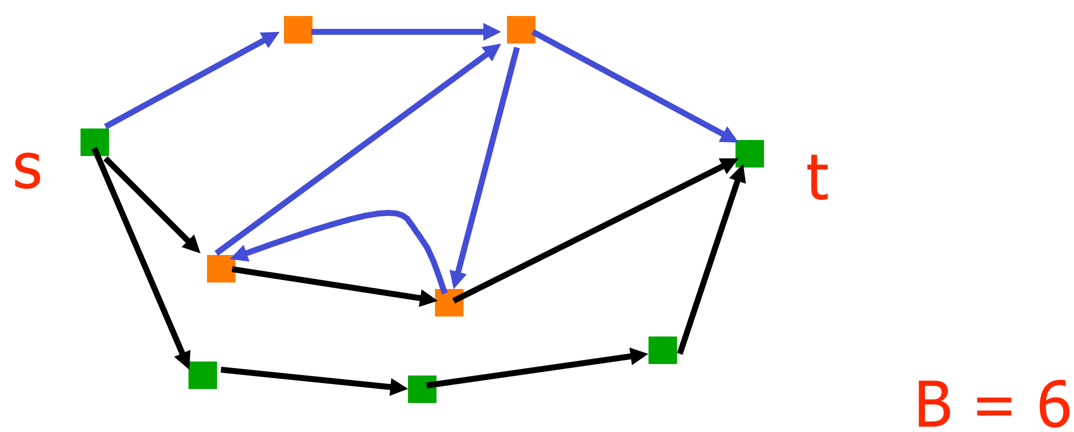


B = 6

Orienteering

Input: Graph (undir or dir) G, nodes s, t and budget B

Goal: find s $\rightarrow$ t walk/path P of length $\leq$ B that maximizes number of nodes in P



Orienteering: known results

Undirected graphs

- Approx. algorithms
  - (2+ε) for points in R^2 [Arkin-Mitchell-Narasimhan'98]
  - 4 [Blum-Chawla-Karger-Lane-Meyerson-Minkoff'03]
  - 3 [Bansal-Blum-Chawla-Meyerson'04]
  - (1+ε) for points in R^d, d fixed [Chen-HarPeled'05]
  - (2+ε) [C-Korula-Pal'08]
- Hardness:
  - APX-hard [BCKLMM'03]

Orienteering: known results

Undirected graphs

- Approx. algorithms
  - (2+ε) for points in R^2 [Arkin-Mitchell-Narasimhan'98]
  - 4 [Blum-Chawla-Karger-Lane-Meyerson-Minkoff'03]
  - 3 [Bansal-Blum-Chawla-Meyerson'04]
  - (1+ε) for points in R^d, d fixed [Chen-HarPeled'05]
  - (2+ε) [C-Korula-Pal'08]
- Hardness:
  - APX-hard [BCKLMM'03]

Directed Graphs

- Approx. algorithms
  - O(log n) in *quasi-poly* time [C-Pal'05]
  - $O(\log^2 n)$ [C-Korula-Pal'08] [Nagarajan-Ravi'07]
- Hardness: APX-hard

Orienteering: known results

Undirected graphs

- Approx. algorithms
  - (2+ε) for points in R^2 [Arkin-Mitchell-Narasimhan'98]
  - 4 [Blum-Chawla-Karger-Lane-Meyerson-Minkoff'03]
  - 3 [Bansal-Blum-Chawla-Meyerson'04]
  - (1+ε) for points in R^d, d fixed [Chen-HarPeled'05]
  - (2+ε) [C-Korula-Pal'08]
- Hardness:
  - APX-hard [BCKLMM'03]

Directed Graphs

- Approx. algorithms
  - O(log n) in *quasi-poly* time [C-Pal'05]
  - $O(\log^2 n)$ [C-Korula-Pal'08] [Nagarajan-Ravi'07]
- Hardness: APX-hard

