Metaheuristics for Network Design

Michel Gendreau
CIRRELT and MAGI
École Polytechnique de Montréal

Spring School on Supply Chain and Transportation Network Design
HEC Montréal – May 12-14, 2010

With the help of …

- Teodor Gabriel Crainic
- Judy Farvolden
- Ilfat Ghamlouche
- Geneviève Hernu
- Xiaorui Fu, Walter Rei, and Stein W. Wallace
- And several others…

Outline

- Network Design
  - Definitions, formulations, the need for heuristics
- Early Heuristics
- Path-based Tabu Search
- Cycle-based Tabu Search
- Path Relinking
- Scatter Search
- Progressive Hedging-based Metaheuristics for Stochastic Network Design
- Conclusion

Network Design

Network Design

- Given
  - Network
    - Existing + possible additions
    - Cost structure
    - Capacities
  - Demands

- Determine what part of the possible network to include to satisfy demand at minimum total cost.

Network Design Problems

- Decisions on adding
  - Nodes: Location
  - Arcs: Network design
  - Capacity: Network dimensioning, …
- Cost structure
  - Fixed costs (to install/open/offer/use …)
  - Variable costs (to operate)
  - Budget constraints, …
- Single or multi-commodity
- Capacitated or not, …

Network Design Problems

- Decisions on adding
  - Nodes: Location
  - Arcs: Network design
  - Capacity: Network dimensioning, …
- Cost structure
  - Fixed costs (to install/open/offer/use …)
  - Variable costs (to operate)
  - Budget constraints, …
- Single or multi-commodity
- Capacitated or not, …

Linear Arc-based Formulation

Minimize $z(x,y) = \sum_{(ij)\in A} f_{ij} y_{ij} + \sum_{(ij)\in A} \sum_{p\in P} c_{ij}^p x_{ij}^p$

Subject to $\sum_{j\in N} x_{ji}^p - \sum_{j\in N} x_{ij}^p = d_i^p$ Flow constraints

$\sum_{p\in P} x_{ij}^p \le u_{ij} y_{ij}$ Linking/Capacity

$x_{ij}^p \le u_{ij}^p y_{ij}$ constraints

$x_{ij}^p \ge 0$

$y_{ij} \in \{0,1\}$

$x \in X$

Flow Subproblem for given *y*

Minimize $z(x, \bar{y}) = \sum_{(ij)\in A} \sum_{p\in P} c_{ij}^p x_{ij}^p$

Subject to $\sum_{j\in N} x_{ji}^p - \sum_{j\in N} x_{ij}^p = d_i^p$

$\sum_{p\in P} x_{ij}^p \leq u_{ij}$ for $(i, j) \ni \bar{y}_{ij} = 1$

$x_{ij}^p \leq u_{ij}^p$ for $(i, j) \ni \bar{y}_{ij} = 1$

$x_{ij}^p \geq 0$

Flow Subproblem for given *y*

Minimize $z(x,\bar{y}) = \sum_{(ij)\in A} \sum_{p\in P} c_{ij}^{p} x_{ij}^{p}$

Subject to $\sum_{j\in N} x_{ji}^{p} - \sum_{j\in N} x_{ij}^{p} = d_i^{p}$

$\sum_{p\in P} x_{ij}^{p} \le u_{ij}$ for $(i, j) \ni \bar{y}_{ij} = 1$

$x_{ij}^{p} \le u_{ij}^{p}$ for $(i, j) \ni \bar{y}_{ij} = 1$

$x_{ij}^{p} \ge 0$

Capacitated, multi-commodity network flow problem!

Linear Path-based Formulation

Minimize $z(h, y) = \sum_{(ij)\in A} f_{ij} y_{ij} + \sum_{p\in P} \sum_{l\in L^p} k_l^p h_l^p$

Subject to

$$\sum_{l\in L^p} h_l^p = d^p \qquad ij \in P$$

$$\sum_{p\in P} \sum_{l\in L^p} h_l^p \delta_{ij}^{lp} \leq u_{ij} y_{ij} \qquad ij \in A$$

$$\sum_{l\in L^p} h_l^p \delta_{ij}^{lp} \leq u_{ij}^p y_{ij} \qquad p \in P, ij \in A$$

$$h_l^p \geq 0 \qquad p \in P$$

$$x_{ij}^p = \sum_{l\in L^p} h_l^p \delta_{(ij)l}^p \qquad k_l^p = \sum_{ij\in A} c_{ij}^p \delta_{(ij)l}^p$$

Difficult Problems

- Fixed costs $\leftrightarrow$ Capacities
- Multiple commodity paths
- Difficult (and degenerate) flow subproblems
- Large dimensions
- Large instances must be solved with heuristics

Early Heuristics

- Classical heuristics
  - Greedy approaches
  - "Add/drop" approaches
  - Arc swaps
    - Good for location, bad for design!
- We must come up with better ideas…

Tabu Search for Network Design

Tabu Search

- Proposed by Glover (1986), it is one of the most effective metaheuristics.
- TS can be seen as a generalization/extension of classical local search (improvement) methods :
  - A "current" solution is slowly modified with the goal of ultimately finding a very good solution.
  - The value of the objective function is allowed to degrade to escape local optima.
  - Short-term memory mechanisms prevent cycling.
  - Longer-term memories are used to guide the search in the solution space.

