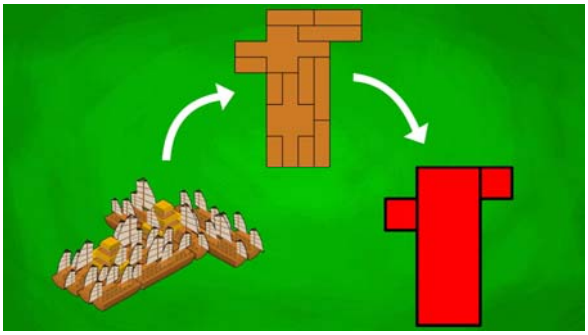
Rectilinear Packing without Rotation

Jimmy Lee & Peter Stuckey

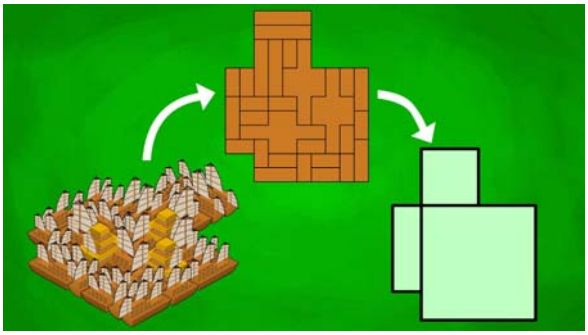
Packing Battle Ships

Northern Provinces Ship Block



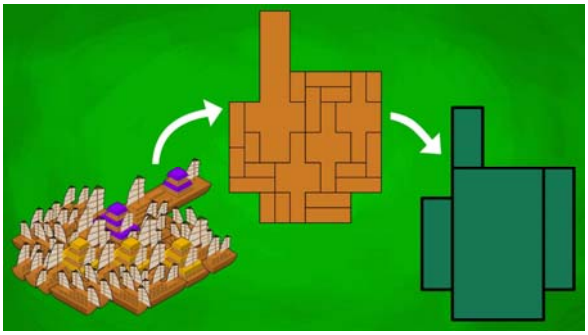
3

Yan Province Ship Block



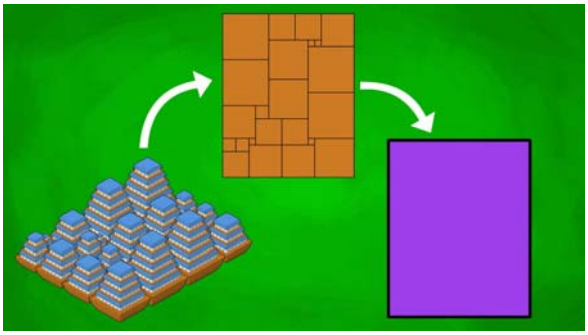
4

Jing Province Ship Block



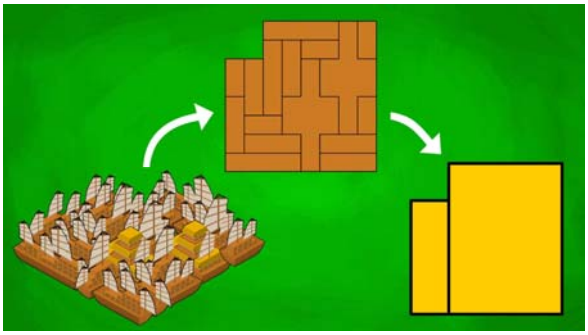
5

House Ship Block



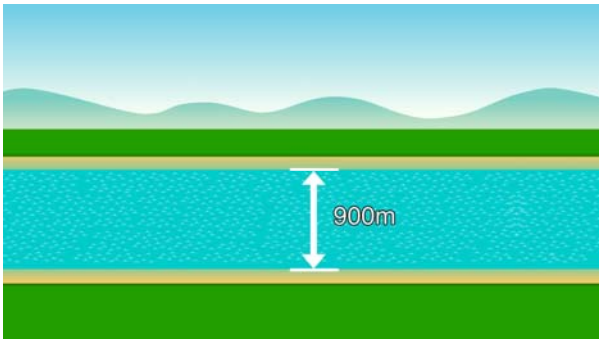
6

Xu Province Ship Block



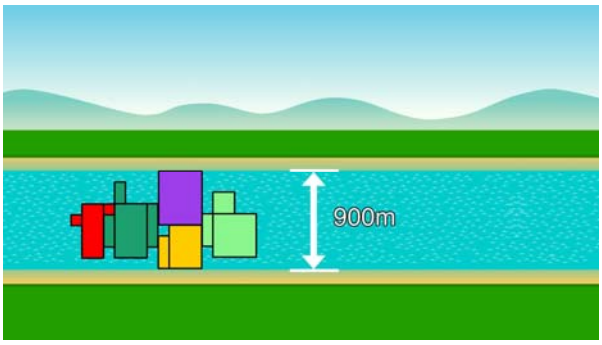
7

A River of Fixed Width



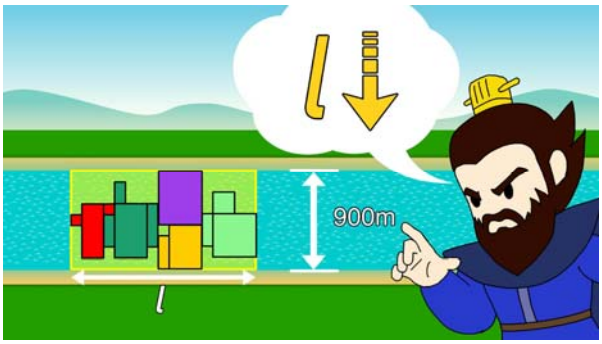
8

Optimizing the Length



9

Optimizing the Length



10

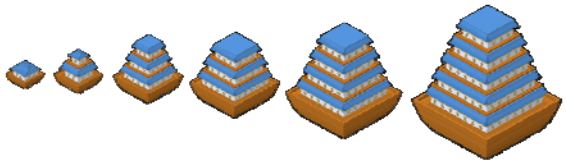
Ship Block Packing

- Given a sequence of rectilinear shapes
- Pack them together so that
  - the resulting height/width does not exceed *h*
  - the resulting length *l* is minimized

11

Ship Block Shapes

- We assume ship blocks are rectilinear
  - House ships are square in shape

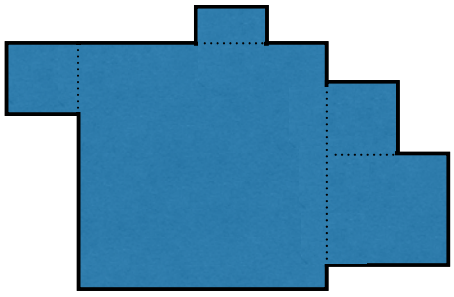


  - Battle ships are rectangular and rectilinear in shape

12

Example Block Shapes

