Long-Term Scheduling for Road Network Disaster Recovery

David Rey [1] and **Hillel Bar-Gera** [2]

[1]School of Civil and Environmental Engineering, UNSW Sydney, Australia
d.rey@unsw.edu.au

[2]Department of Industrial Engineering and Management, Ben-Gurion University, Beer-Sheva, Israel
bargera@bgu.ac.il

Cascading Disasters Workshop
Haifa, Isreal, October 28-29, 2018

Outline

Outline

1. Network Disaster Recovery
2. Problem Formulation
3. Numerical Results

Problem Background

Transportation needs following a disaster depend on time scale:

- Short term ($\sim$12-24 hours) - evacuation
- Mid term ($\sim$12-24 days) - very complex
- $\rightarrow$ Long term ($\sim$12-24 months) - recovery

Cascading effects are due to interactions between decisions.
Interactions can be negative or positive, of different types:

- $\rightarrow$ Objective function
- $\rightarrow$ Feasibility of other decisions
- Demand for transportation

Problem Background (cont)

Network recovery can be organized into projects on damaged roads

Each project is assumed to have the following information available:

- Set of network links
- Project duration (may depend on other projects)
- Link capacity reduction
- Free-flow travel time augmentation

Method to measure congestion impacts $\rightarrow$ traffic assignment

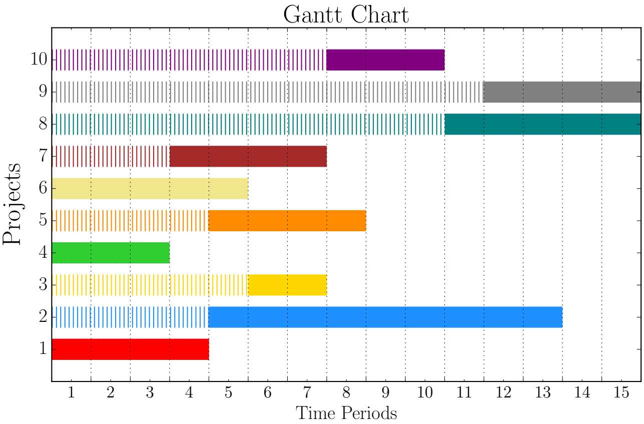
$\rightarrow$ Poor coordination of projects can lead to severe delays due to nonlinear congestion effects under User Equilibrium (UE)

Modelling Approach (1)

We make the following assumptions:

- Recovery projects must be scheduled in consecutive time periods
- Each link of the network is affected by at most one project over the planning period
- The planning period can be discretized, *e.g.* weeks, months
- Travellers' make route choice decisions based on the UE conditions
- System performance can be measured by the total network (i.e. total system travel time) over the planning horizon

Illustration: 10 projects, 15 time periods



→ At each time period we determine the total network delay based on damaged and under repair links/projects — e.g. at time period 9, network delay is influenced by Projects 2 and 10 which are under repair and Projects 8 and 9 which are still damaged

Modelling Approach (2)

Our objective is to minimize the total network delay: at each period of the planning horizon, the network delay is determined by the traffic flow pattern resulting from links current state

→ We use a bi-level programming formulation to represent this disaster recovery scheduling problem

- Upper-level: Scheduling problem - Minimize total network delay of project schedule subject to recovery resource availability constraints
- Lower-level: Traffic Assignment Problem (TAP) - User Equilibrium with adjusted link state

Outline

Traffic Assignment Problem (TAP)

Traffic assignment problem notation:

N	set of nodes
A	set of links
W	set of OD pairs
Q_w	demand of OD pair $w \in W$
Π_w	set of paths for $w \in W$
f_k^w	flow on path k for OD w
$\delta_{a,k}^w$	link-path binary matrix
x_a	link flow on $a \in A$
m_a	state of link $a \in A$
$t_a(x_a, m_a)$	travel time function on link $a \in A$ (e.g. BPR)

Link travel time is represented as a function of flow x_a and link state m_a (i.e. *damaged*, *under repair* or *repaired*).

