Centre for
Data Analytics

Insight

Three generalizations of the FOCUS constraint

Nina Narodytska[1], Thierry Petit[2], Mohamed Siala[3], Toby Walsh[4]

September 1, 2017

[1] VMware Research, USA
[2] School of Business, Worcester Polytechnic Institute, Worcester, USA
[3] University College Cork, Ireland
[4] TU Berlin, UNSW, and Data61

- Three generalizations of the Focus constraint, IJCAI’13, August 2013, Beijing, China
- Three generalizations of the Focus constraint, CoRR abs/1304.5970 (2013)
- **Three Generalizations of the Focus Constraint, Constraints. October 2016**

FOCUS Constraint

Focus Constraint

- The concept of concentrating the high values in a sequence of variables to a small number of intervals

Focus Constraint

- The concept of concentrating the high values in a sequence of variables to a small number of intervals
- Useful for instance in cumulative scheduling problems where some excess of capacity can be tolerated to obtain a solution

Focus Constraint

- The concept of concentrating the high values in a sequence of variables to a small number of intervals
- Useful for instance in cumulative scheduling problems where some excess of capacity can be tolerated to obtain a solution
- In practice, excess might be tolerated by renting a new machine to produce more resource

FOCUS Constraint

Focus Constraint

$[x_0, .., x_6]$ 3 4 3 1 3 4 2

- Resource capacity (k): 2

Focus Constraint

$[x_0, .., x_6]$ 3 4 3 1 3 4 2

- Resource capacity (k): 2
- Number of subsequences with excess (y_c): 2
- Length of each subsequence bounded (len): 3

FOCUS Constraint

- $X = [x_0, x_1, \ldots, x_{n-1}]$: a sequence of integer variables
- $s_{i,j} = [i, i+1, \ldots, j]$, $0 \leq i \leq j < n$.
- len, k: integers
- y_c: integer variable

FOCUS Constraint

- $X = [x_0, x_1, \ldots, x_{n-1}]$: a sequence of integer variables
- $s_{i,j} = [i, i+1, \ldots, j]$, $0 \leq i \leq j < n$.
- len, k: integers
- y_c: integer variable

Definition

FOCUS(X, y_c, len, k) iff there exists a set S_X of *disjoint* sequences of indices $s_{i,j}$ such that three conditions are all satisfied:

1. $|S_X| \leq y_c$
2. $\forall x_l \in X$, $x_l > k \Leftrightarrow \exists s_{i,j} \in S_X$ such that $l \in s_{i,j}$
3. $\forall s_{i,j} \in S_X$, $j - i + 1 \leq len$

FOCUS Constraint

$[x_0, .., x_6]$ 3   4   3   1   3   4   2

- Resource capacity (k): 2
- Number of subsequences with excess (y_c): 2
- Length of each subsequence bounded (len): 3
- FOCUS($X, [2, 2], 3, 2$) is satisfied

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

$[x_0, .., x_9]$: 4 2 4 2 2 0 0 0 0 0

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

$[x_0, .., x_9]$: 4 2 4 2 2 0 0 0 0 0

- A new activity to schedule that requires 5 days and consumes 2 units each day

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

$[x_0, .., x_9]$:	4	2	4	2	2	0	0	0	0	0

- A new activity to schedule that requires 5 days and consumes 2 units each day

$[x_0, .., x_9]$	4	4	6	4	4	2	0	0	0	0
$[x_0, .., x_9]$:	4	2	4	2	2	2	2	2	2	2

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

$[x_0, .., x_9]$:	4	2	4	2	2	0	0	0	0	0

- A new activity to schedule that requires 5 days and consumes 2 units each day

$[x_0, .., x_9]$	4	4	6	4	4	2	0	0	0	0
$[x_0, .., x_9]$:	4	2	4	2	2	2	2	2	2	2

- WEIGHTEDFOCUS

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day *i*

$[x_0, .., x_9]$:	4	2	4	2	2	0	0	0	0	0

- A new activity to schedule that requires 5 days and consumes 2 units each day

$[x_0, .., x_9]$	4	4	6	4	4	2	0	0	0	0
$[x_0, .., x_9]$:	4	2	4	2	2	2	2	2	2	2

- WEIGHTEDFOCUS
- SPRINGYFOCUS

FOCUS is great, but..

