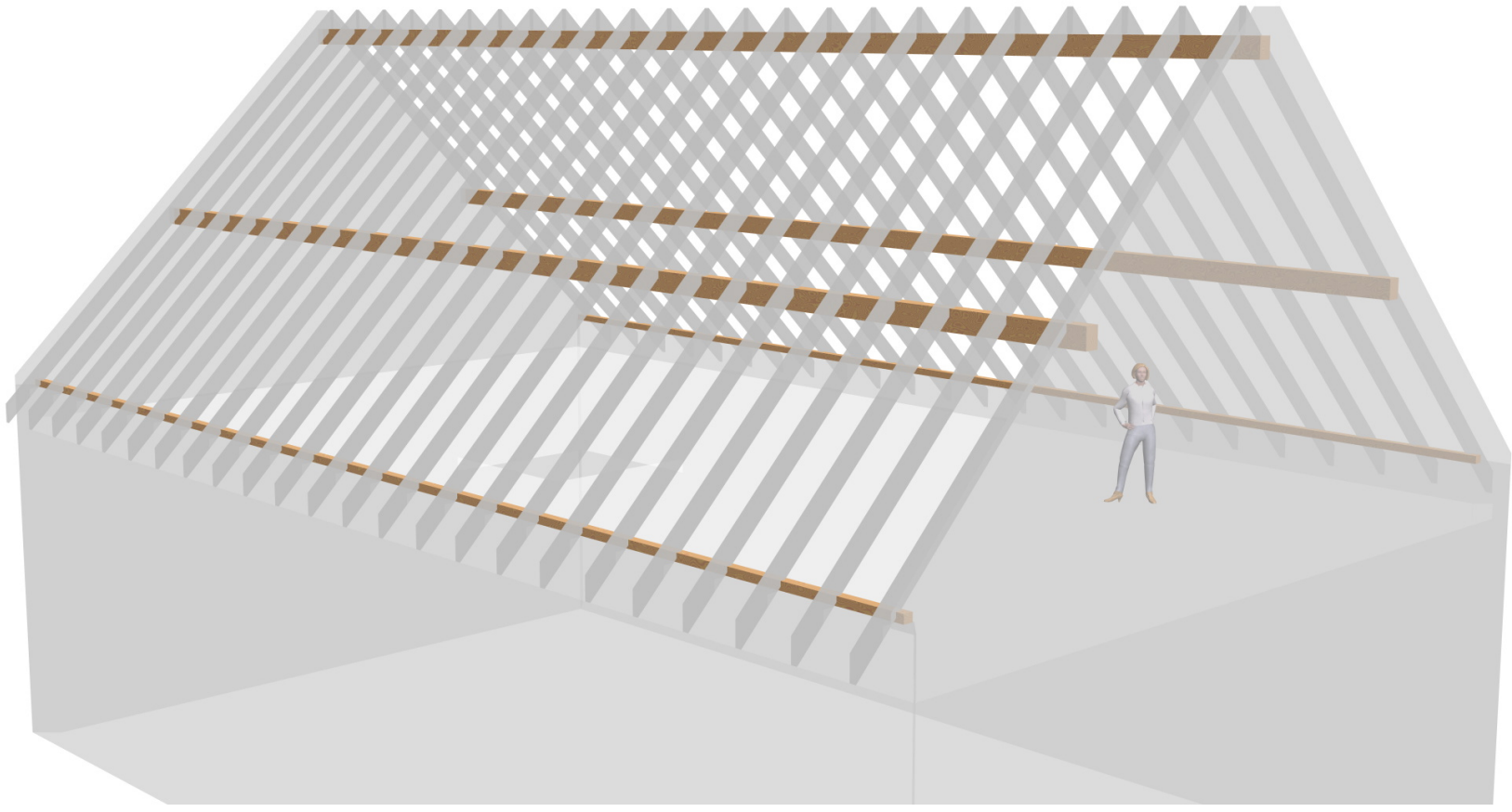
Optimization of a purlin punching process

Jan Pöschko

Institute of Optimization and Discrete Mathematics

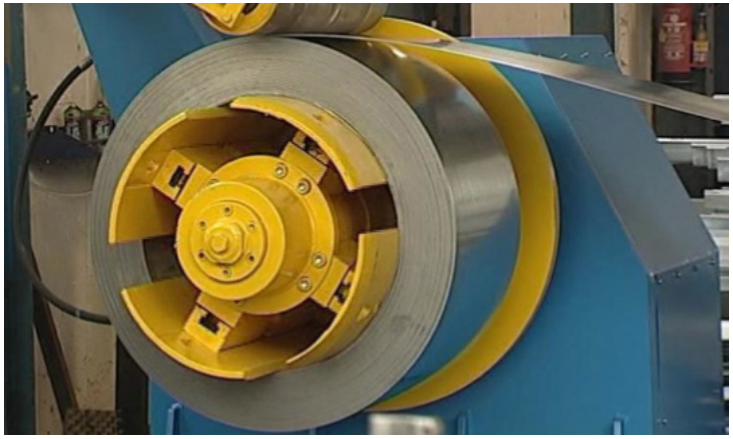
Purlins



(http://de.wikipedia.org/w/index.php?title=Datei:Pfetten_in_Pfettendach.jpg)

