last days:

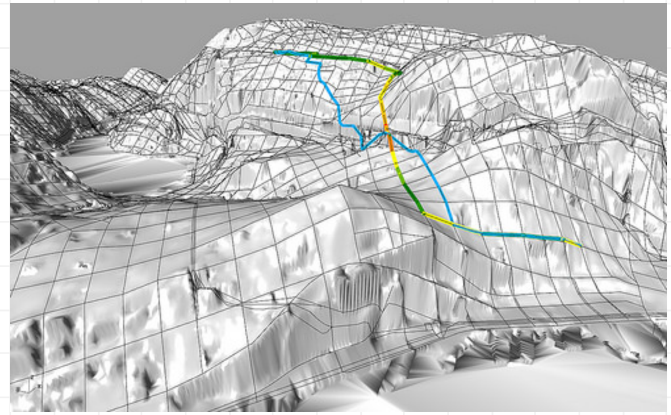
- some basic geometric shortest path algorithms
- some possible papers to present

today:

- more papers
- basic graph algorithms

shortest path on a polyhedral surface

generalizes polyhedral domain

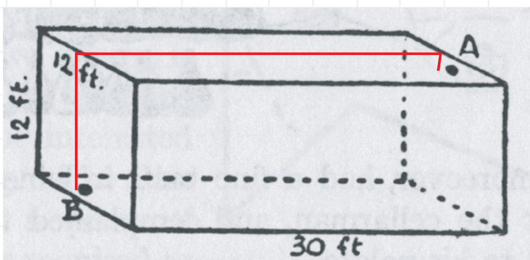


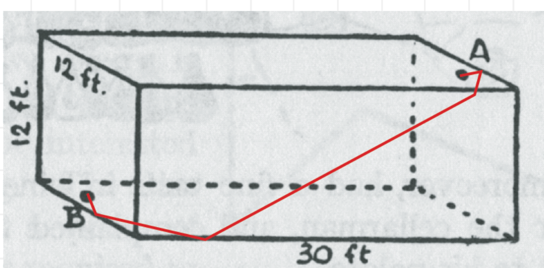
shortest paths follow straight lines in *unfolding*

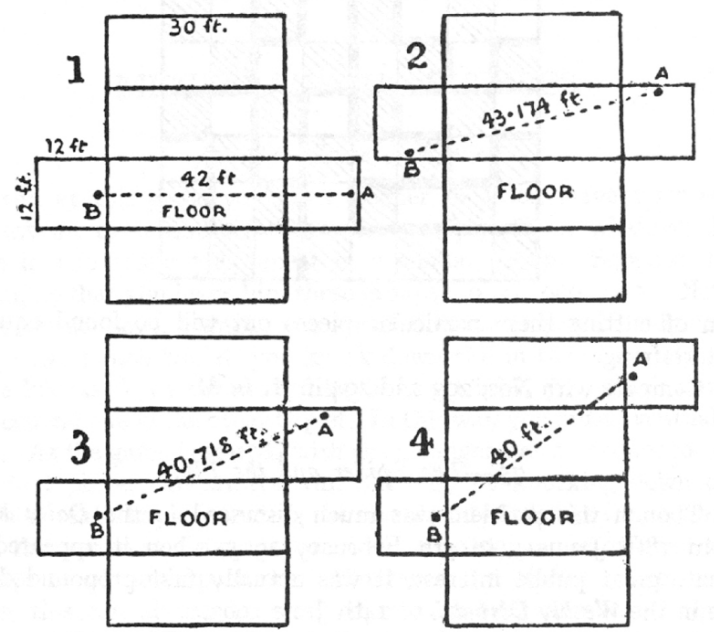
the spider and the fly problem

Dudeney, The Canterbury Puzzles, 1958

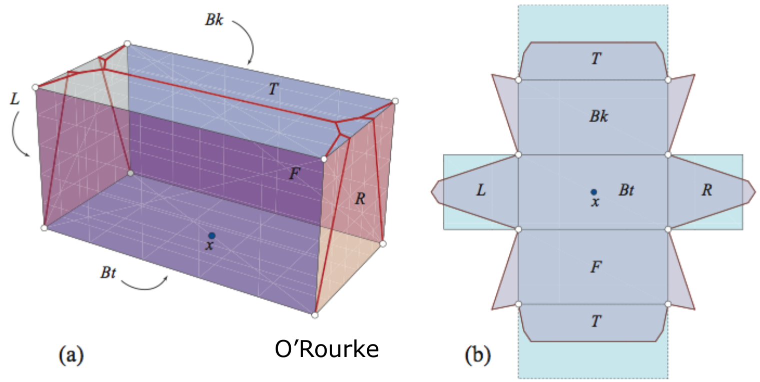
?