- Resource R with capacity k=3
- $[x_0, .., x_9]$ to model the amount of consumed capacity at day i

$[x_0, .., x_9]$:	4	2	4	2	2	0	0	0	0	0

- A new activity to schedule that requires 5 days and consumes 2 units each day

$[x_0, .., x_9]$	4	4	6	4	4	2	0	0	0	0
$[x_0, .., x_9]$:	4	2	4	2	2	2	2	2	2	2

- WEIGHTEDFOCUS
- SPRINGYFOCUS
- WEIGHTEDSPRINGYFOCUS

SpringyFocus

Definition

SpringyFocus(X, y_c, len, h, k) iff there exists a set S_X of *disjoint* sequences of indices $s_{i,j}$ such that four conditions are all satisfied:

1. $|S_X| \leq y_c$
2. $\forall x_l \in X, x_l > k \Rightarrow \exists s_{i,j} \in S_X$ such that $l \in s_{i,j}$
3. $\forall s_{i,j} \in S_X, j - i + 1 \leq len, x_i > k$ and $x_j > k$.
4. $\forall s_{i,j} \in S_X, |\{l \in s_{i,j}, x_l \leq k\}| \leq h$

WeightedFocus

Definition

WeightedFocus(X, y_c, len, k, z_c) iff there exists a set S_X of *disjoint* sequences of indices $s_{i,j}$ such that four conditions are all satisfied:

1. $|S_X| \leq y_c$
2. $\forall x_l \in X, x_l > k \Leftrightarrow \exists s_{i,j} \in S_X$ such that $l \in s_{i,j}$
3. $\forall s_{i,j} \in S_X, j - i + 1 \leq len$
4. $\sum_{s_{i,j} \in S_X} |s_{i,j}| \leq z_c$.

WeightedSpringyFocus

Definition

WeightedSpringyFocus(X, y_c, len,h, k, z_c) iff there exists a set S_X of *disjoint* sequences of indices $s_{i,j}$ such that five conditions are all satisfied:

1. $|S_X| \leq y_c$
2. $\forall x_l \in X, x_l > k \Rightarrow \exists s_{i,j} \in S_X$ such that $l \in s_{i,j}$
3. $\forall s_{i,j} \in S_X, |\{l \in s_{i,j}, x_l \leq k\}| \leq h$
4. $\forall s_{i,j} \in S_X, j - i + 1 \leq len, x_i > k$ and $x_j > k$.
5. $\sum_{s_{i,j} \in S_X} |s_{i,j}| \leq z_c$.

Contributions

Contributions

- BC algorithm for SpringyFocus $O(n)$ time

Contributions

- BC algorithm for SpringyFocus $O(n)$ time
- BC algorithm for WeightedFocus in $O(n \max(z_c))$ time

Contributions

- BC algorithm for SPRINGYFOCUS $O(n)$ time
- BC algorithm for WEIGHTEDFOCUS in $O(n \max(z_c))$ time
- BC algorithm for WEIGHTEDSPRINGYFOCUS in $O(n \max(z_c))$ time

Contributions

- BC algorithm for SpringyFocus $O(n)$ time
- BC algorithm for WeightedFocus in $O(n \max(z_c))$ time
- BC algorithm for WeightedSpringyFocus in $O(n \max(z_c))$ time
- But also:
  - ILP formulation for SpringyFocus ($O(n^2)$ time):

Contributions

- BC algorithm for SpringyFocus $O(n)$ time
- BC algorithm for WeightedFocus in $O(n \max(z_c))$ time
- BC algorithm for WeightedSpringyFocus in $O(n \max(z_c))$ time
- But also:
  - ILP formulation for SpringyFocus ($O(n^2)$ time):
  - Decomposition for WeightedFocus based on Focus

Contributions

- BC algorithm for SpringyFocus $O(n)$ time
- BC algorithm for WeightedFocus in $O(n \max(z_c))$ time
- BC algorithm for WeightedSpringyFocus in $O(n \max(z_c))$ time
- But also:
  - ILP formulation for SpringyFocus ($O(n^2)$ time):
  - Decomposition for WeightedFocus based on Focus
  - Decomposition of WeightedSpringyFocus based on GCC

Filtering for WeightedFocus

Filtering for WEIGHTEDFOCUS

Finding a (bound) support

- Based on a Dynamic program

Filtering for WeightedFocus

Finding a (bound) support

- Based on a Dynamic program
- For each prefix of variables $[x_0, x_1, \ldots, x_j]$, we consider the construction of a "solution" $S_{c,j}$ to WeightedFocus restricted to $[x_0, x_1, \ldots, x_j]$ of cost $c + NB_1$.

