http://www.cs.unh.edu/~ruml/cs758

1 handout: slides
Backtracking

- Hardness
- Optimization
- Backtracking
- Depth-first Search
- DFS Order
- Problems
- ILDS
- ILDS Order
- Break

Local Search
Hardness

NPC: SAT, vertex cover, clique, subset sum, ...

greedy: local choice is optimal

DP: poly number of options to track

search: exponential number of options, often combinations
A tree representation of alternatives in a small combinatorial problem.
Backtracking

- depth-first search
- child ordering
- lower bounds
- branch-and-bound
Depth-first Search

DFS \((node)\)
1. If is-leaf\((node)\)
2. Visit\((node)\)
3. else
4. For \(i\) from 0 to \(num-children\)
5. DFS(child\((node, i)\))
Depth-first Search Order

Backtracking
- Hardness
- Optimization
- Backtracking
- Depth-first Search

DFS Order
- Problems
- ILDS
- ILDS Order
- Break

Local Search
Problems Are Hard

Backtracking
■ Hardness
■ Optimization
■ Backtracking
■ Depth-first Search
■ DFS Order
■ Problems
■ ILDS
■ ILDS Order
■ Break
Local Search

13,509 US cities (W. Cook)
Problems Are Hard

Backtracking
- Hardness
- Optimization
- Backtracking
- Depth-first Search
- DFS Order

Problems
- ILDS
- ILDS Order
- Break

Local Search

Wheeler Ruml (UNH)
Problems Are Hard

Backtracking
- Hardness
- Optimization
- Backtracking
- Depth-first Search
- DFS Order

Problems
- ILDS
- ILDS Order
- Break

Local Search

(S. LaValle)

Wheeler Ruml (UNH)
ILDS \((node, allowance, remaining)\)

1. If is-leaf\((node)\)
2. Visit\((node)\)
3. else
4. If \(allowance > 0\)
5. ILDS\(\text{child}(node, 1), allowance - 1, remaining - 1\)\)
6. If \(remaining > allowance\)
7. ILDS\(\text{child}(node, 0), allowance, remaining - 1\)\)

start with ILDS\((\text{root}, iteration, max-depth)\)
The second pass of ILDS visits all leaves with one discrepancy in their path from the root.
asst 14
final exam: Wed Dec 17, 1-3pm, Kingsbury N113
last year’s final at recitation this Fri Dec 5
review on Fri Dec 12
Local Search
A graph representing an improvement-based search.
hill climbing
simulated annealing
large neighborhood search
genetic algorithms
particle swarm optimization
Max Cut

- maximize weight of edges crossing the cut $w(A, B)$
- decision version is NP-complete
- simple local search:
  - move vertex $u$ from $A$ to $B$ iff
    $$\sum_{v \neq u \in A} w_{uv} > \sum_{v \in B} w_{uv}$$
- it’s possible to bound suboptimality of local minima under this neighborhood!
for any $u$ in $A$, 
\[
\sum_{v \neq u \in A} w_{uv} \leq \sum_{v \in B} w_{uv}
\]

summing over all $u$ in $A$,
\[
2 \sum_{(u,v) \in A} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B)
\]

summing over all $u$ in $B$,
\[
2 \sum_{(u,v) \in B} w_{uv} \leq \sum_{u \in A, v \in B} w_{uv} = w(A, B)
\]

add:
\[
2 \sum_{(u,v) \in A} w_{uv} + 2 \sum_{(u,v) \in B} w_{uv} \leq 2w(A, B)
\]
divide by 2:

\[
\sum_{(u,v) \in A} w_{uv} + \sum_{(u,v) \in B} w_{uv} \leq w(A, B)
\]

eg, more weight crossing than within partitions

let \( W \) be sum of all weight in graph.

add crossing weight to both sides:

\[
W \leq 2w(A, B)
\]

\[
W/2 \leq w(A, B)
\]

note optimal is at most \( W \)
For example:

- What’s still confusing?
- What question didn’t you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.

*Thanks!*