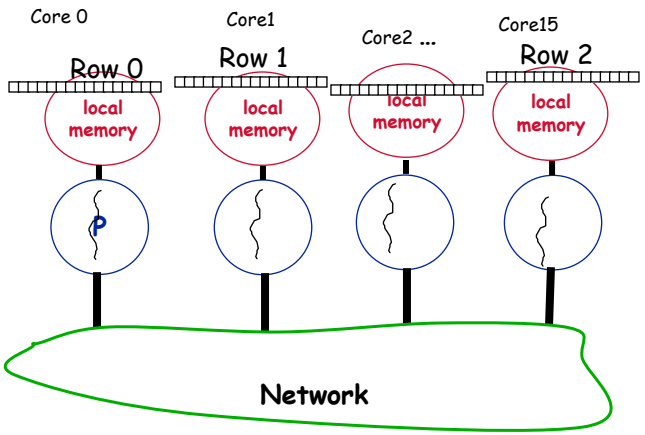
How to Divvy up the Pie in the Face of Dynamic Computations – E.g., TSP

Agarwal

6.173
Fall 2010
L05

Recall, for Static Problems



Load balancing can occur from a good static partitioning of static computations – i.e., were we know a priori the amount of computation and communication

Dynamic Computations

In many computations, the amount of computation and communication are data dependent, and hard to predict statically.

E.g., TSP

Other examples:

Searching

Sorting

Graph problems

Adaptive algorithms

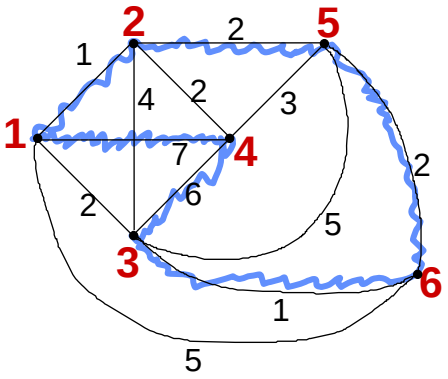
Adaptive grid refinement

Data dependent computations

Here is where we have some fun!

Let's look at our familiar TSP example

Quickly Review TSP B&B

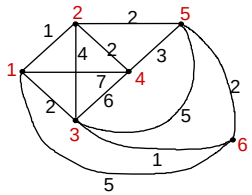


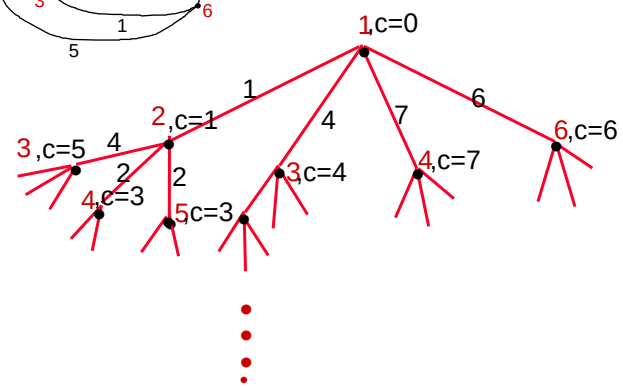
Find shortest tour

Example tour: (1 2 5 6 3 4 1)

***assume missing edges imply infinite cost**

Exhaustive Search

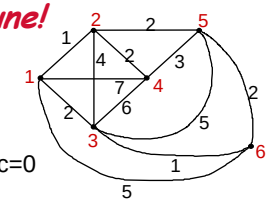




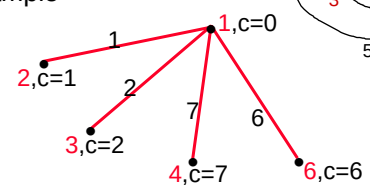
Find lowest cost (c) path,

- But $\frac{(n-1)!}{2}$ tours!!!
- Can exploit parallelism though!

Branch and Bound – *Prune!*



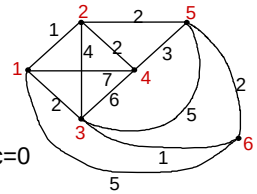
Simple example

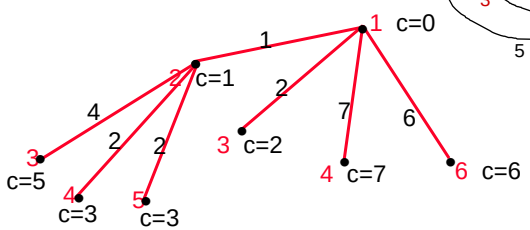


Idea: find an upper bound on cost,
then use that to prune.

