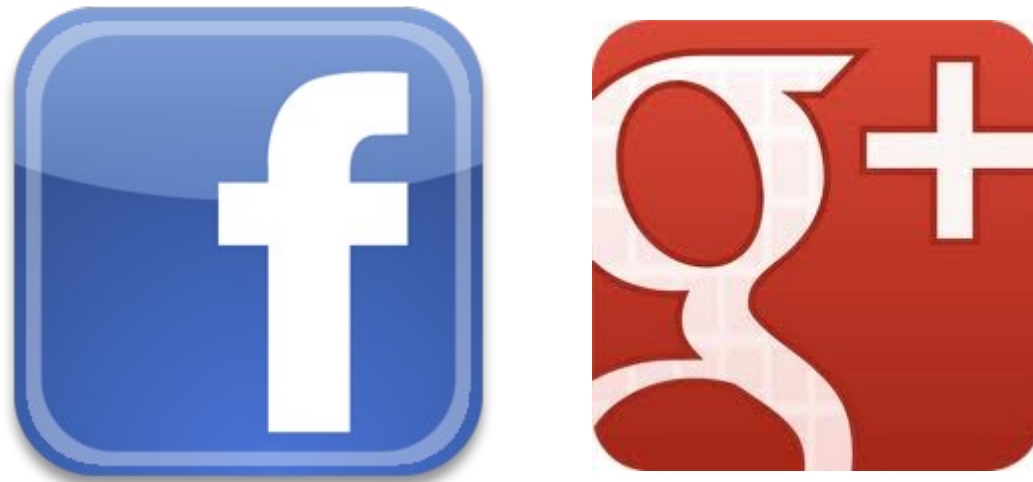


Events Organizer on Social Networks



Julien Odent
Dec 4, 2012

Events Organizer: Overview

- Modeling: variables and constraints
- LS concepts
 - Objective function
 - Neighborhood
 - Legal moves
 - Selection
- Implementation
- Improvements and 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 $n \times m$

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

d : exclusion matrix of size $m \times m$

$d_{ij} = 1$ 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:

cf : $n \times n$ adjacency matrix of “close friends” graph

$cf_{ij} = 1$ iff participant i has participant j among its close friends
= 0 otherwise

$emax$: 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 $n \times m$

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

0 otherwise

Events Organizer: Constraints

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

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

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

Max events constraint $\sum_j^m s_{ij} \leq \text{emax}_i \forall i$

Min events constraint (soft) $\sum_j^m s_{ij} \geq \text{emin}_i \forall i$

Events Organizer: LS concepts

Neighborhood $N(s)$: set of neighboring solutions of s

Legal moves $L(N, t)$: filters out forbidden moves from N

Selection $S(L, s)$: how to choose the neighbor

Objective $O(s)$: defines the score of s

Events Organizer: Objectives

Objectives

Maximize the number of connections

$$\sum_j^m \sum_{i,k \neq i}^n cf_{ik} s_{ij} s_{kj}$$

Maximize the minimal preferences
quotient equals to 1 if no preference

$$\sum_i^n \min\left(1, \frac{\sum_j^m s_{ij}}{emin_i}\right)$$

Objective (maximization):

$$\sum_j^m \sum_{i,k \neq i}^n cf_{ik} s_{ij} s_{kj} + \sum_i^n \min\left(1, \frac{\sum_j^m s_{ij}}{emin_i}\right)$$

Events Organizer: Neighborhood

- Neighborhood $N(s)$: set of neighboring solutions of s
- Operations
 - ADD $s_{ij}=0 \quad s'_{ij}=1$
 - REMOVE $s_{ij}=1 \quad s'_{ij}=0$
 - MOVE $s_{ij}=1 \wedge s_{ik}=0 \quad s_{ij}=0 \wedge s_{ik}=1$
 $\text{move}(i, j, k) = \text{remove}(i, j) + \text{add}(i, k)$
 - REPLACE $s_{hj}=1 \wedge s_{ij}=0 \quad s_{hj}=0 \wedge s_{ij}=1$
 $\text{replace}(h, i, j) = \text{remove}(h, j) + \text{add}(i, j)$
 - SWAP $s_{hj}=1 \wedge s_{hk}=0 \wedge s_{ik}=1 \wedge s_{ij}=0 \quad s_{hj}=0 \wedge s_{hk}=1 \wedge s_{ik}=0 \wedge s_{ij}=1$
 $\text{swap}(h, i, j, k) = \text{move}(h, j, k) + \text{move}(i, k, j)$

Events Organizer: Legal moves

- Legal moves $L(N, t)$: filters out forbidden moves from N
- A move is legal if none of its parameters (participant indices) is within the tabu list

$$L(N, t) = \{n \in N : n \notin t\}$$

- Aspiration criterion: Same as above OR $O(s) > O(s^*)$

Events Organizer: Selection

- Selection $S(L, s)$: how to choose the neighbor
- Best Neighbor heuristic (randomly)

$$S(L, s) = \{n \in L : O(n) = \max_{s \in L} O(s)\}$$

Events Organizer: Implementation

- Takes function identifiers as arguments: initial solution, objective, neighborhood, is_legal, selection
- Provides bottom-up and top-down initial solution (randomly)
- Provides the best neighbor heuristic (selection function)
- Provides 4 objective functions (incrementally)
- Provides is_legal_not_tabu and is_legal_not_tabu_aspiration functions
- Class Status: keeps track of parameters (p, c, d, cf, emin, emax but also attempts, tenure)

Events Organizer: Improvements

- Implement and test other selection strategies (epsilon)
- Implement and test random restarts
 - After n (unfruitful or not) iterations from a new initial solution
 - After n (unfruitful or not) iterations from another selection criterion
- Find the optimal weight vector for multiobjective function (uniformly normalized?)

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