Young Researchers CERMICS Seminar

Solving the Firefighter problem on trees

Adèle Pass-Lanneau

October 10, 2017

Internship Director:

Frédéric Meunier

École des Ponts ParisTech

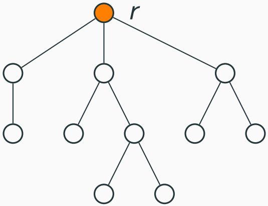
Introduction to Firefighting

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$t = 0$

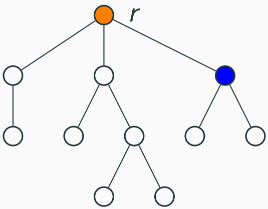
Introduction to Firefighting

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$t = 1$

protection

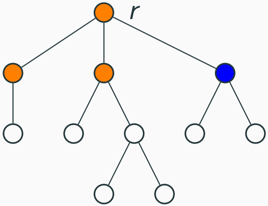
Introduction to Firefighting

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$t = 1$

propagation

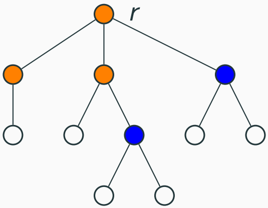
Introduction to Firefighting

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$t = 2$

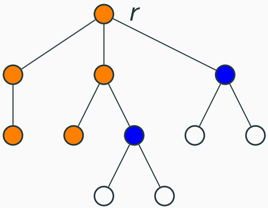
protection

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$t = 2$

propagation

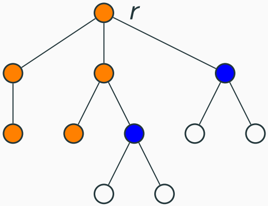
Introduction to Firefighting

The **Firefighter problem** (Hartnell, 1995):

models **propagation of a fire** in a network

(or disease in a population, virus in computer network, etc)

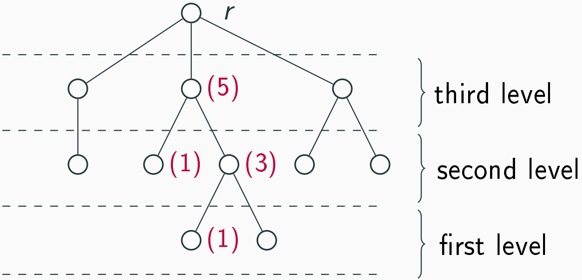
- A graph, for us: a tree T with root r
- A fire starts at r
- At each time step
  - 1 new vertex can be protected
  - The fire spreads
- **Objective:** maximize the number of saved vertices



$\longrightarrow$ 6 vertices saved

An Integer Programming formulation

- *Level*: set of vertices with same distance to the root
- *Weight*: if v protected, w_v vertices saved



An Integer Programming formulation

Decision variables: $x_v = 1$ iff v is protected

$$
\begin{aligned}
\max \quad & \sum_{v \in V \setminus \{r\}} x_v w_v \\
\text{s. t.} \quad & x_u + x_v \leq 1 \qquad \forall\ u, v \text{ on same level} \qquad (1) \\
& x_u + x_v \leq 1 \qquad \forall\ u \text{ ancestor of } v \qquad (2) \\
& x_v \in \{0, 1\}
\end{aligned}
$$

(1): 1 vertex protected per level

(2): a vertex already saved must not be protected

Solving the Firefighter problem

- Firefighter problem is **NP-hard** on trees (Finbow et al., 2009)
- We are interested in **methods for solving it**

Different approaches:

- Polynomial algorithms in special cases
- Methods based on integer programming
- Lagrangian relaxation

Outline

1. Firefighting and Stable Set problems

2. Perfect graphs: a polynomial case for the Firefighter problem

3. Facets of polyhedra: description of the Firefighter polytope

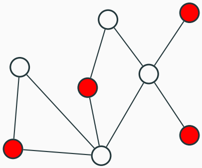
4. Numerical resolution through Lagrangian relaxation

Firefighting and Stable Set problems

Stable Sets

Stable set

Given a graph $G = (V, E)$, a subset of vertices $S \subset V$ is *stable* if $\forall u, v \in S$, u and v are not adjacent, i.e., $(u, v) \notin E$.



- Classical problems: stable set of **maximum size**, of **maximum weight** $\sum_{v \in S} w_v$
- Max Stable Set is **NP-complete**; even **hard in practice** with a thousand vertices.

Applications of Stable Set problems

- **Task scheduling**: given a set of jobs with begin/end date, find the max number of tasks that can be scheduled on a single machine.

