Inspection Route Optimization

Bernard Gendron* Thibaut Vidal+

August 11, 2017

CRM Industrial Problem Solving Workshop, Montréal, Canada

* CIRRELT and Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada

+ PUC-Rio - Pontificia Universidade Catòlica do Rio de Janeiro

Outline

Students

1. Steven Lamontagne, Université de Montréal
2. Belhal Karimi, Polytechnique Paris
3. David Hagenimana, University of Nairobi
4. Jalal Ahammad, Memorial University of Newfoundland
5. Monuara Gagum, Memorial University of Newfoundland
6. Dena Kazerani, Polytechnique Paris
7. Ilya Chugunov, UC Berkeley
8. Maikel Geagea, Polytechnique Montréal
9. Mickael Albertus, Université Paul Sabatier Toulouse
10. Bora Yongacoglu, Queens University

Coordinators

- Bernard Gendron
- Thibaut Vidal
- Mathieu Giguère, The Co-Operators
- Jean-Michel Plante, The Co-Operators

The Co-Operators

The Co-operators

- Insurance & Financial Services
- 1,000,000 Clients
- 5,000 employees

We're a Coop!

- Cooperative Values
- Owned by 35 members (who are also cooperatives!)

We like Data!

- 150 Professionals
- 2017 Science & Analytics Summit

Description of the Application

- Locations (buildings, stores, farms,...) all across Canada
- The company has to decide if a particular location is worth to insure or not
- A risk score is computed following an off-site inquiry
- When the risk score is high, an inspector is sent to visit the location (on-site inquiry)
- More than 5,000 locations need to be visited (over a year)
- Not enough inspectors to visit all of them!
- **Which locations should be visited over the next year?**

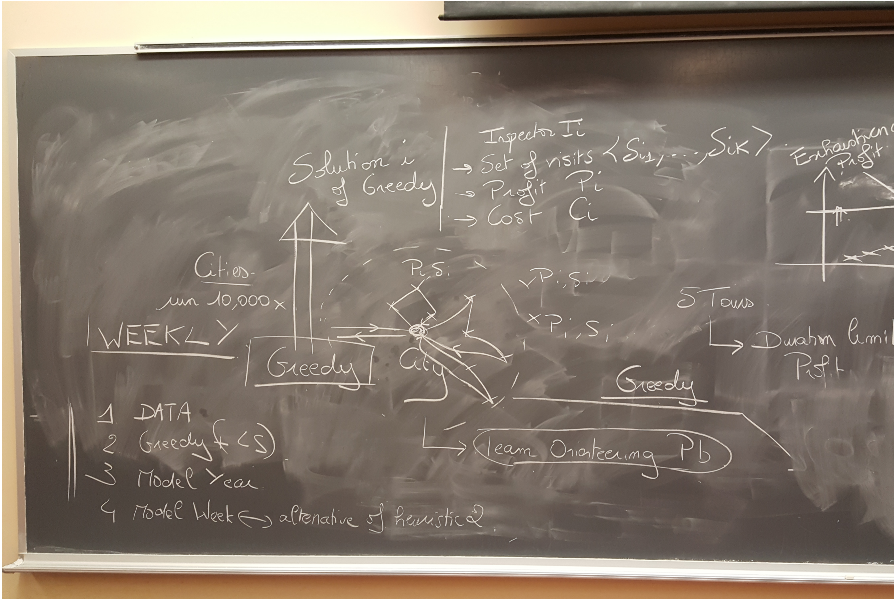
Methods Implemented by the Company

- Old method
  - 11 inspectors: assign each of them to a territory (divided by province) based on their home
  - For each inspector, assign the locations to be inspected in descending order of risk scores
  - **Drawback: too greedy!**
- New method
  - Same assignment of inspectors to territories
  - Decompose each territory by cities to which locations are assigned
  - For each city, find a tour that maximizes risk scores and minimizes distances
  - **Drawbacks: 1) arbitrary decomposition 2) time limits not always respected**

High-Level View of the Problem

- Objective: **maximize risk scores**
- Constraints: **time limits**
  - Each day: every inspector makes a tour (starting and ending at a hotel) whose total duration (travel + inspection) should not exceed 7 hours
  - Each week: every inspector makes up to 5 daily tours before coming back home
  - Over the year: every inspector works (about) 46 weeks
- **Decomposition into weekly problems “assembled” in an annual problem**

Solutions of the Weekly Problem



Assignment of Tasks to Team Members

1. Extract and analyze data given by the company
2. Develop and implement model for the annual problem
3. Develop and implement model for the weekly problem
4. Develop and implement heuristic methods for the weekly problem

Annual Problem: Assumptions and Data

- Assume we know all weekly solutions: not realistic, but it helps to model the problem!
- Each weekly solution has an inspector traveling from/to home and making 5 daily tours
- I: set of weekly solutions
- J: set of locations
- K: set of inspectors
- $\delta_{ij} = 1$ if location $j \in J$ belongs to weekly solution $i \in I$, 0 otherwise
- $\theta_{ik} = 1$ if inspector $k \in K$ is assigned to weekly solution $i \in I$, 0 otherwise
- p_i: profit (risk score) of weekly solution $i \in I$
- W_k: annual number of weeks worked by inspector $k \in K$