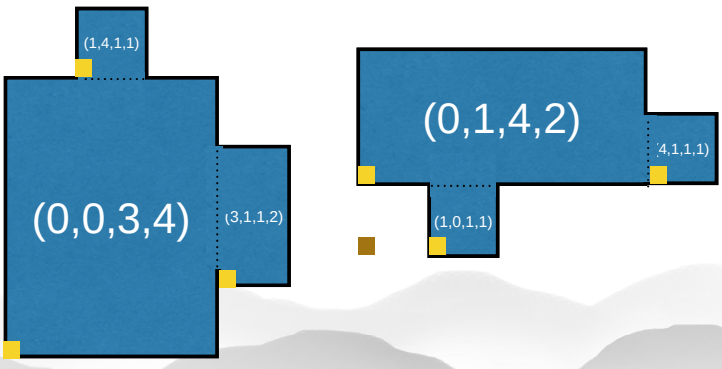
- A complex block shape of 5 rectangles



13

Representing Block Shapes

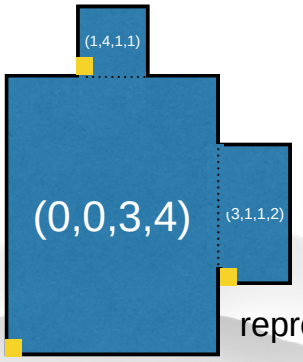
- Rectangles at offset to shape bottom left
  - (x offset, y offset, x size, y size)
- The offsets are different even when the orientation is different (more later)



14

Representing Block Shapes

- Number the rectangle offsets
- A block (of a specific orientation) is
  - a set of rectangles with offsets
- Rectangle offsets can be shared if possible
- E.g. a list of offsets (ordering unimportant)
  - **1: 0,0,3,4**
  - 2: 0,1,4,2
  - **3: 1,4,1,1**
  - **4: 3,1,1,2**
  - 5: 4,1,1,1
  - 6: 1,0,1,1

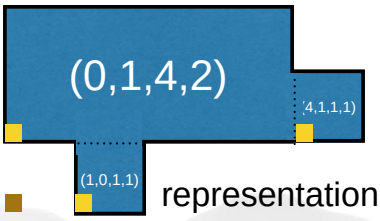


representation {1,3,4}

15

Representing Block Shapes

- Number the rectangle offsets
- A block (of a specific orientation) is
  - a set of rectangles with offsets
- Rectangle offsets can be shared if possible
- E.g. a list of offsets (ordering unimportant)
  - 1: 0,0,3,4
  - **2: 0,1,4,2**
  - 3: 1,4,1,1
  - 4: 3,1,1,2
  - **5: 4,1,1,1**
  - **6: 1,0,1,1**



representation {2,5,6}

16

Ship Block Packing (sbpnorotate.mzn)