Tabu Search Template

1. Build an initial solution
2. Local search
   a. Find the best non-tabu solution in the neighborhood of the current solution
   b. Update
      - Best known solution and value (if appropriate)
      - Short, medium and long-term memories
   c. Check the stopping criterion: go to a or 3
3. Intensification
4. Diversification

Tabu Search Key Features (1)

- Intensification:
  - A more thorough exploration of areas of the search space that have been identified as “promising”.
  - Usually based on medium-term memories, e.g., recency memory.
  - Often involves “freezing” some components of the solution.

Tabu Search Key Features (2)

- Diversification:
  - A forced move to unexplored areas of the search space.
  - Usually based on long-term memories, e.g., frequency memory.
  - May be implemented as a separate phase or in a continuous fashion.
  - In many implementations, the most important feature besides the basic local search.

Failed Tabu Search Approach

- Applying TS with "add/drop" or swap neighborhoods.
- These neighborhoods are too limited to lead to a purposeful local search of the solution space.
- Even TS cannot help here!

Path-based Tabu Search (1)

- Crainic, Gendreau and Farvolden (2000)
- Approach based on a fixed cost vision of the problem.
- The key variables are the commodity flow variables; the design variables are simply obtained from flow variables.
- The neighborhood used relies on the path-based formulation of the problem.
- Local search moves correspond to pivots in the path-based formulation of the multi-commodity network flow problem in which all arcs are open.
- Fixed costs of arcs with positive flow are added to the cost of the solution.

Path-based Tabu Search (2)

- To avoid having to consider all possible paths, column generation is used to create the path flow variables.
  - Column generation is invoked only when a local optimum w.r.t. to the paths already generated.
  - Fixed costs are linearized when solving the column generation subproblem.
- Tabus are implemented by forbidding paths that have exited the basis to go back into it for a random number of iterations.
- Diversification is invoked after a set number of column generation phases without improvement of the objective.
- Diversification is induced by closing a small subset of often-used arcs for some time.

Path-based Tabu Search (3)

- The method was tested on sets of instances having:
  - between 20 and 100 nodes,
  - between 100 and 700 arcs,
  - between 10 and 400 commodities,
  - different fixed to variable costs ratios;
  - loose or tight capacity constraints.

- Overall assessment:
  - The method worked fine and was at the time the best available one by far, but some questions remained with respect to its performance on the larger instances…

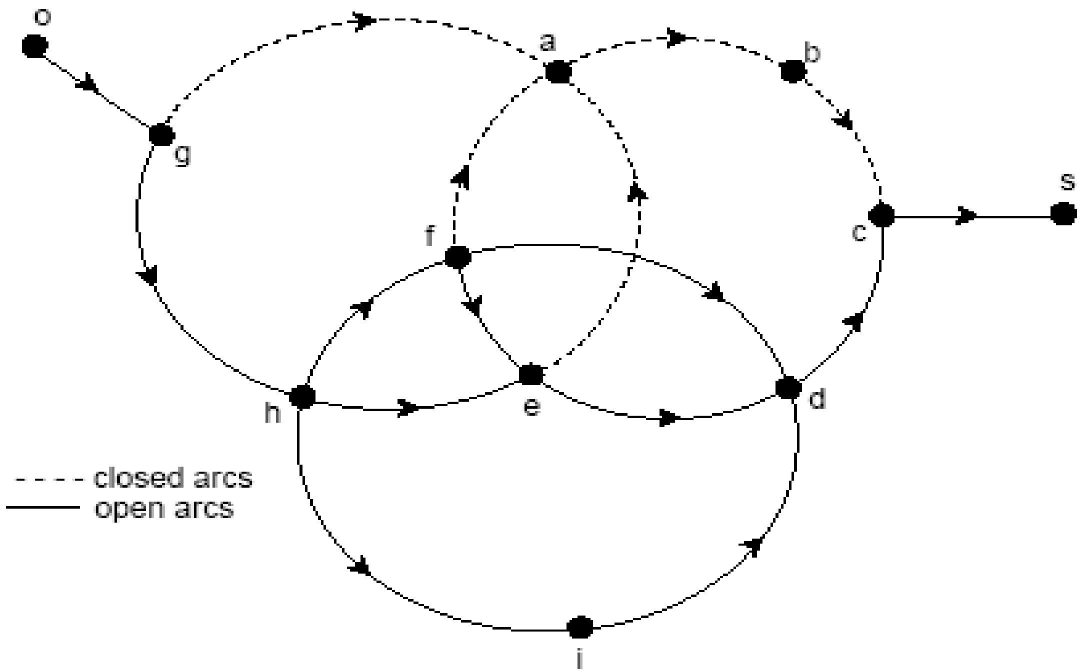
Cycle-based Tabu Search (1)

