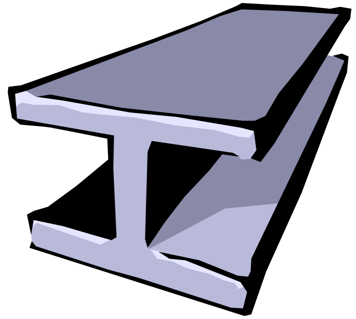
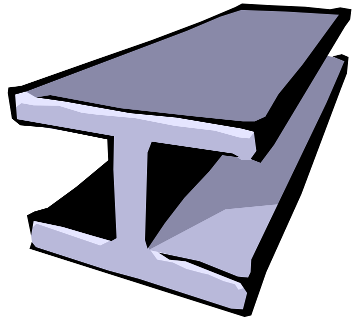


Modelling a Steel Mill Slab Design Problem

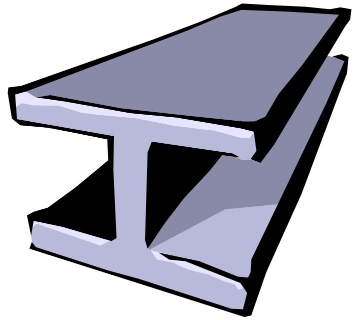
Ian Miguel
Alan M Frisch

Slab Design



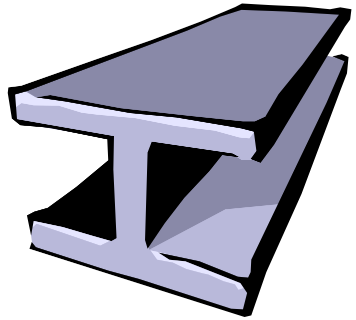
- The mill can make σ different slab sizes.
- Given *d* input orders each sepcifying:
  - A *colour* (route through the mill).
  - A *weight*.
- Pack orders onto slabs such that the total slab capacity is minimised, subject to:
  - Capacity constraints.
  - Colour constraints.

Slab Design Constraints



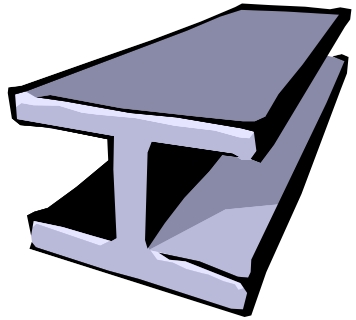
- **Capacity Constraints:**
  - Total weight of orders assigned to each slab cannot exceed the slab's capacity.
- **Colour Constraints:**
  - Each slab can contain at most p of k total colours.
  - Reason: expensive to cut slabs up to send them to different parts of the mill.

What's Interesting about this Problem?



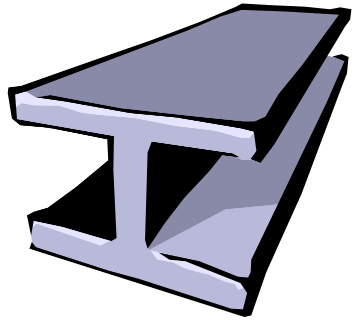
- Many problems exhibit flexibility in portions of their structure.
- Example: the number required of a certain type of variable.
- Flexibility must be resolved during the solution process.
- Slab design is a representative example of this type of problem:
  - Number of slabs is unknown.

What's Interesting about this Problem?