TAP Representation (Beckmann, 1956)

$$
\begin{aligned}
\boldsymbol{x}^\star(\boldsymbol{m}) = \arg\min \quad & \sum_{a \in A} \int_0^{x_a} t_a(v, m_a) dv \\
\text{s.t.} \quad & \sum_{k \in \Pi_w} f_k^w = Q_w && \forall w \in W \\
& \sum_{w \in W} \sum_{k \in \Pi_w} f_k^w \delta_{a,k}^w = x_a && \forall a \in A \\
& f_k^w \geq 0 && \forall w \in W, k \in \Pi_w \\
& x_a \geq 0 && \forall a \in A
\end{aligned}
$$

State-dependent Network Delay under UE

$$
\mathcal{D}(\boldsymbol{m}) = \boldsymbol{x}^\star(\boldsymbol{m})^\intercal \, \boldsymbol{t}(\boldsymbol{x}^\star(\boldsymbol{m}), \boldsymbol{m}) = \sum_{a \in A} x_a^\star(m_a) t_a(x_a^\star(m_a), m_a)
$$

Project Scheduling

P	set of projects
D_p	duration of $p \in P$
T	set of time periods ($\|T\|$ planning horizon)
F_p	feasible start times for $p \in P$: $F_p \equiv \{0, \dots, \|T\| - D_p\}$

Start time variables

$$g_{s,p} \equiv \begin{cases} 1 \text{ if project } p \text{ starts at time period } s \\ 0 \text{ otherwise} \end{cases} \quad \forall p \in P, \forall s \in F_p$$

Schedule constraints

$$\sum_{s \in F_p} g_{s,p} = 1 \quad \forall p \in P$$

Projects and Affected Links

A_p	set of links affected by $p \in P$
m_a	state of link $a \in A_p$

Affected links have 3 states: damaged $\rightarrow$ under repair $\rightarrow$ repaired

- At $t = 0$, all links $a \in A_p, p \in P$ are damaged (other links remain unaffected)
- If a recovery project p is **active**, all links $a \in A_p$ are under repair
- If a recovery project p is completed, *i.e.* has been **active** for D_p consecutive time periods, all links $a \in A_p$ are repaired

Project Pattern Representation

The network experiences a varying TSTT depending on which projects are active $\rightarrow$ projects combinations. The number of distinct flow patterns is $2^{|P|}$

Let $\sigma \in \{1, \ldots, 2^{|P|}\} = \Sigma$ be a *pattern* of projects

Pattern variables

$$k_{\sigma}^{t} \equiv \begin{cases} 1 \text{ if pattern } \sigma \text{ is selected at time } t \\ 0 \text{ otherwise} \end{cases} \qquad \forall \sigma \in \Sigma, \forall t \in T$$

Pattern selection constraints

$$\sum_{\sigma \in \Sigma} k_{\sigma}^{t} = 1 \quad \forall t \in T$$

Linking Schedule and Pattern Variables

To link the project schedule variables with the pattern selection variables, we introduce two binary matrices $[\gamma_{\sigma,p}]$ and $[\omega^t_{s,p}]$:

$$\gamma_{\sigma,p} \equiv \begin{cases} 1 & \text{if project } p \text{ is active in pattern } \sigma \\ 0 & \text{otherwise} \end{cases} \quad \forall p \in P, \forall \sigma \in \Sigma$$

$$\omega^t_{s,p} \equiv \begin{cases} 1 & \text{if } 0 \leq t \leq s + D_p - 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall p \in P, \forall s \in F_p, \forall t \in T$$

Linking constraints

$$\sum_{\sigma \in \Sigma} k^t_\sigma \gamma_{\sigma,p} = \sum_{s \in F_p} g_{s,p} \omega^t_{s,p} \quad \forall p \in P, \forall t \in T$$

Linking Schedule and Pattern Variables (2)

For instance, if $P = \{A, B, C\}$, $T = \{0, \dots, 7\}$ and $D_A = 5$

$$[\gamma_{\sigma,p}] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad [\omega^t_{s,A}] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Project A can start at $t = 0, 1, 2$ or 3

Recovery Resources Availability

Each project $p \in P$ requires a known amount of recovery resources per time period denoted R_p. The total amount of recovery resources available at time period $t \in T$ is denoted R_t.

