Heuristics for Solving a Partial Interdiction Problem with Capacitated Facilities and Demand Outsourcing

Deniz Aksen, Koç University, İstanbul, Turkey

İlknur Çoğal, Boğaziçi University, İstanbul, Turkey

Necati Aras, Boğaziçi University, İstanbul, Turkey

Vocational Training Center of Volos

26 June 2014
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation
- Recent Solution Procedures
  - Progressive Grid Search (PGS)
  - Multi-start Revised Simplex Search (MSS)
- New Solution Procedures
  - Fully Informed Particle Swarm Optimization (FIPSO)
  - Electromagnetism-like Algorithm (EMLA)
- Computational Results
- Conclusions
Introduction to BPIP

- Static Stackelberg game between two actors:

  **System Planner (Follower)**
  - minimizes the total demand-weighted transportation cost.
  - responsible for satisfying the overall demand.
  - considers possible capacity reductions.

  **Attacker (Leader)**
  - causes maximum disruption in the service level.
  - forces the system planner to meet customer demands with a higher cost due to outsourcing.

- Fits the Bilevel Programming (BP) framework.
Introduction to BPIP

- Static Stackelberg game between two actors:

  System Planner (Follower)  Attacker (Leader)

- Fits the Bilevel Programming (BP) framework.

- Recently developed solution methods for BPIP:
  - Progressive Grid Search (PGS) (not practical on large size problems)
  - Multi-start Revised Simplex Search (MSS) (developed to overcome the exponential time complexity of PGS)

- Newly tried solution methods for BPIP:
  - Fully Informed Particle Swarm Optimization (FIPSO) (faster than MSS only in certain instances, but not as good as MSS)
  - Electromagnetism-like Algorithm (EMLA) (significantly faster, and also slightly better than MSS)
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation
- Recent Solution Procedures
  - Progressive Grid Search (PGS)
  - Multi-start Revised Simplex Search (MSS)
- New Solution Procedures
  - Fully Informed Particle Swarm Optimization (FIPSO)
  - Electromagnetism-like Algorithm (EMLA)
- Computational Results
- Conclusions
Motivation

- New capabilities for attack and the improvements in the efficiency of the existing methods

- More sophisticated, better planned and coordinated strikes

- Soft targets
  - An ambulance station in North Ireland (1999)
  - An electric grid (Motto et al., *IEEE Trans Power Sys* 2005)
  - Telecommunication towers in Afghanistan (2008)
  - Pipelines on the Kirkuk-Yumurtalık Crude Oil Transfer Network between Iraq and Turkey
    - 37 bomb attacks alone in July and August 2013, cutting the oil flow in half.
    - Largest combo attack on 21 Aug 2013 totally paralysing the oil flow for 4 days.

growth in the number of terrorist attacks
Motivation

- Harm to the public
  - More than 40 large scale terrorist attacks and Iraq attributed to Al-Qaeda
Availability of public information about critical infrastructure

- An adversary's ability to collect information about an infrastructure system and use that information to identify weak spots in the system's architecture.

- Al Qaeda training manual (Department of Justice, 2004) advises: "Using public sources openly and without resorting to illegal means, it is possible to gather at least 80% of information about the enemy.

- "One can often find all the information necessary to plan a disruptive attack on an infrastructure system."

- 2D and even 3D visual aids publicly available in Google Maps / Google Street View and Google Earth.
Motivation

- Partial disruption
  - Addressed only once before in the literature.
  - The facilities may operate at a reduced service capacity following an attack.

- Bilevel programming model and new solution methods proposed to guide system planners and policy makers.
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation
- Recent Solution Procedures
  - Progressive Grid Search (PGS)
  - Multi-start Revised Simplex Search (MSS)
- New Solution Procedures
  - Fully Informed Particle Swarm Optimization (FIPSO)
  - Electromagnetism-like Algorithm (EMLA)
- Computational Results
- Conclusions
Facility Interdiction Problems

- The identification of critical system components
  - Their failures result in the deterioration of the system performance
- The effects of disorders can be mitigated if they are considered in the design phase of the network
- Models on intentional (malicious) attacks through the use of interdiction
- Interdiction models differ in the objective
- \textit{Wollmer (OR 1964)} \rightarrow Interdiction of supply lines, removing of arcs
  \rightarrow Minimizing the network flow capacity
  \rightarrow Maximizing the shortest path between origin and destination
- Facility Interdiction models \rightarrow Facilities (nodes) are potential targets

\textbf{Note that}

Intentional attacks are different from the natural catastrophic events. Attackers intelligently hit the vital system components.
Interdiction Problems for Facility Locations

- $r$-interdiction covering problem ($\text{RIC}$). (Church et al., 2004)
- $r$-interdiction median problem ($\text{RIM}$). (Church et al., 2004)
- Interdiction median problem with fortification ($\text{IMF}$) (Church et al., 2007)
  - Maximal covering model with precedence constraints ($\text{MCPC}$). (Scaparra and Church, 2008a)
  - $r$-interdiction median problem with fortification ($\text{RIMF}$). (Scaparra and Church, 2008b)
- Budget constrained RIMF with capacity expansion. (Aksen et al., 2010)
- A bilevel fixed charged location problem. (Aksen et al., 2011)
- The fortification $r$-interdiction median problem with facility recovery time. (Losada et al., 2012)
Interdiction Problems for Facility Locations

- Protecting supply systems to mitigate potential disaster: a model to fortify capacitated facilities. *(Scaparra and Church, KBS Working Paper 2010; IRSR 2012)*

- Hedging against disruptions with ripple effects in location analysis *(Liberatore, Scaparra and Daskin, OMEGA 2012)*

  - A trilevel capacitated facility fortification-interdiction problem with full interdiction and partial collateral damages of a deterministic degree.

- A bilevel partial interdiction problem (BPIP) with capacitated facilities and demand outsourcing *(Aksen, Akca and Aras, Comp.&OR 2014)*
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation
- Recent Solution Procedures
  - Progressive Grid Search (PGS)
  - Multi-start Revised Simplex Search (MSS)
- New Solution Procedures
  - Fully Informed Particle Swarm Optimization (FIPSO)
  - Electromagnetism-like Algorithm (EMLA)
- Computational Results
- Conclusions
Problem Formulation

System Planner

- Has perfect information

Attacker

- Leader who decides what facilities to disrupt at what fraction
- Tries to reduce capacity of the facilities
- Has limited budget
Problem Formulation

Attacker
- Leader who decides what facilities to disrupt at what fraction
- Tries to reduce capacity of the facilities
- Has limited budget
- Has perfect information

System Planner
- Follower who concerns the post-attack operation efficiency
- Assigns customers to surviving facilities regarding their remaining capacities
- Minimizes the unsatisfied demand
- Has perfect information
Bilevel Programming

- What is Bilevel Programming?
  - Optimization problems where one optimization problem is embedded in another one.