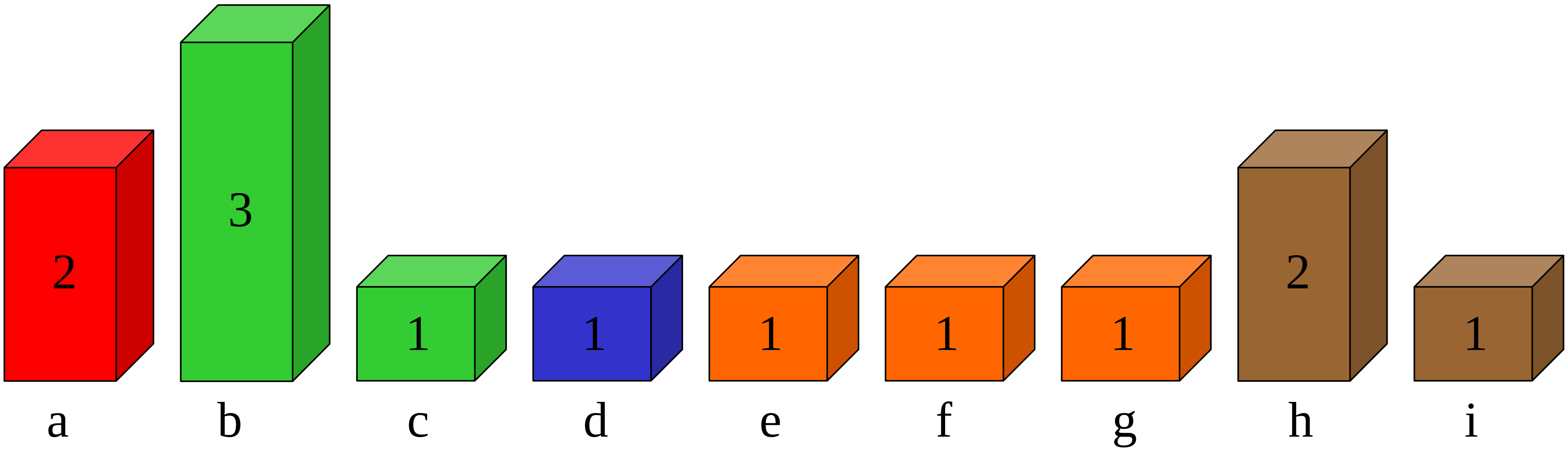


- Symmetry.
  - Some instance-based.
- Channelling.
  - Between matrices of decision variables.
- Implied Constraints.
  - Some of which follow from the symmetry-breaking.

An Example

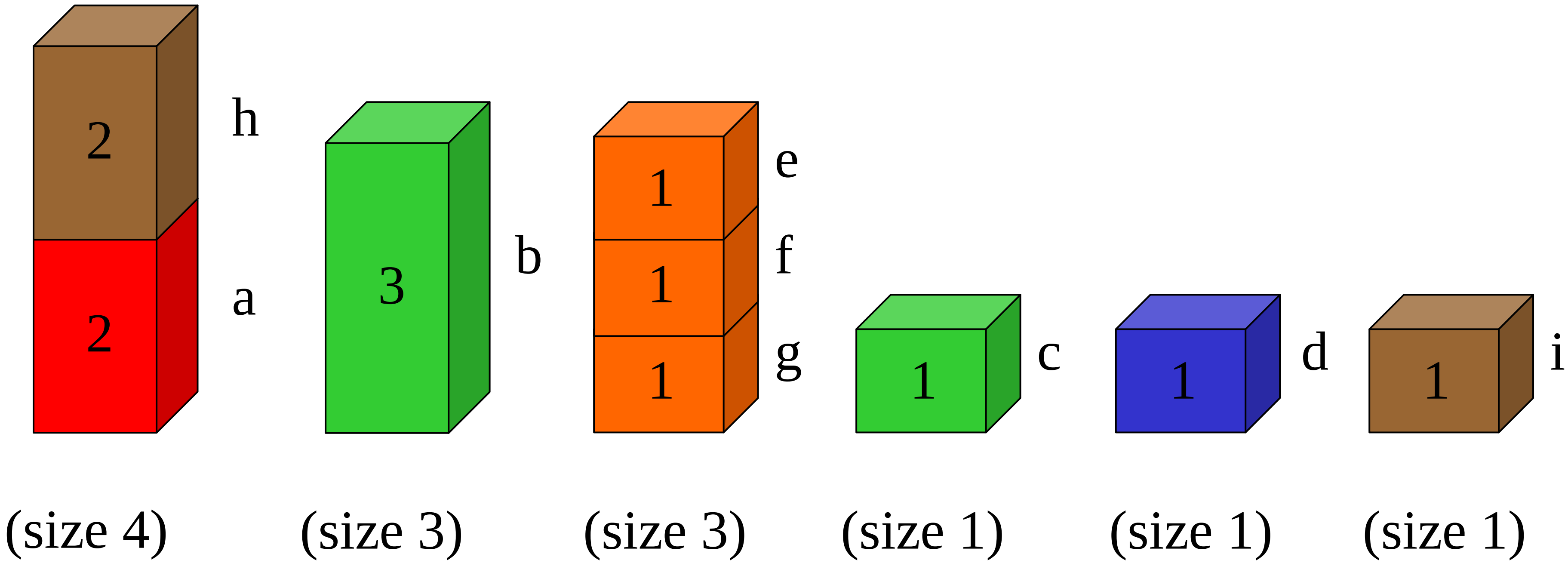


- Slab Sizes: {1, 3, 4} ($\sigma = 3$)
- Orders: {o_a, …, o_i} as shown below ($d = 9$)
- Colours: {red, green, blue, orange, brown} ($k = 5$)
- $p = 2$



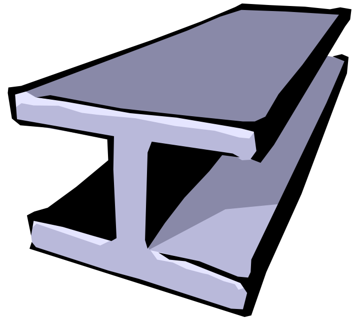
A Solution

- Use 6 Slabs:



This is an optimal solution as it produces no wastage.