Close gap for directed graphs

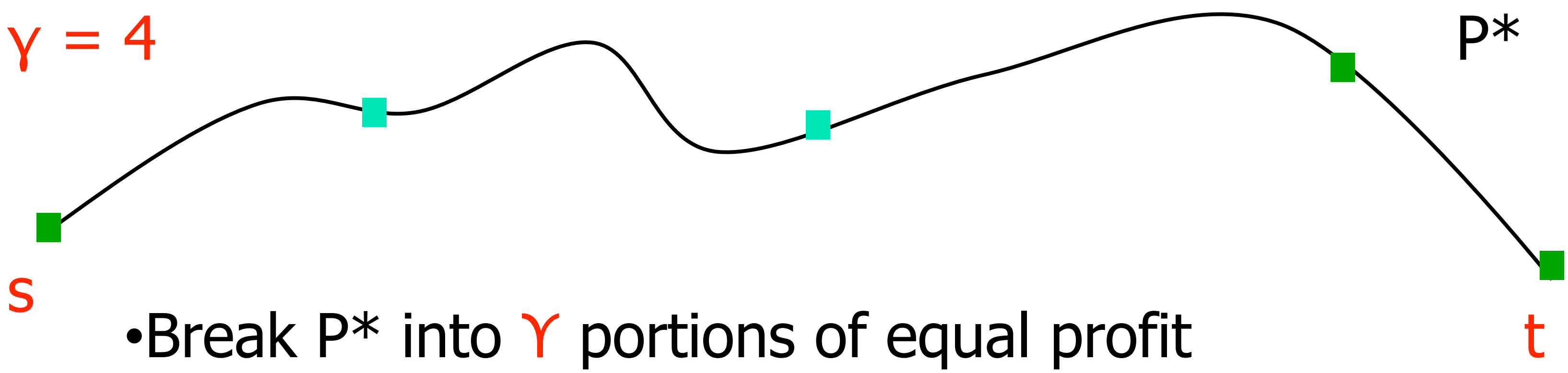
Orienteering: Key Idea [BCKLMM]

- Reduce to k-Stroll problem via the intermediate problem called *min-excess* problem
- The k-Stroll problem
  - Input: Graph G, nodes s, t and integer k
  - Goal: Find min-cost s-t walk/path that visits k nodes
- Min-excess problem
  - Input: Graph G, nodes s, t and integer k
  - Goal: Find s-t walk/path P that visits k nodes and minimizes *excess of P* = len(P) – dist(s,t)

Orienteering via Min-Excess

[BCKLMM'03, BBCM'04]

Theorem: γ approx for Min-Excess implies
ceiling(γ) approx for Orienteering

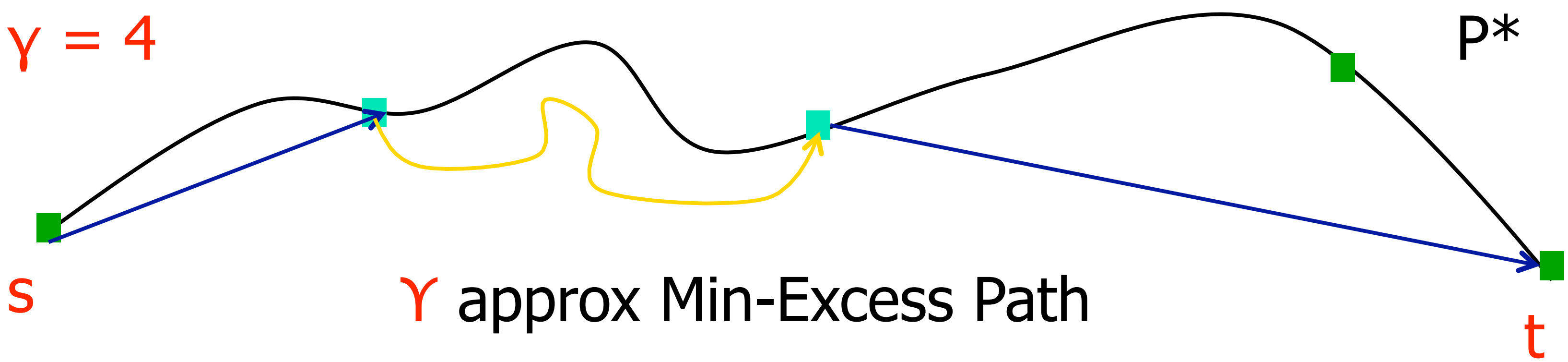


- Break P* into Υ portions of equal profit
- One of the portions has ≤ 1/Υ excess(P*)

