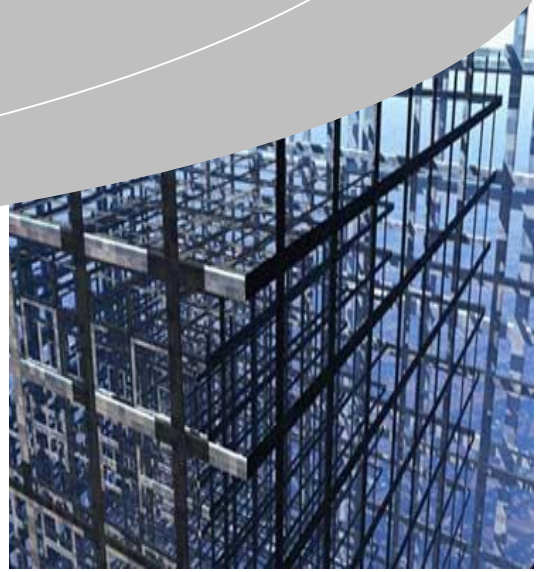
POLYTECHNIQUE MONTRÉAL

Optimizing drone routing for post-disaster search and rescue

Sean Grogan, PhD Candidate
Michel Gamache, Robert Pellerin; Professors
Département de Mathématiques et de Génie Industriel
École Polytechnique de Montréal

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux



Contents

Context

- In the aftermath of a disaster, often rescuers are tasked with locating victims trapped and in need of aid:
  - Most search and rescue procedures are incredibly manual with rescuers approaching collapsed structures;
  - After a disaster, human rescuers using tools such as dogs, acoustic sensors, and snake cameras to locate people;
- Explosion of cell phone usage around the world:
  - Cellphones emit a host of signals by virtue of being on and active.
  - Research has begun to explore the ability to passively or actively detect in interact these signals.

FEMA Urban Search and Rescue team members search house to house for survivors in tornado devastated neighborhood in Moore, Oklahoma. Andrea Booher for FEMA

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Introduction

- Some organizations and research institutions have begun to look at the use of drones in the search and rescue operations;
- Attaching a wireless sensor, a device to detect cell phones, to the done can allow the drone to locate cell phone users in a region;
- Drones have an obvious utility with the ability to fly over a disaster area;
- Drones have a hard endurance constraint (flight time) and cannot usually cover the entire search area alone.

Problem Formulation and Constraints (1/4)

- Envision the search area discretized in into scanning areas,
- A drone with a wireless sensor attached will fly to and scan a scanning area,
- Drones have a hard *endurance constraint*, flight time,
- Drones will consume flight time along the route *and* at each scanning site

Problem Formulation and Constraints (2/4)

- Formulation envisioned as a distance-constrained capacity vehicle routing problem (DCVRP).
- Began with formulation from Kek, et al (2008) and modified to suit the problem here,
- Notable change: the DCVRP is usually outlined as having a different resource consumed at the nodes (customers) and on the arc (enroute).
- In a post-disaster context: time windows less important than overall service time.

Kek, A. G., Cheu, R. L., & Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling, 47*(1-2), 140-152. doi:10.1016/j.mcm.2007.02.007

Problem Formulation and Constraints (3/4)

Sets:

$i \in N$	Set of customers
$i \in H$	Set of hubs (where drones depart)
$i \in V = N \cup H$	Set of hubs and customers
$k \in K$	Set of Drones available

Parameters:

c_{ij}	Cost (time) of going from i to j
d_i	Demand (time) of scanning at site i
Q	The upper limit of time a drone can fly (endurance)
K	The upper limit on the number of drones available
M	Big ass number, specifically $Q + \max_j [d_j]$

Variables:

$x_{ijk} \in \{0,1\}$	Arc i to j is traveled by drone route k
$y_k \in \{0,1\}$	Indicator variable that drone route k is used
$z_{ik} \geq 0$	Load variables specifying the total load serviced by vehicle since its last visit to a depot by the time it reaches customer node. Read as maximum load on route k.
$q_k^R \geq 0$	Amount of endurance consumed by routes
$q_k^S \geq 0$	Amount of endurance consumed by scanning