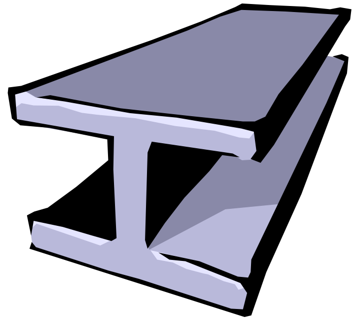
How Many Slabs?



- Number of slabs is not fixed.
  - If there are d orders, then an optimal solution uses no more than d slabs.
- An array, S, of slab variables: $s[1], \ldots, s[d]$
  - Domains size σ.
- Objective

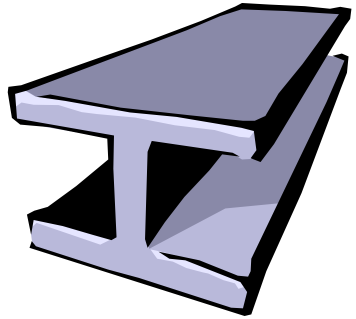
$$\text{minimise} \quad \sum_i s[i] = optVar$$

Slab Variable Redundancy



- Some slab variables may be *redundant*:
  - 0 is added to the domain of each *s[i]*.
  - If slab *i* is not necessary to solve the problem, *s[i]* = 0.

The Order Matrix: O



$O[i,j]$= 1 iff order i is placed on slab j.

	o_a	o_b	o_c	o_d
s_1	0	0	1	1
s_2	0	1	0	0
s_3	1	0	0	0
s_4	0	0	0	0

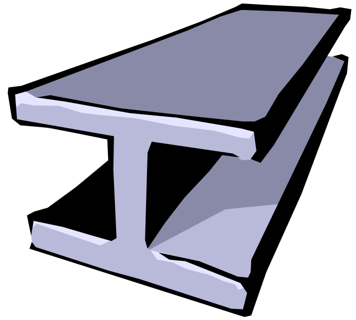
Capacity constraints:

$$\forall j : \sum_i weight(o_i) \times O[i, j] \leq s[j]$$

$$\forall i : \sum_j O[i, j] = 1$$ Every order goes on exactly 1 slab

(Note: is row j column i of the matrix).

The Colour Matrix: C



	Red	Green	Blue	Orange
s_1	0	0	1	1
s_2	0	1	0	0
s_3	1	0	0	0
s_4	0	0	0	0

Colour constraints

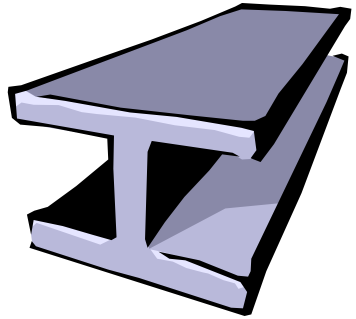
$$\forall j \sum_i C[i, j] \leq p$$

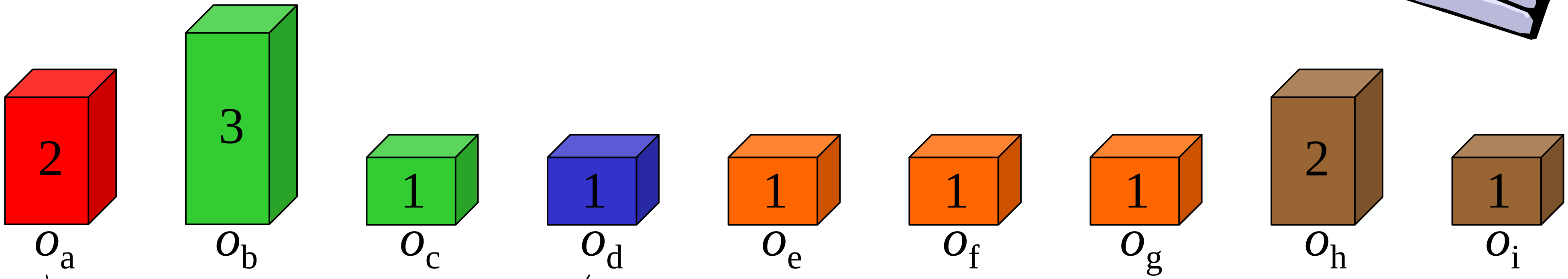
- Channelling: for all i, j

$$O[i, j] = 1 \rightarrow C[colour(i), j] = 1$$

- Why $\rightarrow$ and not $\leftrightarrow$?

