EDA HW5

B02901178 江誠敏

December 8, 2015

1 Problem 1.

1. No, since we don't know the exact amount of iteration r.
2. Yes, as long as we know W or w_i is bounded (so that $W < n \max w_i$.

2 Problem 2.

	swap	gain	Σ gain	locked	A	B
1	(c, d)	0	0	c, d	a, b, d	c, e, f
2	(b, f)	-39	-39	b, c, d, f	a, d, f	c, b, e
3	(a, e)	39	0	a, b, c, d, e, f	d, e, f	a, b, c

3 Problem 3.

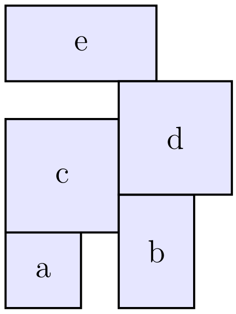
We define

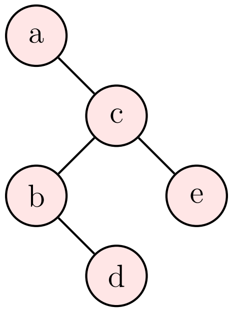
1. Solution space: Each partition that split the vertice into to group with same size is in the solution space.
2. Neighborhood structure: Let (A, B) be a balanced partitioning. For all $a \in A, b \in B$, we swap these vertice and become a new partition (A', B'), then all these partition are the neighborhood of (A, B).
3. Cost Function: We define the total cost (cut size) to be our cost function.

Then we could preform annealing. That is, set a random initial state and temperature. Then each time, we pick a random neighbor, calculate the cost difference, and decide if we should go to the new state.

4 Problem 4.

First we compact the placement, and then derive the B-tree.

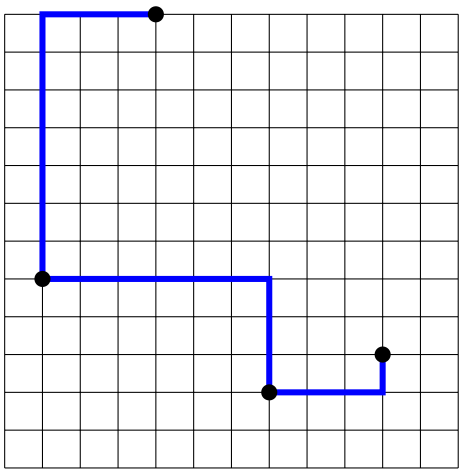


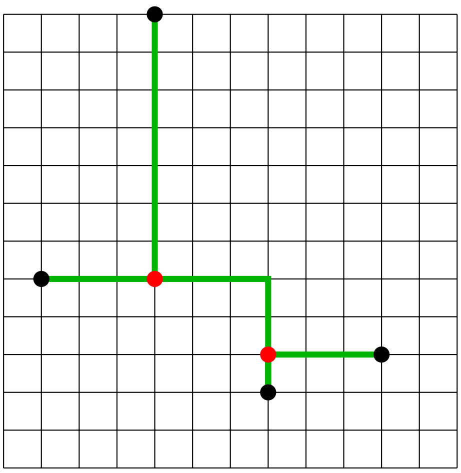


1. (a): $[(0,0),(0,2),(2,2),(2,0)]$.
2. (c): $[(0,0),(0,5),(3,5),(3,0)]$.
3. (b): $[(0,0),(0,5),(3,5),(3,3),(5,3),(5,0)]$.
4. (d): $[(0,0),(0,5),(3,5),(3,6),(6,6),(6,0)]$.
5. (e): $[(0,0),(0,8),(4,8),(4,6),(6,6),(6,0)]$.

Area cost $= 48$. Area efficiency $= 36/48 \approx 0.75\%$.

5 Problem 5.





$$m = 19$$
$$n = 23$$
$$p = 19$$
$$q = 424$$
$$r \approx 176.67$$
$$s \approx 20.10$$

6 Problem 6.

(a) $a = 10, b = 10, \text{err} = 0\%$.

(b) $a = 10, b = 12, \text{err} = 16.7\%$.

(c) $a = 10, b = 14, \text{err} = 28.6\%$.

(d) As the degree increase, the error rate increases.
Using MST estimation or fit by some polynomial (i.e, $\hat{b} = pa + q$, where a is HPWL, $\hat{b}$ is the new estimate, and p, q are some parameter.) may give a better result.

