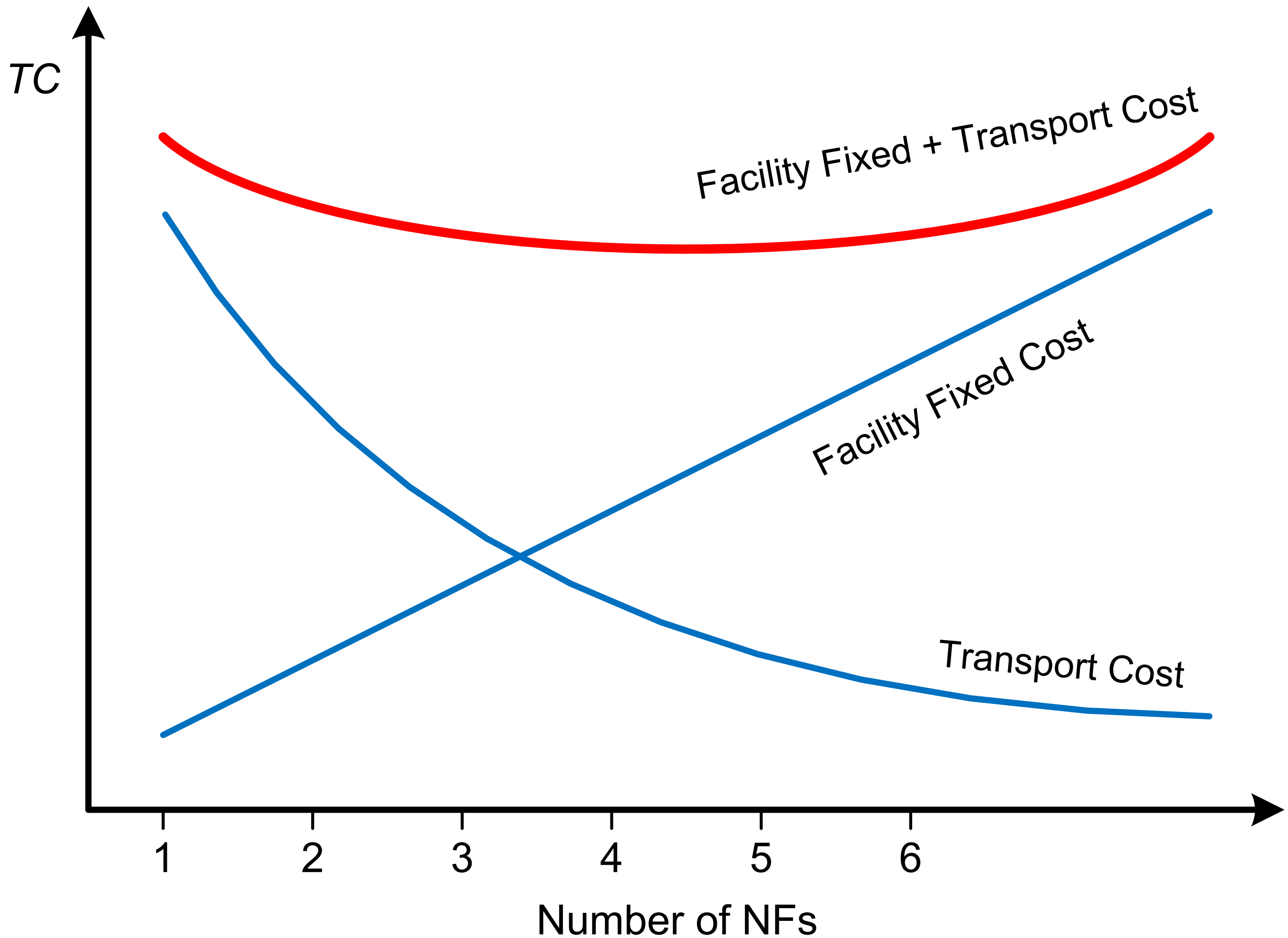
Optimal Number of NFs



Uncapacitated Facility Location (UFL)