\[
\begin{align*}
\text{Max} & \quad F(x,y) \\
& x \in X \\
\text{s.t.} & \quad G(x,y) = B \\
\text{where} \quad Y \text{ solves:} \\
\text{Min} & \quad f(x,y) \\
& y \in Y \\
\text{s.t.} & \quad g(x,y) = b
\end{align*}
\]
Problem Formulation

Parameters:

\[ a_i = \text{demand of customer } i \]
\[ c_i = \text{cost of shipping customer } i\text{'s unit demand per unit distance} \]
\[ d_{ij} = \text{Euclidean distance between customer } i \text{ and facility } j \]
\[ c_p = \text{cost of outsourcing customer } i\text{'s unit demand (independent of distance)} \]
\[ e_j = \text{cost of interdicting full capacity of facility } j \]
\[ e_{tot} = \text{total interdiction budget of the attacker} \]
\[ q_j = \text{capacity of facility } j \]

Decision Variables:

\[ V_{ij} = \begin{cases} 
1 & \text{if customer } i \text{ is assigned to facility } j \text{ after attack} \\
0 & \text{otherwise} 
\end{cases} \]
\[ S_j = \text{fraction of facility } j\text{'s capacity lost due to attack} \]
Problem Formulation

Leader’s problem

\[
\max \sum_{s} Z_{att} = \sum_{i \in I} \sum_{j \in J} a_i c_i d_{ij} V_{ij} + c_p \sum_{i \in I} a_i (1 - \sum_{j \in J} V_{ij})
\]

s.t.: Budget constraint

Defender’s problem

\[
\min \sum_{V} Z_{def} = \sum_{i \in I} \sum_{j \in J} a_i c_i d_{ij} V_{ij} + c_p \sum_{i \in I} a_i (1 - \sum_{j \in J} V_{ij})
\]

s.t.: Capacity constraints

Assignment constraints
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation

Recent Solution Procedures
- Progressive Grid Search (PGS)
- Multi-start Revised Simplex Search (MSS)

New Solution Procedures
- Fully Informed Particle Swarm Optimization (FIPSO)
- Electromagnetism-like Algorithm (EMLA)

Computational Results

Conclusions
Progressive Grid Search (PGS)

- Divides (dices) the search space into equidistant intervals (hypercubes)
- Evaluates the objective value at each point
- Exponential growth in the number of the function evaluations
- Prohibitively long computational times
- Consists of three steps:

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>K = 0.2</td>
<td>K = 0.1</td>
<td>K = 0.05</td>
</tr>
<tr>
<td>D = u_j − l_j = 1</td>
<td>D’ = 2K</td>
<td>D’ = 2K</td>
</tr>
<tr>
<td>6^m support points</td>
<td>l_j = S*_j − 0.2</td>
<td>l_j = S’*_j − 0.1</td>
</tr>
<tr>
<td>S*: best support point of Step-1</td>
<td>u_j = S*_j + 0.2</td>
<td>u_j = S’*_j + 0.1</td>
</tr>
<tr>
<td>5^m support points</td>
<td>S’*: best support point of Step-2</td>
<td>5^m support points</td>
</tr>
<tr>
<td>S’’*: best support point of Step-3 and the solution of PGS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Grid Search in 1D Space

Step 1

Step 2

Step 3
Grid Search in 2D Space
Nelder and Mead Simplex Search (NMSS)

- Direct search method
  - The derivative of the function is difficult or impossible to evaluate
  - The gradient of the function is not available
  - Estimation of the values via calculation at the several points

- Generating a sequence of changing simplices

- Traverse the solution space

- The simplex search method of Spendley, Hext and Himsworth (1962)

- Further developed by Nelder and Mead (1965)

- Four main construction operations.
NMSS

- Gurson (2000) → Small ambiguities are solved
- Brooks (2000) → Problems in NMSS
- Wolff (2004) → Simple bounds and constraints
- Luersen et al. (2004) → Restart option

Having capacitated facilities and budget constraint → many complications in the application of NMSS
Nelder-Mead Simplex search over Himmelblau function

(c) P.A. Simionescu 2006
Premature Convergence

- NMSS is susceptible to premature termination at a local optimum
- An effective action is to restart the search
- Revised Simplex Search (RSS) of Humphrey and Wilson (2000)
  - The search is restarted from the ending values of the previous phase
  - The size of the initial simplex of a new phase is reduced linearly
  - The final estimated optimum is the best ending value among the solution of the phases
Multi-start Revised Simplex Search (MSS)

- The RSS algorithm is modified
- MSS is started $K$ times from different points in the search space
- The best solution of the each start is cached
- The final estimated optimum is the best ending value of the $K$ runs

- Initial simplex generation: excessive sensitivity to starting values
  - Deterministic initial vertex generation
  - Randomization scheme for initial vertex generation
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation

Recent Solution Procedures
- Progressive Grid Search (PGS)
- Multi-start Revised Simplex Search (MSS)

New Solution Procedures
- Fully Informed Particle Swarm Optimization (FIPSO)
- Electromagnetism-like Algorithm (EMLA)

Computational Results

Conclusions
Particle Swarm Optimization (PSO)

- Particle Swarm Optimization (PSO) is a stochastic and population-based technique developed by Kennedy and Eberhart (1995).

- Evolutionary computation technique inspired by the movement of organisms in a flock of birds / school of fish → swarm intelligence.

- Does not require complex mathematical equations.

- Each individual represents a solution in the population (swarm) and moves in the search space with a velocity:

  Velocity of the particle adjusted using its current and best positions as well as the best position of all particles so far → Standard PSO
Particle Swarm Optimization (PSO)

- Velocity update and position equations:

\[ v_i(t+1) = w v_i(t) + c_1 r_1(p_i(t) - x_i(t)) + c_2 r_2(p_{best}(t) - x_i(t)) \]
\[ x_i(t+1) = x_i(t) + v_i(t+1) \]

- \( x_i \) current position of particle \( i \)
- \( v_i \) current velocity of particle \( i \)
- \( p_i \) previous best position of particle \( i \)
- \( p_{best} \) best remembered position in the population
- \( w \) inertia weight
- \( c_1, c_2 \) cognitive and social parameters
- \( r_1, r_2 \) random number between 0 and 1
- \( t_{max} \) maximum number of iterations
- \( k \) number of particles.
Fully Informed Particle Swarm Optimization (FIPSO)

- Fully Informed Particle Swarm (FIPS)) is a variant of PSO proposed by Mendes (2002).
- Social influence comes from the group norm.
- The particle is affected by all its neighbors.
- The velocity update and position equations change as:

\[
v_i(t+1) = w v_i(t) + \frac{1}{k_i} \sum_{n=1}^{k_i} c_i r (p_{i,n}(t) - x_i(t))
\]
\[
x_i(t+1) = x_i(t) + v_i(t+1)
\]