Applications of Stable Set problems

- **Pilot-Copilot allocation**: given a set of persons speaking different languages, find maximum number of pairs speaking the same language. 🯅

Firefighter and Stable Set

IP for stable set

$$\begin{aligned} \max \quad & \sum_{v\in V} x_v w_v \\ \text{s. t.} \quad & x_u + x_v \leq 1 \quad \forall\, (u,v) \in E \\ & x_v \in \{0,1\} \end{aligned}$$

IP for Firefighter

$$\begin{aligned} \max \quad & \sum_{v\in V\setminus\{r\}} x_v w_v \\ \text{s. t.} \quad & x_u + x_v \leq 1 \quad \forall\, u,v \text{ on same level} \\ & x_u + x_v \leq 1 \quad \forall\, u \text{ ancestor of } v \\ & x_v \in \{0,1\} \end{aligned}$$

Firefighter and Stable Set

IP for stable set

$$\begin{aligned} \max \quad & \sum_{v \in V} x_v w_v \\ \text{s. t.} \quad & x_u + x_v \leq 1 \quad \forall\, (u,v) \in E \\ & x_v \in \{0,1\} \end{aligned}$$

IP for Firefighter

$$\begin{aligned} \max \quad & \sum_{v \in V \setminus \{r\}} x_v w_v \\ \text{s. t.} \quad & x_u + x_v \leq 1 \quad \forall\, u, v \text{ on same level} \\ & x_u + x_v \leq 1 \quad \forall\, u \text{ ancestor of } v \\ & x_v \in \{0,1\} \end{aligned}$$

Observation

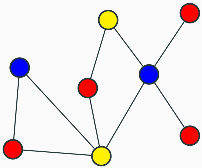
The Firefighter problem is a Max Weight Stable Set problem in an appropriate graph, called ***Firefighter graph***, with weights w.

Perfect (Firefighter) graphs

Introduction to perfect graphs

$G = (V, E)$ is a graph.

- ***Induced subgraph*** $G[X]$ ***of*** G: take a subset of vertices X and all edges of G between them.
- **Clique-number** $\omega(G)$: size of the biggest *clique* in G, i.e., induced subgraph where all edges exist.
- **Chromatic number** $\chi(G)$: minimum number of colors needed in a *valid coloring* of G. In a valid coloring, every vertex has one color and every color form a stable set.



$\omega(G) = 3$
Valid coloring with 3 colors: $\chi(G) \leq 3$
Always $\omega(G) \leq \chi(G)$
Hence $\chi(G) = 3$

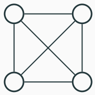
Perfect graphs

Definition (Claude Berge, 1960)

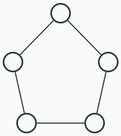
A graph G is *perfect* if for every induced subgraph H of G, the equality $\chi(H) = \omega(H)$ holds.

Are they perfect?

- A clique?



- A cycle of length 5?

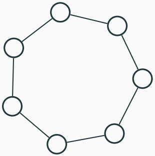


The Strong Perfect Graph Theorem

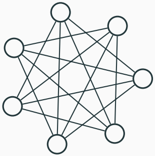
SPGT

A graph is perfect if and only if it contains no induced odd hole nor odd antihole.

- Conjectured by Berge in 60s, remained open for 40 years
- Proven by Chudnovsky, Robertson, Seymour and Thomas in 2002-2006



Hole of length 7



Antihole of length 7

Polyhedral characterization of perfect graphs

Another characterization of perfect graphs is on their **Stable Set polytope** (in next section).

[Grötschel, Lovász, Schrijver, 1988]

The Max Weight Stable Set problem can be solved in polynomial time in perfect graphs.

- Reminder: Firefighter $\iff$ Max Weight Stable Set in Firefighter graphs
- **Perfect Firefighter graphs are a polynomial case**
- But polynomiality is provided by big theoretical result: no specialized algorithm
- **Our main result**: a combinatorial polynomial algorithm in this case.

The Stable Set polytope of Firefighter graphs

Linear Programming vs. Integer Programming

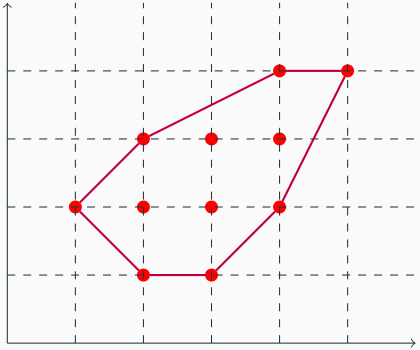
Continous relaxation of the IP: linear program obtained by dropping integrality constraints.

$$\begin{aligned} \max \quad & \sum_{v \in V_G} x_v w_v \\ \text{s. t.} \quad & x_u + x_v \leq 1 \quad \forall\, (u, v) \in E_G \\ & x_v \in \{0, 1\} \end{aligned}$$

- Linear Programming is easy:
  simplex, integer points methods; polynomiality.
- Integer Programming is hard:
  ex. NP-hard for Stable Set problem.