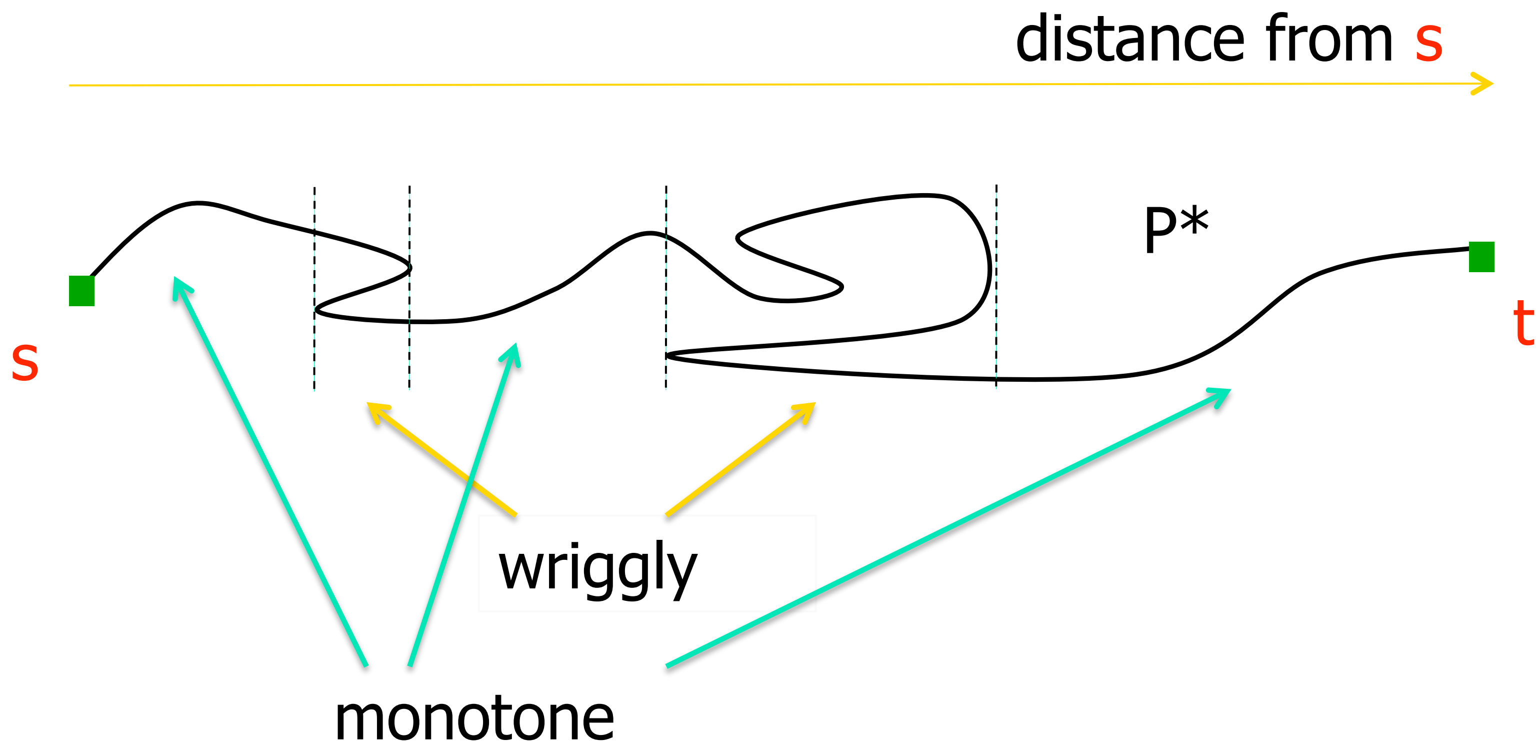
Orienteering via Min-Excess

[BCKLMM'03, BBCM'04]