- \( k_i \) the number of neighbors of particle \( i \).
- \( p_{i,n} \) the position of the \( n \)th neighbor of particle \( i \).
- \( r \) random number between 0 and 1.
Electromagnetism Like Algorithm (EMLA)

- Proposed first by İlker Birbil and Shu-Cherng Fang (*JGO* 2003)
- A population-based metaheuristic using an attraction-repulsion mechanism to move the points (particles) to obtain a near-optimal solution.
- Every point (particle) charged with electrons and protons represents a solution.
- The charge of a point is related to the objective function value of that point, and sets the magnitude of attraction or repulsion of the point.
- Better solutions are charged more than the others.
- After calculating the charges of the particles, they need a direction to move towards the better areas of the search space.
Electromagnetism Like Algorithm (EMLA)

\[ f(x_2) > f(x_3) \]
\[ f(x_3) > f(x_1) \]

I don't talk to the other protons.
Electromagnetism Like Algorithm (EMLA)

- The magnitude and direction of the total force exerted on each solution point (particle) are evaluated by using the interactions (charges) among the particles.

\[
q_i = \exp \left( -m \frac{f(x_i) - f(x_{\text{best}})}{\sum_{j=1}^{k}(f(x_j) - f(x_{\text{best}}))} \right) \quad \forall i
\]

- In a maximization problem:

\[
F_i = \sum_{i \neq j}^{k} \begin{cases} 
(x_j - x_i) \frac{q_i q_j}{||x_j - x_i||^2} & \text{if } f(x_j) > f(x_i) \\
(x_i - x_j) \frac{q_i q_j}{||x_j - x_i||^2} & \text{if } f(x_j) \leq f(x_i)
\end{cases}, \forall i
\]

- CASE I: A point with a better objective value attracts the other one.
- CASE II: A point with a worse objective value repels the other one.
Electromagnetism Like Algorithm (EMLA)

Initialize parameters and variables
1. Randomly initialize $k$ points.
2. Compute the objective function values of all the points.

Charge Calculation
Compute charges of all points in the population.

Force Calculation
1. Using computed charges, calculate the forces applied on the points.
2. Compute the net applied force for all the points.

\[ \begin{align*}
\vec{F}_i^c &= \frac{F_i^c}{\|F_i^c\|} \\
q_i &= \exp(-m \frac{f(x_i) - f(x_{best})}{\sum_{z=1}^{k} (f(x_z) - f(x_{best}))})
\end{align*} \]

$\lambda \in U[0, 1]$ is a random step length;
$u$ is the upper bound vector, and $l$ is the lower bound vector of $x$.

\[ \begin{align*}
x_i &= \begin{cases} 
    x_i + \lambda \vec{F}_i^c (u - x_i) & \text{if } F_i^c > 0 \\
    x_i + \lambda \vec{F}_i^c (x_i - l) & \text{otherwise}
  \end{cases} \\
& \quad i \in [1, k] \end{align*} \]

Report the point with the best objective function value.
Handling the budget (hyperplane) constraint:

- Budget Adjustment in EMLA
  - The total interdiction budget of the attacker may not be fully utilized;

\[ e_{tot} > \sum_{j \in J^l} e_j S_j \text{ or } e_{tot} < \sum_{j \in J^l} e_j S_j \]

If there is an overutilization

- decrease \( S_j \) variables iteratively starting from the facility with the highest cost of interdicting its full capacity

If there is an underutilization

- increase \( S_j \) variables iteratively starting from the facility with the lowest cost of interdicting its full capacity
Handling the budget (hyperplane) constraint:

for each particle do
  if \((e_{used} - e_{tot}) < 0\) then
    Sort \(e_j\) as ascending order
    for each facility \(j\) beginning one with the smallest \(e_j\) do
      while \(e_{used} - e_{tot} < -\varepsilon\) and \(e_{used} - e_{tot} > \varepsilon\) do
        Compute \(avInc\);
        if \(S_j + avInc \geq 1\) then
          Set \(S_j = 1\)
        else
          Add \(avInc\) to \(S_j\)
        end if
      end while
    end for
  else if \((e_{used} - e_{tot} > 0)\) then
    for each facility \(j\) beginning one with the biggest \(e_j\) do
      while \(e_{used} - e_{tot} < -\varepsilon\) and \(e_{used} - e_{tot} > \varepsilon\) do
        Compute \(avInc = (e_{tot} - e_{used}) / e_j, j \in J'\);
        if \(S_j + avInc \leq 0\) then
          Set \(S_j = 0\)
        else
          Add \(avInc\) to \(S_j\)
        end if
      end while
    end for
  end if
end for
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation

Recent Solution Procedures
- Progressive Grid Search (PGS)
- Multi-start Revised Simplex Search (MSS)

New Solution Procedures
- Fully Informed Particle Swarm Optimization (FIPSO)
- Electromagnetism-like Algorithm (EMLA)

Computational Results

Conclusions
Computing Platform

- Cplex 12.1 Concert Technology within MS-Visual Studio 2008.
- Intel Xeon W3690 3.47 GHz Hexa-Core Processor
  - PassMark Score (www.cpubenchmark.net): 9,736
  - Overall Rank in www.cpubenchmark.net: 52 (as of June 24, 2014)
- 24 GB DDR3 ECC RAM
Random Data Generation Table

- 72 BPIP instances which differ in customer and facility configurations
- The data generation scheme of Aksen et al. (2010)
- $4 \leq m \leq 10 \rightarrow$ small-size instances
- $11 \leq m \leq 15 \rightarrow$ large-size instances

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of facilities ($m$)</td>
<td>$4,5,\ldots,15$</td>
</tr>
<tr>
<td>Number of customers ($n$)</td>
<td>$40,50,\ldots,150$</td>
</tr>
<tr>
<td>Customer $i$’s demand ($a_i$)</td>
<td>$5,10,\ldots,100$</td>
</tr>
<tr>
<td>Unit shipment cost ($c_d$)</td>
<td>0.1</td>
</tr>
<tr>
<td>Outsourcing cost ($c_p$)</td>
<td>100</td>
</tr>
<tr>
<td>Unit capacity acquisition cost ($h_j$)</td>
<td>2500</td>
</tr>
<tr>
<td>Capacity multiplier ($q$)</td>
<td>250</td>
</tr>
<tr>
<td>Full interdiction cost of a facility $j$ ($e_j$)</td>
<td>$15000,16000,\ldots,30000$</td>
</tr>
<tr>
<td>Interdiction budget ($e_{tot}$)</td>
<td>$\eta \sum_{j \in J} e_j$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0.2</td>
</tr>
</tbody>
</table>
### PGS, MSS, FIPSO and EMLA Results