For modeling this constraint, we introduce a binary matrix $[\zeta^t_{s,p}]$ (similar to $[\omega^t_{s,p}]$):

$$\zeta^t_{s,p} \equiv \begin{cases} 1 & \text{if } s \leq t \leq s + D_p - 1 \\ 0 & \text{otherwise} \end{cases} \qquad \forall p \in P, \forall s \in F_p, \forall t \in T$$

Resource Availability Constraint

$$\sum_{p \in P} \sum_{s \in F_p} g_{s,p} \zeta^t_{s,p} R_p \leq R_t \quad \forall t \in T$$

Disaster Recovery Scheduling Problem

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{\sigma \in \Sigma} k_\sigma^t \mathcal{D}(\sigma) & \\
\text{s.t.} \quad & \sum_{s \in F_p} g_{s,p} = 1 & \forall p \in P \\
& \sum_{\sigma \in \Sigma} k_\sigma^t = 1 & \forall t \in T \\
& \sum_{\sigma \in \Sigma} k_\sigma^t \gamma_{\sigma,p} = \sum_{s \in F_p} g_{s,p} \omega_{s,p}^t & \forall p \in P, \forall t \in T \\
& \sum_{p \in P} \sum_{s \in F_p} g_{s,p} \zeta_{s,p}^t R_p \le R_t & \forall t \in T \\
& g_{s,p} \in \{0,1\} & \forall p \in P, s \in F_p \\
& k_\sigma^t \in \{0,1\} & \forall t \in T, \sigma \in \Sigma
\end{aligned}
$$

$\rightarrow$ TAP (lower-level) implicitly represented in the objective

Disaster Recovery Scheduling Problem

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{\sigma\in\Sigma} k_\sigma^t \mathcal{D}(\sigma) & \\
\text{s.t.} \quad & \sum_{s\in F_p} g_{s,p} = 1 & \forall p \in P \\
& \sum_{\sigma\in\Sigma} k_\sigma^t = 1 & \forall t \in T \\
& \sum_{\sigma\in\Sigma} k_\sigma^t \gamma_{\sigma,p} = \sum_{s\in F_p} g_{s,p}\omega_{s,p}^t & \forall p \in P, \forall t \in T \\
& \sum_{p\in P}\sum_{s\in F_p} g_{s,p}\zeta_{s,p}^t R_p \le R_t & \forall t \in T \\
& g_{s,p} \in \{0,1\} & \forall p \in P, s \in F_p \\
& k_\sigma^t \in [0,1] & \forall t \in T, \sigma \in \Sigma
\end{aligned}
$$

$\rightarrow$ Proved that integrality restrictions on k_σ^t can be relaxed

Solution Approach

If the number of projects is "manageable" → all project combinations can be enumerated and the corresponding TAP solved to obtain the associated network delay — **highly parallelizable task**, up to $2^{|P|}$ threads

Then, solve Disaster Recovery Scheduling Problem using Mixed-Integer Linear Programming (MILP) → efficient commercial software available, e.g. CPLEX

Else, heuristic, Branch-and-Price approach proposed by Rey *et al.* (2016)[1] based on a statistical approximation of TSTT values and Column (pattern variable) Generation

[1] Rey, D., Bar-Gera, H., Dixit, V.V., Waller, S.T., 2016. A branch and price algorithm for the bi-level network maintenance scheduling problem, in: 2016 INFORMS Annual Meeting. Nashville, USA. — journal paper under review

Scheduling Heuristics

We benchmark the MILP formulation against classical scheduling heuristics

- **Shortest Processing Time** (SPT):
  sort projects by increasing duration D_p
- **Largest Average First-Order** (LAFO):
  sort projects by decreasing $\bar{\Delta}_p$
- **Approximated Smith's Ratio**:
  sort projects by decreasing $\frac{\bar{\Delta}_p}{D_p}$

Outline

Test Data

- Berlin MPF network: 975 nodes and 2,184 links, available at https://github.com/bstabler/TransportationNetworks
- 10-project instance with 100 links affected per project
- 26 time periods (in weeks), uniformly distributed project duration (maximum duration is half of planning period)
- damaged state: link capacity reduction: 50%, link free-flow travel time augmentation: 20%
- All projects require a unit resource $R_p = 1$
- all numerical results presented in terms of normalized network delay $\rightarrow \widehat{\mathcal{D}}(\sigma) = \frac{\mathcal{D}(\sigma)}{\mathcal{D}(\sigma_0)}$
- All TAP solved with TAPAS (Bar-Gera, 2010)[2]