Theorem: γ approx for Min-Excess implies
ceiling(γ) approx for Orienteering

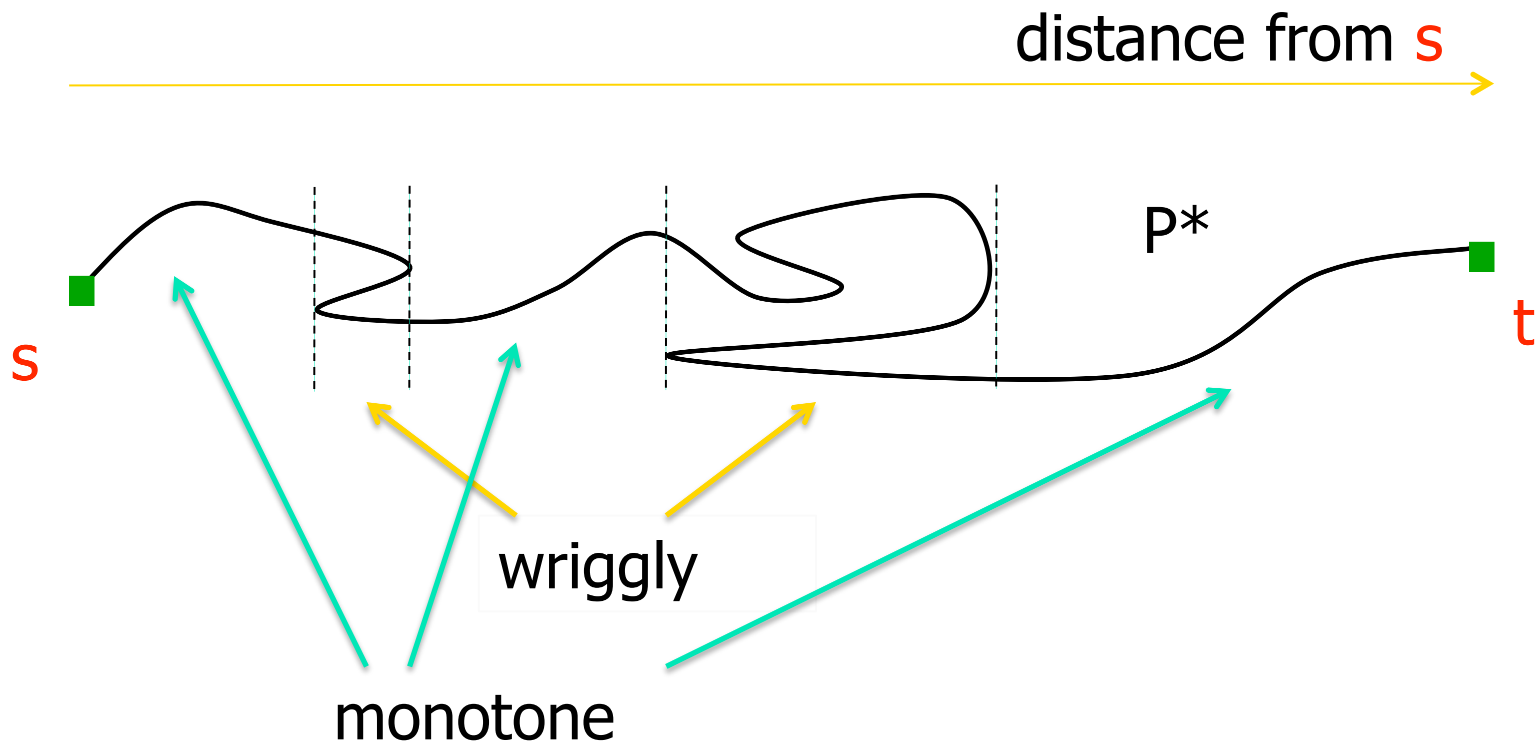


Min-Excess via (approx) k-Stroll



wriggly portions have large excess: use k-stroll approx
monotone portions: use exact algorithm
stitch via dynamic programming

Min-Excess via (approx) k-Stroll



[BCKLMM'03]

Theorem: β approx for k-Stroll implies O(β) for min-excess

k-Stroll and Orienteering

[BCKLMM'03]

Theorem: α approx for k-Stroll implies $O(\alpha)$ approx for Orienteering

Algorithms for k-Stroll

- Undir graphs: (2+ϵ) [Chaudhuri-Godfrey-Rao-Talwar'03]
- Directed graphs: ??

Is there a non-trivial approx. for dir k-Stroll?
Is the problem very hard?