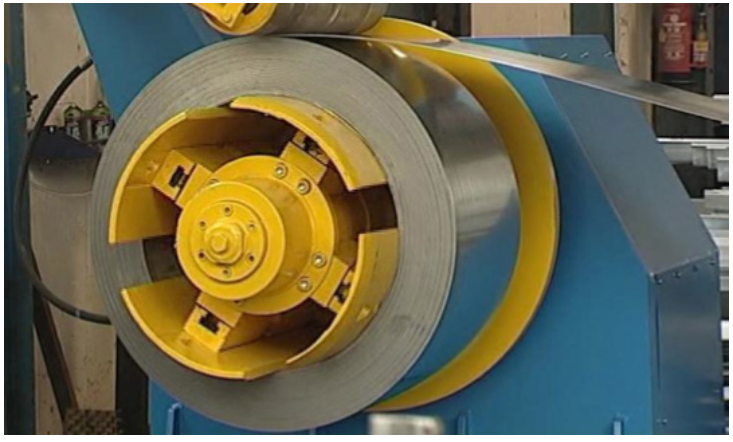
Production

1. sheet of metal runs from coil



Production

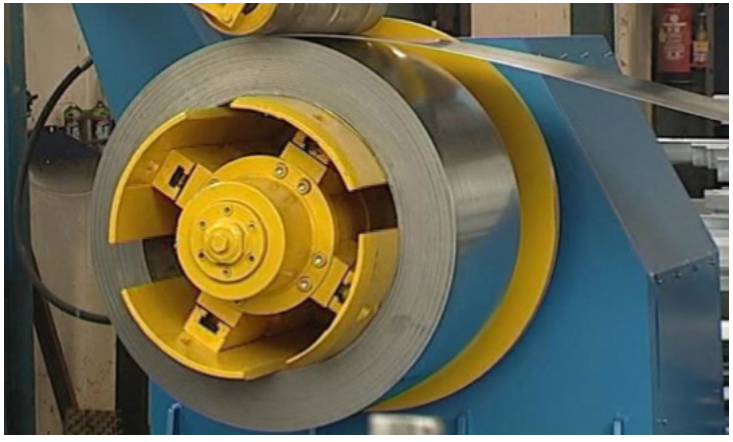
1. sheet of metal runs from coil



2. holes are punched

Production

1. sheet of metal runs from coil

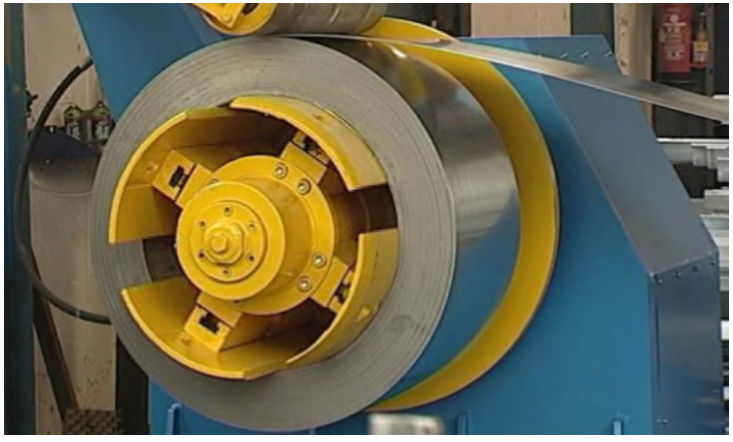


2. holes are punched

3. metal is cut

Production

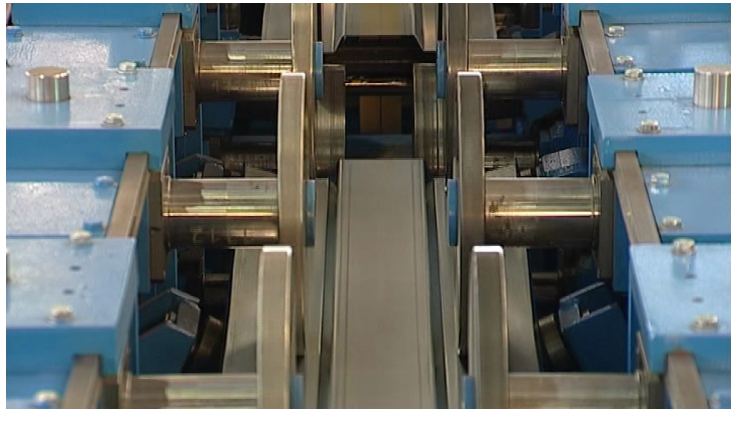
1. sheet of metal runs from coil



2. holes are punched

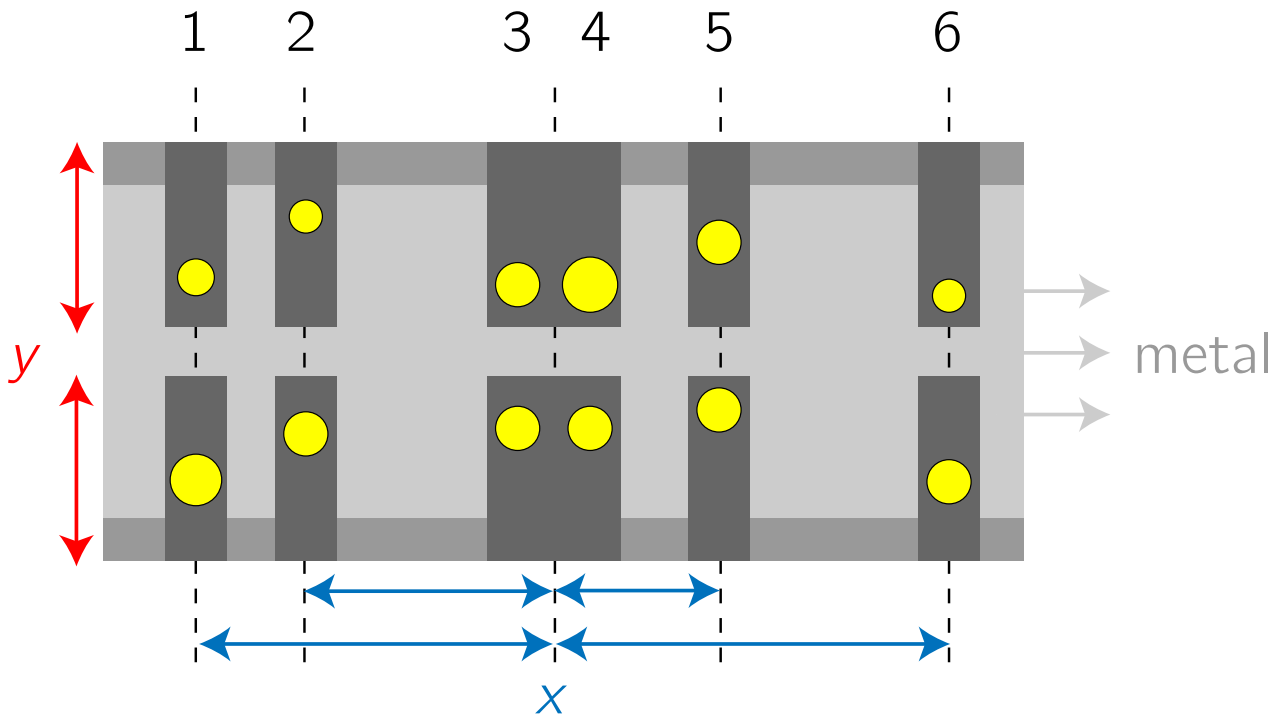
3. metal is cut

4. metal is folded



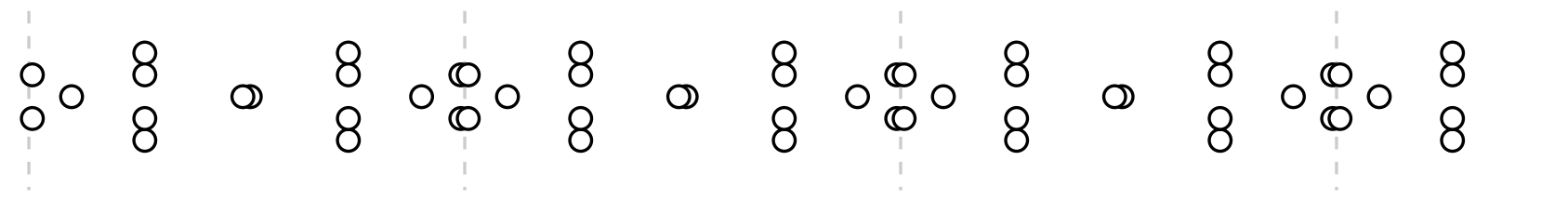
Punching machine

- 6 pairs of stamps
- constant assignment of tools (diameters/shapes) to stamps
- x-positions dynamic with restrictions
- y-positions fully dynamic



Optimization problem