shortest path on a polyhedral surface




O'Rourke

**The discrete geodesic problem**

JSB Mitchell, DM Mount, CH Papadimitriou - SIAM Journal on Computing, 1987 - SIAM
We present an algorithm for determining the shortest path between a source and a destination on an arbitrary (possibly nonconvex) polyhedral surface. The path is constrained to lie on the surface, and distances are measured according to the Euclidean metric. Our **...**
Cited by 472 Related articles All 39 versions Cite Save

From: http://scholar.google.ca/scholar?hl=en&q=THE+DISCRETE+GEODESIC+PROBLEM&btnG=&as_sdt=1%2C5&as_sdtp=

$O(n^2 \log n)$

**Fast exact and approximate geodesics on meshes**

V Surazhsky, T Surazhsky, D Kirsanov… - ACM Transactions on …, 2005 - dl.acm.org
From: http://scholar.google.ca/scholar?hl=en&q=Fast+exact+and+approximate+geodesics+on+meshes&btnG=&as_sdt=1%2C5&as_sdtp=

**Shortest paths on a polyhedron, Part I: Computing shortest paths**

J Chen, Y Han - International Journal of Computational Geometry & …, 1996 - World Scientific
We present an algorithm for determining the **shortest path** between any two points along the surface of a **polyhedron** which need not be convex. This algorithm also computes for any source point on the surface of a **polyhedron** the inward layout and the subdivision of the **...**
Cited by 115 Related articles All 4 versions Cite Save

From: http://scholar.google.ca/scholar?q=Shortest+paths+on+a+polyhedron%2C+Part+I%3A&btnG=&hl=en&as_sdt=0%2C5

$O(n^2)$

an algorithm by Kapoor claimed $O(n \log^2 n)$ but it's in doubt

## shortest path on a polyhedral surface

**An optimal-time algorithm for shortest paths on a convex polytope in three dimensions**

$O(n \log n)$

cont. Dijkstra

Y Schreiber, M Sharir - Twentieth Anniversary Volume:, 2009 - Springer

Abstract We present an **optimal-time algorithm** for computing (an implicit representation of) the **shortest-path** map from a fixed source s on the surface of a **convex polytope** P in **three dimensions**. Our **algorithm** runs in O (nlogn) time and requires O (nlogn) space, where n is **...**

Cited by 40 Related articles All 18 versions Cite Save

From: http://scholar.google.ca/scholar?q=An+optimal-time+algorithm+for+shortest+paths+on+a+convex+polytope+in+three+dimensions&btnG=&hl=en&as_sdt=0%2C5

## more papers listed are here:

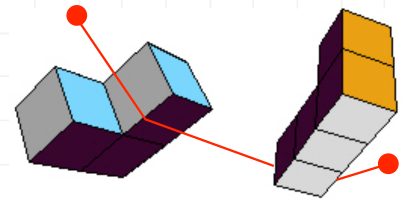
**A survey of geodesic paths on 3D surfaces**

P Bose, A Maheshwari, C Shu, S Wuhrer - Computational Geometry, 2011 - Elsevier

From: http://scholar.google.ca/scholar?hl=en&q=A+survey+of+geodesic+paths+on+3D+surfaces&btnG=&as_sdt=1%2C5&as_sdtp=

shortest paths in 3D

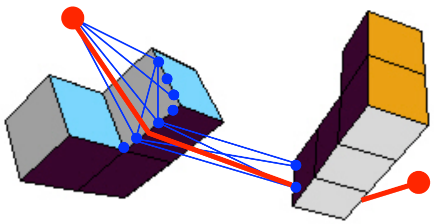
NP-hard:



**New lower bound techniques for robot motion planning problems**
J Canny, J Reif - … of Computer Science, 1987., 28th Annual …, 1987 - ieeexplore.ieee.org
Cited by 456Related articlesAll 8 versionsCiteSave