- The path-based TS had some limitations:
  - The way the neighborhood was defined implied that we only considered flow modifications of a single commodity at the time.
  - Too myopic for instances with several commodities.
- New neighborhood structures were required.
- We went back to a search based on design variables (y) → these variables completely define a solution, if one solves the associated minimum cost flow problem.
- Paper by Ghamlouche, Crainic and Gendreau (2003)

Cycle-based Neighborhood

- In a network design problem, in order to significantly modify a solution, one must be able to open and close sequences of arcs that make up subpaths.
- Flow movements take place along cycles in the residual graph.
- The new neighborhood is based on the identification of cycles in residual graphs of a given capacities.
- This allows for significant modifications of the current solution at each iteration:
  - several arcs / several products simultaneously.

Cycles



Cycle-based Tabu Search (2)

- At each iteration, the cycle-based neighborhood is partially examined by considering sets of candidates arcs as starting points for creating cycles.
- A labeling heuristic is used to identify low cost cycles containing a given arc.
- For each solution, once the cycle has been identified, the flow pattern is adjusted by solving exactly the associated MCNF problem.
  - Infeasible solutions may be found because commodities are aggregated in the determination of the best cycle → restoration might be needed.
- Fairly simple search strategy:
  - Intensification is invoked when very good solutions are obtained.
  - Implemented by modifying the flows of individual commodities.
  - Simple stopping criteria (elapsed CPU time or number of iterations)

Cycle-based Tabu Search (3)

- The method was tested on the same sets of instances as the path-based TS.
- The cycle-based TS outperformed the path-based one on 34 out of 43 instances:
  - Path-based was better only on 9 of the smaller, “easy” instances that can be solved optimally using B&B.
  - The optimality gap on instances with high fixed costs was reduced dramatically.

Path Relinking for Network Design

Path Relinking

- Another technique proposed by Glover.
- Basic idea:
  - explore in the solution space trajectories that link good known solutions with the objective of finding even better ones (intensification concept).
- The "good" (or "elite") solutions to be provided must be provided by some other method.
- Compromise between the quality and the diversity of the solutions considered.

Path Relinking for Network Design (1)

- Paper by Ghamlouche, Crainic and Gendreau (2004).
- Implementation based on the cycle-based TS.
- We tested several strategies for building the reference set of solutions to be considered and for selecting the initial and guiding the initial and the guiding solutions of trajectories.
- Trajectories are explored using the one-commodity-at-the-time variant of the cycle-based neighborhood.

Path Relinking for Network Design (2)

- Tested on the same instances as the other methods.
- Excellent results:
  - Path relinking improves the best solutions found by the cycle-based method for 37 instances out of 41!
  - Probably the best heuristic currently available for the capacitated, multi-commodity network design problem.

Scatter Search for Network Design

Scatter Search

- Yet another technique proposed by Glover (1977) for combining good known solutions for a problem.
- Basic idea:
  - linearly combine vectors describing solutions.
- We want to explore the space of the (binary) design variables → How can this concept be applied?!?

Scatter Search Template

1) Generate an initial population of diverse and good solutions.
2) Select a subset = Reference Set (RS).
3) Extract *L* solutions from RS = Candidate Set (CS).
4) Create new solutions by combining solutions in CS
   → Feasibility!
5) Improve the new solutions.
6) Update the Reference Set.

Scatter Search for Network Design

- Paper by Crainic and Gendreau (2007)
- We want to explore the space of the (binary) design variables

  → How can we apply the basic SS concepts ?!?
- We combine solutions by rounding some variables and fixing them to set values and letting others be free.

Initialization

- Initial population
  - Solutions generated by Cycle-based tabu search
- Reference Set initialized with improving local optima, i.e., local optima better than solutions already in RS.
- Candidate sets
  - Best solution in RS
  - The solution farthest away from it (Hamming distance)
  - Randomly selected solutions

Creating a Solution (2)

- Two thresholds are associated with arc selection

$$0 \le t_c \le t_o \le 1$$

- In the new solution
  - Close arc *(i,j)* when $0 \le m_{ij} < t_c$
  - Open arc *(i,j)* when $t_o < m_{ij} \le 1$
  - Arc *(i,j)* is undecided when $t_c \le m_{ij} \le t_o$

Creating a Solution

- L solutions in a Candidate Set.
- Compute a desirability factor for arc (i, j) as

$$m_{ij} = \sum_{l \in CS} w_l y_{ij}^l / \sum_{l \in CS} w_l \qquad (i, j) \in A$$

$$0 \leq m_{ij} \leq 1$$

- The weights w_l are particular to each mechanism.

