Enhancing Finite Set Constraint Programming with Probe Backtrack Search

Stefano Gualandi, Luis Quesada

Universite' Catholique de Louvain, Belgium

Introduction

- **Motivations**
  - to use **Finite Set constraint programming** for modeling ***combinatorial optimization problems*** because is a very expressive formal language
  - to improve the efficiency of Finite Set constraint programming by using a **Probe Backtrack Search**
- Finite Set constraint language signature:

$$F_{FS} = \{\emptyset, U, \cap, \cup, /, \#)$$

$$C_{FS} = \{\equiv, \supset, \supseteq, \subset, \subseteq, \in, \notin\}$$

  - The elements of the set U are the natural numbers
  - The cardinality (i.e., #) symbols establishes a relation between FS and FD constraint programming

Probe backtrack search

- Probe backtrack search (PBT) [Sakkout&Wallace99] is a general search method involving two phases: *probe* and *search*
- A **probe** is a solution of a **relaxation** of the original problem. The prober is an algorithm that solves efficiently the relaxation. E.g., a relaxation could defined by considering a **subset of constraints**. The choice of the relaxation is extremely important and it does influence the performance of PBT
- Using PBT:
  - If the relaxation does not admit a solution, the main problem does not admit a solution either (**pruning**)
  - If a relaxation does admit a solution, its cost is a lower or upper bound of the main problem (**bound**)
  - Otherwise, the probe is used to guide the search (**heuristic**)

FS with Probe backtrack search

- The main idea for enhancing Finite Set Constraint Programming with Probe Backtrack Search is to define **relaxations** of Finite Set-based models using the **cardinalities** of the Finite Set variables

Example: Network routing

- Network topology formalized with graph $G=(N, E)$, where each link $e \in E$ has a limited capacity c_e and a propagation delay d_e
- Set of demands K, where each $k \in K$ has:

  s_k : source node

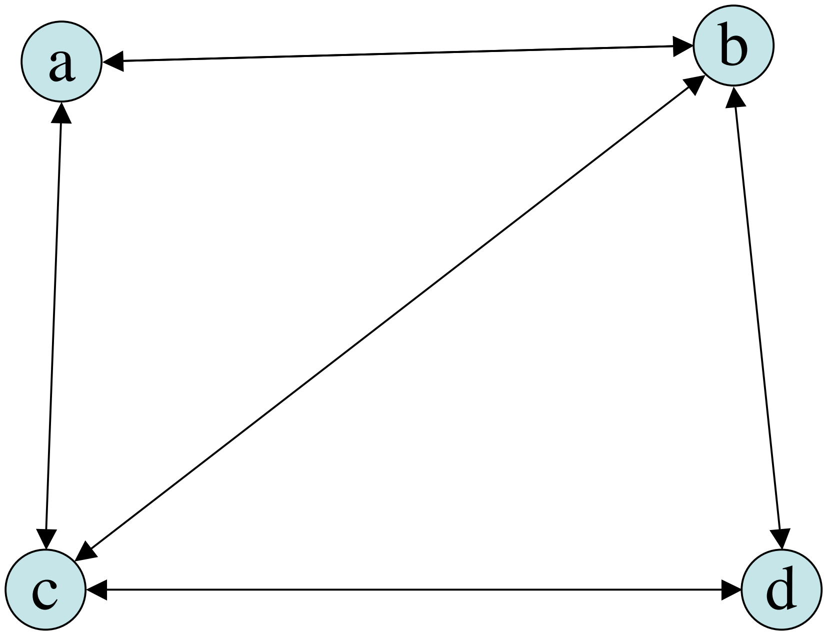
  t_k : destination node

  q_k : bandwidth

  d_k : maximum propagation delay
- Set of demands RK that must be routed
- Set of routed demands S
- The total placed bandwidth is maximized:

$$max \sum_{k \in S} q_k$$

Example: Network Routing



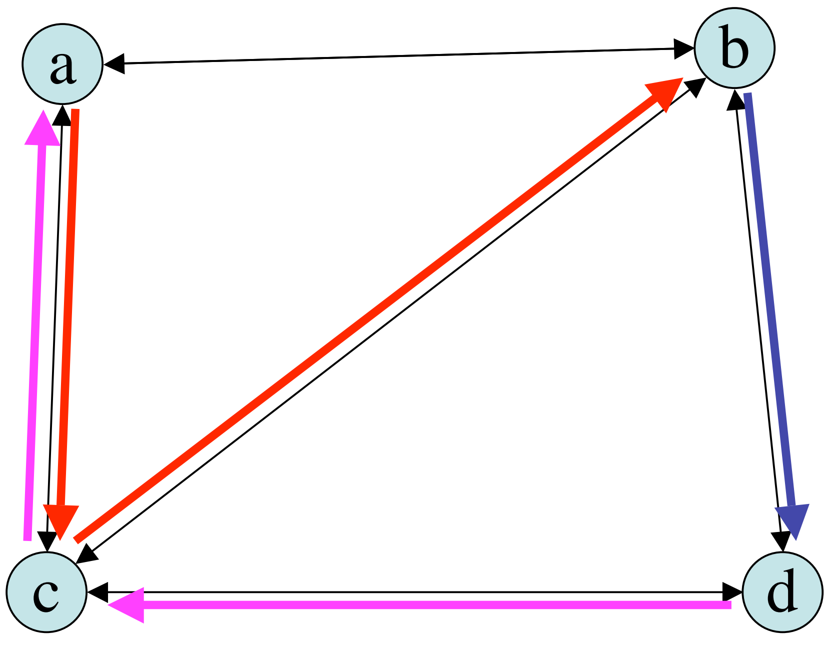
k	s_k	t_k	q_k	d_k
1	**a**	**b**	**2**	**5**
2	**a**	**d**	**3**	**5**
3	**b**	**d**	**1**	**5**
4	**c**	**a**	**1**	**2**
5	**d**	**a**	**2**	**3**

s_k = source; t_k = destination
q_k = bandwidth; d_k = max delay

Edge capacities: c_{ab}=1, c_{ac}=2, c_{cb}=3, c_{bd}=2, c_{cd}=2

Edge delaies: d_{ab}=2, d_{ac}=1, d_{cb}=1, d_{bd}=2, d_{cd}=2

Example: Network Routing



K	s_k	t_k	q_k	d_k
1	**a**	**b**	**2**	**5**
2	a	d	3	5
3	**b**	**d**	**1**	**5**
4	c	a	1	2
5	**d**	**a**	**2**	**3**

s_k = source; t_k = destination
q_k = bandwidth; d_k = max delay

Edge capacities: $c_{ab}=1$, $c_{ac}=2$, $c_{cb}=3$, $c_{bd}=2$, $c_{cd}=2$

Edge delaies: $d_{ab}=2$, $d_{ac}=1$, $d_{cb}=1$, $d_{bd}=2$, $d_{cd}=2$

Comparing FD & FS models

- The Finite Domain-based model has a Boolean FD variable x_{ek} per each demand k per each edge e:

$$\forall e \in E. \forall k \in K. \; x_{ek} = 0 \Leftrightarrow e \text{ not in } Path_k$$
$$x_{ek} = 1 \Leftrightarrow e \text{ in } Path_k$$

- The Finite Set-based model has a Finite Set variable S_e per each edge e:

$$\forall e \in E. \; k \in S_e \Leftrightarrow e \in P_k$$

(i.e., S_e is the set of demands traversing the edge e)

Finite Set-based model

S, S_e, P_k are Finite Set variables
RK, K, E, and M_i are sets
S_i^{in}, S_j^{out} incoming-outgoing edge sets
Kt_n set of demands from node n
Ks_n set of demands to node n

$$max \sum_{k \in S} q_k$$

(i) $RK \subseteq S \subseteq K$

(ii) $\forall e \in E.\ S_e \subseteq S$

(iii) $\forall k \in K.\ P_k \subseteq E$

(iv) $\forall n \in N.\ partition([S_1^{in} \ldots S_v^{in}], M_1) \wedge$
$partition([S_1^{out} \ldots S_u^{out}], M_2) \wedge$
$M_1 \setminus Ks_n \equiv M_2 \setminus Kt_n$

(v) $\forall k \in S.\ \forall e \in E.(k \in S_e \Leftrightarrow e \in P_k)$

(vi) $\forall e \in E.\ \sum_{k \in Se} q_k \leq c_e$

(vii) $\forall k \in S.\ \sum_{e \in Pk} d_e \leq d_k$

Finite Set-based model

$max\ \sum_{k \in S} q_k$

(i) $RK \subseteq S \subseteq K$

(ii) $\forall e \in E.\ S_e \subseteq S$

(iii) $\forall k \in K.\ P_k \subseteq E$

(iv) $\forall n \in N.\ partition([S_1^{in} \ldots S_v^{in}], M_1) \wedge$
$partition([S_1^{out} \ldots S_u^{out}], M_2) \wedge$
$M_1 \setminus Ks_n \equiv M_2 \setminus Kt_n$

(v) $\forall k \in S.\ \forall e \in E.(k \in S_e \Leftrightarrow e \in P_k)$

(vi) $\forall e \in E.\ \sum_{k \in Se} q_k \leq c_e$

(vii) $\forall k \in S.\ \sum_{e \in Pk} d_e \leq d_k$

S, S_e, P_k are Finite Set variables
RK, K, E, and M_i are sets
S_i^{in}, S_j^{out} incoming-outgoing edge sets
Kt_n set of demands from node n
Ks_n set of demands to node n