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Problem Formulation and Constraints (3/4)

Sets:

$i \in N$	Set of customers
$i \in H$	Set of hubs (where drones depart)
$i \in V = N \cup H$	Set of hubs and customers
$k \in K$	Set of Drones available

Parameters:

$\boldsymbol{c_{ij}}$	**Cost (time) of going from i to j**
$\boldsymbol{d_i}$	**Demand (time) of scanning at site i**
Q	The upper limit of time a drone can fly (endurance)
K	The upper limit on the number of drones available
M	Big ass number, specifically $Q + \max_j [d_j]$

Variables:

$x_{ijk} \in \{0,1\}$	Arc i to j is traveled by drone route k
~~$y_k \in \{0,1\}$~~	~~Indicator variable that drone route k is used~~
$\mathbf{z_{ik} \geq 0}$	**Load variables specifying the total load serviced by vehicle since its last visit to a depot by the time it reaches customer node. Read as maximum load on route k.**
$\mathbf{q_k^R \geq 0}$	**Amount of endurance consumed by routes**
$\mathbf{q_k^S \geq 0}$	**Amount of endurance consumed by scanning**

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Problem Formulation and Constraints (4/4)

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk} + \sum_{i\in N}\sum_{k\in K} d_i x_{ij} \qquad (1)$$

$$\sum_{\substack{j\in V \\ j\neq i}}\sum_{k\in K} x_{ijk} \leq 1 \qquad i \in N \qquad (2)$$

$$\sum_{i\in H}\sum_{j\in N} x_{ijk} = y_k \qquad k \in K \qquad (3)$$

$$\sum_{\substack{i\in V \\ i\neq j}} x_{ijk} - \sum_{\substack{j\in V \\ j\neq i}} x_{jik} = 0 \qquad \begin{matrix} k \in K \\ j \in V \end{matrix} \qquad (4)$$

$$z_{ik} = 0 \qquad \begin{matrix} i \in H \\ k \in N \end{matrix} \qquad (5)$$

$$\left(z_{ik} + d_j - z_{jk}\right) \leq M\left(1 - x_{ijk}\right) \qquad \begin{matrix} i \in V \\ j \in N \\ k \in K \\ i \neq j \end{matrix} \qquad (6)$$

$$\left(z_{ik} + d_j - z_{jk}\right) \geq -M\left(1 - x_{ijk}\right) \qquad \begin{matrix} i \in V \\ j \in N \\ k \in K \\ i \neq j \end{matrix} \qquad (7)$$

$$z_{ik} \leq q_k^S \qquad k \in K \qquad (8)$$

$$\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ijk} + \sum_{i\in V}\sum_{j\in V} d_i x_{ijk} \leq q_k^R \qquad k \in K \qquad (9)$$

$$q_k^S + q_k^R \leq Q \qquad k \in K \qquad (10)$$

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Problem Formulation and Constraints (4/4)

$$\min \sum_{i\in V}\sum_{j\in V}\sum_{k\in K} c_{ij}x_{ijk} + \sum_{i\in N}\sum_{k\in K} d_i x_{ij} \qquad (1)$$

$$\sum_{\substack{j\in V \\ j\neq i}}\sum_{k\in K} x_{ijk} \leq 1 \qquad i \in N \qquad (2)$$

~~$$\sum_{i\in H}\sum_{j\in N} x_{ijk} = y_k \qquad k \in K \qquad (3)$$~~

$$\sum_{\substack{i\in V \\ i\neq j}} x_{ijk} - \sum_{\substack{j\in V \\ j\neq i}} x_{jik} = 0 \qquad \begin{matrix} k \in K \\ j \in V \end{matrix} \qquad (4)$$

$$z_{ik} = 0 \qquad \begin{matrix} i \in H \\ k \in N \end{matrix} \qquad (5)$$

$$\left(z_{ik} + d_j - z_{jk}\right) \leq M\left(1 - x_{ijk}\right) \qquad \begin{matrix} i \in V \\ j \in N \\ k \in K \\ i \neq j \end{matrix} \qquad (6)$$