Combination Mechanisms

- Voting (V): $w_l = 1$ for all l
- Distance (H): $w_l = 1$ / (Hamming distance between solution l and best solution)
- Cost (C): $w_l = 1$ / (cost difference between solution l and best solution)
- Frequency (F): w_l = frequency of arc (i,j) in best solutions

Processing a Solution

- The incomplete solution obtained from the combination operation is first sent to CPLEX with:
  - Closed arcs are given a 0 capacity.
  - Undecided arcs are given a $f_{ij}/u_{ij}+c_{ij}$ variable cost.
- TabuCycle is launched from the resulting solution
  - The procedure includes a mending phase for unfeasible solutions.
  - An intensification phase (moves involving a single commodity) may be called from the resulting solution before starting the tabu search proper.

Preliminary Computational Testing

- Tests conducted on a small (10) subset of problems.
- Experiment with 2 different values for t_c and t_o
  - $t_c = 0.25$, $t_o = 0.75$
  - $t_c = 0.40$, $t_o = 0.60$

Preliminary Computational Testing

- Tests conducted on a small (10) subset of problems.
- Experiment with 2 different values for t_c and t_o
  - $t_c = 0.25$, $t_o = 0.75$
  - $t_c = 0.40$, $t_o = 0.60$
- All further experiments ran with these values.

Experiments

- Set of 43 problem instances
  - 20 to 100 nodes
  - 100 to 700 arcs
  - Different fixed to variable cost ratios
  - Different capacity to demand ratios
- Test the
  - Four combination mechanisms
  - Four Candidate Set cardinality values = 2,3,4,5
  - Intensify or not at the start of TabuCycle
  - Different sizes for the Reference Set
- Same tabu search parameters used for path relinking.

Comparisons to Path Relinking (Gaps in %)

	L = 3		L = 4		L = 5	
	no intens	intens	no intens	intens	no intens	intens
(V)	-3.86 Min 6.27 Max 0.57 Avg	-4.20 Min 7.15 Max 0.30 Avg	-3.17 Min 9.09 Max 0.43 Avg	-2.92 Min 4.55 Max 0.40 Avg	-2.78 Min 9.90 Max 0.64 Avg	-3.24 Min 8.30 Max 0.53 Avg
(C)	-3.86 Min 7.87 Max 0.58 Avg	-4.10 Min 11.48 Max 0.74 Avg	-3.56 Min 8.64 Max 0.94 Avg	-3.73 Min 8.58 Max 0.64 Avg	-3.85 Min 7.59 Max 0.72 Avg	-3.71 Min 4.52 Max 0.65 Avg
(F)	-1.97 Min 10.35 Max 1.68 Avg	-2.03 Min 9.44 Max 1.46 Avg	-2.00 Min 5.14 Max 1.35 Avg	-2.61 Min 5.49 Max 1.62 Avg	-2.53 Min 8.85 Max 1.92 Avg	-2.59 Min 8.85 Max 1.88 Avg
(H)	-3.92 Min 5.65 Max 0.44 Avg	-3.48 Min 8.11 Max 0.66 Avg	-2.79 Min 8.54 Max 0.56 Avg	-3.43 Min 5.55 Max 0.65 Avg	-3.48 Min 10.74 Max 1.31 Avg	-1.78 Min 10.74 Max 1.11 Avg

Results

- Results with $L = 2$ are not competitive.
- Results obtained with less than 20 elements in the reference set are significantly poorer.

Discussion

- All combinations are capable of producing solutions better than those obtained with Path Relinking on some instances.
- On average, no combination can beat PR, but the best ones are getting close.
- Difficult to discriminate between inclusion or not of Intensification phase; since it requires computing effort, one may eventually abandon it.
- It seems that $L = 3$ or 4 is sufficient.

Progressive Hedging-based Metaheuristics for Stochastic Network Design

Network Design in Presence of Uncertainty

- When using a deterministic model, one assumes that all necessary information concerning the different parameters is readily available.
- This may not always be the case!
- Uncertainty is observed in practice:
  - Demands:
    - Uncertain Volumes
    - Uncertain Origins or Destinations
  - Costs
  - Arc Failures

Stochastic Programming Models

- One can account for uncertainty in the parameters of an optimization problem through stochastic programming models.
- Two or more decision stages:
  - 1st-stage decisions are made NOW, before uncertainty is resolved.
  - Decision in following stages are based on new information regarding the values of the uncertain parameters (recourse)
- Objective: Find the best decisions to be made NOW to minimize the sum of costs in the 1st stage and expected cost of recourse in later stages.