So, let's first find a bound using a greedy algorithm

Find an Upper Bound

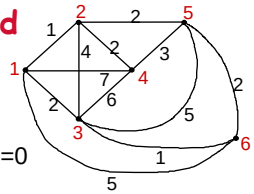


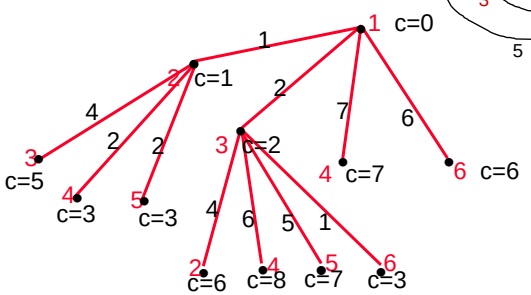


- (1, 2 ...) most promising, because it has lowest cost so far. So follow that lead -- greedy

Next, (1, 3 ...) most promising

Find an Upper Bound

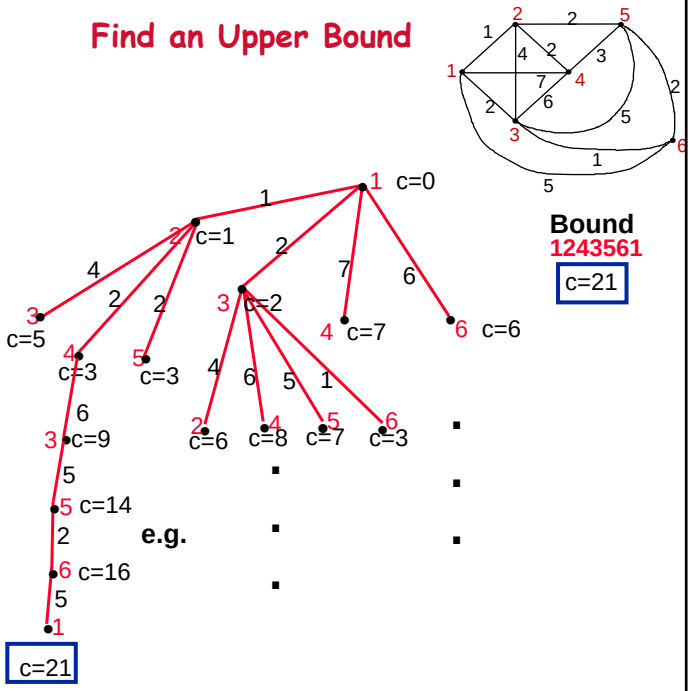




- (1, 2 ...) most promising, follow that lead -- greedy
- next, (1, 3 ...) most promising