Filtering for WEIGHTEDFOCUS

Finding a (bound) support

- Based on a Dynamic program
- For each prefix of variables $[x_0, x_1, \ldots, x_j]$, we consider the construction of a "solution" $S_{c,j}$ to WEIGHTEDFOCUS restricted to $[x_0, x_1, \ldots, x_j]$ of cost $c + NB_1$.
- It turns out that from $S_{c,j}$, one can construct $S_{c,j+1}$ and $S_{c+1,j+1}$.

Filtering for WeightedFocus

Filtering for WEIGHTEDFOCUS

- $S_{c,j}$ is a set of subsequences

Filtering for WeightedFocus

- $S_{c,j}$ is a set of subsequences
- $last(S_{c,j})$: the last sequence in $S_{c,j}$

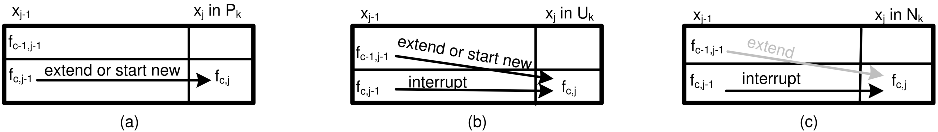
Filtering for WEIGHTEDFOCUS

- $S_{c,j}$ is a set of subsequences
- $last(S_{c,j})$: the last sequence in $S_{c,j}$
- A variable $x_l \in X$ is:
  - Penalizing, (P_k), iff $\min(x_l) > k$.
  - Neutral, (N_k), iff $\max(x_l) \leq k$.
  - Undetermined, (U_k), otherwise.

Filtering for WEIGHTEDFOCUS

- $S_{c,j}$ is a set of subsequences
- $last(S_{c,j})$: the last sequence in $S_{c,j}$
- A variable $x_l \in X$ is:
  - Penalizing, (P_k), iff $\min(x_l) > k$.
  - Neutral, (N_k), iff $\max(x_l) \leq k$.
  - Undetermined, (U_k), otherwise.
- Dynamic programming table f, where $f_{c,j} = \{q_{c,j}, l_{c,j}\}$,
- $q_{c,j} = |S_{c,j}|$
- $l_{c,j} = |last(S_{c,j})|$

Dynamic Programming



Example

c	$D(x_0)$ $[1,1]$	$D(x_1)$ $[0,1]$	$D(x_2)$ $[1,1]$	$D(x_3)$ $[1,1]$	$D(x_4)$ $[0,1]$	$D(x_5)$ $[1,1]$	$D(x_6)$ $[0,1]$	$D(x_7)$ $[1,1]$
0	$\{1,1\}$	$\{1,\infty\}$	$\{2,1\}$	$\{2,2\}$	$\{2,\infty\}$	$\{3,1\}$	$\{3,\infty\}$	$\{4,1\}$
1		$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{1,\infty\}$	$\{2,1\}$	$\{2,\infty\}$	$\{3,1\}$
$z_c^U = 2$					$\{1,5\}$	$\{2,1\}$	$\{2,2\}$	$\{2,3\}$

Bound Consistency

Bound Consistency

- Failure triggering

Bound Consistency

- Failure triggering
- We compute a similar table b on the reverse sequence

Bound Consistency

- Failure triggering
- We compute a similar table b on the reverse sequence
- To check a support for $x_i = v$, $v > k$, for each $f_{c_1,i-1}$ it is sufficient to consider only one element $b_{c_2,n-i-2}$ such that $b_{c_2,n-i-2}$ is non-dummy and c_2 is the maximum value that satisfies inequality $c_1 + c_2 + 1 \leq z_c^U$.

Bound Consistency

- Failure triggering
- We compute a similar table b on the reverse sequence
- To check a support for $x_i = v$, $v > k$, for each $f_{c_1,i-1}$ it is sufficient to consider only one element $b_{c_2,n-i-2}$ such that $b_{c_2,n-i-2}$ is non-dummy and c_2 is the maximum value that satisfies inequality $c_1 + c_2 + 1 \leq z_c^U$.
- To check a support for $x_i = v$, $v \leq k$, for each $f_{c_1,i-1}$ it is sufficient to consider only one element $b_{c_2,n-i-2}$ such that $b_{c_2,n-i-2}$ is non-dummy and c_2 is the maximum value that satisfies inequality $c_1 + c_2 \leq z_c^U$.