Stochastic Network Design

Problem considered:

- Uncertainty: Commodity volume demand
  - Stochastic demand vector ***d*** with given density function
  - Second stage random events: $\omega \in \Omega$
  - For a given realization ω, demand becomes known

$$d(\omega)_i^k, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}$$

- Problem has 2 decision stages:
  - 1^{st} Stage: Network design
  - 2^{nd} Stage: Operation decisions to satisfy demands

Formulation

$$\min \sum_{(i,j)\in\mathcal{A}} f_{ij}y_{ij} + \mathbf{E}_{\mathbf{d}}[Q(y,d(\omega))]$$

$$\text{s.t.} \quad y_{ij} \in \{0,1\},\ \forall (i,j) \in \mathcal{A}$$

where

$Q(y,d(\omega))$ = total distribution cost given the design decisions made (i.e., y) and the observed demands (i.e., $d(\omega)$).

$Q(y,d(\omega))$ is a performance measure on the operations performed

$Q(y,d(\omega)) \Rightarrow$ Recourse that is considered

Recourse Strategy

Assume $\mathcal{A} = \{\mathcal{A}^A, \mathcal{A}^D\}$

Where,

$\mathcal{A}^A$ = design arcs

$\mathcal{A}^D$ = dummy arcs linking each pair of origin and destination nodes for which demand is defined (i.e., external direct transportation service)

Distribution decisions:

- Use the network designed in Stage 1 as well as possible
- Call upon extra capacity (from the dummy arcs) at a (unit) price

2nd-stage Model

$$Q(y, d(\omega)) = \min \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}^+(i)} x_{ij}^k - \sum_{j \in \mathcal{N}^-(i)} x_{ji}^k = d(\omega)_i^k, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}$$

$$\sum_{k \in \mathcal{K}} x_{ij}^k \le u_{ij} y_{ij}, \ \forall (i,j) \in \mathcal{A}$$

$$x_{ij}^k \ge 0, \ \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}$$

One obtains:

Capacitated Multicommodity Minimum Cost Flow Problem

Recourse Revisited

- One could also consider adding capacity to the arcs to handle capacity if needed.
- This yields a more complex and more difficult 2^{nd}-stage problem.

Monte Carlo Sampling

Given the complexity of computing $\mathbf{E}_{\mathbf{d}}[Q(y, d(\omega))]$

Monte Carlo Sampling may be applied:

- $\mathcal{S}$ = set of representative scenarios of the random event

Model becomes:

$$\min \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} + \sum_{s\in\mathcal{S}} p^s \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c_{ij}^k x_{ij}^{ks}$$

s.t. $$\sum_{j\in\mathcal{N}^+(i)} x_{ij}^{ks} - \sum_{j\in\mathcal{N}^-(i)} x_{ji}^{ks} = d_i^{ks}, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}$$

$$\sum_{k\in\mathcal{K}} x_{ij}^{ks} \leq u_{ij} y_{ij}, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S}$$

$$y_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{A}$$

$$x_{ij}^{ks} \geq 0, \ \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}$$

Monte Carlo Sampling

For all $s \in \mathcal{S}$:

- $x_{ij}^{ks}, \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}$: recourse decisions that are taken if scenario s is observed
- $d_i^{ks}, \forall i \in \mathcal{N}, \forall k \in \mathcal{K}$: demand values if scenario s is observed

Observations:

- Previous model is deterministic
- Very large:
  - Bloc structure (i.e., $|\mathcal{S}|$ blocs)
- Solution approaches $\Rightarrow$ decomposition techniques

Scenario Decomposition

Define $y_{ij}^s \in \{0,1\}$, $\forall (i,j) \in \mathcal{A}$ and $\forall s \in \mathcal{S}$

Non-anticipativity Constraints:

- $y_{ij}^s = y_{ij}^t$, $\forall s,t \in \mathcal{S}$ and $s \neq t$

Model:

$$\min \sum_{s\in\mathcal{S}} p^s \left(\sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij}^s + \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} c_{ij}^k x_{ij}^{ks} \right)$$

$$\text{s.t.} \sum_{j\in\mathcal{N}^+(i)} x_{ij}^{ks} - \sum_{j\in\mathcal{N}^-(i)} x_{ji}^{ks} = d_i^{ks}, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}$$

$$\sum_{k\in\mathcal{K}} x_{ij}^{ks} \leq u_{ij} y_{ij}^s, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S}$$

$$y_{ij}^s = y_{ij}^t, \ \forall s,t \in \mathcal{S}, \ s \neq t$$

$$y_{ij}^s \in \{0,1\}, \ \forall (i,j) \in \mathcal{A}, \ \forall s \in \mathcal{S}$$

$$x_{ij}^{ks} \geq 0, \ \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}$$