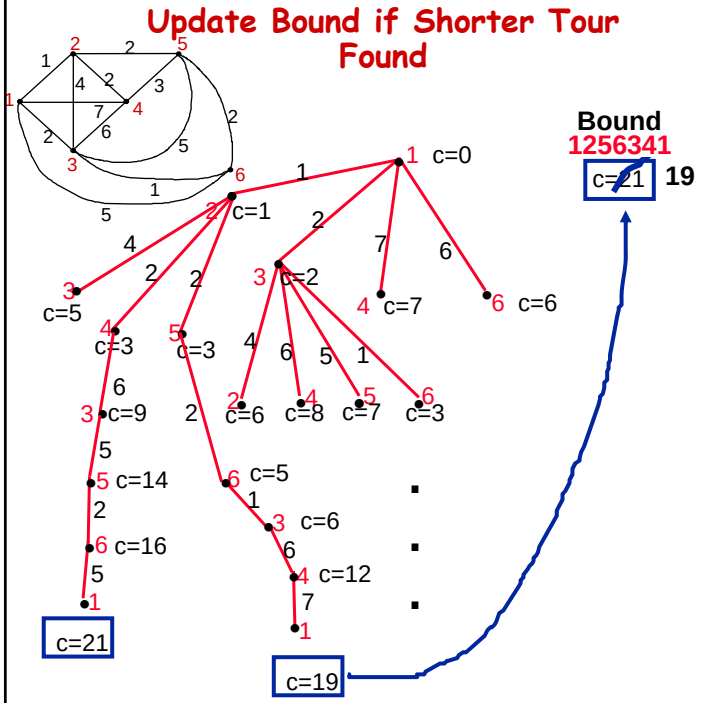
Next, (1, 2, 4 ...) most promising

Find an Upper Bound



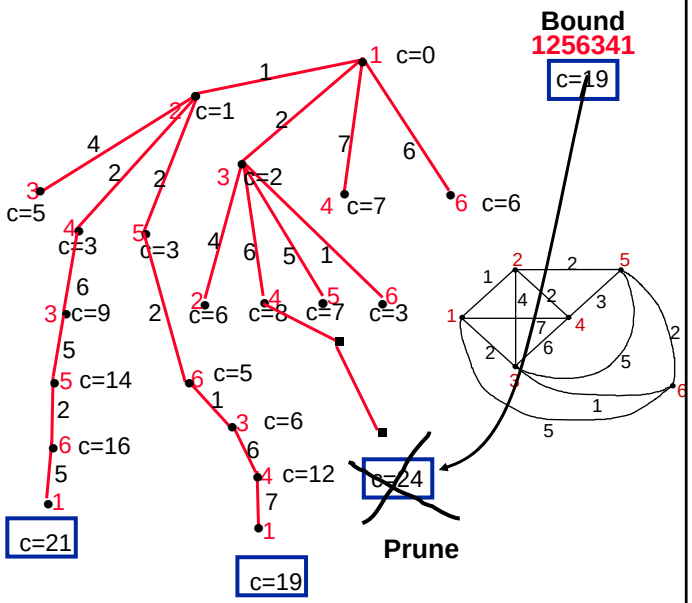
- Continue … till a tour is found
- Bound = 21: (1 2 4 3 5 6 1)

Update Bound if Shorter Tour Found



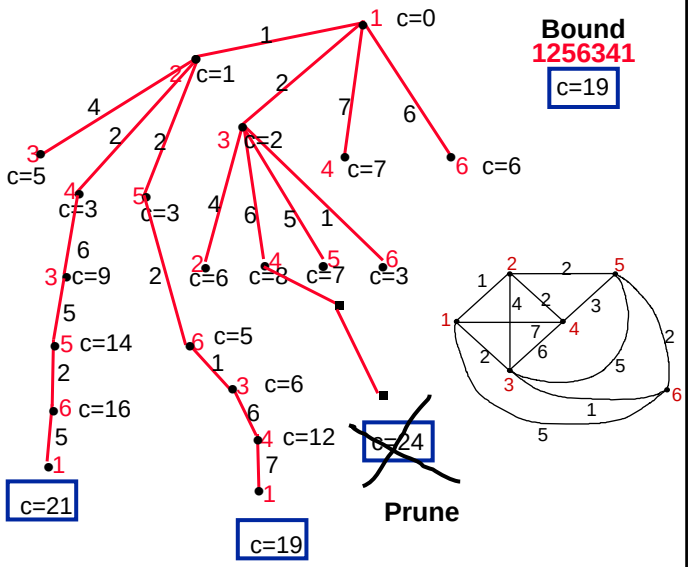
- New bound = 19: (1 2 5 6 3 4 1)

Now Prune Future Paths if Partial Path Cost Exceeds Bound



- Prune if partial path cost is greater than the bound

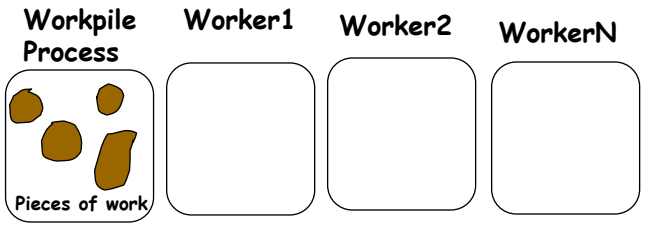
Branch and Bound Algorithms



Discuss

- What are the opportunities for parallelism?