From: http://scholar.google.ca/scholar?q=New+lower+bound+techniques+for+robot+motion+planning+problems&btnG=&hl=en&as_sdt=0%2C5

don't present



**An algorithm for shortest-path motion in three dimensions**
CH Papadimitriou - Information Processing Letters, 1985 - Elsevier
Abstract We describe a fully polynomial approximation scheme for the problem of finding the shortest distance between two points in **three-dimensional** space in the presence of polyhedral obstacles. The fastest **algorithm** known for the exact solution of this problem is **...**
Cited by 197Related articlesAll 2 versionsCiteSave

From: http://scholar.google.ca/scholar?q=An+algorithm+for+shortest-path+motion+in+three+dimensions&btnG=&hl=en&as_sdt=0%2C5

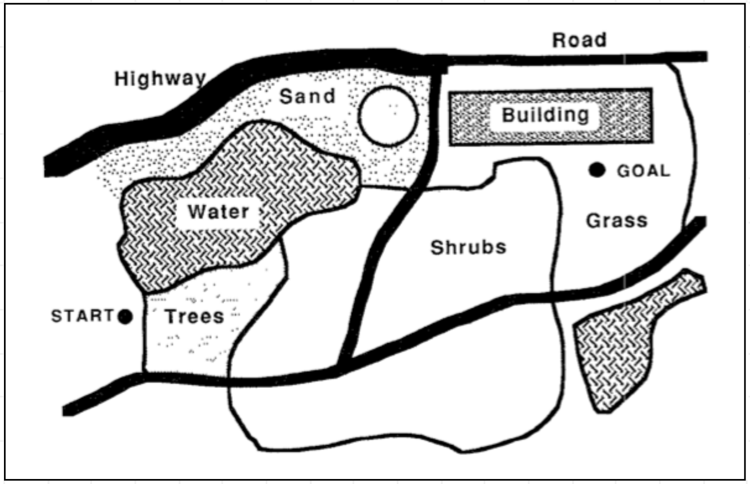
fixes above:

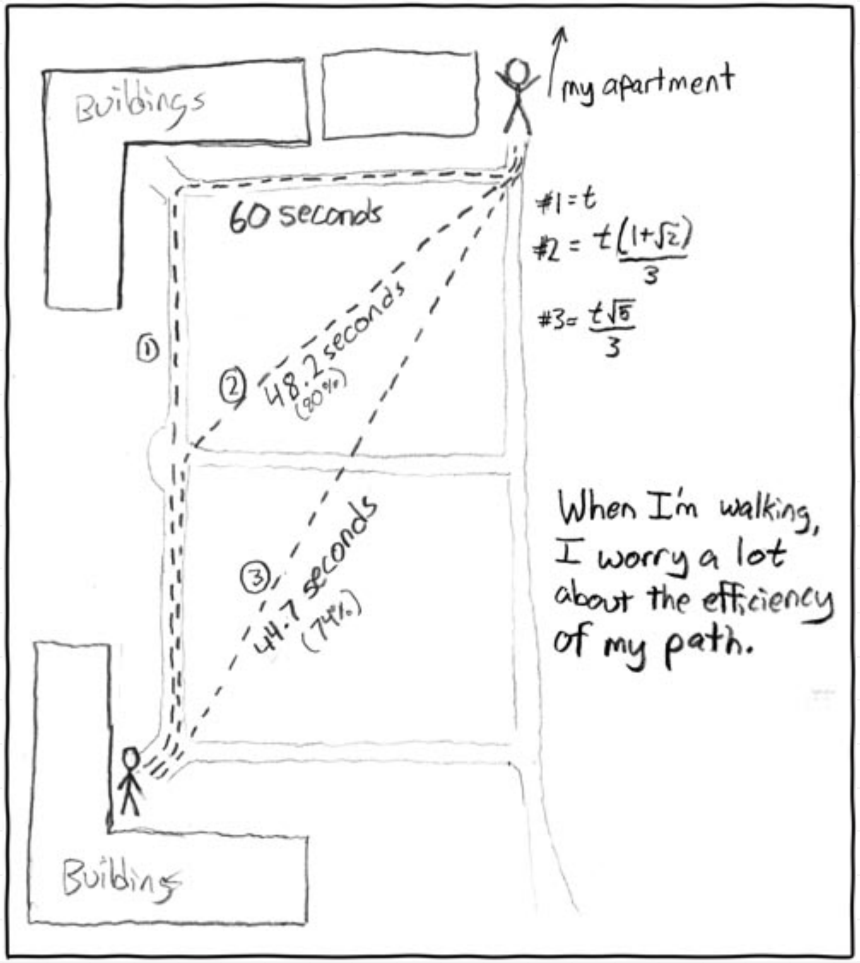
**Precision-sensitive Euclidean shortest path in 3-Space**
J Sellen, J Choi, CK Yap - SIAM Journal on Computing, 2000 - SIAM
From: http://scholar.google.ca/scholar?q=Precision-sensitive+Euclidean+shortest+path+in+3-Space&btnG=&hl=en&as_sdt=0%2C5

