Events Organizer on Social Networks

Julien Odent
Nov 6, 2012

Events Organizer: Overview

- Modeling
  - Variables
  - Constraints
  - Objectives
  - Neighborhood
- Implementation
- Improvements
- Extensions

Events Organizer: Variables

Assume n, the number of participants and m, the number of events

INPUTS:

c: capacity vector of length m

c_i = maximum capacity of event i (0 if unlimited)

p: preferences matrix of size n x m

p_{ij} = 1 if participant i wishes to attend event j

0 otherwise

d: exclusion matrix of size m x m

d_{ij} = 1 if event i is exclusive with event j

0 otherwise

Events Organizer: Variables

Assume n, the number of participants and m, the number of events

INPUTS:

G = (V, E): connections' graph

$\forall i : participant : i \in V$

$\forall i < j : (i, j) \in E$ iff participant i is friend with participant j

emax: vector of length n

$emax_i$ = maximum # events participant i wishes to attend

emin: vector of length n

$emin_i$ = minimum # events participant i wishes to attend

Events Organizer: Variables

Assume n, the number of participants and m, the number of events

OUTPUT:

s: attending matrix of size n x m

s_{ij} = 1 if participant i attends event j

0 otherwise

Events Organizer: Constraints

Preferences constraint $\forall i, j : s_{ij} - p_{ij} \leqslant 0$

Max capacity constraint $\forall j, c_j > 0 : \Sigma_i^m s_{ij} - c_j \leqslant 0$

“Non-ubiquity” constraint $\forall i < j, k : s_{ki} + s_{kj} + d_{ij} \leqslant 2$

$emin_i \leqslant emax_i \; \forall i$

Max events constraint $\sum_j^m s_{ij} \leqslant emax_i \; \forall i$

Min events constraint (soft) $\sum_j^m s_{ij} \geqslant emin_i \; \forall i$

Events Organizer: Objectives

Objectives

- Maximize the number of attendees $\sum_i^n \sum_j^m s_{ij}$
- Maximize the number of connections $\sum_i^m |E_i|$
  - where E_i = edges of $G_i \subseteq G$, the connections' graph between attending participants of event i
- Maximize the minimal preferences $\sum_i^n \frac{\sum_j^m s_{ij}}{emin_i}$
  - quotient equals to 1 if no preference
  - quotient equals to 1 if it's greater than 1
- Objective:

$$\sum_i^n \sum_j^m s_{ij} \; + \; \sum_i^m |E_i| \; + \; \sum_i^n \frac{\sum_j^m s_{ij}}{emin_i}$$

Events Organizer: Neighborhood

- Defined by the solutions obtained by applying one of the operations
- Operations
  - ADD $s_{ij}=0$ $s'_{ij}=1$
  - REMOVE $s_{ij}=1$ $s'_{ij}=0$ (useful?)
  - SWAP

    for one participant, between two events

    $s_{ik}=1 \wedge s_{il}=0$ $s_{ik}=0 \wedge s_{il}=1$

    for one event, between two participants

    $s_{ki}=1 \wedge s_{li}=0$ $s_{ik}=0 \wedge s_{il}=1$

Events Organizer: Implementation

- Copy p into s
- Make s consistent with capacity vector
- Make s consistent with exclusive events (d matrix) by removing the problematic attendees (one for each overlap)

  => s is the initial solution
- Tabu elements (features): participants
- Features expire at each iteration

Events Organizer: Improvements

- Find the best linear combination of objectives (1 + 1 + 1)
- Find the best value for the number of iterations (attemps = 100)
- Find the best initial value when adding in tabu list (attemps / 4)
- Find the best size of the tabu list (|p| / 2)
- Find the amount of decreasing value when expiring features and also the rate at which expiring features (1, at each iteration)
- Or perhaps another mean to expire features? (when tabu list is full, on time, something else...)

Events Organizer: Extensions

- Min attendees for an event to actually occur

  => Drop a few events in order to maximize the others (threshold)
- Total ordering of the preferences of a given participant
- Possibility to include more constraints
  - Minimum/maximum age requirement (hard)
  - Average age of participants (soft)
  - Girl to boy ratio (soft)
- Real-time computation, handling of events