- NFs can only be located at discrete set of sites
  - Allows inclusion of fixed cost of locating NF at site $\Rightarrow$ opt number NFs
  - Variable costs are usually transport cost from NF to/from EF
  - Total of $2^n - 1$ potential solutions (all nonempty subsets of sites)

$$M = \{1, \ldots, m\}, \quad \text{existing facilites (EFs)}$$

$$N = \{1, \ldots, n\}, \quad \text{sites available to locate NFs}$$

$$M_i \subseteq M, \quad \text{set of EFs served by NF at site } i$$

$$c_{ij} = \text{variable cost to serve EF } j \text{ from NF at site } i$$

$$k_i = \text{fixed cost of locating NF at site } i$$

$$Y \subseteq N, \quad \text{sites at which NFs are located}$$

$$\begin{aligned} Y^* &= \arg\min_Y \left\{ \sum_{i \in Y} k_i + \sum_{i \in Y} \sum_{j \in M_i} c_{ij} : \bigcup_{i \in Y} M_i = M \right\} \\ &= \text{min cost set of sites where NFs located} \end{aligned}$$

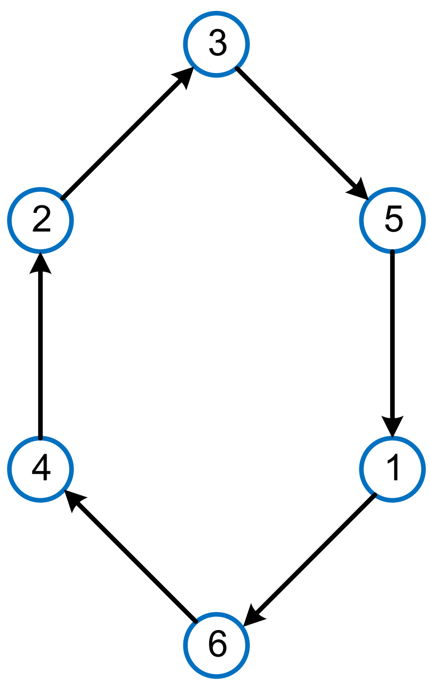
$$\left| Y^* \right| = \text{number of NFs located}$$

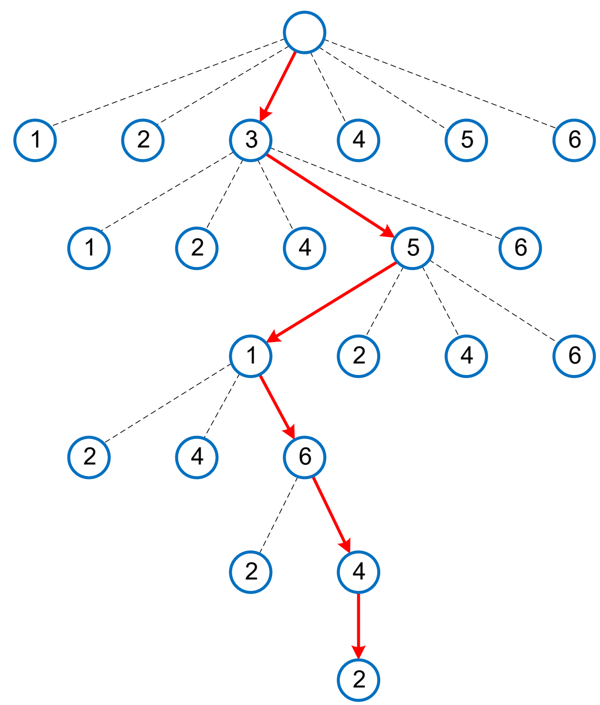
Heuristic Solutions

- Most problems in logistics engineering don’t admit optimal solutions, only
  - Within some bound of optimal (provable bound, opt. gap)
  - Best known solution (stop when need to have solution)
- Heuristics - computational effort split between
  - Construction: construct a feasible solution
  - Improvement: find a better feasible solution
- Easy construction:
  - any random point or permutation is feasible
  - can then be improved $\Rightarrow$ *construct-then-improve* multiple times
- Hard construction:
  - almost no chance of generating a random feasible solution due to constraints on what is a feasible solution
  - need to include randomness at decision points as solution is generated in order to construct multiple different solutions (which “might” then be able to be improved)

Heuristic Construction Procedures

- Easy construction:
  - any random permutation is feasible and can then be improved
- Hard construction:
  - almost no chance of generating a random solution in a single step that is feasible, need to include randomness at decision points as solution is constructed