PBT + Finite Set

- In order to use Probe Backtrack Search with Finite Set, the relaxation we choose is the maximal flow over the cardinalities of the set variables S_e associated with the edges:

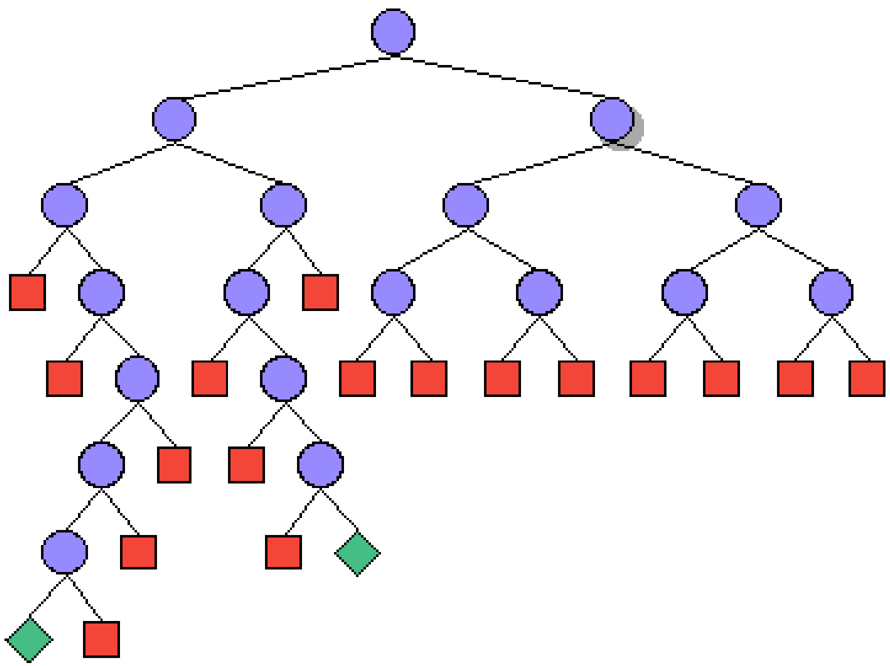
$$\max \sum_{e \in outEdge(s)} q_{max}\, x_e$$
$$\forall n.\ \sum_{e' \in out(n)} x_{e'} - \sum_{e'' \in in(n)} x_{e''} = b_n$$
$$\forall e.\ 0 \leq q_{max}\, x_e \leq c_e$$

- **$x_e = \#S_e$ the cardinality of the set**

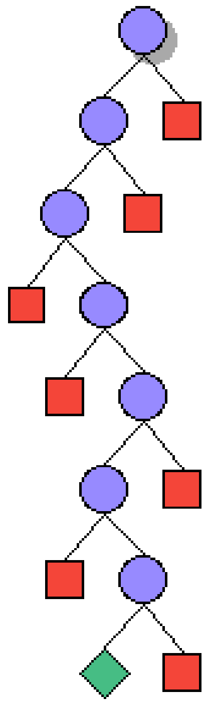
- b_n is the difference between the flow produced and consumed over node n; q_{max} is the maximal capacity of the demands being not part of the solution so far

Qualitative comparison

Search tree obtained by using only Finite Set Constraint Programming



Search tree obtained by using Finite Set + PBT



Ongoing work

- To evaluate the impact on real life problems [Kamarainen04] with a topology:
  - 38 router (38 nodes)
  - 86 bi-directional links (172 edges)
  - 1406 demands
  - 141 mandatory demands
- To consider different relaxations

Questions

Related work

- Probe backtrack search:
  - [Sakkout&Wallace99] introduction of probe backtrack search for minimal perturbation in dynamic scheduling
  - [Liatsos&all03] solve the network routing with FD+PBT using a single shortest path relaxation
  - [Kamarainen04] uses a local search prober
- For improving Finite Set:
  - [Lagoon&Stuckey04] propose a new representation for set variables which leads to new propagators
  - [Gervet01] propose new n-ary propagators

Choosing a relaxation

- The relaxation we choose is the maximal flow over the set variables S_e associated with the edges:

$$max \sum_{e \in outEdge(s)} q_{max} * x_e$$

$$\sum_{e' \in out(n)} x_{e'} - \sum_{e'' \in in(n)} x_{e''} = 0 \quad \forall n \in N \setminus \{s,t\}$$

$$0 \leq q_{max} * x_e - q_{max} * \#glb(S_e) \leq C_e - Cu_e \qquad \forall e \in E$$

$x_e = \#S_e$ the cardinality of the set

q_{max} is the maximal capacity of the demands being not part of the solution so far

Cu_e is the used capacity of the edge