- Given *n* blocks defined by fixed shapes. Place the shapes in a river of width *h* so they don't overlap with the length *l* used minimized.

```
int: n; % number of blocks
set of int: BLOCK = 1..n;
int: m; % number of rectangles/offsets
set of int: ROFF = 1..m;
array[ROFF,1..4] of int: d; % defns
array[BLOCK] of set of ROFF: shape;
int: h; % width of river
int: maxl; % maximum length of river
```

17

Ship Block Packing Data (sbpnorotate.dzn)

```
n = 5; m = 12; h = 9; maxl = 16;
d = [| 1,0,2,5 % (xoffset,yoffset,xsize,ysize)
     | 3,4,1,1
     | 0,3,1,1
     | 1,4,2,2
     | 0,1,1,3 % shared by blocks 2 & 3
     | 1,0,4,4
     | 1,5,1,2
     | 1,0,3,5
     | 4,1,1,4
     | 0,0,1,3
     | 1,0,3,4
     | 0,0,4,5 |];
shape = [{1,2,3}, {4,5,6}, {5,7,8,9}, {12},
         {10,11}];
```

18

Packing Decisions + Objective (sbpnorotate.mzn)

- For each block
  - x position of its base
  - y position of its base

```
array[BLOCK] of var 0..maxl: x;
array[BLOCK] of var 0..h: y;

var 0..maxl: l; % length of river used

solve minimize l;
```

19

Packing Constraints (sbpnorotate.mzn)

- For each rectangle/offset in each block
  - it fits within the river

```
forall(i in BLOCK)(forall(r in ROFF)
   (r in shape[i] ->
      (x[i] + d[r,1] + d[r,3] <= l /\
       y[i] + d[r,2] + d[r,4] <= h)));
```

- Can a rectangle stick out the bottom or left?
- No, since offsets are positive

20

Packing Constraints (sbpnorotate.mzn)

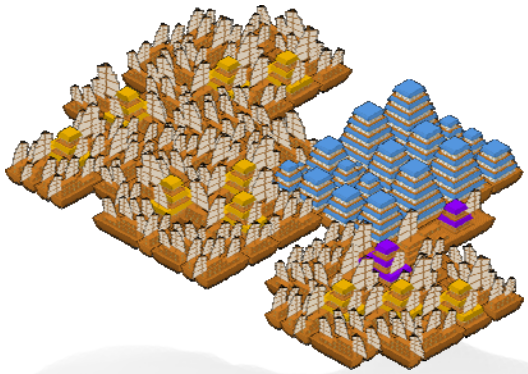
- Rectangle/offsets don't overlap

```
forall(i,j in BLOCK where i < j)
   (forall(r1,r2 in ROFF)
       (r1 in shape[i] /\
        r2 in shape[j] ->
(x[i] + d[r1,1] + d[r1,3] <= x[j] + d[r2,1]
                     \/
 x[j] + d[r2,1] + d[r2,3] <= x[i] + d[r1,1]
                     \/
 y[i] + d[r1,2] + d[r1,4] <= y[j] + d[r2,2]
                     \/
 y[j] + d[r2,2] + d[r2,4] <= y[i] + d[r1,2]
)));
```

21

Solving the Model

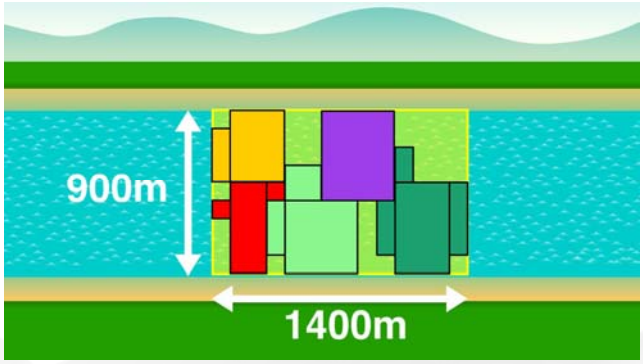
```
l = 14;
x = [0, 3, 9, 6, 0];
y = [0, 0, 0, 4, 5]
----------
==========
```



22

Solving the Model

```
l = 14;
x = [0, 3, 9, 6, 0];
y = [0, 0, 0, 4, 5]
----------
==========
```



23

Summary

- Complex packing problems
  - make shapes from components
  - ensure components don't overlap
- Packing bulk mineral cargoes onto storage pads at ports is an example of complex rectilinear packing (without rotation)
- Problem gets even more complicated when orientation/rotation is taken into account!

24

Image Credits

All graphics by Marti Wong, ©The Chinese University of Hong Kong and the University of Melbourne 2016