Workpile Partitioning Approach



A "master" process (aka "workpile" process) first creates the work items. It can then:

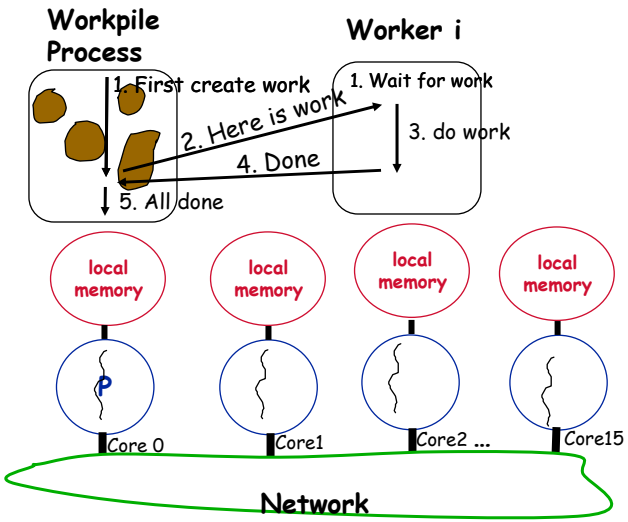
Statically assign work to each of the workers so estimated load is balanced

OR

Dynamically negotiate with the workers for dynamic load balancing

Workpile Partitioning Approach

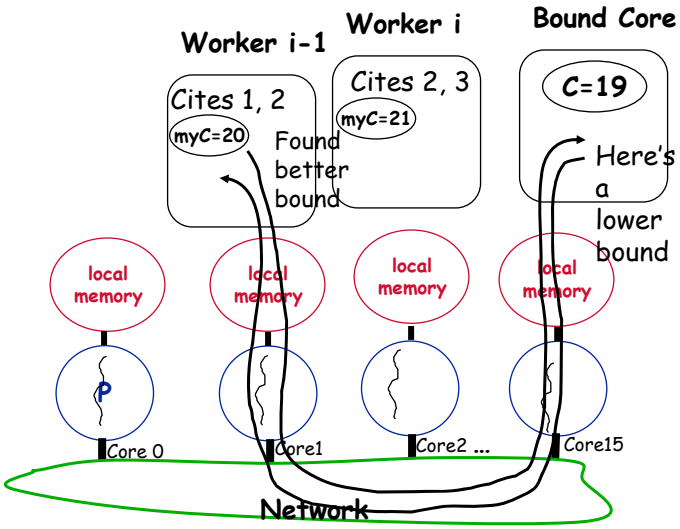
Static Load Balancing



- Create just as much work as there are cores and dole out this work to each core (then sit around and watch the fun)
- Discuss approaches to statically partition TSP

Workpile Partitioning Approach

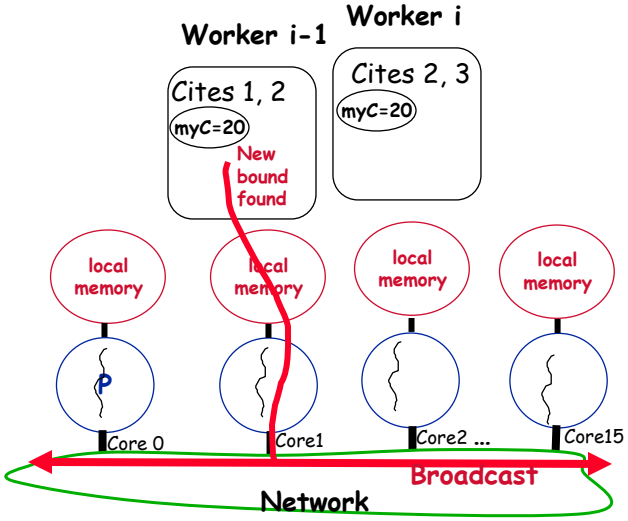
Static Load Balancing, Bound Core



- Workpile core creates just as much work as there are worker cores and dole out this work to each core (then sits around and watches the fun)
- Discuss approaches to statically partition TSP
  - Statically, give a set of starting cities to each core
  - Can create a "bound core" to keep track of the min cost in message passing model – example of program partitioning!
  - You can stop searching whenever you think you have a good-enough tour (but not necessarily optimal)

Workpile Partitioning Approach

Static Load Balancing, Broadcast



- Workpile core creates just as much work as there are worker cores and dole out this work to each core (then sits around and watches the fun)
- Discuss approaches to statically partition TSP
  - Statically, give a set of starting cities to each core
  - If you had a fast broadcast, you could also broadcast new bounds when you found them