given holes by $x_n \in \mathbb{R}$, $t_n \in T \subset \mathbb{Z}$

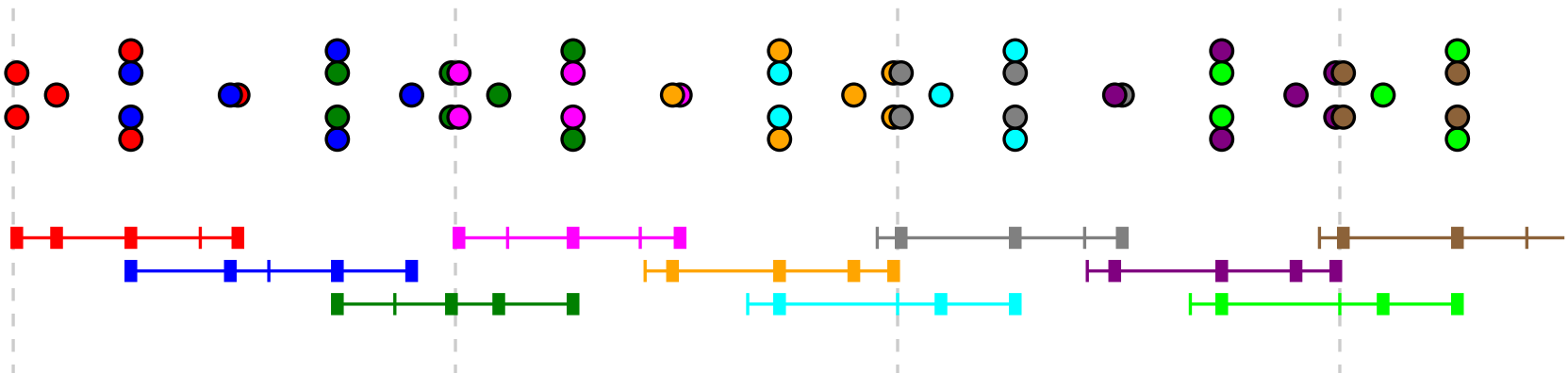


Optimization problem

given holes by $x_n \in \mathbb{R}$, $t_n \in T \subset \mathbb{Z}$
find

- equipping of the machine: $e_j \in T$, $j \in S = \{1, \ldots, 6\}$
- subsequent positions of stamps and punched holes:

$$s_{ij} \in \mathbb{R}, \quad h_{ijn} = \begin{cases} 1 & \text{hole } n \text{ punched by stamp } j \text{ in turn } i \\ 0 & \text{otherwise} \end{cases}$$



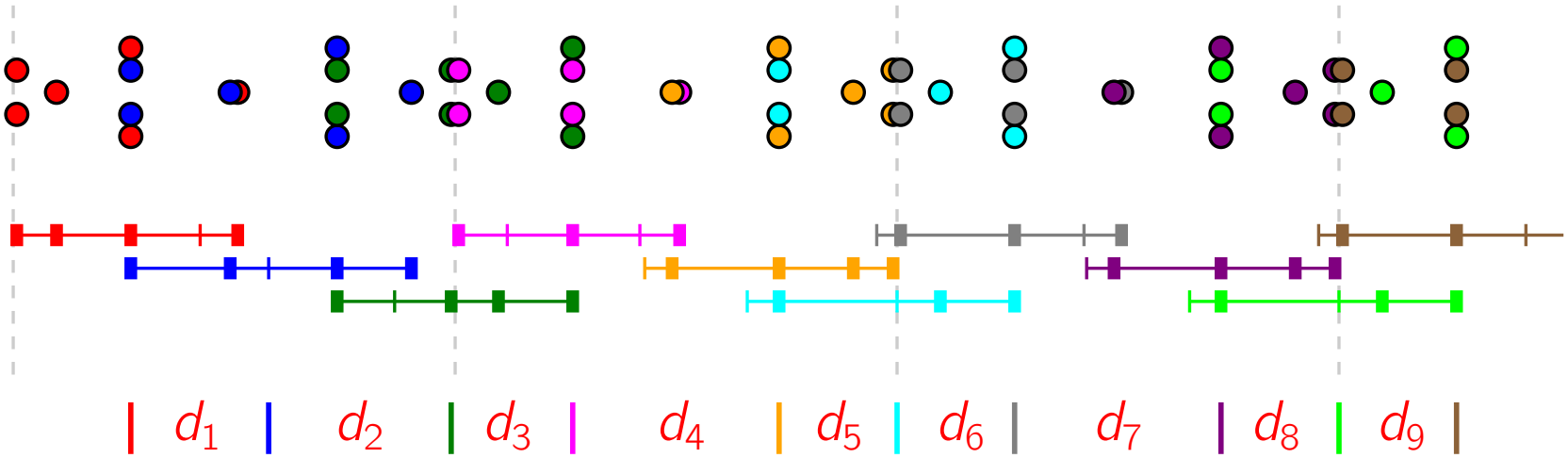
Optimization problem

given holes by $x_n \in \mathbb{R}$, $t_n \in T \subset \mathbb{Z}$
find

- equipping of the machine: $e_j \in T$, $j \in S = \{1, \ldots, 6\}$
- subsequent positions of stamps and punched holes:

$$s_{ij} \in \mathbb{R}, \quad h_{ijn} = \begin{cases} 1 & \text{hole } n \text{ punched by stamp } j \text{ in turn } i \\ 0 & \text{otherwise} \end{cases}$$

maximizing displacements $d_i = s_{i,4} - s_{i-1,4}$



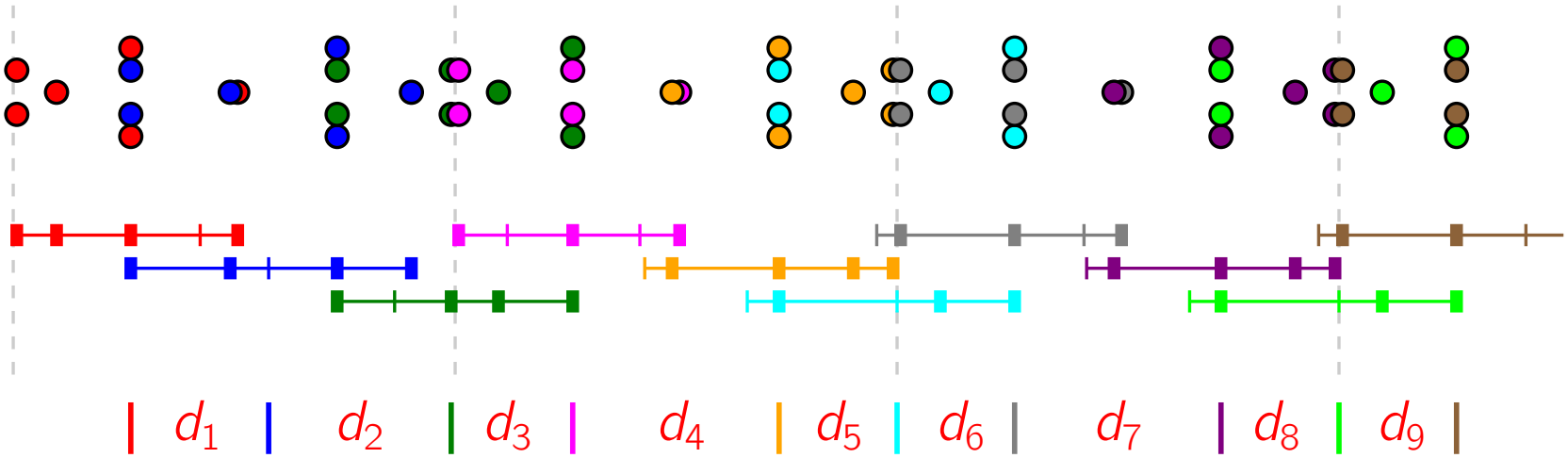
Optimization problem

given holes by $x_n \in \mathbb{R}$, $t_n \in T \subset \mathbb{Z}$
find

- equipping of the machine: $e_j \in T$, $j \in S = \{1, \ldots, 6\}$
- subsequent positions of stamps and punched holes:

$$s_{ij} \in \mathbb{R}, \quad h_{ijn} = \begin{cases} 1 & \text{hole } n \text{ punched by stamp } j \text{ in turn } i \\ 0 & \text{otherwise} \end{cases}$$

maximizing displacements $d_i = s_{i,4} - s_{i-1,4}$: $\min \sum \frac{1}{d_i}$



Program formulation

$$\min \sum \frac{1}{d_i}$$

s. t.

$$\sum_n t_n h_{ijn} = e_j \quad \forall i \; \forall j \in A_i \quad \text{(active stamps have correct tools)}$$

$$\sum_n x_n h_{ijn} = s_{ij} \quad \forall i \; \forall j \in A_i \quad \text{(active stamps are on correct positions)}$$

$$\sum_i \sum_{j \in A_i} h_{ijn} = 1 \quad \forall n \quad \text{(each hole is punched exactly once)}$$

where $A_i = \{j \mid \exists n : h_{ijn} = 1\}$ is the set of active stamps in turn i

Program formulation

$$\min \sum \frac{1}{d_i}$$

s. t.

$$\sum_n t_n h_{ijn} = e_j \quad \forall i \; \forall j \in A_i \quad \text{(active stamps have correct tools)}$$

$$\sum_n x_n h_{ijn} = s_{ij} \quad \forall i \; \forall j \in A_i \quad \text{(active stamps are on correct positions)}$$

$$\sum_i \sum_{j \in A_i} h_{ijn} = 1 \quad \forall n \quad \text{(each hole is punched exactly once)}$$

where $A_i = \{j \mid \exists n : h_{ijn} = 1\}$ is the set of active stamps in turn i and machine constraints

$$c_j \leq \overline{s}_{ij} - \overline{s}_{i,j-1} \leq C_j \quad \text{(distances between stamps are feasible)}$$

$$|\overline{s}_{ij} - \overline{s}_{i-1,j}| \leq C \quad (x\text{-movement is not too large})$$