Scenario Decomposition

If Lagrangean relaxation is applied on the non-anticipativity constraints

$$\Downarrow$$

Problem becomes scenario separable

Following this strategy, one must solve:

- $|\mathcal{S}|$ network design problems (i.e. one for each scenario subproblem)

¿From this approach $\Rightarrow$ lower bounds on the original problem can be obtained using the Lagrange Dual

Previous work:

- Carøe and Schultz (1999): Branch and Bound method

Scenario Decomposition

Other approach ⇒ The Progressive Hedging (PH) Strategy

Method proposed by Rockafellar and Wets (1991)

- Continuous variables
- Multistage case
- Convergence results

Applications to integer problems:

- Løkketangen and Woodruff (1996)
  - (0,1) Multistage problems
- Haugen, Løkketangen and Woodruff (2001)
  - Stochastic lot-sizing problem

Scenario Decomposition

Redefine the Non-anticipativity Constraints:

- $y_{ij}^s = \overline{y}_{ij}, \forall (i,j) \in \mathcal{A}$ and $\forall s \in \mathcal{S}$
- $\overline{y}_{ij} \in \{0,1\}, \forall (i,j) \in \mathcal{A}$

Note:

- $\overline{y}_{ij}, \forall (i,j) \in \mathcal{A}$: overall design (i.e., a reference solution)

By applying an augmented Lagrangean strategy:

$$\min \quad \sum_{s\in\mathcal{S}} p^s \left(\sum_{(i,j)\in\mathcal{A}} (f_{ij} + \lambda_{ij}^s - \rho\overline{y}_{ij} + \tfrac{\rho}{2}) y_{ij}^s + \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} c_{ij}^k x_{ij}^{ks} \right)$$

$$- \sum_{(i,j)\in\mathcal{A}} \lambda_{ij}^s \overline{y}_{ij} + \sum_{(i,j)\in\mathcal{A}} \frac{1}{2}\rho\overline{y}_{ij}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}^+(i)} x_{ij}^{ks} - \sum_{j\in\mathcal{N}^-(i)} x_{ji}^{ks} = d_i^{ks}, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}$$

$$\sum_{k\in\mathcal{K}} x_{ij}^{ks} \le u_{ij} y_{ij}^s, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S}$$

$$y_{ij}^s \in \{0,1\}, \ \overline{y}_{ij}^s \in \{0,1\}, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S}$$

$$x_{ij}^{ks} \ge 0, \ \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}.$$

Scenario Decomposition

Observation:

- When $\overline{y}$ is fixed $\Rightarrow$ Problem becomes scenario seperable

Scenario subproblems obtained ($\forall s \in \mathcal{S}$):

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in\mathcal{A}} \left(f_{ij} + \lambda_{ij}^{s} - \rho\overline{y}_{ij} + \tfrac{\rho}{2}\right) y_{ij}^{s} + \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} c_{ij}^{k} x_{ij}^{ks} \\
\text{s.t.} \quad & \sum_{j\in\mathcal{N}^{+}(i)} x_{ij}^{ks} - \sum_{j\in\mathcal{N}^{-}(i)} x_{ji}^{ks} = d_{i}^{ks}, \ \forall i \in \mathcal{N}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S} \\
& \sum_{k\in\mathcal{K}} x_{ij}^{ks} \leq u_{ij} y_{ij}^{s}, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S} \\
& y_{ij}^{s} \in \{0,1\}, \ \forall (i,j) \in \mathcal{A}, \forall s \in \mathcal{S} \\
& x_{ij}^{ks} \geq 0, \ \forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{K}, \forall s \in \mathcal{S}.
\end{aligned}$$

These subproblems are just fixed-charge, multicommodity network design problems with modified costs!

Solution Approach - Motivation

Basic idea:

Solve the scenario subproblems	$\Rightarrow$	Obtain design $\hat{y}^s$
Update the reference point	$\Rightarrow$	new $\overline{y}$
Update the penalties	$\Rightarrow$	differences between $\hat{y}^s$ and $\overline{y}$

Advantages:

- General structure to find consensus between $\hat{y}^s, \forall s \in \mathcal{S}$
- Efficient solution strategies for deterministic CMND (heuristics and exact)

Challenges:

- Finding and updating $\overline{y}$?
- Penalty adjustments ?
- Solving the subproblems?