$$\left(z_{ik} + d_j - z_{jk}\right) \geq -M\left(1 - x_{ijk}\right) \qquad \begin{matrix} i \in V \\ j \in N \\ k \in K \\ i \neq j \end{matrix} \qquad (7)$$

$$\mathbf{z_{ik} \leq q_k^S} \qquad \mathbf{k \in K} \qquad \mathbf{(8)}$$

$$\mathbf{\sum_{i\in V}\sum_{j\in V} c_{ij}x_{ijk} + \sum_{i\in V}\sum_{j\in V} d_i x_{ijk} \leq q_k^R} \qquad \mathbf{k \in K} \qquad \mathbf{(9)}$$

$$\mathbf{q_k^s + q_k^R \leq Q} \qquad \mathbf{k \in K} \qquad \mathbf{(10)}$$

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Model Demonstration (1/2)

- Used a random sampling of nodes from TSPLIP's d18512 file,
- Had to create a demand matrix that was reasonably proportional to the distances in the graph:

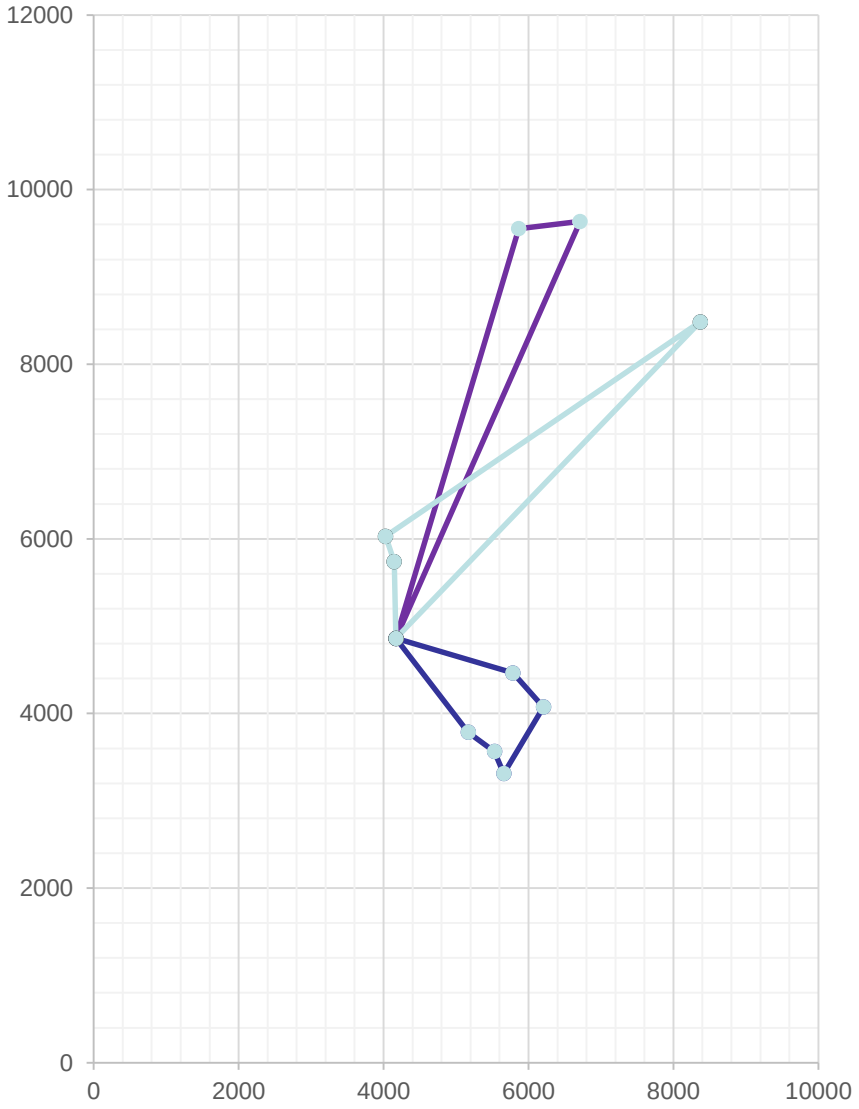
$$d_i = u\{0,1\} \times 0.7 \times \frac{\sum_{\substack{i,j \in N \\ i \neq j}} c_{ij}}{|N|^2 - |N|}$$

- Number of cities : N=10 + 1 depot
- Drone endurance: Q = $\frac{\sum_{\substack{i,j \in N \\ i \neq j}} c_{ij}}{|N|^2 - |N|} \times 6$
- Number of drones : K = $\left\lceil \frac{N}{3} \right\rceil$
- 15 experiments executed.