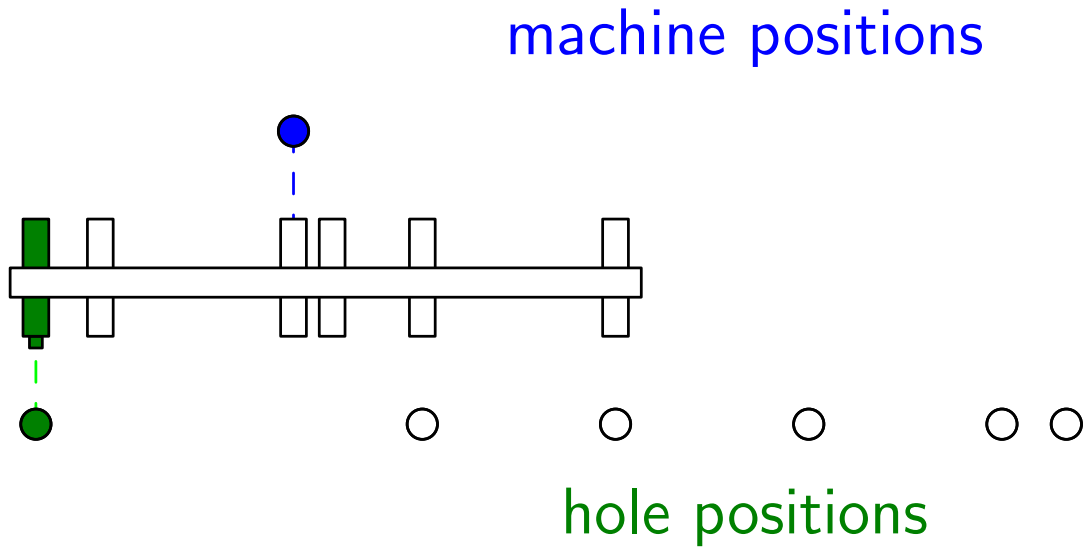
where $\overline{s}_{ij} = s_{ij} - s_{i,4}$ is the relative stamp position

Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\operatorname{argmin}_{P_{ES}} c(P_{ES})$

Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\operatorname{argmin}_{P_{ES}} c(P_{ES})$

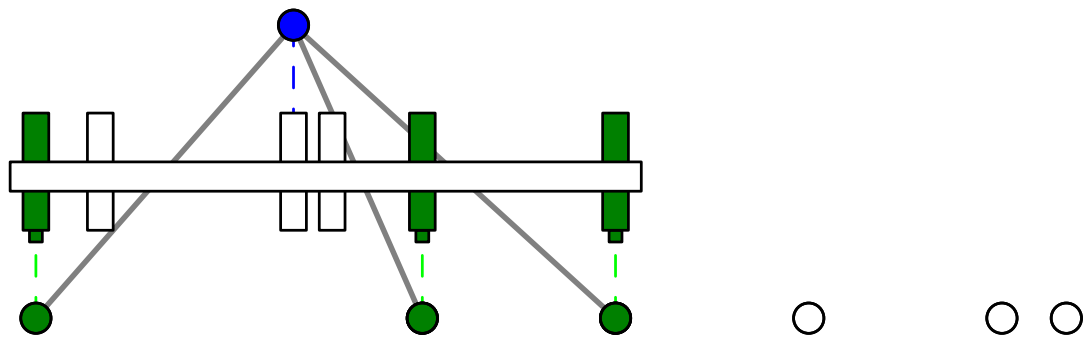


Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\text{argmin}_{P_{ES}}\, c(P_{ES})$

---

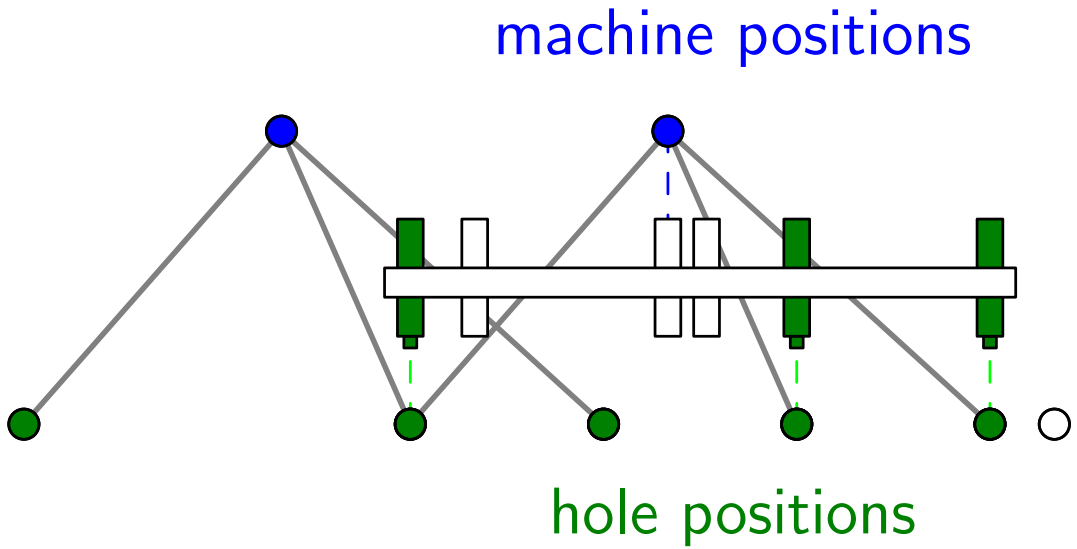
machine positions



hole positions

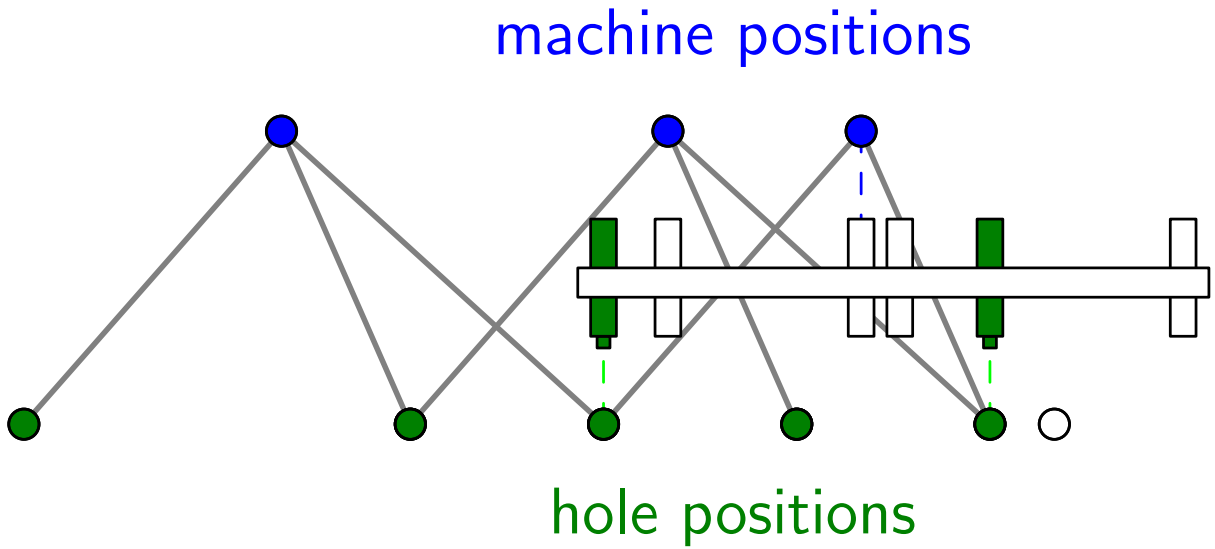
Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\text{argmin}_{P_{ES}}\, c(P_{ES})$