Algorithms for k-Stroll in dir graphs

- k=n is asymmetric TSP Path problem (ATSPP)
  - $O(\sqrt{n})$ approx [Lam-Newman'05]
  - $O(\log n)$ approx [C-Pal'06]
- Bicriteria (α, β) approx: output path with k/α vertices and cost β OPT
  - $(O(\log^2 k), O(1))$ approx [C-Korula-Pal'08] [Nagarajan-Ravi'07] *(different approaches)*
  - Bi-criteria approx sufficient for Orienteering

Improve k-Stroll bi-criteria approx

Orienteering with Time-Windows

Orienteering-TW

- Each node v has a time window [R(v), D(v)]
- v counted only if it is visited in its window

Deadline-TSP: R(v) = 0 for all v

Goal: Find s-t walk to max # of nodes visited

Orienteering with Time-Windows

[Bansal-Blum-Chawla-Meyerson'04]

α approx for Orienteering implies

- $O(\alpha \log n)$ approx for Deadline-TSP
- $O(\alpha \log^2 n)$ approx for Orienteering-TW

$\alpha = O(1)$ for undir and $\alpha = O(\log^2 n)$ in dir graphs

Orienteering with Time-Windows

Conjecture: there is an $O(\log n)$ approx for Orient-TW in undirected graphs

Is the problem $\omega(1)$-factor hard in directed graphs?

Evidence for conjecture:

- $O(\log n)$ approx in quasi-poly time even in directed graphs. [C-Pal'05]
- $O(\alpha \log L_{max})$ approx [C-Korula'07] where L_{max} is max window length assuming integer data

Orienteering with Time-Windows

[C-Korula'07]

Two simple algorithms:

- $O(\alpha \log L_{max})$ approx assume integer data and is L_{max} is max window length
- $O(\alpha \max(\log n, \log (L_{max}/L_{min})))$

Difficult case: L_{max}/L_{min} is super-poly in n

Orienteering with Time-Windows

[C-Korula'07]

Idea for $O(\log L_{max})$ approx

Lemma: Let [a,b] be an interval with a, b integer and $m = b-a$. Then [a,b] can be *partitioned* into at most $2 \log m$ *disjoint* sub-intervals such that

- length of each sub-interval is a power of 2
- sub-interval of length 2^i starts at multiple of 2^i
- at most 2 intervals of each length

Proof of Lemma

- [a, b] interval with a and b integers
- If a, b are *even* integers, recurse on [a/2, b/2] and multiply each interval by 2
- If a, b are *odd,* recurse on [a+1, b-1] and add [a, a+1] and [b-1, b]
- If a is *odd* and b is even, recurse on [a+1, b] and add [a, a+1]
- If a is *even* and b is *odd*, recurse on [a,b-1] and add [b-1, b]

Orienteering with Time-Windows

- Apply lemma to each [R(v), D(v)]
- Consider all sub-intervals of length 2^i.
- These intervals start at a multiple of 2^i hence they are *either disjoint or completely overlap*
- Can use Orienteering in each interval and stitch across disjoint intervals using dynamic prog.

- At most $\log L_{max}$ classes and one of them has $OPT/2\log L_{max}$ profit

Fixed-parameter Tractability

Observation: There is an $O(4^k \text{ poly}(n))$ time algorithm that gives optimum profit if there is a solution that visits at most k nodes.

Follows from “color-coding” scheme of [Alon-Yuster-Zwick]

A more complex path problem

SOP-TW

- $f: 2^V \rightarrow \mathcal{R}^+$ a monotone submodular set function on the nodes V
- Each node v has a *time window* [R(v), D(v)] .