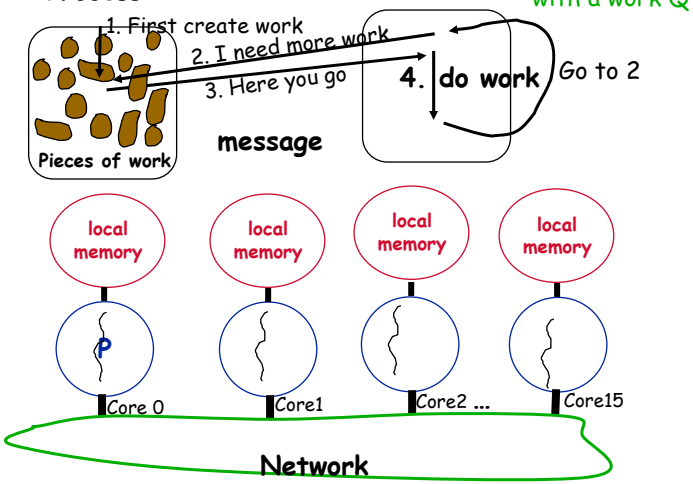
Problems with static partitioning?

Workpile Partitioning Approach

Dynamic Load Balancing

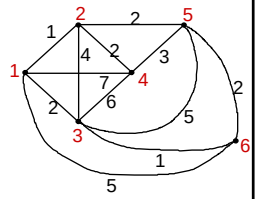
(aka self scheduling)

In shared memory, this would be done with a work Q



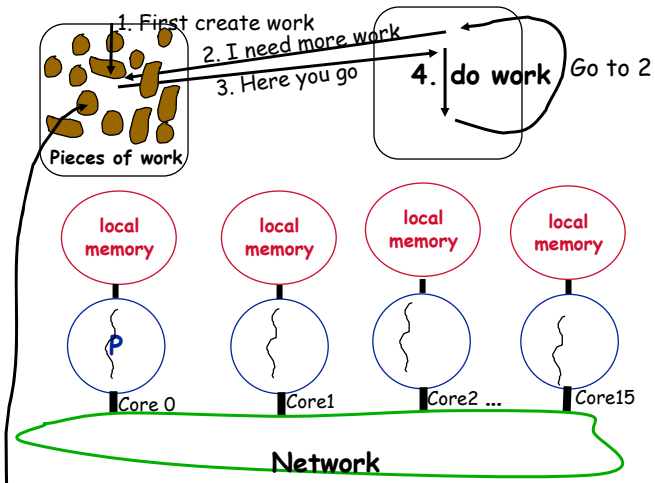
Create many more work items than there are workers

In TSP B&B, what might these work items be? Discuss



Workpile Partitioning Approach

Dynamic Load Balancing in TSP

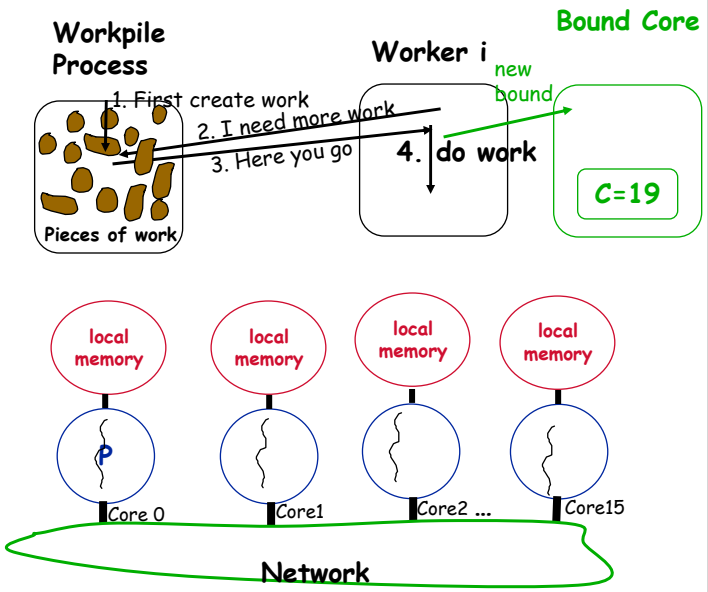


Starting from a given city, enumerate all partial paths of length N

Each worker core picks up one of these partial paths and fleshes it out (bound addressed as before)

Workpile Partitioning Approach

Dynamic Load Balancing in TSP



The Ubiquitous

Simulating Annealing Method

for TSP

"When in doubt use sim. anneal.!"

