TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

CrowdPlanr: Planning Made Easy with Crowd

Ilia Lotosh Tova Milo Slava Novgorodov

Goal

Crowd-sourcing can be used effectively to solve problems that are difficult for computers.

CrowdPlanr uses the power of the crowd to solve a problem of planning a sequence of actions, when the goal is hard to formalize.

An example of a problem

Prepare a schedule for a vacation trip

Input:

Dates, budget, location, interests

Desired output:

A detailed schedule of what to do and when on your vacation

Goal:

Enjoy your vacation the most!

Example input:

I am in Brisbane for an ICDE'13 conference. In my spare time I would like to explore the city. What should I do?

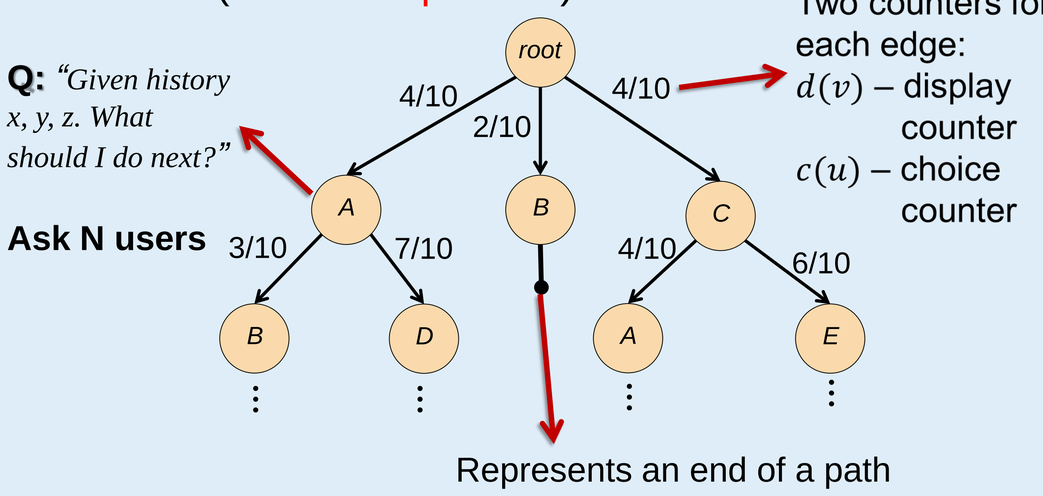
Possible output:

City Hall ⟶ Treasury Building ⟶ Customs House ⟶ City Botanic Gardens ⟶ Brisbane Arcade ⟶ South Bank Parklands ⟶ Wheel of Brisbane ⟶ Streets Beach

Model presentation

We assume that we have a set of possible items **S**.

All possible sequences of items from **S** are presented as a tree (order is important!).