Goal: find path P s.t nodes in P are visited in time windows and f(P) is maximized

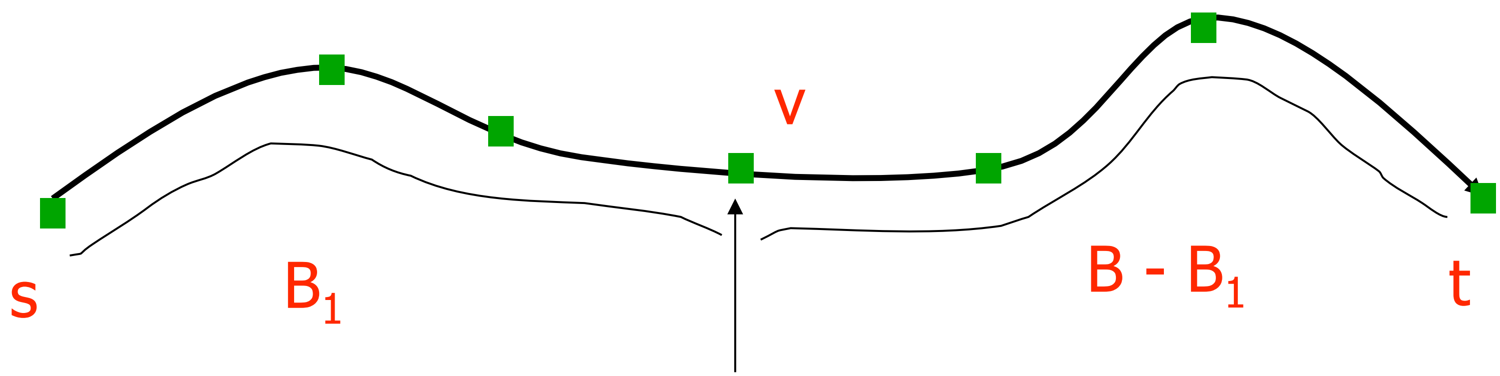
Algorithm for SOP-TW

[C-Pal'05]

Theorem: There is a quasi-poly time O(log n) approx. for SOP-TW

Recursive Greedy Alg: idea

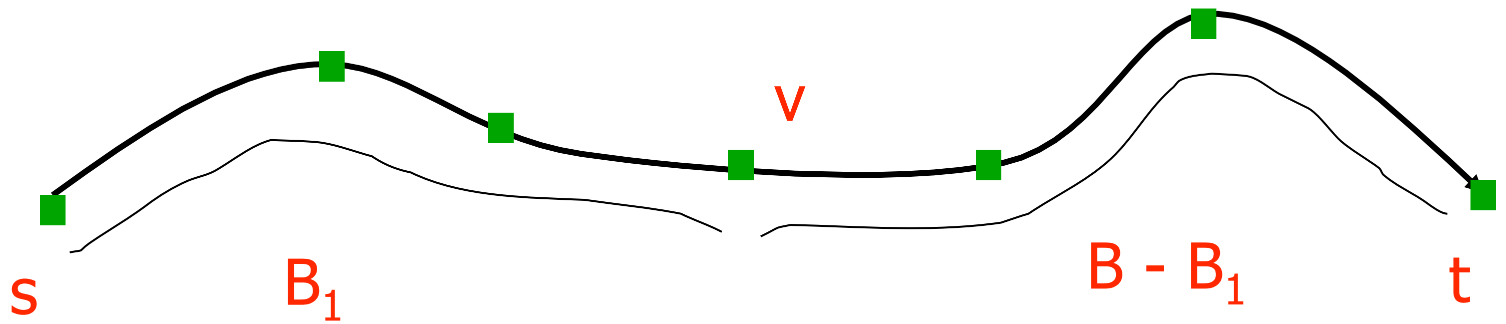
Unknown optimum path P^*



middle node v

time to reach v = B_1

Recursive Greedy Algorithm



Savitch's algo for optimization ?

RG(f, s, t, B, i)

1. "Guess" v and $B_1 \in [R(v), D(v)]$
2. P_1 = RG(f, s, v, B_1, i-1)
3. P_2 = RG(f_{P1}, v, t, B-B_1, i-1)
4. return P = P_1 concat P_2