Very common in industry, especially VLSI CAD, for most search and optimization problems

Basic idea:

Find a workable solution.

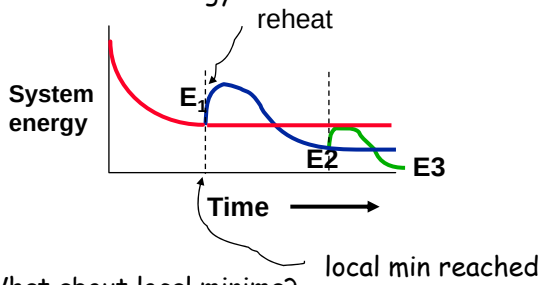
Then try to search for successively better solutions by making small random changes (or "moves") to existing solution

Original Ideas: Kirkpatrick - IBM

Analogy with thermodynamics

Annealing, tempering materials

- Start with a high temperature
- Cool slowly
  - mobility reduced
- Move to reduce energy
- Reduced energy state



What about local minima?
Reheat slightly, and cool again…
Repeat

Need a model for "hopping" out of local mimina…

Simulated Annealing Method Steps

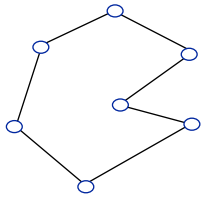
1. Determine a current state of the system and a starting temperature
2. * Create an objective function (system energy)
3. Create a suitable "move"
4. Always accept moves that improve the objective function.

   Also accept with certain probability moves that hurt the objective function (Probability increases with temperature. Temperature is reduced over time so uphill moves become rarer with time)
5. Iterate till satisfied *

Back to TSP and Simulated Annealing
Analogy with thermodynamic cooling

1. Current state → a tour
2. * Objective function (system energy)

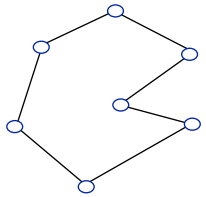
   Current tour cost $\sum$edges in tour



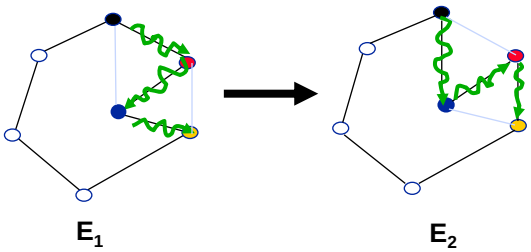
TSP with Simulated Annealing

1. Current state → a tour
2. * Objective function (system energy)

   Current tour cost $\sum$ edges in tour



3. "Move"
   - Reverse a section of paths
   - can also do multiple at a time



TSP with Simulated Annealing

4. "Accepting" moves

$If\ E_2 < E_1 \rightarrow move$

Else move with probability $p = e^{\frac{-(E_2 - E_1)}{kT}}$

Always accept downhill moves

Accept uphill moves sometimes

5. Iterate till the objective function reaches some threshold value *

Reduce temperature T with time

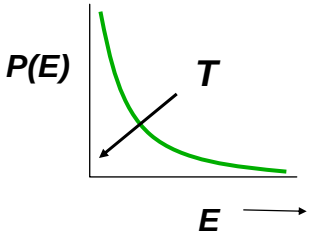
i.e. large uphill moves must become rarer with time

Simulated Annealing*

Analogy with thermodynamics

Use Boltzmann distribution for energies

Probability energy level E occupied = $e^{\frac{-E}{kT}}$



There is a finite probability of "hopping" into a higher energy state given by

$$e^{\frac{-(\Delta E)}{kT}}$$

Simulated annealing -- benefits

Can have very complex, nonlinear, cost functions

Stop whenever you want and get a reasonable solution

Parallel?

(Academic) Issues with Dynamic Partitioning

- How do you define speedup

$$\frac{T(1)}{T(P)}?$$

- But the amount of work done is different
- Synchronization in TSP with message passing
- Communication locality?

  Keep local copy of bound

  Who cares if we do some extra computation – cores are becoming free anyway!

- Use TSP B&B for lab!