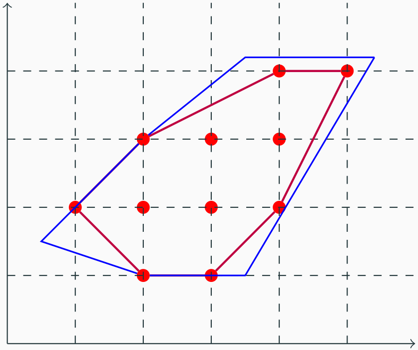
Strengthening a continuous relaxation

- **Objective for solving IPs:** get continuous relaxation as close as possible to convex hull of integer points
- Adding *cuts* given by ***valid inequalities***
- The "strongest" valid inequalities are ***facets***



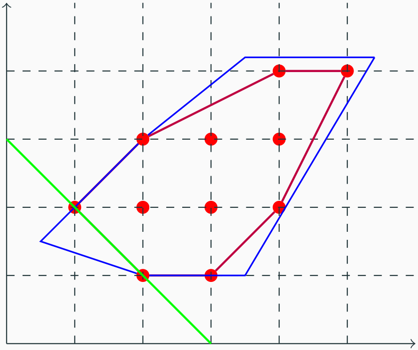
Strengthening a continuous relaxation

- **Objective for solving IPs:** get continuous relaxation as close as possible to convex hull of integer points
- Adding *cuts* given by ***valid inequalities***
- The "strongest" valid inequalities are ***facets***



Strengthening a continuous relaxation

- **Objective for solving IPs:** get continuous relaxation as close as possible to convex hull of integer points
- Adding *cuts* given by ***valid inequalities***
- The "strongest" valid inequalities are ***facets***



Describing the Stable Set polytope

- **Stable set polytope**:

$$STAB(G) = conv\{x \in \{0,1\}^{|V|} \mid x_u + x_v \leq 1 \ \forall (u,v) \in E\}$$

- Polytope of continuous relaxation

$$P = \{x \in \mathbb{R}_+^{|V|} \mid x_u + x_v \leq 1 \ \forall (u,v) \in E\}$$

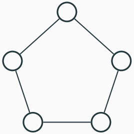
  In most cases, $STAB(G) \subsetneq P$

- **Finding facets of the Stable Set polytope** is a major problem in combinatorial optimization
- Useful in practice to design efficient algorithms
- Results in graph classes, e.g., line-graphs (Edmonds), claw-free graphs, etc.

Valid inequalities for Stable Set

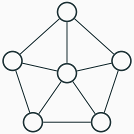
Many known valid inequalities can be added, such as:

Odd cycle inequality:



$$\sum_{v \in \text{cycle}} x_v \leq$$

Odd wheel inequality:

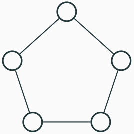


$$\sum_{v \in \text{cycle}} x_v + \qquad \leq$$

Valid inequalities for Stable Set

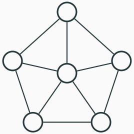
Many known valid inequalities can be added, such as:

Odd cycle inequality:



$$\sum_{v \in \text{cycle}} x_v \leq 2$$

Odd wheel inequality:

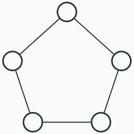


$$\sum_{v \in \text{cycle}} x_v + \qquad \leq 2$$

Valid inequalities for Stable Set

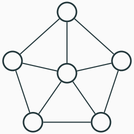
Many known valid inequalities can be added, such as:

Odd cycle inequality:



$$\sum_{v \in \text{ cycle}} x_v \leq 2$$

Odd wheel inequality:



$$\sum_{v \in \text{ cycle}} x_v + 2x_{\text{hub}} \leq 2$$

Facets of the Firefighter polytope

We studied **the Firefighter polytope**, i.e., $STAB(G)$ where G is a Firefighter graph.

Our main results:

- New facets with "handmade" proofs
- Generic methods to compute facets
- Further results on characterizing all facets of $STAB(G)$

A Lagrangian-based exact method

Why Lagrangian relaxation?

