5.345 shift

DESCRIPTION LINKS GRAPH

Origin

N. Beldiceanu

Constraint

`shift`(MIN_BREAK, MAX_RANGE, TASKS)

Arguments

MIN_BREAK	:	int
MAX_RANGE	:	int
TASKS	:	collection(origin−dvar, end−dvar)

Restrictions

MIN_BREAK > 0
MAX_RANGE > 0
required(TASKS, [origin, end])
TASKS.origin $<$ TASKS.end

Purpose

The difference between the end of the last task of a *shift* and the origin of the first task of a *shift* should not exceed the quantity MAX_RANGE. Two tasks t_1 and t_2 belong to the *same shift* if at least one of the following conditions is true:

- Task t_2 starts after the end of task t_1 at a distance that is less than or equal to the quantity MIN_BREAK,
- Task t_1 starts after the end of task t_2 at a distance that is less than or equal to the quantity MIN_BREAK.
- Task t_1 overlaps task t_2.

Example

$$\left(6, 8, \left\langle \begin{array}{ll} \texttt{origin}-17 & \texttt{end}-20, \\ \texttt{origin}-7 & \texttt{end}-10, \\ \texttt{origin}-2 & \texttt{end}-4, \\ \texttt{origin}-21 & \texttt{end}-22, \\ \texttt{origin}-5 & \texttt{end}-6 \end{array} \right\rangle \right)$$

Figure 5.680 represents the different tasks of the example. Each task is drawn as a rectangle with its corresponding `id` attribute in the middle. We indicate the distance between two consecutive tasks of a same shift and note that it is less than or equal to MIN_BREAK $= 6$. Since each shift has a range that is less than or equal to MAX_RANGE $= 8$, the `shift` constraint holds (the *range* of a shift is the difference between the end of the last task of the shift and the origin of the first task of the shift).

Typical

MIN_BREAK > 1
MAX_RANGE > 1
MIN_BREAK $<$ MAX_RANGE
$|$TASKS$| > 2$

Symmetries

- Items of TASKS are permutable.
- One and the same constant can be added to the `origin` attribute of all items of TASKS.

`TASKS`

①	`origin` − 17	`end` − 20	④	`origin` − 21	`end` − 22
②	`origin` − 7	`end` − 10	⑤	`origin` − 5	`end` − 6
③	`origin` − 2	`end` − 4			

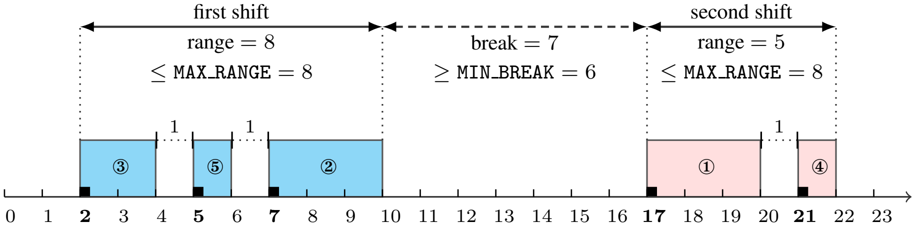

Figure 5.680: The two shifts of the **Example** slot

Usage

The shift constraint can be used in machine scheduling problems where one has to shut down a machine for maintenance purpose after a given maximum utilisation of that machine. In this case the `MAX_RANGE` parameter indicates the maximum possible utilisation of the machine before maintenance, while the `MIN_BREAK` parameter gives the minimum time needed for maintenance.

The shift constraint can also be used for timetabling problems where the rest period of a person can move in time. In this case `MAX_RANGE` indicates the maximum possible working time for a person, while `MIN_BREAK` specifies the minimum length of the break that follows a working time period.

See also

common keyword: `sliding_time_window` (*temporal constraint*).

used in graph description: `range_ctr`.

Keywords

constraint type: scheduling constraint, timetabling constraint, temporal constraint.