Model Demonstration (2/3)

Q = 19600

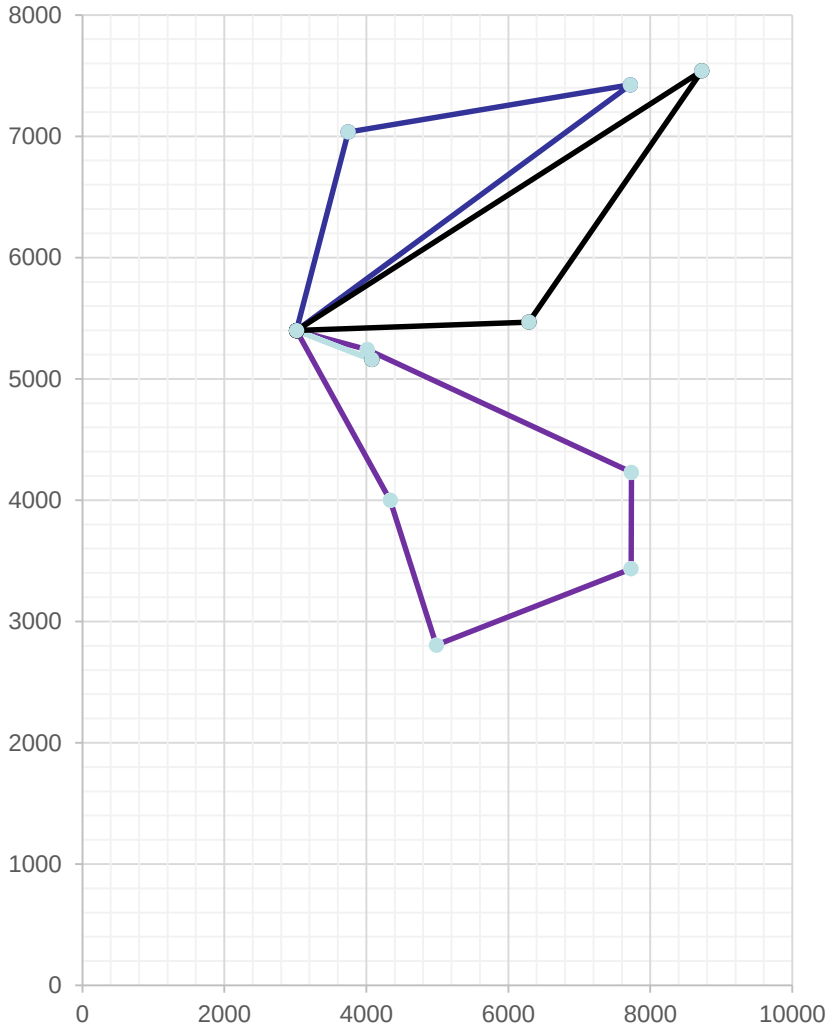
- Tour 1 : 0->3->9->8->5->10->0
  - Transit : 5351
  - Service : 5311
  - Total : 10661
- Tour 2 : 0->1->6->0
  - Transit : 11252
  - Service : 3475
  - Total : 14727
- Tour 3 : 0->2->7->4->0
  - Transit : 11729
  - Service : 3698
  - Total : 15427



Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Model Demonstration (3/3)