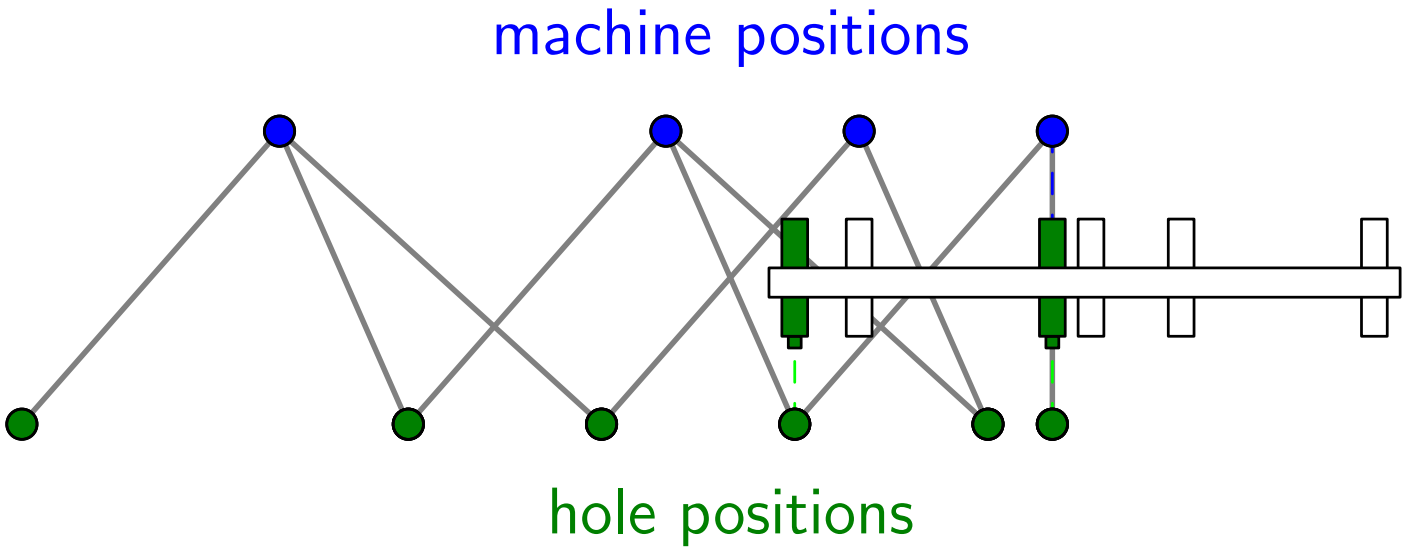
Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\operatorname{argmin}_{P_{ES}} c(P_{ES})$



Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\operatorname{argmin}_{P_{ES}} c(P_{ES})$

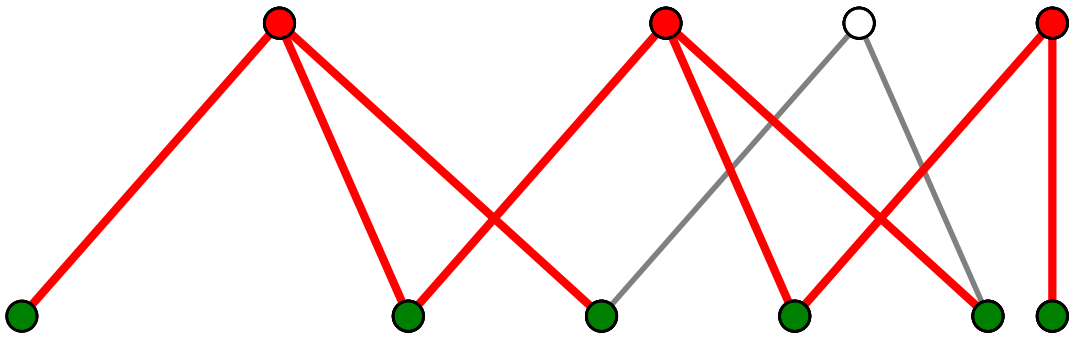


Algorithm: determining a feasible starting solution

for all feasible equippings $E = (e_j) \in T^6$ of the machine **do**
  for random fixed relative stamp positions $S = (\bar{s}_j) \in \mathbb{R}^6$ **do**
    determine a punching plan P_{ES} using a greedy algorithm
  end for
end for
return $\operatorname{argmin}_{P_{ES}} c(P_{ES})$