#### Small-size Low Budget instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>PGS</th>
<th>MSS</th>
<th>FIPSO</th>
<th>EMLA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Z_{\text{att}}$</td>
<td>Time (s)</td>
<td>$Z_{\text{att}}$</td>
<td>Time (s)</td>
</tr>
<tr>
<td>4-1</td>
<td>130,912</td>
<td>3</td>
<td>133,817</td>
<td>56</td>
</tr>
<tr>
<td>4-2</td>
<td>93,591</td>
<td>2</td>
<td>95,897</td>
<td>41</td>
</tr>
<tr>
<td>4-3</td>
<td>92,501</td>
<td>2</td>
<td>95,760</td>
<td>42</td>
</tr>
<tr>
<td>5-1</td>
<td>150,402</td>
<td>10</td>
<td>151,820</td>
<td>97</td>
</tr>
<tr>
<td>5-2</td>
<td>104,077</td>
<td>7</td>
<td>105,359</td>
<td>56</td>
</tr>
<tr>
<td>5-3</td>
<td>122,075</td>
<td>16</td>
<td>122,576</td>
<td>168</td>
</tr>
<tr>
<td>6-1</td>
<td>142,090</td>
<td>49</td>
<td>141,597</td>
<td>156</td>
</tr>
<tr>
<td>6-2</td>
<td>122,557</td>
<td>95</td>
<td>123,886</td>
<td>268</td>
</tr>
<tr>
<td>6-3</td>
<td>132,185</td>
<td>65</td>
<td>132,308</td>
<td>202</td>
</tr>
<tr>
<td>7-1</td>
<td>172,133</td>
<td>292</td>
<td>173,537</td>
<td>373</td>
</tr>
<tr>
<td>7-2</td>
<td>171,586</td>
<td>408</td>
<td>172,353</td>
<td>358</td>
</tr>
<tr>
<td>7-3</td>
<td>159,226</td>
<td>511</td>
<td>161,550</td>
<td>650</td>
</tr>
<tr>
<td>8-1</td>
<td>190,492</td>
<td>2,588</td>
<td>192,478</td>
<td>940</td>
</tr>
<tr>
<td>8-2</td>
<td>184,337</td>
<td>1,439</td>
<td>187,248</td>
<td>399</td>
</tr>
<tr>
<td>8-3</td>
<td>195,543</td>
<td>3,488</td>
<td>196,436</td>
<td>1,565</td>
</tr>
<tr>
<td>9-1</td>
<td>220,723</td>
<td>16,601</td>
<td>220,231</td>
<td>1,382</td>
</tr>
<tr>
<td>9-2</td>
<td>189,298</td>
<td>8,084</td>
<td>191,439</td>
<td>2,479</td>
</tr>
<tr>
<td>9-3</td>
<td>223,521</td>
<td>17,436</td>
<td>221,309</td>
<td>959</td>
</tr>
<tr>
<td>10-1</td>
<td>236,703</td>
<td>112,235</td>
<td>236,522</td>
<td>2,253</td>
</tr>
<tr>
<td>10-2</td>
<td>232,957</td>
<td>127,807</td>
<td>232,791</td>
<td>1,641</td>
</tr>
<tr>
<td>10-3</td>
<td>242,122</td>
<td>96,843</td>
<td>241,720</td>
<td>2,006</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td>167,097</td>
<td>18,475</td>
<td><strong>168,125</strong></td>
<td>766</td>
</tr>
<tr>
<td><strong>No. of Best</strong></td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>11</td>
</tr>
</tbody>
</table>
## PGS, MSS, FIPSO and EMLA Results

<table>
<thead>
<tr>
<th>Instance</th>
<th>PGS</th>
<th>MSS</th>
<th>FIPSO</th>
<th>EMLA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Z_{\text{att}}$</td>
<td>Time (s)</td>
<td>$Z_{\text{att}}$</td>
<td>Time (s)</td>
</tr>
<tr>
<td>4-1</td>
<td>167,380</td>
<td>3</td>
<td>168,174</td>
<td>19</td>
</tr>
<tr>
<td>4-2</td>
<td>139,637</td>
<td>2</td>
<td>140,902</td>
<td>20</td>
</tr>
<tr>
<td>4-3</td>
<td>144,944</td>
<td>3</td>
<td><strong>146,357</strong></td>
<td>21</td>
</tr>
<tr>
<td>5-1</td>
<td>211,799</td>
<td>8</td>
<td><strong>213,249</strong></td>
<td>44</td>
</tr>
<tr>
<td>5-2</td>
<td>166,584</td>
<td>9</td>
<td><strong>167,760</strong></td>
<td>31</td>
</tr>
<tr>
<td>5-3</td>
<td>183,775</td>
<td>11</td>
<td>183,783</td>
<td>60</td>
</tr>
<tr>
<td>6-1</td>
<td>201,060</td>
<td>38</td>
<td><strong>201,362</strong></td>
<td>53</td>
</tr>
<tr>
<td>6-2</td>
<td>200,818</td>
<td>66</td>
<td>202,636</td>
<td>91</td>
</tr>
<tr>
<td>6-3</td>
<td>203,722</td>
<td>107</td>
<td>205,554</td>
<td>89</td>
</tr>
<tr>
<td>7-1</td>
<td>253,888</td>
<td>422</td>
<td>251,692</td>
<td>147</td>
</tr>
<tr>
<td>7-2</td>
<td>247,519</td>
<td>390</td>
<td><strong>248,815</strong></td>
<td>121</td>
</tr>
<tr>
<td>7-3</td>
<td>246,119</td>
<td>379</td>
<td>248,187</td>
<td>276</td>
</tr>
<tr>
<td>8-1</td>
<td>287,732</td>
<td>2,984</td>
<td>286,688</td>
<td>181</td>
</tr>
<tr>
<td>8-2</td>
<td>277,084</td>
<td>1,968</td>
<td>276,822</td>
<td>237</td>
</tr>
<tr>
<td>8-3</td>
<td>284,966</td>
<td>3,100</td>
<td>284,171</td>
<td>208</td>
</tr>
<tr>
<td>9-1</td>
<td>334,386</td>
<td>19,677</td>
<td>333,990</td>
<td>290</td>
</tr>
<tr>
<td>9-2</td>
<td>304,114</td>
<td>22,030</td>
<td>296,831</td>
<td>356</td>
</tr>
<tr>
<td>9-3</td>
<td>326,818</td>
<td>18,018</td>
<td>325,006</td>
<td>353</td>
</tr>
<tr>
<td>10-1</td>
<td>360,092</td>
<td>135,099</td>
<td><strong>360,567</strong></td>
<td>673</td>
</tr>
<tr>
<td>10-2</td>
<td>349,406</td>
<td>131,002</td>
<td>348,330</td>
<td>524</td>
</tr>
<tr>
<td>10-3</td>
<td><strong>358,521</strong></td>
<td>170,838</td>
<td>357,653</td>
<td>343</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td>250,017</td>
<td>24,103</td>
<td>249,930</td>
<td>197</td>
</tr>
<tr>
<td><strong>No. of Best</strong></td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>
### PGS, MSS, FIPSO and EMLA Results