7 Problem 7.

We split a square into 4 nodes. That is, we use (b, d) to represent a node, where b is a square in the graph and d is a direction, which means that "We traversed to the square d with the last direction facing d". Also we change the distance(cost) to a 2-tuple (x, y) such that x is the path length, and y is the number of bends on the path. We define

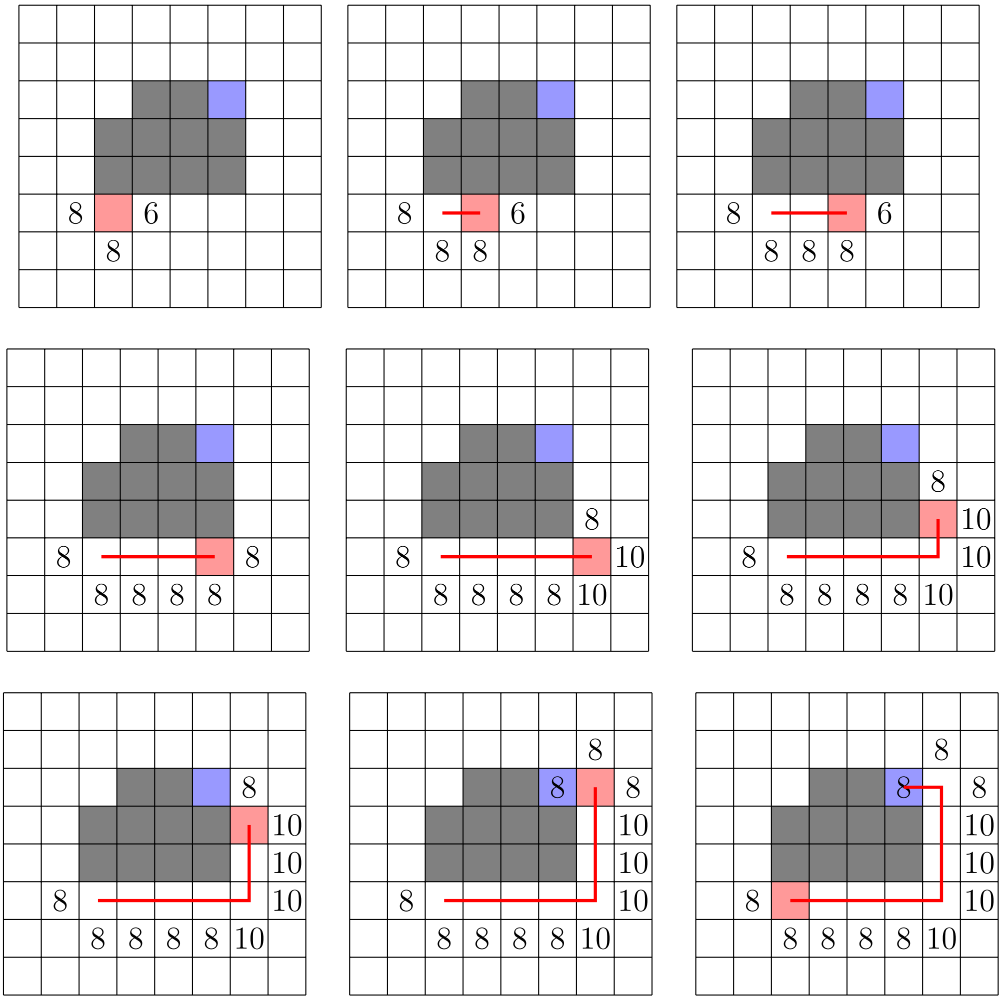
$$(x_1, y_1) < (x_2, y_2) \quad \Leftrightarrow \quad x_1 < x_2 \lor (x_1 = x_2 \land y_1 < y_2)$$

and

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2).$$

The distance from (b_1, d_1) to (b_2, d_2) is set to $(1, 0)$ if $d_1 = d_2$ and $(1, 1)$ if $d_1 \neq d_2$, when b_1, b_2 are adjacent. Now set the distance $d(S, i) = 0$ for each of the 4 distance and find the minimum path to $d(T, i)$ using any shortest path finding algorithm gives the answer.

8 Problem 8.



9 Problem 9.