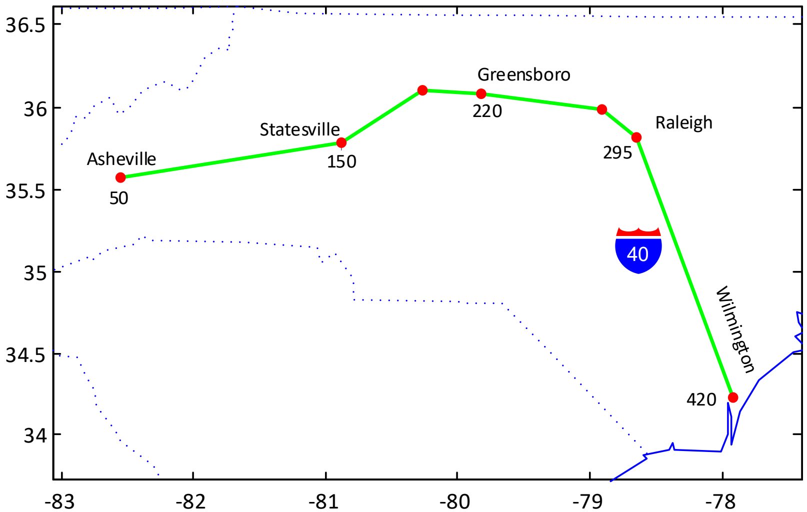




UFL Solution Techniques

- Being uncapacitated allows simple heuristics to be used to solve
  - ADD construction: add one NF at a time
  - DROP construction: drop one NF at a time
  - XCHG improvement: move one NF at a time to unoccupied sites
  - HYBRID algorithm combination of ADD and DROP construction with XCHG improvement, repeating until no change in Y
    - Use as default heuristic for UFL
    - See Daskin [2013] for more details
- UFL can be solved as a MILP
  - Easy MILP, LP relaxation usually optimal (for strong formulation)
  - MILP formulation allows constraints to easily be added
    - e.g., capacitated facility location, fixed number of NFs, some NF at fixed location
  - Will model UFL as MILP mainly to introduce MILP, will use UFL HYBRID algorithm to solve most problems

Ex 8: UFL ADD



$$k = \begin{bmatrix} 150 & 200 & 150 & 150 & 200 \end{bmatrix}$$

$$c_{ij} = w_j d_{ij} = f_j r d_{ij} = (1)(1) d_{ij} = d_{ij}$$

	c_{ij}	1	2	3	4	5
Asheville:	1	0	100	170	245	370
Statesville:	2	100	0	70	145	270
Greensboro:	3	170	70	0	75	200
Raleigh:	4	245	145	75	0	125
Wilmington:	5	370	270	200	125	0

$$Y = \{\ \}$$

Y	1	2	3	4	5	c_{Yj}	k_Y	$c_{Yj} + k_Y$
1	0	100	170	245	370	885	150	1,035
2	100	0	70	145	270	585	200	785
3	170	70	0	75	200	515	150	665
4	245	145	75	0	125	590	150	740
5	370	270	200	125	0	965	200	1,165

$$Y = \{3\}$$

Y	1	2	3	4	5	c_{Yj}	k_Y	$c_{Yj} + k_Y$
3,1	0	70	0	75	200	345	300	645
3,2	100	0	0	75	200	375	350	725
3,4	170	70	0	0	125	365	300	665
3,5	170	70	0	75	0	315	350	665

$$Y = \{3,1\}$$

Y	1	2	3	4	5	c_{Yj}	k_Y	$c_{Yj} + k_Y$
3,1,2	0	0	0	75	200	275	500	775
3,1,4	0	70	0	0	125	195	450	645
3,1,5	0	70	0	75	0	145	500	645

$$Y^* = \{3,1\}$$

UFLADD: Add Construction Procedure

procedure $\mathit{ufladd}(\mathbf{k},\mathbf{C})$

$Y \leftarrow \{\}$

$TC \leftarrow \infty,\ done \leftarrow \text{false}$

repeat

$TC' \leftarrow \infty$

for $i' \in \{1,\dots,n\} \setminus Y$

$$TC'' \leftarrow \sum_{h \in Y \cup i'} k_h + \sum_{j=1}^{m} \min_{h \in Y \cup i'} c_{hj}$$

if $TC'' < TC'$

$TC' \leftarrow TC'',\ i \leftarrow i'$

endif

endfor

if $TC' < TC$

$TC \leftarrow TC',\ Y \leftarrow Y \cup i$

else

$done \leftarrow \text{true}$

endif

until $done = \text{true}$

return Y, TC

```
%% UFLADD Matlab code, given k and C as inputs
y = [];
TC = Inf; done = false;
while ~done
   TC1 = Inf;                      % Stops if y = all NF,
   for i1 = setdiff(1:size(C,1),y) % since i1 = []
      TC2 = sum(k([y i1])) + sum(min(C([y i1],:),[],1));
      if TC2 < TC1
         TC1 = TC2; i = i1;
      end
   end
   if TC1 < TC % not true if y = all NF, since TC1 = Inf
      TC = TC1; y = [y i];
   else
      done = true;
   end
end
y,TC
```