<table>
<thead>
<tr>
<th>Instance</th>
<th>MSS</th>
<th>FIPSO</th>
<th>EMLA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-1</td>
<td>246,034</td>
<td>247,434</td>
<td>250,619</td>
</tr>
<tr>
<td>11-2</td>
<td>242,505</td>
<td>242,236</td>
<td>244,565</td>
</tr>
<tr>
<td>11-3</td>
<td>244,814</td>
<td>245,194</td>
<td>246,424</td>
</tr>
<tr>
<td>12-1</td>
<td>253,023</td>
<td>251,511</td>
<td>253,608</td>
</tr>
<tr>
<td>12-2</td>
<td><strong>277,464</strong></td>
<td>270,332</td>
<td>273,862</td>
</tr>
<tr>
<td>12-3</td>
<td>232,252</td>
<td>230,936</td>
<td><strong>234,342</strong></td>
</tr>
<tr>
<td>13-1</td>
<td><strong>292,787</strong></td>
<td>284,677</td>
<td>289,027</td>
</tr>
<tr>
<td>13-2</td>
<td><strong>283,359</strong></td>
<td>274,093</td>
<td>276,145</td>
</tr>
<tr>
<td>13-3</td>
<td>310,922</td>
<td>306,728</td>
<td><strong>313,907</strong></td>
</tr>
<tr>
<td>14-1</td>
<td><strong>300,496</strong></td>
<td>297,759</td>
<td>298,102</td>
</tr>
<tr>
<td>14-2</td>
<td><strong>291,356</strong></td>
<td>284,855</td>
<td>290,424</td>
</tr>
<tr>
<td>14-3</td>
<td><strong>302,989</strong></td>
<td>296,393</td>
<td>298,806</td>
</tr>
<tr>
<td>15-1</td>
<td>329,193</td>
<td>331,672</td>
<td><strong>334,945</strong></td>
</tr>
<tr>
<td>15-2</td>
<td>352,567</td>
<td>352,518</td>
<td><strong>352,916</strong></td>
</tr>
<tr>
<td>15-3</td>
<td><strong>351,240</strong></td>
<td>339,386</td>
<td>342,920</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td><strong>287,400</strong></td>
<td><strong>283,715</strong></td>
<td><strong>286,707</strong></td>
</tr>
<tr>
<td><strong>No. of Best</strong></td>
<td>7</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>
## PGS, MSS, FIPSO and EMLA Results

### Large-Size High Budget instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>MSS</th>
<th>FIPSO</th>
<th>EMLA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Z_{\text{MSS att}}$</td>
<td>Time (s)</td>
<td>$Z_{\text{FIPSO att}}$</td>
</tr>
<tr>
<td>11-1</td>
<td>382,248</td>
<td>676</td>
<td>382,734</td>
</tr>
<tr>
<td>11-2</td>
<td>375,081</td>
<td>733</td>
<td>377,552</td>
</tr>
<tr>
<td>11-3</td>
<td>380,463</td>
<td>673</td>
<td>381,128</td>
</tr>
<tr>
<td>12-1</td>
<td>404,641</td>
<td>1,043</td>
<td>399,018</td>
</tr>
<tr>
<td>12-2</td>
<td>418,656</td>
<td>640</td>
<td>413,990</td>
</tr>
<tr>
<td>12-3</td>
<td>388,935</td>
<td>699</td>
<td>388,919</td>
</tr>
<tr>
<td>13-1</td>
<td>450,429</td>
<td>1,156</td>
<td>439,206</td>
</tr>
<tr>
<td>13-2</td>
<td>443,106</td>
<td>2,941</td>
<td>438,468</td>
</tr>
<tr>
<td>13-3</td>
<td>455,373</td>
<td>1,873</td>
<td>451,389</td>
</tr>
<tr>
<td>14-1</td>
<td>479,735</td>
<td>2,449</td>
<td>477,640</td>
</tr>
<tr>
<td>14-2</td>
<td>475,530</td>
<td>1,298</td>
<td>468,399</td>
</tr>
<tr>
<td>14-3</td>
<td>475,474</td>
<td>4,816</td>
<td>465,629</td>
</tr>
<tr>
<td>15-1</td>
<td>511,704</td>
<td>1,019</td>
<td>508,109</td>
</tr>
<tr>
<td>15-2</td>
<td>520,933</td>
<td>3,155</td>
<td>528,370</td>
</tr>
<tr>
<td>15-3</td>
<td>515,655</td>
<td>1,781</td>
<td>517,065</td>
</tr>
</tbody>
</table>

**Averages**
- MSS: 445,198, 1,663
- FIPSO: 441,974, 1,597
- EMLA: 446,449, 640

**No. of Best**
- MSS: 5
- FIPSO: 0
- EMLA: 10
New Problem: Two Player Tri-level Game

System Planner
- Leader
- Location of the facilities
- Capacity of the facilities
- Preattack assignment of the customers

Attacker
- Follower
- Indirect action fractions
- Partial indirect action
- Limited budget

System Planner
- Post attack assignment of the customers
- Outsourcing option
Trilevel Problem Formulation

Parameters

\( a_i \) demand of customer \( i \).
\( c_d \) unit shipping cost per unit distance,
\( c_p \) unit cost of outsourcing customer demand (independent of distance),
\( d_{ij} \) Euclidean distance between customer \( i \) and facility at site \( j \),
\( e_j \) cost of interdicting the full capacity of facility site \( j \),
\( e_{tot} \) interdiction budget of the attacker,
\( f_j \) fixed cost of opening a facility at site \( j \),
\( h \) unit cost of capacity acquisition for a facility,
\( q \) capacity multiplier.
Trilevel Problem Formulation

Index Sets

$I$ set of customers, $I = \{1, \ldots, n\}$

$J$ set of candidate facility sites, $J = \{1, \ldots, m\}$

$J'$ set of opened facility sites.
Trilevel Problem Formulation

Decision Variables

\[ X_j = \begin{cases} 
1 & \text{if a facility is opened at site } j. \\
0 & \text{otherwise.} 
\end{cases} \]

\[ U_{ij} = \begin{cases} 
1 & \text{if customer } i \text{ is assigned to facility at site } j \text{ before the attack.} \\
0 & \text{otherwise.} 
\end{cases} \]

\[ V_{ij} = \begin{cases} 
1 & \text{if customer } i \text{ is assigned to facility at site } j \text{ after the attack.} \\
0 & \text{otherwise.} 
\end{cases} \]

\[ Q_j = \text{level of capacity acquisition at facility site } j \text{ (capacity can be acquired in bulk only.).} \]

\[ S_j = \text{fraction of the capacity of facility } j \text{ lost due to interdiction.} \]
Trilevel Problem Formulation