- IP formulation is **too large**: quadratic number of constraints
  even continuous relaxation is unpracticable
- We are interested in **large instances**
  (epidemiology, computer network, etc.)
- **Approximated solutions (with guarantee)** in minutes
  can be better than optimum in hours

Back to the canonical IP

Remember the canonical IP:

$$\begin{aligned} \max \quad & \sum_{v \in V \setminus \{r\}} x_v w_v \\ \text{s. t.} \quad & x(L) \leq 1 && \forall \text{ level } L \\ & x_u + x_v \leq 1 && \forall u \text{ ancestor of } v \\ & x_v \in \{0, 1\} \end{aligned}$$

- Without level constraints, the problem is easy!
- Solvable through **Dynamic Programming**.

Recap on Lagrangian relaxation

Introduce Lagrange multipliers λ associated with level constraints

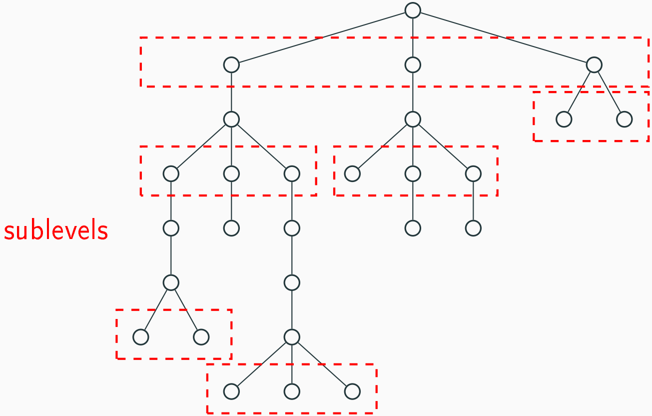
$$
\begin{aligned}
OPT &= \max\{w^T x \mid x \in X,\ x(L) \leq 1\ \forall L\} \\
&= \max_{x \in X} \inf_{\lambda \geq 0} \left(w^T x + \sum_L \lambda_L \cdot (x(L) - 1) \right) \\
&\leq \inf_{\lambda \geq 0} \underbrace{\max_{x \in X} \left(w^T x + \sum_L \lambda_L \cdot (x(L) - 1) \right)}_{\text{dual function } \mathcal{G}(\lambda)}
\end{aligned}
$$

- For every λ, $\mathcal{G}(\lambda)$ is an **upper bound**
- $\mathcal{G}(\lambda)$ is **easy to compute** (solving relaxation with DP)
- Infimum of $\mathcal{G}$ is computed by **gradient descent algorithm**
- ... but bad bound if all levels dualized

An example of stronger relaxation 1/2

Relaxation (R^1): remove level constraints, add *sublevel* constraints:

$$\forall v, \quad \sum_{u \text{ child of } v} x_u \leq 1$$



An example of stronger relaxation 1/2

Relaxation (R^1) is still easy to solve

Let $\mathcal{M}$ be the optimum of (R^1) in the subtree rooted at v, but with $x_v = 0$.

Solving (R) through dynamic programming

Then $OPT = \mathcal{M}(r)$ and for every v,

$$\mathcal{M}(v) = \max \left\{ \underbrace{\sum_{z \in Ch(v)} \mathcal{M}(z)}_{\text{if no child of v is selected}} \quad ; \quad \underbrace{\max_{u \in Ch(v)} \left\{ w(u) + \sum_{\substack{z \in Ch(v) \\ z \neq u}} \mathcal{M}(z) \right\}}_{\text{if the child z is selected}} \right\}$$

An algorithm for the Firefighter problem

- Relaxation (R^1) can be generalized into **a family of relaxations** (R^p) such that:
  - Quality of bound increases with p
  - Computation time increases with p
- By **tuning parameter** p, we found good trade-off between quality of bound and computation time.
- Other features are added:
  - Greedy initialization heuristic
  - Repair lagrangian heuristic
  - Pruning technique to eliminate useless vertices

Numerical results

- Lagrangian method finds optimum and proves optimality (LB = UB)
- It outperforms linear programming
- Heuristics are good; the difficult part is to **certify optimality**: lagrangian relaxation is appropriate

Instance		Greedy	Cplex		Lagrangian method	
n	id	solution	solution	time	solution	time
100	1	79	81*	<1s	81*	<1s
1000	1	828	837*	51s	837*	3s
5000	1	4136	4228*	27m 04s	4228*	25s
10000	1	8309	8495*	4h 26m 11s	8495*	1m 07s

Thank you for your attention!