UFLXCHG: Exchange Improvement Procedure

procedure $uflxchg(\mathbf{k}, \mathbf{C}, Y)$

$TC \leftarrow \sum_{i \in Y} k_i + \sum_{j=1}^{m} \min_{i \in Y} c_{ij}$

$TC' \leftarrow \infty$, $done \leftarrow$ false

while $|y| > 1$ and $done =$ false

  for $i' \in y$

    for $j' \in \{1, \ldots, n\} \setminus Y$

      $Y' \leftarrow Y \setminus i' \cup j'$

      $TC'' \leftarrow \sum_{i \in Y'} k_i + \sum_{j=1}^{m} \min_{i \in Y'} c_{ij}$

      if $TC'' < TC'$

        $TC' \leftarrow TC''$, $i \leftarrow i'$, $j \leftarrow j'$

      endif

    endfor

  endfor

  if $TC' < TC$

    $TC \leftarrow TC'$, $Y \leftarrow Y \setminus i \cup j$

  else

    $done \leftarrow$ true

  endif

endwhile

return Y, TC

```
%% UFLXCHG Matlab code, given k, C, and y as inputs
TC = sum(k(y)) + sum(min(C(y,:),[],1));
TC1 = Inf; done = false;
while length(y) > 1 && ~done
   for i1 = y
      for j1 = setdiff(1:size(C,1),y)
         y1 = [setdiff(y,i1) j1];
         TC2 = sum(k(y1)) + sum(min(C(y1,:),[],1));
         if TC2 < TC1
            TC1 = TC2; i = i1; j = j1;
         end
      end
   end
   if TC1 < TC
      TC = TC1; y = [setdiff(y,i) j];
   else
      done = true;
   end
end
y, TC
```

Modified UFLADD

procedure $ufladd(\mathbf{k},\mathbf{C},\boxed{Y,p})$

$\cancel{Y \leftarrow \{\}}$

$TC \leftarrow \infty,\ done \leftarrow \text{false}$

repeat

$\quad TC' \leftarrow \infty$

$\quad$ **for** $i' \in \{1,\ldots,n\} \setminus Y$

$$\qquad TC'' \leftarrow \sum_{h \in Y \cup i'} k_h + \sum_{j=1}^{m} \min_{h \in Y \cup i'} c_{hj}$$

$\qquad$ **if** $TC'' < TC'$

$\qquad\quad TC' \leftarrow TC'',\ i \leftarrow i'$

$\qquad$ **endif**

$\quad$ **endfor**

$\quad$ **if** $\boxed{\left(p = \{\} \text{ and } TC' < TC\right) \text{ or } \left(p \neq \{\} \text{ and } |Y| < p\right)}$

$\qquad TC \leftarrow TC',\ Y \leftarrow Y \cup i$

$\quad$ **else**

$\qquad done \leftarrow \text{true}$

$\quad$ **endif**

until $done = \text{true}$

return Y, TC

- *Y* input can be used to start UFLADD with *Y* NFs
  - Used in hybrid heuristic
- *p* input can be used to keep adding until number of NFs = *p*
  - Used in p-median heuristic

UFL: Hybrid Algorithm

procedure $ufl(\mathbf{k},\mathbf{C})$
$Y', TC' \leftarrow ufladd(\mathbf{k},\mathbf{C})$
$done \leftarrow \text{false}$
repeat
  $Y, TC \leftarrow uflxchg(\mathbf{k},\mathbf{C},Y')$
  if $Y \neq Y'$
    $Y', TC' \leftarrow ufladd(\mathbf{k},\mathbf{C},Y)$
    $Y'', TC'' \leftarrow ufldrop(\mathbf{k},\mathbf{C},Y)$
    if $TC'' < TC'$
      $TC' \leftarrow TC'', Y' \leftarrow Y''$
    endif
    if $TC' \geq TC$
      $done \leftarrow \text{true}$
    endif
  else
    $done \leftarrow \text{true}$
  endif
until $done = \text{true}$
return Y, TC

```
%% UFL Matlab code, given k and C
[y1,TC1] = ufladd(k,C);
done = false;
while ~done
   [y,TC] = uflxchg(k,C,y1);
   if ~isequal(y,y1)
      [y1,TC1] = ufladd(k,C,y);
      [y2,TC2] = ufldrop(k,C,y);
      if TC2 < TC1
         TC1 = TC2; y1 = y2;
      end
      if TC1 >= TC
         done = true;
      end
   else
      done = true;
   end
end
y, TC
```