Arc input(s) $\mathtt{TASKS}$

Arc generator $SELF \mapsto \mathtt{collection}(\mathtt{tasks})$

Arc arity 1

Arc constraint(s)

- $\mathtt{tasks.end} \geq \mathtt{tasks.origin}$
- $\mathtt{tasks.end} - \mathtt{tasks.origin} \leq \mathtt{MAX\_RANGE}$

Graph property(ies) $\mathbf{NARC} = |\mathtt{TASKS}|$

Arc input(s) $\mathtt{TASKS}$

Arc generator $CLIQUE \mapsto \mathtt{collection}(\mathtt{tasks1}, \mathtt{tasks2})$

Arc arity 2

Arc constraint(s)

$$\bigvee \left(\begin{array}{l} \bigwedge \left(\begin{array}{l} \mathtt{tasks2.origin} \geq \mathtt{tasks1.end}, \\ \mathtt{tasks2.origin} - \mathtt{tasks1.end} \leq \mathtt{MIN\_BREAK} \end{array} \right), \\ \bigwedge \left(\begin{array}{l} \mathtt{tasks1.origin} \geq \mathtt{tasks2.end}, \\ \mathtt{tasks1.origin} - \mathtt{tasks2.end} \leq \mathtt{MIN\_BREAK} \end{array} \right), \\ \bigwedge \left(\begin{array}{l} \mathtt{tasks2.origin} < \mathtt{tasks1.end}, \\ \mathtt{tasks1.origin} < \mathtt{tasks2.end} \end{array} \right) \end{array} \right)$$

Sets

$$\mathsf{CC} \mapsto \left[\begin{array}{l} \mathtt{variables} - \mathtt{col} \left(\begin{array}{l} \mathtt{VARIABLES} - \mathtt{collection}(\mathtt{var} - \mathtt{dvar}), \\ \left[\begin{array}{l} \mathtt{item}(\mathtt{var} - \mathtt{TASKS.origin}), \\ \mathtt{item}(\mathtt{var} - \mathtt{TASKS.end}) \end{array} \right] \end{array} \right) \end{array} \right]$$

Constraint(s) on sets $\mathtt{range\_ctr}(\mathtt{variables}, \leq, \mathtt{MAX\_RANGE})$

Graph model The first graph constraint forces the following two constraints between the attributes of each task:

- The end of a task should not be situated before its start,
- The duration of a task should not be greater than the $\mathtt{MAX\_RANGE}$ parameter.

The second graph constraint decomposes the final graph in connected components where each component corresponds to a given shift. Finally, the **Constraint(s) on sets** slot restricts the stretch of each shift.

Parts (A) and (B) of Figure 5.681 respectively show the initial and final graph associated with the second graph constraint of the **Example** slot. Since we use the set generator CC we show the two connected components of the final graph. They respectively correspond to the two shifts that are displayed in Figure 5.680.

Signature Consider the first graph constraint. Since we use the $SELF$ arc generator on the $\mathtt{TASKS}$ collection the maximum number of arcs of the final graph is equal to $|\mathtt{TASKS}|$. Therefore we can rewrite the graph property $\mathbf{NARC} = |\mathtt{TASKS}|$ to $\mathbf{NARC} \geq |\mathtt{TASKS}|$ and simplify $\underline{\overline{\mathbf{NARC}}}$ to $\overline{\mathbf{NARC}}$.

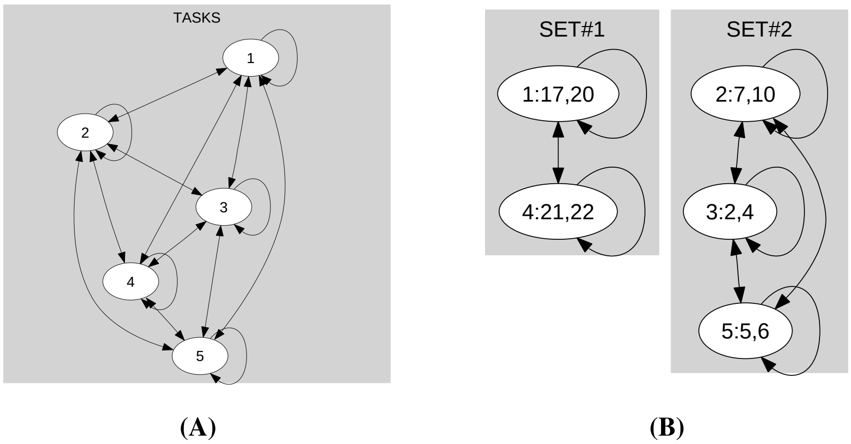

Figure 5.681: Initial and final graph of the `shift` constraint