Example

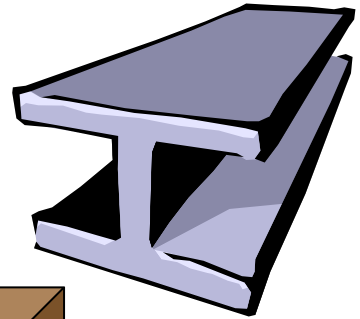


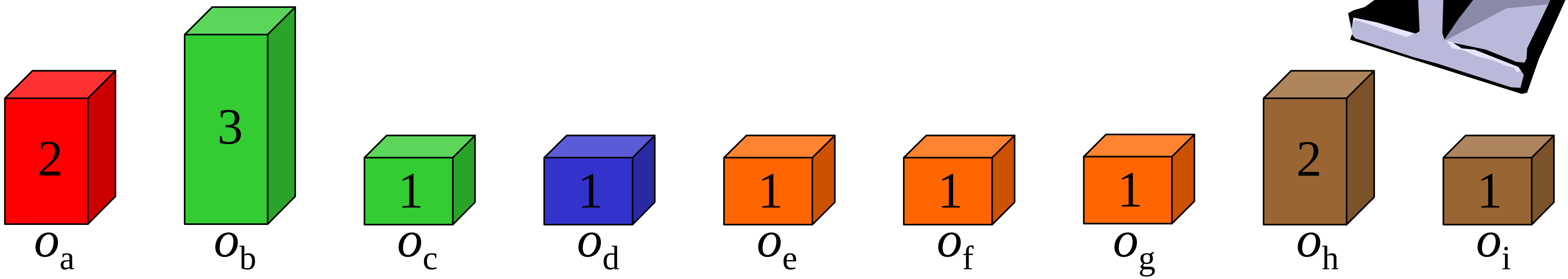


	o_a	o_b	o_c	o_d	o_e	o_f	o_g	o_h	o_i
s_1	1								
s_2				1					
…									

	Red	Green	Blue	Orange	Brown
s_1	1				
s_2			1		
…					

Solution



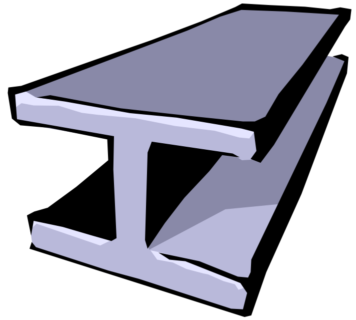


S									
	4	3	3	3	0	0	0	0	0

O	o_a	o_b	o_c	o_d	o_e	o_f	o_g	o_h	o_i
s_1	0	0	0	0	0	0	**1**	**1**	**1**
s_2	**1**	0	**1**	0	0	0	0	0	0
s_3	0	**1**	0	0	0	0	0	0	0
s_4	0	0	0	**1**	**1**	**1**	0	0	0
…	0	0	0	0	0	0	0	0	0

C	Red	Green	Blue	Orange	Brown
s_1	0	0	0	**1**	**1**
s_2	**1**	**1**	0	0	0
s_3	0	**1**	0	0	0
s_4	0	0	**1**	**1**	0
…	0	0	0	0	0

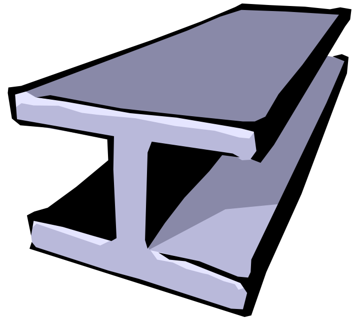
Symmetry



- Slabs are indistinguishable.
- So model suffers from symmetry:
  - Counteract with binary symmetry-breaking constraints:

    $s[1] \geq s[2], \quad s[2] \geq s[3], \ldots$

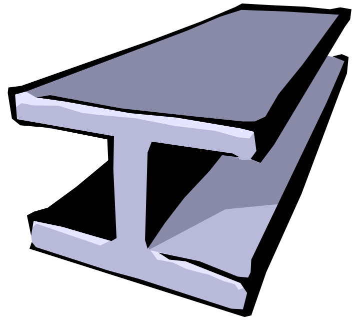
Symmetry



- Slab variables assigned the same size are indistinguishable.
- Break this symmetry with the constraint: If *s*[*i*] and *s*[*i+1*] have the same assignment, then row *i* of the order matrix must be lexicographically less than or equal to row and *i+1*.
  E.g. 1001 ≥ 0110.