---

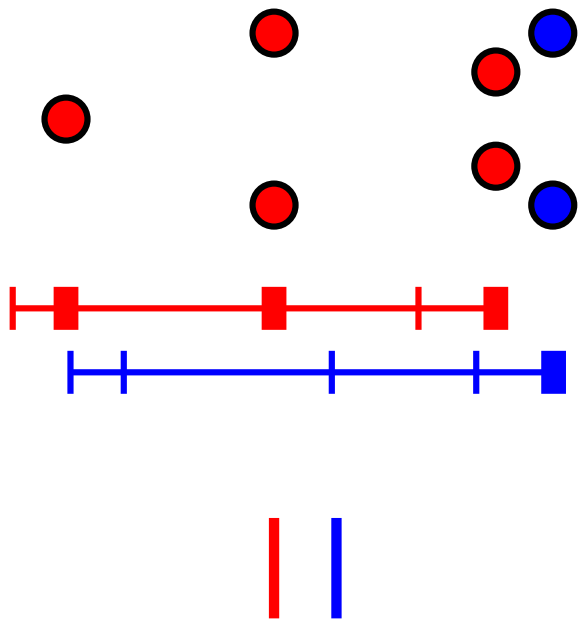
cover of holes by machine positions



hole positions

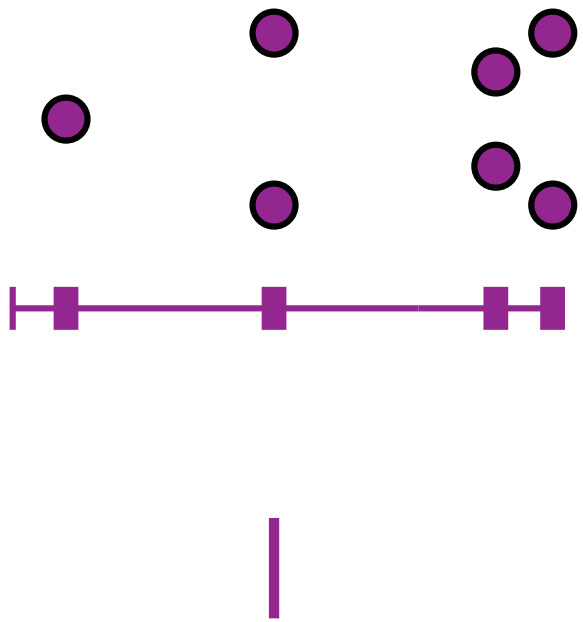
Algorithm: local search for an improved solution

- combine punching steps
- change stamp positions and assignments
- exchange order of punching steps



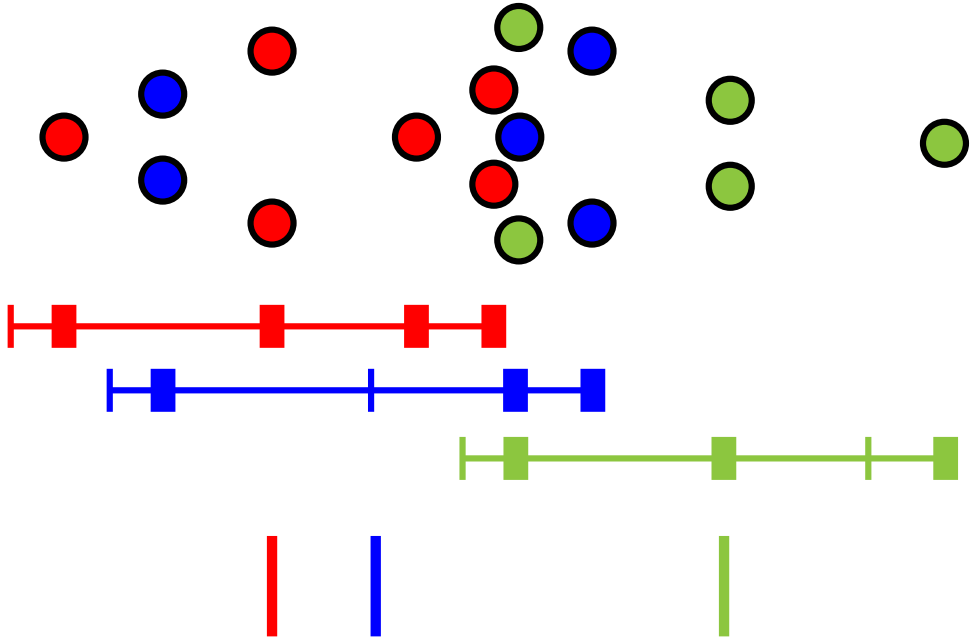
Algorithm: local search for an improved solution

- combine punching steps
- change stamp positions and assignments
- exchange order of punching steps



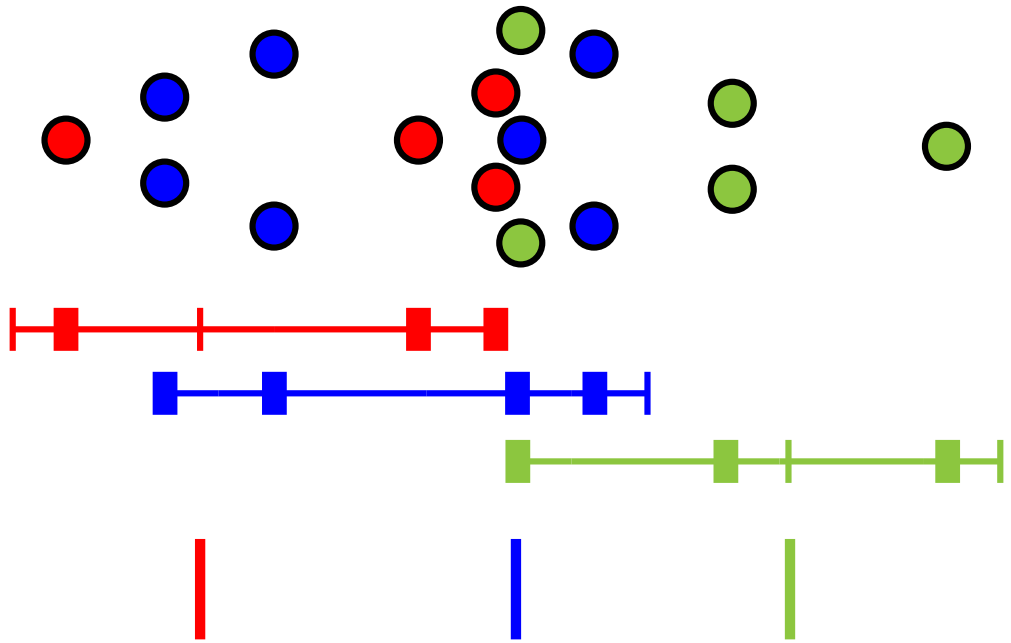
Algorithm: local search for an improved solution

- combine punching steps
- change stamp positions and assignments
- exchange order of punching steps