Solution Approach

General Structure

Description:

- Phase 1: search for consensus using the progressive hedging strategy
- Phase 2: restrict the search space and complete the design

Obtaining consensus

- Defining the overall design

  Aggregation operator, at iteration ν:

$$\overline{y}_{ij}^{\nu} = \sum_{s \in \mathcal{S}} p_s y_{ij}^{s\nu}, \quad \forall (i,j) \in \mathcal{A}$$

Observations:

- $\overline{y}$ is not necessarily a feasible design
- $\overline{y}$ gives a trend

Solution Approach

- Obtaining consensus (cont'd)
  - Defining the overall design (cont'd)

    Obtaining a feasible design *max design*:

$$y_{ij}^{M\nu} = \bigvee_{s \in \mathcal{S}} y_{ij}^{s\nu}, \quad \forall (i,j) \in \mathcal{A}$$

  - Strategies for penalty adjustments

    For iteration ν and $\forall s \in \mathcal{S} \Rightarrow \min \sum_{(i,j)\in\mathcal{A}} f_{ij}^{\nu s} y_{ij}^{s} + \sum_{(i,j)\in\mathcal{A}} \sum_{k\in\mathcal{K}} c_{ij}^{k} x_{ij}^{ks}$

    Progressive Hedging:

    - $\lambda_{ij}^{s\nu} \leftarrow \lambda_{ij}^{s\nu-1} + \rho^{\nu-1}(y_{ij}^{s\nu} - \overline{y}_{ij}^{\nu-1})$, $\forall (i,j) \in \mathcal{A}$ and $\forall s \in \mathcal{S}$
    - $\rho^{\nu} \leftarrow \alpha\rho^{\nu-1}$
    - $f_{ij}^{s\nu} \leftarrow f_{ij} + \lambda_{ij}^{s\nu-1} - \rho^{\nu-1}\overline{y}_{ij}^{\nu-1} + \frac{\rho^{\nu-1}}{2}$, $\forall (i,j) \in \mathcal{A}$

Solution Approach

- Obtaining consensus (cont'd)
  - Strategies for penalty adjustments (cont'd)
    Heuristic Cost Adjustment:
    Global adjustments:

$$f_{ij}^{\nu} = \begin{cases} \beta f_{ij}^{\nu-1} & \text{if } \overline{y}_{ij}^{\nu-1} < c^{low} \\ \frac{1}{\beta} f_{ij}^{\nu-1} & \text{if } \overline{y}_{ij}^{\nu-1} > c^{high} \\ f_{ij}^{\nu-1} & \text{otherwise,} \end{cases}$$

Where, $\beta > 1$, $0 < c^{low} < 0.5$ and $0.5 < c^{high} < 1$

Local adjustments:

$$f_{ij}^{s\nu} = \begin{cases} \beta f_{ij}^{\nu} & \text{if } |y_{ij}^{s\nu-1} - \overline{y}_{ij}^{\nu-1}| \geq c^{far} \text{ and } y_{ij}^{s\nu-1} = 1 \\ \frac{1}{\beta} f_{ij}^{\nu} & \text{if } |y_{ij}^{s\nu-1} - \overline{y}_{ij}^{\nu-1}| \geq c^{far} \text{ and } y_{ij}^{s\nu-1} = 0 \\ f_{ij}^{\nu} & \text{otherwise,} \end{cases}$$

Where, $0.5 < c^{far} < 1$

Solution Approach

- Complete Design

  Stop Phase 1 when either:

  - Max number of iterations
  - Max number of iterations without improvement
  - Max time is reached

  Phase 2:

  - Fix status for consensus arcs
  - Solve (using CPLEX) restricted problem

  Implementation:

  - Cycle-based tabu search: Ghamlouche, Crainic and Gendreau 2003
  - Parallel (master-slave synchronous strategy)