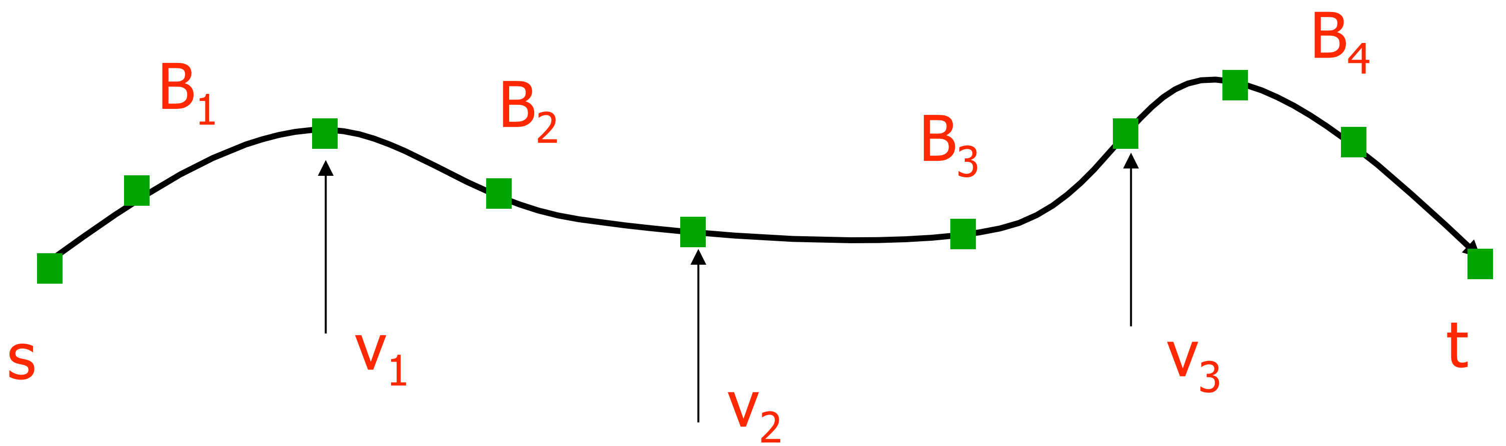
Analysis

Theorem: RG(f,s,t,B,log n) yeilds an O(log n) approximation

Running time with recursion depth i: $(nB)^{O(i)}$

Can improve to $(n \log B)^{O(i)}$: quasi-poly

Guessing more



Running time $O(n^{a \log n})$

Approximation: $\log n / \log (a+1)$

$\log^{1-\varepsilon} n$ approximation in $\exp(n^{\varepsilon})$ time
(sub-exponential time)

Applications

Quasi-poly algorithms:

- $O(\log^2 n)$ approx for group Steiner problem in undir graphs. Current approx. is $O(\log^3 n)$ and hardness is $\Omega(\log^{2-\epsilon} n)$. SOP-TW is hard to within $\Omega(\log^{1-\epsilon} n)$ factor.
- $O(\log n)$ approx for Orienteering with time varying profits at nodes
- $O(\log n)$ approx for Orienteering with multiple disjoint time windows for each node v.

Questions

Obvious: change quasi-poly to poly.

Conjecture: $O(\log^2 n)$ approx. for group Steiner via LP.

Is there a non-trivial poly-time (poly-log?) approx for Orienteering with multiple time windows?

Group Steiner problem

Set cover + Steiner tree = group Steiner

Undirected graph $G = (V, E)$

Groups: $S_1, S_2, \ldots, S_k$, each $S_i \subseteq V$

Goal: find minimum cost tree $T = (V', E')$ such that

$|V' \cap S_i| \geq 1$ for $1 \leq i \leq k$

Group Steiner problem

$O(\log^2 n)$ approx if G is a tree

$O(\log^3 n)$ approx for general graphs

[Garg-Konjevod-Ravi'98 + ...]

$\Omega(\log^{2-\varepsilon} n)$ approx not possible even on trees unless NP contained in quasi-polynomial time

[Halperin-Krauthgamer'03]

SOP and group Steiner

Simple observation:

α-approx for SOP implies $2\alpha \log k$ approx for group Steiner problem

Consequences:

$O(\log^2 n)$ approx for group Steiner problem in quasi-poly time

$\Omega(\log^{1-\varepsilon} n)$ hardness for SOP unless NP is contained in quasi-poly time