\[
\begin{align*}
\min_{X, Q, U} & \quad \sum_{j \in J} f_j X_j + h \sum_{j \in J} Q_j + c_d \sum_{i \in I} \sum_{j \in J} a_{ij} d_{ij} U_{ij} \\
& \quad + c_d \sum_{i \in I} \sum_{j \in J} a_{ij} d_{ij} (1 - U_{ij}) V_{ij} + c_p \sum_{i \in I} a_i (1 - \sum_{j \in J} V_{ij}) \\
\text{s.t.} & \\
\sum_{j \in J} U_{ij} &= 1 \quad \forall i \in I \\
\sum_{i \in I} a_i U_{ij} &\leq q Q_j \quad \forall j \in J \\
\sum_{i \in I} a_i U_{ij} &\geq q (Q_j - 1) \quad \forall j \in J \\
U_{ij} &\leq X_j \quad \forall i \in I, \forall j \in J \\
Q_j &\leq \left( \frac{\sum_{i \in I} a_i}{q} + 1 \right) X_j \quad \forall j \in J \\
X_j, U_{ij} &\in \{0, 1\} \quad \forall i \in I, \forall j \in J \\
Q_j &\in \mathbb{Z}^+ \quad \forall j \in J
\end{align*}
\]
Trilevel Problem Formulation

\[ \max_S Z_{att}(S) = c_d \sum_{i \in I} \sum_{j \in J'} a_{ij} d_{ij} (1 - U_{ij}) V_{ij} + c_p \sum_{i \in I} a_i (1 - \sum_{j \in J'} V_{ij}) \]

s.t.

\[ \sum_{j \in J'} S_j e_j \leq e_{tot} \]

\[ 0 \leq S_j \leq 1, \quad \forall j \in J' \]

\[ \min_V Z_{att}(S) = c_d \sum_{i \in I} \sum_{j \in J'} a_{ij} d_{ij} (1 - U_{ij}) V_{ij} + c_p \sum_{i \in I} a_i (1 - \sum_{j \in J'} V_{ij}) \]

s.t.

\[ \sum_{i \in I} a_i V_{ij} \leq (1 - S_j) q Q_j \quad \forall j \in J' \]

\[ \sum_{j \in J'} V_{ij} \leq 1 \quad \forall i \in I \]

\[ V_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J' \]
Our Tabu Search Algorithm with Hashing

- In our trilevel MIP:

  - Tabu Search with Hashing (TSH) to locate the facilities and Cplex to find their capacities and assignment of customers in the upper level,

  - EMLA to obtain the best destructive strategy with the interdiction fractions of the attacker in the middle level,

  - Cplex to determine the best allocation plan of the system planner in the lower level
Neighborhood Structure in Tabu Search

– Move types:
  • 1-Add : One of the closed facilities is opened.
  • 1-Drop : One of the opened facilities is closed.
  • 1-Swap : One of the opened facilities is replaced by the one of the closed one.

– The number of opened facilities may change every iteration since the number of opened facilities is not fixed and determined by the trade off between the various cost components.

– The number of move types to be executed:
  • 1-Add : $m-p$
  • 1-Drop : $p-1$
  • 1-Swap : $p(m-p)$
Our Tabu Search Algorithm with Hashing

Hash Strategy

- to prevent visiting the same solutions (cycling)
- Hash value of the solution is computed and stored sequentially in a list as:

\[ \text{Hash}(\sigma) = \sum_{j=1}^{m} \begin{cases} 2^{m-j} & \text{if facility } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases} \]

- In the hash list, the objective value and move type are stored beside the hash value → to recall and compare the objective value of the candidate solution with the best neighboring solution without computation
- Searching the hash value starting from both head and tail and if not found, shrink the list by removing current head and tail → continue with the new head and tails of the smaller list.
## Results – TDAP-PI

<table>
<thead>
<tr>
<th>Instance</th>
<th>Ratio=1/10</th>
<th># of Opened Fac</th>
<th>Ratio=1/1</th>
<th># of Opened Fac</th>
<th>Diff(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,40,high,17000)</td>
<td>326,156</td>
<td>36</td>
<td>326,156</td>
<td>34</td>
<td>0</td>
</tr>
<tr>
<td>(5,50,high,23000)</td>
<td>449,483</td>
<td>85</td>
<td>449,483</td>
<td>87</td>
<td>0</td>
</tr>
<tr>
<td>(6,60,high,27600)</td>
<td>474,762</td>
<td>218</td>
<td>474,762</td>
<td>238</td>
<td>0</td>
</tr>
<tr>
<td>(7,70,high,35600)</td>
<td>671,136</td>
<td>541</td>
<td>671,136</td>
<td>525</td>
<td>-3</td>
</tr>
<tr>
<td>(8,80,high,35200)</td>
<td>669,944</td>
<td>1680</td>
<td>669,944</td>
<td>2071</td>
<td>0</td>
</tr>
<tr>
<td>(9,90,high,37800)</td>
<td>789,086</td>
<td>1962</td>
<td>789,086</td>
<td>3731</td>
<td>0</td>
</tr>
<tr>
<td>(10,100,high,39800)</td>
<td>931,524</td>
<td>2155</td>
<td>905,459</td>
<td>4111</td>
<td>-2.8</td>
</tr>
<tr>
<td>(11,110,high,46200)</td>
<td>1,004,769</td>
<td>6488</td>
<td>1,011,166</td>
<td>51,416</td>
<td>0.6</td>
</tr>
<tr>
<td>(12,120,high,58000)</td>
<td>1,151,048</td>
<td>3313</td>
<td>1,131,927</td>
<td>39,593</td>
<td>-1.7</td>
</tr>
<tr>
<td>(13,130,high,60400)</td>
<td>1,248,756</td>
<td>19880</td>
<td>1,248,609</td>
<td>9734</td>
<td>0</td>
</tr>
<tr>
<td>(14,140,high,61400)</td>
<td>1,303,613</td>
<td>24785</td>
<td>1,303,613</td>
<td>9963</td>
<td>0</td>
</tr>
<tr>
<td>(15,150,high,65000)</td>
<td>1,427,257</td>
<td>47618</td>
<td>1,427,257</td>
<td>8979</td>
<td>0</td>
</tr>
</tbody>
</table>

### Averages

<table>
<thead>
<tr>
<th>Ratio=1/10</th>
<th># of Opened Fac</th>
<th>Ratio=1/1</th>
<th># of Opened Fac</th>
<th>Diff(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>870,628</td>
<td>9063</td>
<td>867,383</td>
<td>10,874</td>
<td>-0.3</td>
</tr>
</tbody>
</table>