new!

**An approximation algorithm for computing shortest paths in weighted 3-d domains** — generalizes next topic
L Aleksandrov, H Djidjev, A Maheshwari… - Discrete & **Computational** …, 2013 - Springer
From: http://scholar.google.ca/scholar?q=An+approximation+algorithm+for+computing+shortest+paths+in+weighted+3-d+domains&btnG=&hl=en&as_sdt=2005&sciodt=0%2C5&cites=13772326056646696620&scipsc=

weighted region problem

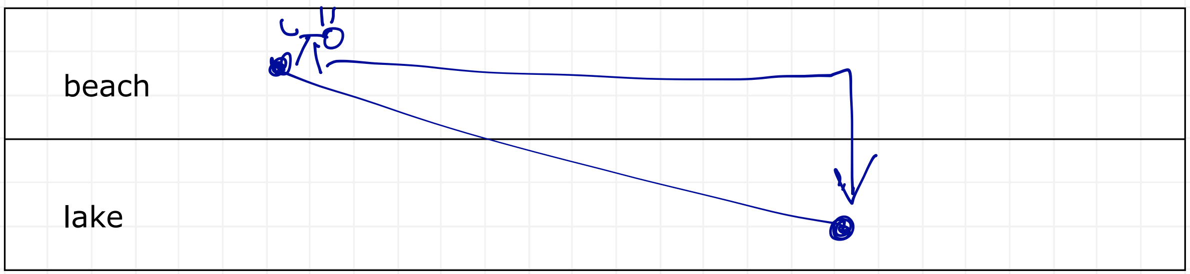




Buildings
my apartment
60 seconds
#1 = t
#2 = t(1+√2)/3
#3 = t√5/3
①
② 48.2 seconds (80%)
③ 44.7 seconds (74%)
When I'm walking, I worry a lot about the efficiency of my path.
Buildings