Reduction size lower bound

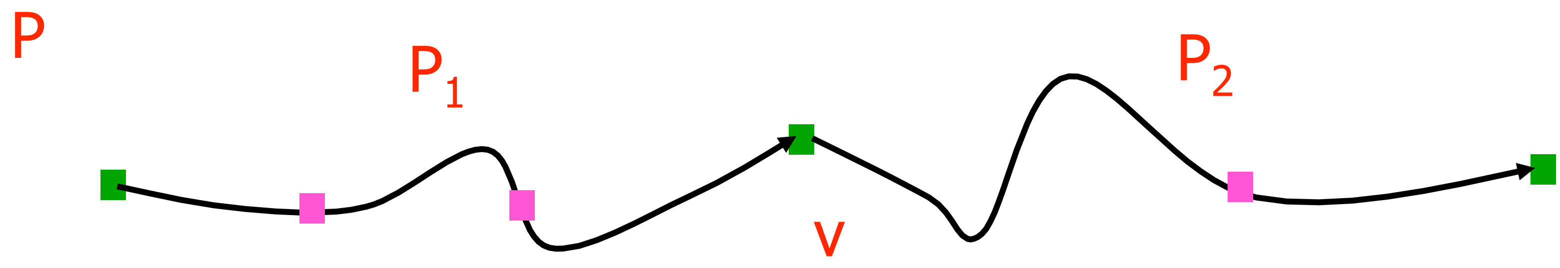
Unless NP $\subseteq$ quasi-polytime no $\log^{2-\varepsilon} n$ approx. for group Steiner problem [Halperin-Krauthgamer'03]

Can we obtain $\log^{2-\varepsilon} n$ hardness under P $\neq$ NP ?
Can reduction size by polynomial?

No, unless NP $\subseteq$ sub-exponential time

From $\log^{1-\varepsilon} n$ approx in subexp time for SOP

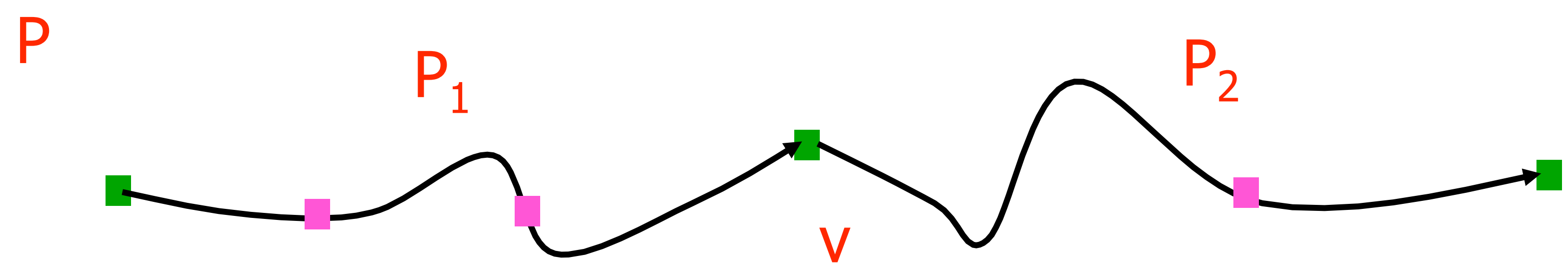
Proof



$|P_1| \geq |P_1^*| / \log (k/2)$

$|P_2| \geq ? / \log (k/2)$

Proof



$|P_1| \geq |P_1^*| \,/\, \log(k/2)$

$|P_2| \geq |P_2^* \setminus P_1| \,/\, \log(k/2)$

$\geq (|P_2^*| - |P_1|) \,/\, \log(k/2)$

Proof contd

$|P| \geq (|P^*| - |P|) / \log(k/2)$

$|P| \geq |P^*| / (1 + \log(k/2))$

$\geq |P^*| / \log k$

Lemma: α approx for recursive step implies $\alpha+1$ approx for greedy step

[Fisher-Nemhauser-Wolsey'78]

Open Problems: Summary

	Undir Graphs	Dir Graphs
Orienteering	$2+\epsilon$	$O(\log n)$* $O(\log^2 n)$
k-Stroll	$2+\epsilon$	?
Orienteering-TW	$O(\log^2 n)$ $O(\log L_{max})$	$O(\log n)$* $O(\log^4 n)$ $O(\log^2 n \log L_{max})$
Multiple TWs/node	$O(\log n)$*	$O(\log n)$*

Only APX-hardness for all of the above problems!

* : quasi-poly running time