P-Median Location Problem

- Similar to UFL, except
  - Number of NF has to equal *p* (discrete version of ALA)
  - No fixed costs

$$p = \text{number of NFs}$$

$$Y^* = \arg\min_Y \left\{ \sum_{i \in Y} \sum_{j \in M_i} c_{ij} : \bigcup_{i \in Y} M_i = M, |Y| = p \right\}$$

$$
\begin{aligned}
&\textbf{procedure } \mathit{pmedian}(p, \mathbf{C}) \\
&Y \leftarrow \mathit{ufladd}(0, \mathbf{C}, \{\}, p) \\
&Y, TC \leftarrow \mathit{uflxchg}(0, \mathbf{C}, Y) \\
&Y' \leftarrow \mathit{ufldrop}(0, \mathbf{C}, \{\}, p) \\
&Y', TC' \leftarrow \mathit{uflxchg}(0, \mathbf{C}, Y) \\
&\textbf{if } TC' < TC \\
&\quad TC \leftarrow TC', Y \leftarrow Y' \\
&\textbf{endif} \\
&\textbf{return } Y, TC
\end{aligned}
$$

Bottom-Up vs Top-Down Analysis

- Bottom-Up: HW3 Q3

$$\boxed{r \begin{array}{l} \nearrow TC_{\mathrm{NEW}_1} \\ \searrow TC_{\mathrm{NEW}_2} \end{array}}$$

$$\mathbf{P}_{3\times 2} = \text{lon-lat of EFs}$$

$$\mathbf{f} = \left[48, 24, 35\right] \quad \text{(TL/yr)}$$

$$r = 2 \quad \text{(\$/TL-mi)}$$

$$g = \frac{1}{3}\left[\frac{d_{RD}(\mathbf{P}_1,\mathbf{P}_2)}{d_{GC}(\mathbf{P}_1,\mathbf{P}_2)} + \frac{d_{RD}(\mathbf{P}_2,\mathbf{P}_3)}{d_{GC}(\mathbf{P}_2,\mathbf{P}_3)} + \frac{d_{RD}(\mathbf{P}_3,\mathbf{P}_1)}{d_{GC}(\mathbf{P}_3,\mathbf{P}_1)}\right]$$

$$TC(\mathbf{x}) = \sum_{i=1}^{3} f_i\, r\, g\, d_{GC}(\mathbf{x},\mathbf{P}_i) \quad \text{(outbound trans. costs)}$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} TC(\mathbf{x})$$

$$TC^* = TC(\mathbf{x}^*)$$

$$\mathbf{x}^{\mathrm{cary}} = \text{lon-lat of Cary}$$

$$TC^{\mathrm{cary}} = TC(\mathbf{x}^{\mathrm{cary}})$$

$$\Delta TC = TC^{\mathrm{cary}} - TC^*$$

- Top-Down: estimate *r* (circuity factor cancels, so not needed, HW4 Q6)

$$\boxed{TC_{\mathrm{OLD}} \to r_{\mathrm{nom}} \to TC_{\mathrm{NEW}}}$$

$$TC^{\mathrm{cary}} = \text{current known } TC$$

$$10 \text{ ton/TL} = \text{known tons per truckload}$$

$$\mathbf{f} = \left[480, 240, 350\right] \quad \text{(ton/yr)}$$

$$r_{\mathrm{nom}} = \frac{TC^{\mathrm{cary}}}{\sum_{i=1}^{3} f_i\, d_{GC}(\mathbf{x}^{\mathrm{cary}},\mathbf{P}_i)} \quad \text{(\$/ton-mi)}$$

$$TC(\mathbf{x}) = \sum_{i=1}^{3} f_i\, r_{\mathrm{nom}} d_{GC}(\mathbf{x},\mathbf{P}_i)$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} TC(\mathbf{x})$$

$$TC^* = TC(\mathbf{x}^*)$$

$$\Delta TC = TC^{\mathrm{cary}} - TC^*$$