Score can be calculated for every path $p = (u_1, u_2, \dots, u_n)$

$$score(p) = \prod_{i=2}^{n} \frac{c(u_i)}{d(u_{i-1})}$$

We shall return a path with a highest score

Challenges

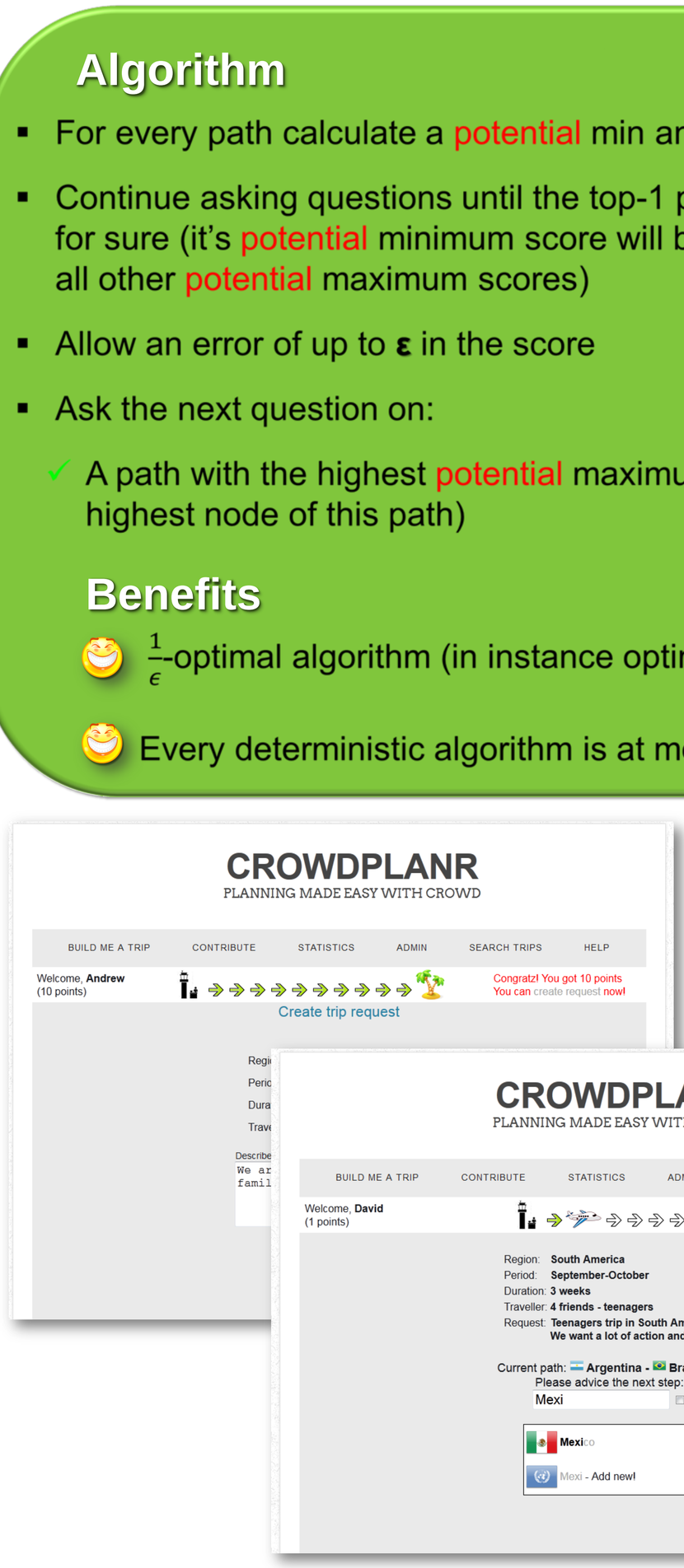
- **The tree of all the plans is exponential in the length of a plan**
- **Most of the plans are irrelevant (will have very low score)**
- **Asking questions is expensive (in time and possibly money)**

Algorithm

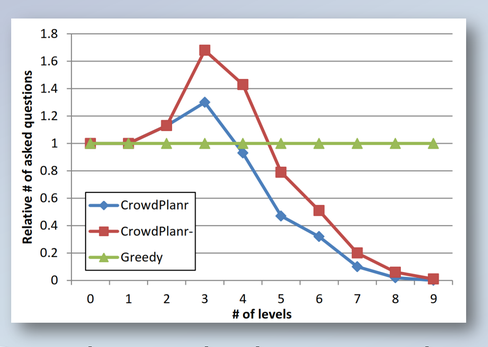
- For every path calculate a potential min and max scores
- Continue asking questions until the top-1 path is known for sure (it's potential minimum score will be higher than all other potential maximum scores)
- Allow an error of up to **ε** in the score
- Ask the next question on:
  - ✓ A path with the highest potential maximum score (a highest node of this path)

Benefits

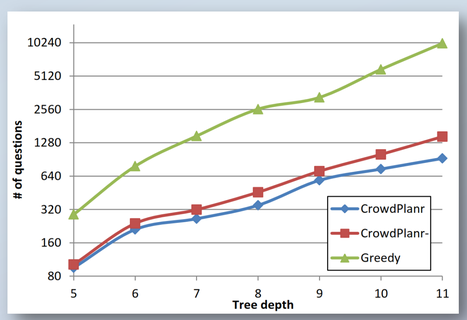
- $\frac{1}{\epsilon}$-optimal algorithm (in instance optimality sense)
- Every deterministic algorithm is at most $\frac{1}{\epsilon}$-optimal



Experimental Results



Questions asked on every level



#Questions function of tree depth