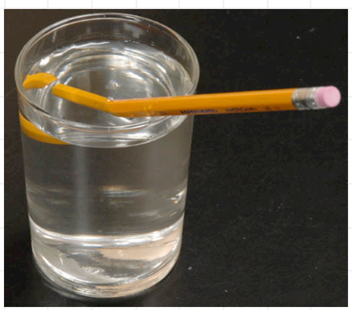
xkcd

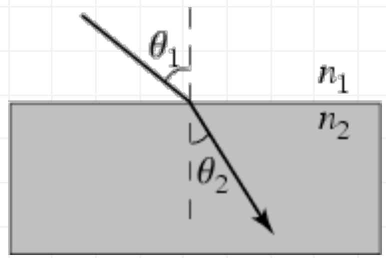
weighted region problem

intuition

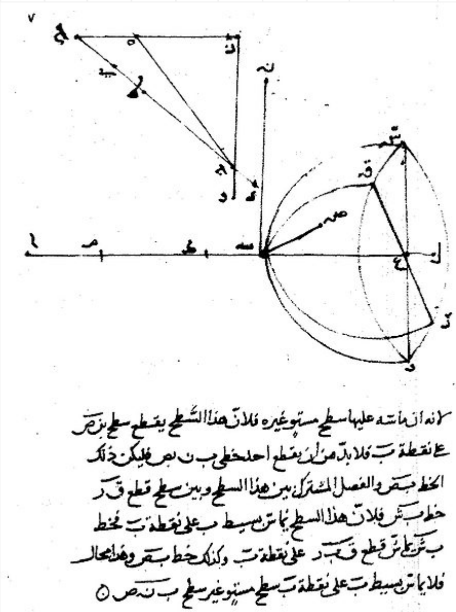



Snell's law






$$n_1 \sin \theta_1 = n_2 \sin \theta_2,$$



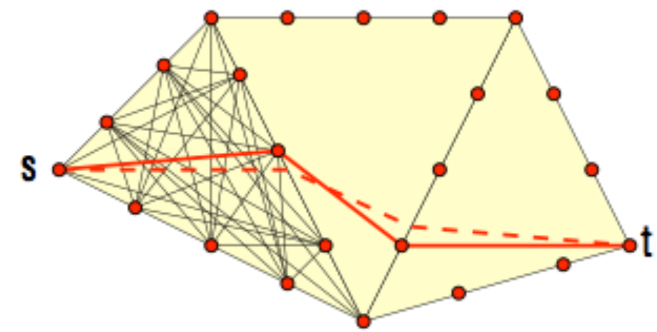
- Ibn Sahl, (Baghdad), On Burning Mirrors and Lenses, 984
- Willebrord Snellius, 1621
- René Descartes, 1637

weighted region problem

[The **weighted region** problem: finding shortest paths through a **weighted** planar subdivision]
JSB Mitchell, CH Papadimitriou - Journal of the ACM (JACM), 1991 - dl.acm.org
Cited by 264 Related articles All 4 versions Cite Save
From: http://scholar.google.ca/scholar?hl=en&q=weighted+region&btnG=&as_sdt=1%2C5&as_sdtp=

OPEN polytime? NP-hard?

main approach: approximation via Steiner points

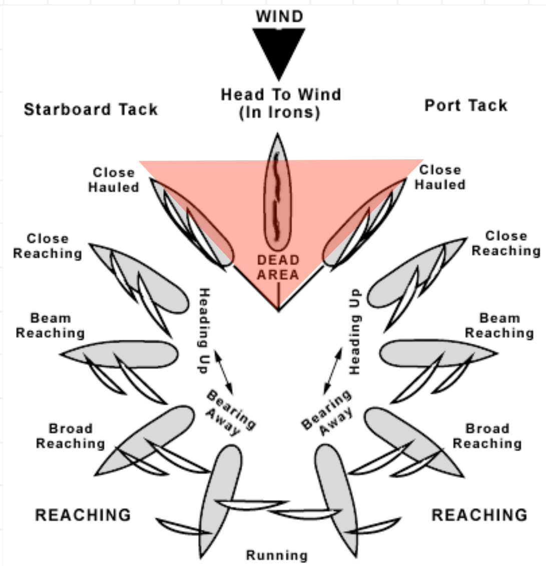


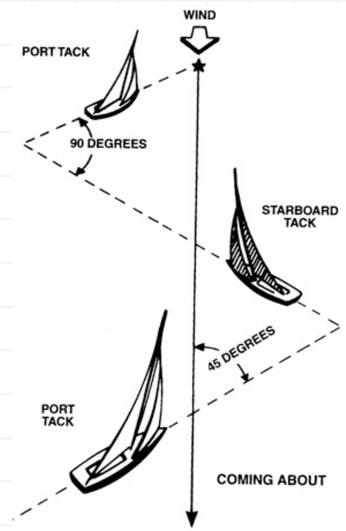

**Determining approximate shortest paths on weighted polyhedral surfaces**
L Aleksandrov, A Maheshwari, JR Sack - Journal of the ACM (JACM), 2005 - dl.acm.org
From: http://scholar.google.ca/scholar?q=Determining+approximate+shortest+paths+on+weighted+polyhedral+surfaces&btnG=&hl=en&as_sdt=0%2C5