$$s[i] = s[i+1] \rightarrow O[\_, i] \geq_{lex} O[\_, i+1]$$

Instance Symmetry

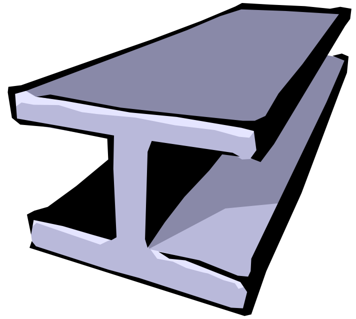


- Orders with same weight, colour are symmetrical.
- Order symmetrical columns lexicographically.
- For all *i*, *j* such that *i*<*j* and orders *i* and *j* have the same weight and colour, add the constraint

$$O[i, \_] \geq_{lex} O[i+1, \_]$$

- Could restrict this to a “sequence” of constraints.

Implied Constraints

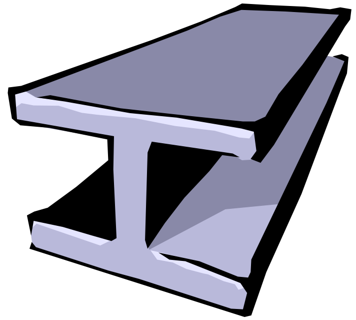


- Combined weight of input orders is a lower bound on optimisation variable:

$$\sum_i weight(o_i) \leq optVar$$

- How does this help?
  - If find a solution of this quality, can stop immediately.

Implied Constraints

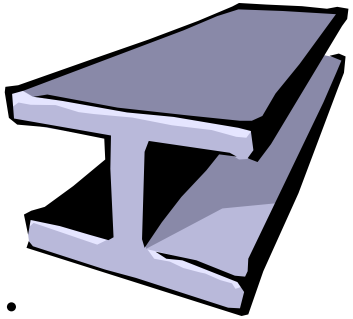


- Lower bound on number of slabs required:

$$minslabs = \frac{\sum_i weight(o_i)}{\text{largest slab size}}$$

- With symmetry-breaking constraints, implies unary constraints on slab variables:
  - Have: $s[1] \geq s[2]$, $s[2] \geq s[3]$, …
  - Know that we need at least *minslabs* slabs.
  - So first *minslabs* slab variables are non-zero.
- Notice that the implied constraint follows from the symmetry-breaking constraint (among others).

Implied Constraints

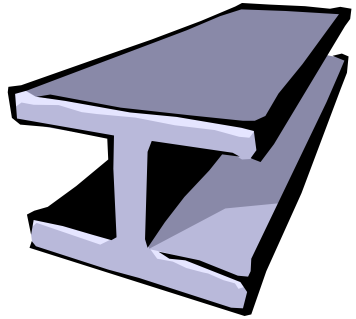


- $AssWt_i$ is the weight of orders assigned to slab i.
  - Prune domains by reasoning about reachable values.
    - Also gives a maximum assigned weight.
    - So, revise lower bound on number of slabs required:

$$\frac{\sum_i weight(o_i)}{\mathrm{maxAssWt}}$$

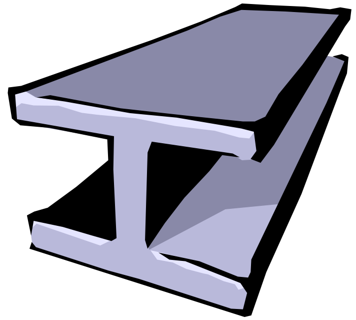
- Incorporate both size and colour information.
- More powerful if done during search (future work).

Implied Constraints



- $Waste_i = s[i] - AssWt_i$
  - Unused portion of a slab.
- Upper bound on total waste:
  - Assume each order is assigned to an individual slab, with smallest size able to hold it.
  - Sum waste in each case: leads to upper bound for optimisation variable.
  - Upper bound on $Waste_i$ is the worst of these cases.

References



- Modelling a Steel Mill Slab Design Problem, A. M. Frisch, I. Miguel, T. Walsh. *Proceedings of the IJCAI Workshop on Modelling and Solving Problems with Constraints*, pp. 39-45, 2001.
- Symmetry and Implied Constraints in the Steel Mill Slab Design Problem. A. M. Frisch, I. Miguel, T. Walsh. *Proceedings of the CP Workshop on Modelling and Problem Formulation*, pp. 8-15, 2001.
- www.csplib.org (Problem 38).