### High Fixed Cost

\[
\frac{100(OFV^1 - OFV^{0.1})}{OFV^{0.1}}
\]

\[
\frac{100(CPU^1 - CPU^{0.1})}{CPU^{0.1}}
\]
## Results – TDAP-PI

<table>
<thead>
<tr>
<th>Instance</th>
<th>Ratio=1/10</th>
<th></th>
<th>Ratio=1/1</th>
<th></th>
<th>Diff(%)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OFV</td>
<td>CPU(s)</td>
<td># of Opened Fac</td>
<td>OFV</td>
<td>CPU(s)</td>
<td># of Opened Fac</td>
</tr>
<tr>
<td>(4,40,low,17000)</td>
<td>246,446</td>
<td>35</td>
<td>2</td>
<td>246,446</td>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>(5,50,low,23000)</td>
<td>268,431</td>
<td>88</td>
<td>2</td>
<td>268,604</td>
<td>89</td>
<td>2</td>
</tr>
<tr>
<td>(6,60,low,27600)</td>
<td>345,852</td>
<td>211</td>
<td>3</td>
<td>345,852</td>
<td>234</td>
<td>3</td>
</tr>
<tr>
<td>(7,70,low,35600)</td>
<td>499,014</td>
<td>375</td>
<td>3</td>
<td>498,530</td>
<td>529</td>
<td>3</td>
</tr>
<tr>
<td>(8,80,low,35200)</td>
<td>490,022</td>
<td>2947</td>
<td>3</td>
<td>490,522</td>
<td>1886</td>
<td>3</td>
</tr>
<tr>
<td>(9,90,low,37800)</td>
<td>602,188</td>
<td>2651</td>
<td>3</td>
<td>602,468</td>
<td>7286</td>
<td>3</td>
</tr>
<tr>
<td>(10,100,low,39800)</td>
<td>715,158</td>
<td>3001</td>
<td>3</td>
<td>715,158</td>
<td>4373</td>
<td>3</td>
</tr>
<tr>
<td>(11,110,low,46200)</td>
<td>744,310</td>
<td>4366</td>
<td>4</td>
<td>743,824</td>
<td>36,208</td>
<td>4</td>
</tr>
<tr>
<td>(12,120,low,58000)</td>
<td>850,930</td>
<td>5510</td>
<td>4</td>
<td>851,586</td>
<td>42,735</td>
<td>4</td>
</tr>
<tr>
<td>(13,130,low,60400)</td>
<td>923,577</td>
<td>46,142</td>
<td>4</td>
<td>920,170</td>
<td>47,096</td>
<td>4</td>
</tr>
<tr>
<td>(14,140,low,61400)</td>
<td>967,838</td>
<td>39,580</td>
<td>5</td>
<td>967,252</td>
<td>44,208</td>
<td>5</td>
</tr>
<tr>
<td>(15,150,low,65000)</td>
<td>1,118,007</td>
<td>15,497</td>
<td>5</td>
<td>1,121,089</td>
<td>26,095</td>
<td>4</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td>647,648</td>
<td>10,034</td>
<td>3</td>
<td>647,625</td>
<td>17,565</td>
<td>3</td>
</tr>
</tbody>
</table>
Results – TDAP-PI

- Effect of attacker’s budget

<table>
<thead>
<tr>
<th>Instance</th>
<th>$\eta = 0.2$ (%)</th>
<th>$\eta = 0.1$ (%)</th>
<th>$\eta = 0.05$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,40,high,1/10)</td>
<td>73.9</td>
<td>37.0</td>
<td>18.5</td>
</tr>
<tr>
<td>(5,50,high,1/10)</td>
<td>82.1</td>
<td>41.1</td>
<td>20.5</td>
</tr>
<tr>
<td>(6,60,high,1/10)</td>
<td>48.4</td>
<td>47.6</td>
<td>23.8</td>
</tr>
<tr>
<td>(7,70,high,1/10)</td>
<td>62.5</td>
<td>59.3</td>
<td>37.1</td>
</tr>
<tr>
<td>(8,80,high,1/10)</td>
<td>69.0</td>
<td>58.7</td>
<td>29.3</td>
</tr>
<tr>
<td>(9,90,high,1/10)</td>
<td>87.9</td>
<td>44.0</td>
<td>37.8</td>
</tr>
<tr>
<td>(10,100,high,1/10)</td>
<td>94.8</td>
<td>51.0</td>
<td>43.3</td>
</tr>
<tr>
<td>(11,110,high,1/10)</td>
<td>94.3</td>
<td>53.7</td>
<td>46.2</td>
</tr>
<tr>
<td>(12,120,high,1/10)</td>
<td>75.3</td>
<td>52.7</td>
<td>26.9</td>
</tr>
<tr>
<td>(13,130,high,1/10)</td>
<td>75.5</td>
<td>43.8</td>
<td>30.8</td>
</tr>
<tr>
<td>(14,140,high,1/10)</td>
<td>71.4</td>
<td>53.9</td>
<td>33.4</td>
</tr>
<tr>
<td>(15,150,high,1/10)</td>
<td>95.6</td>
<td>70.7</td>
<td>35.3</td>
</tr>
</tbody>
</table>

- Attacker capability percentage of being able to destroy

\[
\left( \sum_{j \in J} S_j e_j \right) / e_{tot}
\]

- $\eta = 0.2$ gives the highest percentage → not opened all the potential facilities in the system so the attacker can damage the system more.

- If the budget is decreased with changing $\eta$ to 0.05, the percentage decreases
### Results – TDAP-PI Problem

#### Comparing with Complete Enumeration Results

- TS is much better than complete enumeration with respect to CPU time.
- Also, both algorithm finds the same objective values.

<table>
<thead>
<tr>
<th>Instance</th>
<th>TS (\text{OFV} , \text{CPU(s)})</th>
<th>ENUM (\text{OFV} , \text{CPU(s)})</th>
<th>Diff(%) (\text{OFV} , \text{CPU(s)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4, high, 17000)</td>
<td>326,156 36</td>
<td>326,156 44</td>
<td>0 22</td>
</tr>
<tr>
<td>(5, high, 23000)</td>
<td>449,483 85</td>
<td>449,482 96</td>
<td>0 13</td>
</tr>
<tr>
<td>(6, high, 27600)</td>
<td>474,762 218</td>
<td>474,762 429</td>
<td>0 97</td>
</tr>
<tr>
<td>(7, high, 35600)</td>
<td>671,136 541</td>
<td>671,136 355</td>
<td>0 58</td>
</tr>
<tr>
<td>(8, high, 35200)</td>
<td>669,944 1680</td>
<td>669,944 24,040</td>
<td>0 1331</td>
</tr>
<tr>
<td>(9, high, 37800)</td>
<td>789,086 1962</td>
<td>789,086 5963</td>
<td>0 204</td>
</tr>
<tr>
<td>(10, high, 39800)</td>
<td>931,524 2155</td>
<td>931,524 11,602</td>
<td>0 438</td>
</tr>
<tr>
<td>(11, high, 46200)</td>
<td>1,004,769 6488</td>
<td>1,011,167 86,150</td>
<td>0.6 1228</td>
</tr>
<tr>
<td>(4, low, 17000)</td>
<td>246,446 35</td>
<td>246,681 44</td>
<td>0.1 26</td>
</tr>
<tr>
<td>(5, low, 23000)</td>
<td>268,431 88</td>
<td>268,604 112</td>
<td>0.1 27</td>
</tr>
<tr>
<td>(6, low, 27600)</td>
<td>345,852 211</td>
<td>345,853 430</td>
<td>0 104</td>
</tr>
<tr>
<td>(7, low, 35600)</td>
<td>499,014 375</td>
<td>499,315 354</td>
<td>0.1 128</td>
</tr>
<tr>
<td>(8, low, 35200)</td>
<td>490,022 2947</td>
<td>491,274 23,869</td>
<td>0.3 710</td>
</tr>
<tr>
<td>(9, low, 37800)</td>
<td>602,188 2651</td>
<td>602,468 5925</td>
<td>0.1 124</td>
</tr>
<tr>
<td>(10, low, 39800)</td>
<td>715,158 3001</td>
<td>715,158 11,613</td>
<td>0 287</td>
</tr>
<tr>
<td>(11, low, 46200)</td>
<td>744,310 4366</td>
<td>745,833 87,372</td>
<td>0.2 1901</td>
</tr>
<tr>
<td><strong>Averages</strong></td>
<td><strong>576,768 1677</strong></td>
<td><strong>577,403 16,212</strong></td>
<td><strong>0.1 419</strong></td>
</tr>
</tbody>
</table>
OVERVIEW