**On finding approximate optimal paths in weighted regions**
Z Sun, JH Reif - Journal of Algorithms, 2006 - Elsevier
From: http://scholar.google.ca/scholar?q=On+finding+approximate+optimal+paths+in+weighted+regions&btnG=&hl=en&as_sdt=0%2C5

problems related to weighted region problem

anisotropic path problem



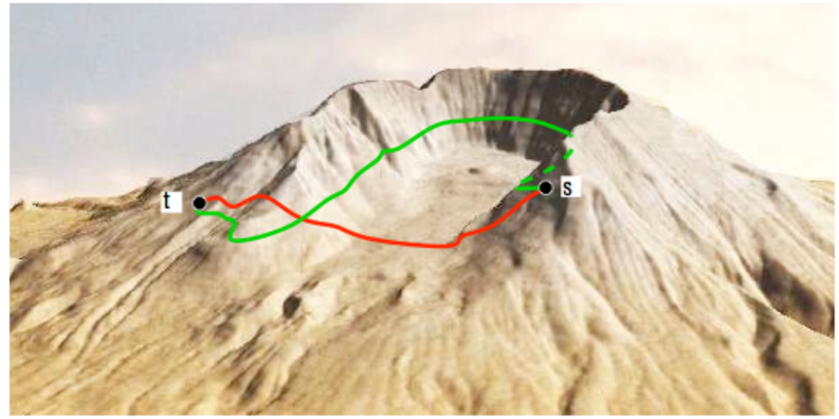




cost of travel depends on direction (may be different in different faces)

very engineering oriented papers

problems related to weighted region problem

shortest descending path on a terrain




OPEN poly time? NP-hard?

also related to watersheds

anisotropic

*approximation algorithms*

weighted region

shortest descending path

problems related to weighted region problem

shortest descending path on a terrain

alg. to decide: is there an s→t path

**Trekking** in the **Alps without freezing** or getting **tired**
M De Berg, M van Kreveld - Algorithmica, 1997 - Springer
From: http://scholar.google.ca/scholar?q=trekking+in+the+alps+without+freezing&btnG=&hl=en&as_sdt=0%2C5

[HTML] **Approximation algorithms** for **shortest descending paths** in **terrains**
M Ahmed, S Das, S Lodha, A Lubiw… - … of Discrete **Algorithms**, 2010 - Elsevier
From: http://scholar.google.ca/scholar?hl=en&q=Approximation+algorithms+for+shortest+descending+paths+in+terrains&btnG=&as_sdt=1%2C5&as_sdtp=

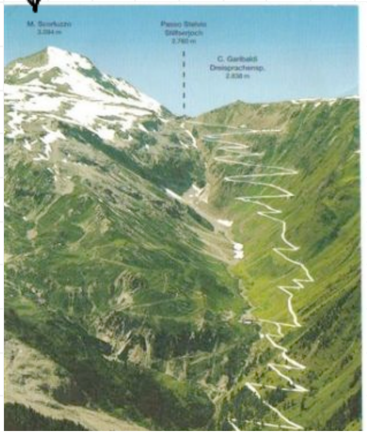
includes

**Shortest descending paths**: **Towards** an **exact algorithm**
M Ahmed, A Lubiw - International Journal of Computational …, 2011 - World Scientific
From: http://scholar.google.ca/scholar?q=Shortest+descending+paths%3A+Towards+an+exact+algorithm&btnG=&hl=en&as_sdt=2005&sciodt=0%2C5&cites=15807659110083868733&scipsc=

* new!

**Approximate shortest descending paths**
SW Cheng, J Jin - SIAM Journal on Computing, 2014 - SIAM
From: http://scholar.google.ca/scholar?q=Approximate+shortest+descending+paths&btnG=&hl=en&as_sdt=2005&sciodt=0%2C5&cites=15807659110083868733&scipsc=

shortest *gently* descending path on a terrain



discussion of Dijkstra's paper

- reviews well done on the whole
- many did more than necessary
- be careful to acknowledge sources

other problem is min spanning tree — algorithm is known as Prim's algorithm though actually due to others earlier:

Jeff Erickson's algorithms notes:

### 20.4 Jarník's ('Prim's') Algorithm

The next oldest minimum spanning tree algorithm was first described by the Czech mathematician Vojtěch Jarník in a 1929 letter to Borůvka; Jarník published his discovery the following year. The algorithm was independently rediscovered by Kruskal in 1956, by Prim in 1957, by Loberman and Weinberger in 1957, and finally by Dijkstra in 1958. Prim, Loberman, Weinberger, and Dijkstra all (eventually) knew of and even cited Kruskal's paper, but since Kruskal also described two other minimum-spanning-tree algorithms in the same paper, *this* algorithm is usually called 'Prim's algorithm', or sometimes 'the Prim/Dijkstra algorithm', even though by 1958 Dijkstra already had another algorithm (inappropriately) named after him.