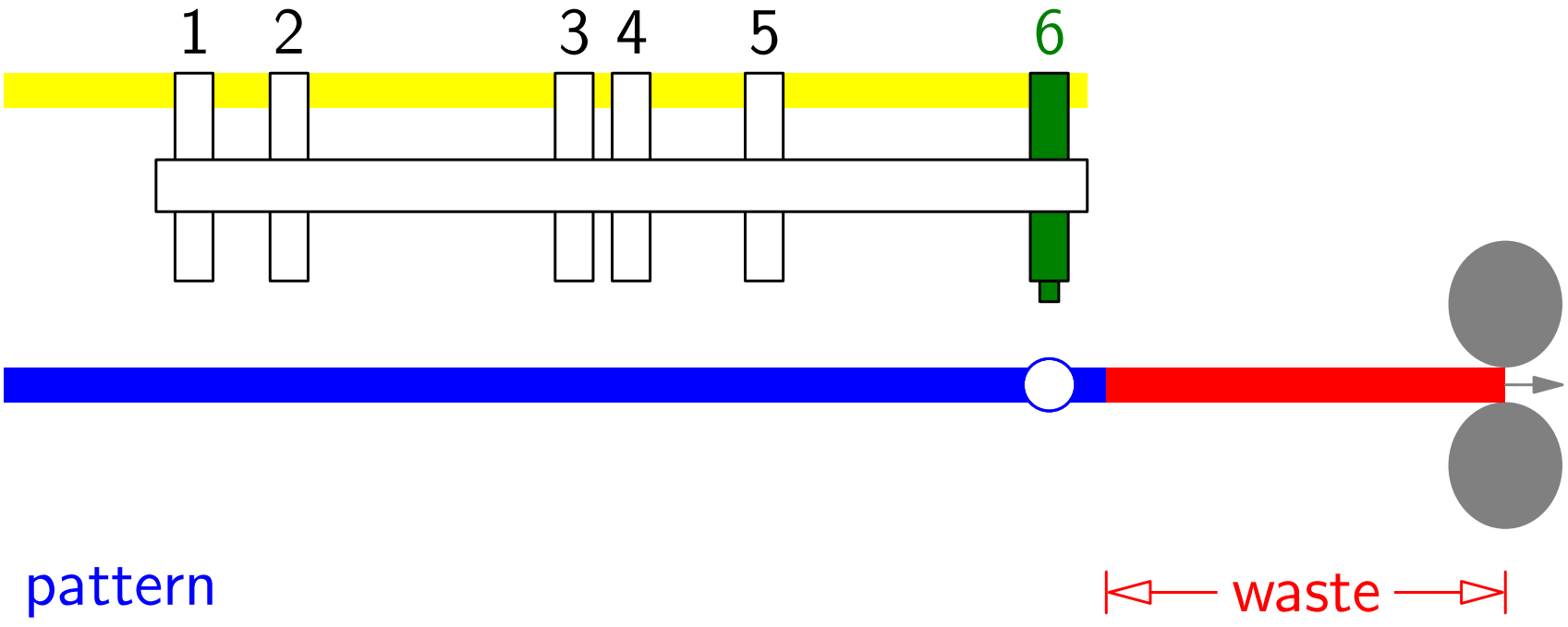
Algorithm: local search for an improved solution

- combine punching steps
- change stamp positions and assignments
- exchange order of punching steps



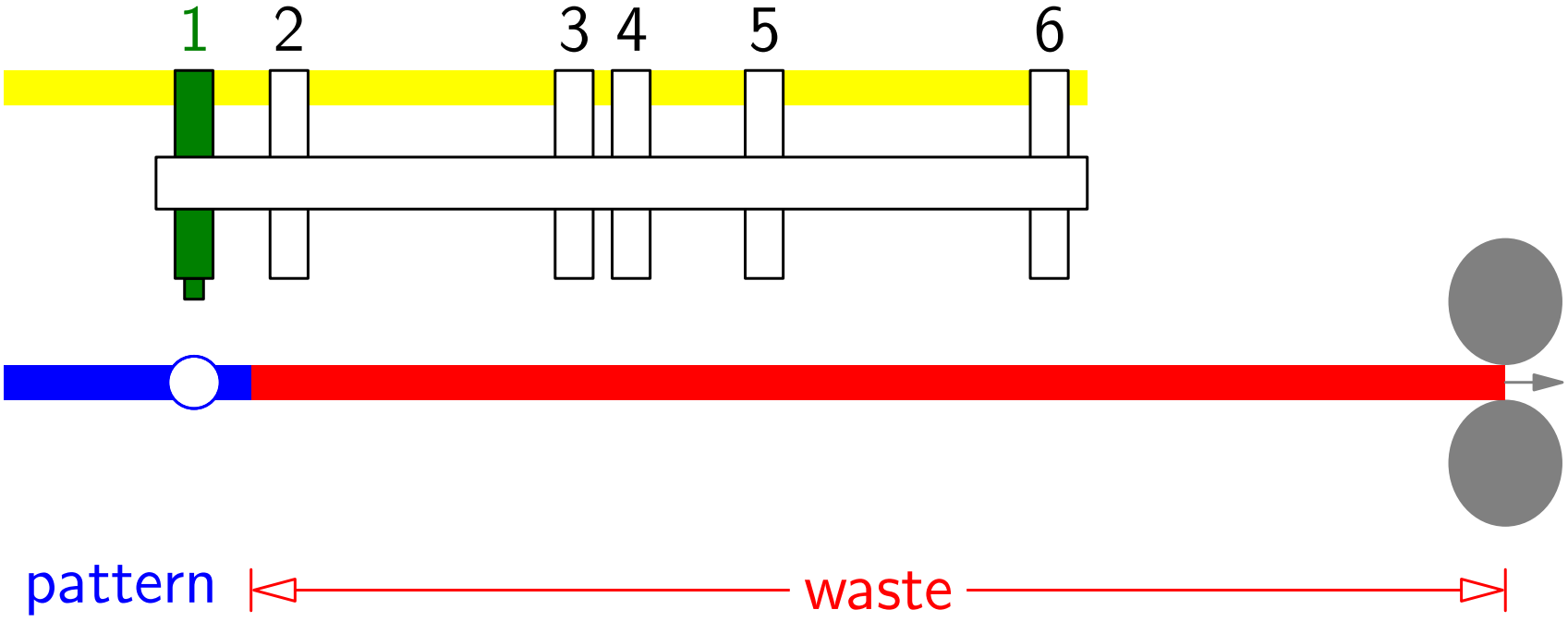
Waste

waste results from used stamp and its position at first step



Waste

waste results from used stamp and its position at first step



Evaluation

- evaluated using 250 given test instances
- solution helps saving both time and material
- in production use in Austria, Poland, Iran, and India

Thank you for your attention!

Jan Pöschko

poeschko@tugraz.at

TU Graz