- Introduction to BPIP
- Motivation
- Literature Survey on Facility Interdiction Problems
- Problem Formulation
- Recent Solution Procedures
  - Progressive Grid Search (PGS)
  - Multi-start Revised Simplex Search (MSS)
- New Solution Procedures
  - Fully Informed Particle Swarm Optimization (FIPSO)
  - Electromagnetism-like Algorithm (EMLA)
- Computational Results
- Conclusions
Conclusions

- EMLA outperforms MSS and FIPS.

- Partial interdiction and capacitated facilities make this model realistic.

- Our problem TDAP-PI is modeled as a trilevel MIP with two players.

- Integrating various decisions in the same model:
  - Facility location and capacity acquisition decisions for the system planner,
  - Customer assignments to the facilities for the system planner,
  - Interdiction fractions of the attacker.

- If the neighborhood size can be efficiently reduced, better results are attainable in a shorter time.
Questions?
NMSS move operations

*(Reflection is best so far)*

\[ f(x_1) = 10 \]
\[ f(x_0) = 8 \]
\[ f(x_r) = 15 \]
\[ f(x_2) = 12 \]
NMSS move operations

(Expansion is even better)

\[ f(x_1) = 10 \]
\[ f(x_0) = 8 \]
\[ f(x_2) = 12 \]
\[ f(x_e) = 18 \]
\[ f(x_r) = 15 \]
NMSS move operations

(Reflection is better than Expansion)

\[ f(x_1) = 10 \]
\[ f(x_2) = 8 \]
\[ f(x_r) = 15 \]
\[ f(x_e) = 14 \]
NMSS move operations

(Reflection w/o Expansion)

\[ f(x_1) = 10 \]
\[ f(x_0) = 8 \]
\[ f(x_2) = 12 \]
\[ f(x_r) = 11 \]
NMSS move operations

(Reflection doesn’t help, Outer Contraction does)

\[ f(x_1) = 10 \]
\[ f(x_0) = 8 \]
\[ f(x_2) = 12 \]
\[ f(x_r) = 9 \]
\[ f(x_{oc}) = 10 \]
NMSS move operations

(Neither Reflection nor Contraction helps: Shrinkage!)

\[ f(x_0) = 8 \]
\[ f(x_1) = 10 \]
\[ f(x_r) = 9 \]
\[ f(x_{oc}) = 8 \]
\[ f(x_2) = 12 \]
NMSS move operations

(Reflection is worst, Inner Contraction helps)

\[ f(x_1) = 10 \]
\[ f(x_0) = 8 \]
\[ f(x_{ic}) = 8 \]
\[ f(x_{r}) = 7 \]
\[ f(x_2) = 12 \]
NMSS move operations

(Reflection is worst, Inner Contraction doesn’t help: Shrinkage!)

\[ f(x_1) = 10 \]
\[ f(x_2) = 8 \]
\[ f(x_{ic}) = 7 \]
\[ f(x_r) = 7 \]
\[ f(x_{ic}) = 7 \]
\[ f(x_2) = 12 \]
Handling the Bounded Variables

- Less than the lower bound of the variable

\[ x_1 = [-0.1, 0.2] \]

\[ x_1 = [0, 0.2] \]

- Greater than the upper bound of the variable

\[ x_1 = [1.1, 0.2] \]

\[ x_1 = [1.0, 0.2] \]

\[ F'(X_1) = F(X_1) - \text{Penalty} \]
Collapse of a simplex

\[ x_1 = [1.1, 0.2] \]

\[ x_0 = [1, 3, 0.2] \quad x_2 = [1.3, 1] \]

\[ x_1 = [1, 0.2] \]

\[ x_0 = [1, 0.2] \quad x_2 = [1, 1] \]

\[ x_1 = [1, 0.2] \]

\[ x_2 = [1, 1] \]
Plateau Problem

\[ f(x_1) = 8 \]
\[ x_1 = [1, 0.2] \]
\[ x_0 = [0.9, 0.2] \]
\[ x_2 = [0.8, 1] \]
\[ f(x_0) = 8 \]
\[ f(x_2) = 8 \]

- All vertices of the simplex have the same objective value.
- No memory beyond the current simplex points.
- If no improvement in \( m+2 \) iteration \( \rightarrow \) caught in plateau.
- Brooks (2000) \( \rightarrow \) check only in the reflection step.
- Stop the current phase and start a new one.
Cycling

- The reflected point is not reflected at the next two iterations
- The second worst point is reflected
Single MSS run

- Phase counter $\tau \rightarrow 1$
- Iteration counter $\varphi \rightarrow 1$
- $\alpha = 1$, $\beta = 2$, $\gamma = 0.5$, $\delta = 0.5$
- Initial simplex is generated.
- Objective values of the $m+1$ points are calculated and sorted in non-increasing order.
- Plateau check ($\omega > m+2$ ?).
- One NM-SS iteration.
- The new point is checked for the bounds.
- Budget feasibility is also verified.
- If reflection move operation is chosen, cycling check is applied.
Termination Criteria

- Termination criteria are checked
  - Termination of a phase → Reaching a maximum iteration number
    → Small volume of the simplex
    → Getting caught in plateau
  - Start a new phase → Scaling procedure is applied as in the initial simplex generation methods
    → Constructing a smaller new simplex around the best solution obtained in the previous phase

\[ u_\tau = \frac{1}{2} u(\tau-1) \]

→ Increasing the shrinkage coefficient linearly
\[ \delta_\tau = \delta(\tau-1) + 0.2 \]

- Termination of a single MSS run → phase counter \( \tau > 3 \)?
- Termination of MSS → \( K > 6 \)