http://web.engr.illinois.edu/~jeffe/teaching/algorithms/notes/20-mst.pdf shortest paths in graphs — NP-complete in general

single source
- Dijkstra — no neg. weights.
- Bellman-Ford — no neg. weight cycle

all pairs
- Floyd-Warshall

---

all use "relaxation", "label correcting"

Maintain tentative distances $d(v)$, <u>over</u> estimates.

goal $\forall$ edge $(u,v)$ $\quad d(v) \le d(u) + w(u,v)$

weight of edge

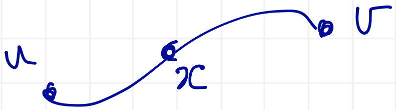
Find edge where this is violated & update $d(v)$ to satisfy it

Dijkstra — one vertex order suffices

Bellman-Ford — dynamic programming. (<u>the</u> original dyn. prog. alg.)

Bellman-Ford - single source shortest path algorithm for no negative weight cycle

idea of dynamic prog. for shortest paths




if shortest uv path goes thru x (we can try all x)
then it consists of

$\underbrace{\text{shortest path } ux + \text{shortest path } xv}_{\text{these are subproblems}}$

how are they smaller?

1. fewer vertices.
2. do not use x

} 2 possibilites lead to different algs.

Bellman Ford - 1 Try paths of ≤ 1 edge, ≤ 2 edges ...

Floyd-Warshall - 2

Bellman-Ford - single source shortest path algorithm for no negative weight cycle

$d_i(v)$ — length of shortest path $s \to v$ using $\leq i$ edges

initialize

$$d_1(v) = \begin{cases} w(s,v) & \text{if } (s,v) \in E \\ \infty & \text{else} \end{cases}$$

$d_1(s) = 0$

Want $d_{n-1}(v)$ — no neg. weight cycle

Compute $d_i$ from $d_{i-1}$

For $i = 2 \ldots n-1$

For each $v$

For each edge $(u,v)$

$d_i(v) \leftarrow \min \{ d_{i-1}(v), \; d_{i-1}(u) + w(u,v) \}$

i.e. $\forall$ edge $(u,v)$

end

Bellman-Ford - single source shortest path algorithm for no negative weight cycle

don't need $i$

initialize

$d(v) = \infty$, $d(s) = 0$

For $i = 1 .. n-1$

for each edge $(u,v)$

$d(v) \leftarrow \min \{ d(v), d(u) + w(u,v) \}$

end

could use any edge order

$O(n \cdot m)$ — $n$ = # vertices, $m$ = # edges

best known strongly poly. alg.

EX. Find actual path

EX. Test for neg. cycles. Can we do this faster?

all pairs shortest path algorithms. Given digraph G with weights w:E -> R, and no negative weight cycle, find shortest path from i to j for all vertices i, j.

Floyd-Warshall

$V = \{1 .. n\}$

$d_i(j,k)$ = length of shortest path $j \to k$ using intermediate vertices in $\{1, .., i\}$

dyn. prog. — solve $i = 0, .. n$ $d_i(j,k)$ $\forall j, k$

Want $d_n$

initially $d_0(j,k) = \begin{cases} w(j,k) & \text{if } (j,k) \in E \\ \infty & \text{else} \end{cases}$

$d_i(j,k) = \min \begin{cases} d_{i-1}(j,k) & \text{— do not use } i \\ d_{i-1}(j,i) + d_{i-1}(i,k) & \text{— use vertex } i \end{cases}$

just reuse space $d(j,k)$

all pairs shortest path algorithms. Given digraph G with weights w:E -> R, and no negative weight cycle, find shortest path from i to j for all vertices i, j.

Floyd-Warshall

initialize $d(j,k)$ as above
for $i = 1 \cdots n$
  for $j = 1 \cdots n$
    for $k = 1 \cdots n$
      $d(j,k) \leftarrow \min\{d(j,k),\ d(j,i) + d(i,k)\}$
end

$O(n^3)$
Space $O(n^2)$

Ex: detect neg. cycles
Ex: find actual path