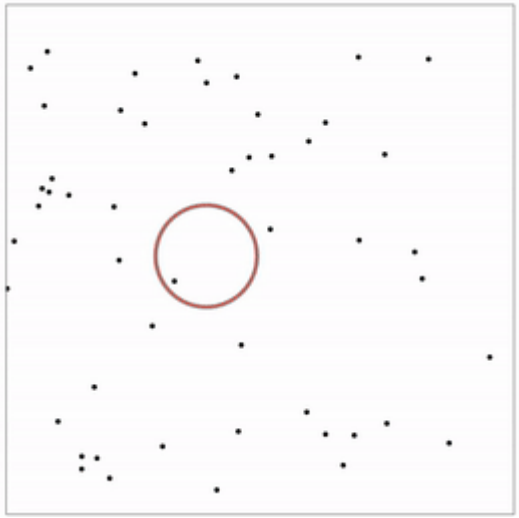
- For N=11:
  - Average time to solve 70 seconds
  - Average tours was 3.2 tours



Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Proposed Heuristic – Elastic Net and Self Ordering Map

- Began as a way of constructing a solution for TSPs,
- Geometrically inspired, consider a rubber band that is slowly drawn to nodes of the graph,
- Internal forces of the rubber band force the tour to remain small,
- The force of the nodes on the rubber band draw the band to the nodes,
- The algorithm terminates when the band arrives sufficiently close to all the cities in the tour.



Animation captured from https://larose.github.io/tsp/

Proposed Heuristic – Elastic Net and Self Ordering Map

- There is a lack of research surrounding the use of neural networks in operations research:
  - Geometrically inspired heuristics don't often work well for TSPs and VRPs in "real world cases"
- However:
  - For a flying vehicle, routing is "as the crow flies"
- Generally:
  - "The incremental development of neural networks [elastic nets and self ordering maps] may lead, over time, to more competitive approaches for the VRP."
  - "SOM may become competitive in the future when massively parallel computer systems will become widely available."

Créput, J.-C., & Koukam, A. (2008). The memetic self-organizing map approach to the vehicle routing problem. *Soft Computing*, *12*(11), 1125–1141. http://doi.org/10.1007/s00500-008-0281-4

Pseudocode

```
done = false
while not done:
    for each m in M:
        MoveNetPoints(m)
    for each k in K:
        TestTours(k)
        if each n in M is at least ε units
          from any m in M:
            done = true
```

```
def MoveNetPoints(point M):
```

$$P_{m,k}^{M} = \alpha \sum_{i \in N} \left[\mathbf{weight}(i,m) \times \left(P_i^N - P_{j,k}^M \right) \right] + \beta\varepsilon \left(P_{j+1,k}^M + P_{j-1,k}^M - 2 \times P_{j,k}^M \right)$$

```
def weight(point n, point m):
    c = Phi(EuclDist(n, m), ε)
    s = 0
    for j in M:
        s += Phi(EuclDist(n, j), ε)
    return c/s

def Phi(d, K):
```

$$\text{return } e^{-\frac{d^2}{2K^2}}$$

```
def EuclDist(point X1, point X2):
```

$$\text{return } \sqrt{(X1_x - X2_x)^2 + \left(X1_y - X2_y\right)^2}$$

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Pseudocode

```
done = false
while not done:
    for each m in M:
        MoveNetPoints(m)
    for each k in K:
        TestTours(k)
        if each n in M is at least ε units
          from any m in M:
            done = true
```

```
def MoveNetPoints(point M):
```

$$P_{m,k}^{M} = \alpha \sum_{i \in N} \left[\mathbf{weight}(i,m) \times \left(P_i^N - P_{j,k}^M \right) \right] + \beta\varepsilon \left(P_{j+1,k}^M + P_{j-1,k}^M - 2 \times P_{j,k}^M \right)$$

```
def weight(point n, point m):
    c = Phi(EuclDist(n, m), ε)
    s = 0
    for j in M:
        s += Phi(EuclDist(n, j), ε)
    return c/s

def Phi(d, K):
```

$$\text{return } e^{-\frac{d^2}{2K^2}}$$

```
def EuclDist(point X1, point X2):
```

$$\text{return } \sqrt{(X1_x - X2_x)^2 + \left(X1_y - X2_y\right)^2}$$

Chaire de recherche Jarislowsky / SNC-Lavalin en gestion de projets internationaux

Questions


Chaire de recherche
Jarislowsky / SNC-Lavalin
en gestion de projets internationaux