Bound Consistency

- Failure triggering
- We compute a similar table b on the reverse sequence
- To check a support for $x_i = v$, $v > k$, for each $f_{c_1,i-1}$ it is sufficient to consider only one element $b_{c_2,n-i-2}$ such that $b_{c_2,n-i-2}$ is non-dummy and c_2 is the maximum value that satisfies inequality $c_1 + c_2 + 1 \leq z_c^U$.
- To check a support for $x_i = v$, $v \leq k$, for each $f_{c_1,i-1}$ it is sufficient to consider only one element $b_{c_2,n-i-2}$ such that $b_{c_2,n-i-2}$ is non-dummy and c_2 is the maximum value that satisfies inequality $c_1 + c_2 \leq z_c^U$.
- Complexity: $O(n \max(z_c))$ time

Empirical Results

Empirical Results

- Purpose: evaluate propagators with decompositions

Empirical Results

- Purpose: evaluate propagators with decompositions
- Only WEIGHTEDFOCUS and WEIGHTEDSPRINGYFOCUS

Empirical Results

- Purpose: evaluate propagators with decompositions
- Only WeightedFocus and WeightedSpringyFocus
- Implemented in choco 2.1.5

Empirical Results

- Purpose: evaluate propagators with decompositions
- Only WeightedFocus and WeightedSpringyFocus
- Implemented in choco 2.1.5
  - Sports league scheduling
  - Cumulative scheduling with rentals
  - Sorting chords

Cumulative Scheduling with Rentals

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$
- Rent a machine between l_i and u_i times within each time interval $[s_i, e_i]$.

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$
- Rent a machine between l_i and u_i times within each time interval $[s_i, e_i]$.
- The cost of the rental period is proportional to its length

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$
- Rent a machine between l_i and u_i times within each time interval $[s_i, e_i]$.
- The cost of the rental period is proportional to its length
- Each time the machine is rented we pay a fixed cost

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$
- Rent a machine between l_i and u_i times within each time interval $[s_i, e_i]$.
- The cost of the rental period is proportional to its length
- Each time the machine is rented we pay a fixed cost
- Minimize simultaneously the number of rental periods and their total length

Cumulative Scheduling with Rentals

- Given a horizon of n days and a set of time intervals $[s_i, e_i]$, $i \in \{1, 2, \ldots, p\}$
- Rent a machine between l_i and u_i times within each time interval $[s_i, e_i]$.
- The cost of the rental period is proportional to its length
- Each time the machine is rented we pay a fixed cost
- Minimize simultaneously the number of rental periods and their total length

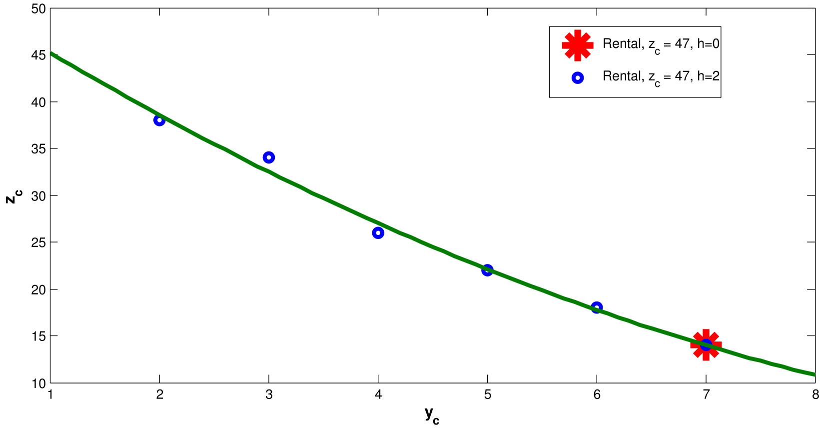
CP Model

- $\forall d \in 1..m,\ l[d] \leq \sum_{i=s[d]}^{i=e[d]} X[i] \leq u[d]$;
- $\textsc{WeightedSpringyFocus}(X, y_c, len, h, 0, z_c)$;

Cumulative Scheduling with Rentals

- Pareto frontier over two cost variables
- First minimizing y_c, then immediately minimize z_c while fixing y_c to its minimum
- Increment y_c by 1 and repeat the same process

Pareto Frontier



Conclusions & Future Research

- Three generalizations of the Focus Constraint
- Complete filtering in polynomial time
- Flexible tools to capture the concept of concentrating costs
- There is a fourth extension of Focus (perhaps later in the session?)

Thank you