Computational Results

- Problem set **S**
  Created by Hoff, Lium, Løkkentagen and Crainic, 2007

Problems			CPLEX				Tabu-PH			Tabu-HC		
$\|\mathcal{N}\|$	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	L.B.	U.B.	Gap	Time	Val.	Gap	Time	Val.	Gap	Time
16	14	10	4909.3	4909.3	0.00%	0.19	4909.3	0.00%	0.34	4909.3	0.00%	0.34
16	14	20	4990.1	4990.1	0.00%	0.79	4990.1	0.00%	0.66	4990.1	0.00%	0.65
30	14	10	5198.6	5198.6	0.00%	0.18	5198.6	0.00%	2.69	5198.6	0.00%	2.58
30	14	20	5218.6	5218.6	0.00%	0.43	5218.6	0.00%	5.96	5218.6	0.00%	5.88
16	40	20	15184.9	15184.9	0.00%	76.16	15243.1	0.38%	4.07	15243.1	0.38%	3.79
16	40	60	15112.8	15244.7	0.87%	600.11	15196.3	0.55%	11.45	15196.3	0.55%	12.40
16	40	90	15103.9	15204.8	0.67%	600.32	15194.3	0.60%	18.53	15194.3	0.60%	20.28
30	40	20	14056.5	14301.0	1.74%	600.17	14498.9	3.15%	133.65	14321.9	1.89%	132.61
30	40	60	13409.3	14723.1	9.80%	600.48	14350.1	7.02%	199.98	14317.9	6.78%	201.96
30	40	90	12787.0	14723.0	15.14%	600.81	14321.4	12.00%	230.03	14287.8	11.74%	232.61
16	80	20	26773.0	27167.5	1.47%	600.23	27464.4	2.58%	5.71	27359.9	2.19%	13.25
16	80	60	26330.8	28621.4	8.70%	600.51	27272.0	3.57%	216.30	27190.2	3.26%	513.33
16	80	90	25709.6	28621.1	11.32%	600.55	27347.4	6.37%	522.49	27371.2	6.46%	524.64
30	80	20	29303.9	31408.3	7.18%	600.46	31010.6	5.82%	217.06	30913.5	5.49%	217.28
30	80	60	27491.8	31412.7	14.26%	600.86	30874.2	12.30%	317.47	30829.7	12.14%	346.34
30	80	90	27473.0	31412.4	14.34%	1201.78	30704.5	11.76%	287.27	30627.4	11.48%	312.68
Average			16815.82	18021.34	5.34%	455.25	17737.11	4.13%	135.85	17698.11	3.94%	158.79

Computational Results

Problem set **R**

Created by Ghamlouche, Crainic and Gendreau, 2003

Attributes of the instances:

Name	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|\mathcal{K}\|$
r04	10	60	10
r05	10	60	25
r06	10	60	50
r07	10	82	10
r08	10	83	25
r09	10	83	50
r10	20	120	40
r11	20	120	100

- 5 different combined levels of fixed cost and capacity ratios
- 3 different levels of positive correlations
- $|\mathcal{S}|$ = 16, 32 and 64

A total of 360 instances

Computational Results

Problem set **R** (cont'd)

General observations for problem difficulty:

- Correlations have very little impact over problem difficulty
- Increasing $|S|$ makes the instances much more difficult

General observations for solution methods:

- Great variability on running times for CPLEX
- Local adjustments have a very limited impact on the quality of the heuristics
- Heuristics are very efficient to solve these problems:
  - Same quality of results as CPLEX but using much less time
  - Better results for comparable solution times
- Performance is quite similar for PH vs HC (slight advantage for HC)

Computational Results

- Problem set **R** (cont'd)

  Aggregated results set **R**: Corr: 0.2, CPLEX vs. Hedging strategies (without local adjustments)

		CPLEX					Tabu-PH			Tabu-HC		
ratio	$\lvert\mathcal{S}\rvert$	L.B.	U.B.	Sol.	Gap	Time	Val.	Gap	Time	Val.	Gap	Time
	16	157719	158019	7	0.06%	63.64	158111	0.09%	5.75	158523	0.22%	6.86
1	32	157021	168161	7	2.07%	66.30	158258	0.23%	47.94	158355	0.28%	27.80
	64	151659	168563	7	3.39%	83.43	167127	3.11%	67.85	158083	1.32%	66.99
	16	371541	524263	5	13.67%	244.48	413942	4.28%	88.16	413942	4.28%	78.99
3	32	354945	544883	4	20.68%	312.54	492793	13.46%	222.55	491877	13.18%	198.99
	64	310325	577554	3	41.22%	359.40	504508	25.67%	269.31	505724	26.00%	269.35
	16	332082	392174	5	5.67%	265.79	346561	2.05%	121.171	346561	2.05%	118.10
5	32	314460	413412	4	13.06%	291.18	386151	7.52%	227.087	386151	7.52%	227.15
	64	286764	428767	4	24.53%	375.06	402844	17.81%	263.033	402844	17.81%	263.06
	16	351995	352271	7	0.08%	81.32	352370	0.19%	30.13	352370	0.19%	18.22
7	32	355578	355929	7	0.10%	103.34	355917	0.10%	70.66	355917	0.10%	70.87
	64	353454	356607	7	0.95%	141.87	354837	0.42%	90.84	354857	0.46%	92.15
	16	859958	870762	6	1.08%	204.11	870750	1.05%	66.19	870750	1.05%	39.87
9	32	854934	884663	4	3.11%	291.18	875958	2.19%	127.53	875972	2.40%	140.12
	64	832381	892986	3	6.37%	387.23	883504	5.75%	248.03	883305	5.73%	250.13

Conclusion

Conclusion

- We have a large set of approaches for tackling network design problems.
- For the deterministic problem, the best approach currently is probably Path Relinking, but one might possibly be able to still improve upon it.
- Much work remains to be done on stochastic variants!