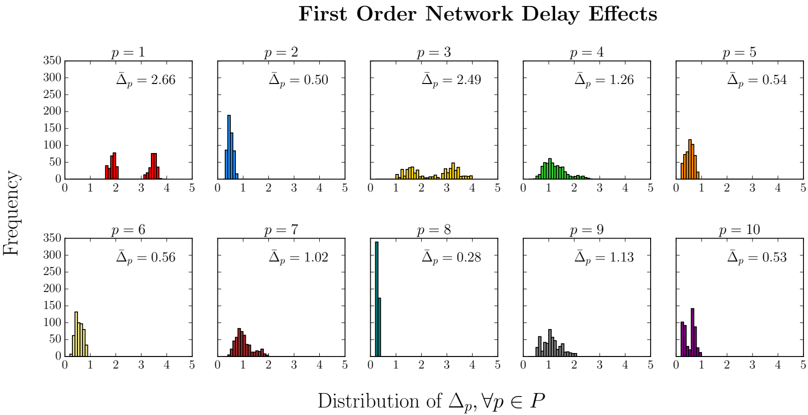
[2] Bar-Gera, H. (2010). Traffic assignment by paired alternative segments. Transportation Research Part B: Methodological, 44(8-9), 1022-1046.

Berlin MPF map and Recovery Projects

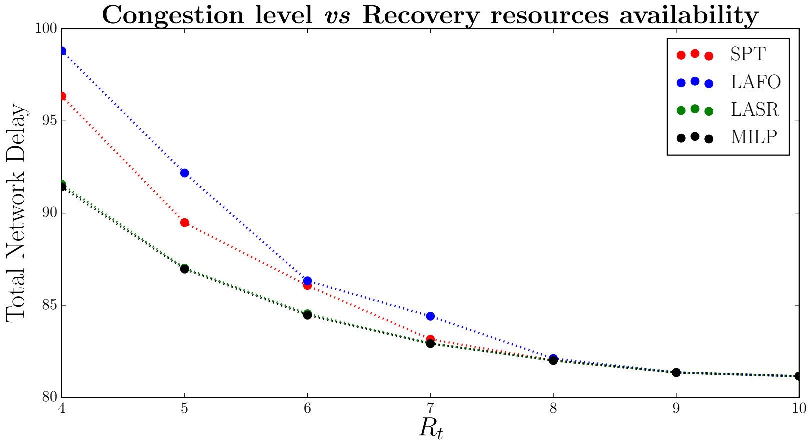
First-order Network Delay Effects

Let φ_p be a binary variable representing project activeness. For each $p \in P$ and for each of the $2^{|P|-1} = 512$ sub-patterns $\tilde{\sigma}$:

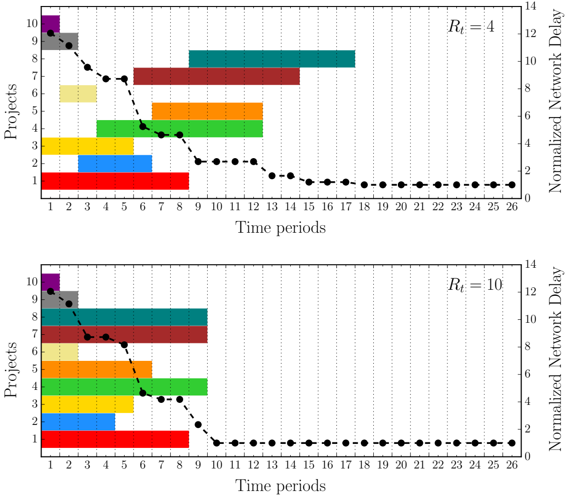
$$\Delta_{p,\tilde{\sigma}} \equiv \mathcal{D}([\varphi_p = 1] \oplus \tilde{\sigma}) - \mathcal{D}([\varphi_p = 0] \oplus \tilde{\sigma})$$



Congestion Level *vs* Recovery Resources



Analysis of MILP solution for $R_t = 4$ and $R_t = 10$



Summary

New mixed integer programming formulation for the disaster recovery scheduling problem

Competitive scheduling heuristics based on first-order network delay effects

Future work focused on improving proposed approach for large number of projects and approximation algorithms

Thank you for your attention