Annual Problem: Model

- $x_i = 1$, if weekly plan $i \in I$ is selected, 0 otherwise

$$\max \sum_{i \in I} p_i x_i$$

$$\sum_{i \in I} \delta_{ij} x_i \leq 1 \quad j \in J$$

$$\sum_{i \in I} \theta_{ik} x_i \leq W_k \quad k \in K$$

$$x_i \in \{0, 1\} \quad i \in I$$

- (Constrained) Set Packing Problem: NP-hard

Annual Problem: Solution Methods

- Column generation based on linear programming + Branch-and-Price
  - Exact method!
  - It takes some time to develop...
- Column generation based on solving lots of weekly problems + Branch-and-Bound
  - Heuristic method
  - Doable in a few days/weeks
  - Can be used as warmstart to speedup exact method

Formalization of the Weekly Problem

- Construct a set of $d \leq 5$ daily tours such that:
  - Overall profit (risk score) is maximized
  - The daily time limit is respected
- This is a variant of Vehicle Routing Problem (VRP) known in the literature as the **Team Orienteering Problem (TOP)**
- Once a TOP is solved (through a model or by using heuristic methods):
  - Complete the week if necessary (by solving another TOP)
  - Assign set of 5 daily tours to nearby inspectors
  - This gives a few complete **weekly solutions**

TOP: Assumptions and Data

- d tours to construct
- Locations: $1, \ldots, n$
- Duplicate depot (hotel): origin 0 and destination $n+1$
- $G = (V, A)$:
  - $V = \{0, 1, \ldots, n, n+1\}$
  - $A = \{(i,j) | i \in V^{n+1}, j \in V^0\}$ ($V^i = V \setminus \{i\}$)
- p_i: profit at location $i \in V$ ($p_0 = p_{n+1} = 0$)
- t_{ij}: time at arc $(i,j) \in A$ (travel from i to j + inspection at j; $t_{0(n+1)} = 0$)
- T: daily time limit

TOP: Generic Model

- $y_i^l = 1$, if $i \in V$ is selected in tour l, 0 otherwise
- $x_{ij}^l = 1$, if $(i,j) \in A$ is selected in tour l, 0 otherwise

$$\max \sum_{i \in V} \sum_{l=1}^{d} p_i y_i^l$$

$$\sum_{j \in V^i} x_{ij}^l = y_i^l \quad i \in V^{n+1}, l = 1, \ldots, d$$

$$\sum_{j \in V^i} x_{ji}^l = y_i^l \quad i \in V^0, l = 1, \ldots, d$$

$$y_0^l = y_{n+1}^l = 1 \quad l = 1, \ldots, d$$

$$\sum_{l=1}^{d} y_i^l \leq 1 \quad i \in V \setminus \{0, n+1\}$$

TOP: Generic Model (continued)

$$\sum_{(i,j)\in A} t_{ij} x_{ij}^l \leq T \quad l = 1, \ldots, d$$

$$y_i^l \in \{0, 1\} \quad i \in V, l = 1, \ldots, d$$

$$x_{ij}^l \in \{0, 1\} \quad (i, j) \in V, l = 1, \ldots, d$$

- Symmetry Breaking Constraints:

$$x_{0i}^l \leq x_{0j}^{l+1} \quad i \in V^0, j \in V^0, i \leq j, l = 1, \ldots, d-1$$

- Subtour Elimination Constraints (SEC):
  - Classical SEC involve exponential number of constraints: can be handled by cutting-plane method (a lot of time to develop)
  - "Compact" (polynomial size) formulation can be obtained through **multicommodity flows**

SEC: Multicommodity Flows

- w_{ij}^{kl}: flow on $(i,j) \in A$ destined to $k \in V^0$ in tour l

$$\sum_{j \in V^i} w_{ij}^{kl} - \sum_{j \in V^i} w_{ji}^{kl} = \begin{cases} y_k^l, & i = 0 \\ -y_k^l, & i = k \\ 0, & i \neq 0, k \end{cases} \quad i \in V, k \in V^0, l = 1, \dots, d$$

$$0 \leq w_{ij}^{kl} \leq x_{ij}^l \quad (i,j) \in A, k \in V^0, l = 1, \dots, d$$

TOP: Heuristic Methods

- Heuristic methods are often preferred to solving a model, as they can provide (effective) solutions much faster
- Here, we have to solve a **large** number of TOPs to "feed" the annual problem: it is crucial to solve them quickly!
- Even if we could solve the above multicommodity flow model, heuristic methods are interesting, as they provide starting solutions that speedup the solution of the model
- **Local Search** methods are among the most efficient and effective heuristics for VRPs

TOP: Local Search

- Location-based moves:
  - **Add** (i, j): add location i (not in tour) immediately after j (in tour)
  - **Switch** (i, j): insert location i (not in tour) in place of j (in tour)
- Routing-based moves:
  - **2-opt** (i, j): replace 2 arcs (one starting at i, the other ending at j) in a tour by 2 other arcs (intra-route)
  - **2-opt*** (i, j): replace 2 arcs (one starting at i, the other ending at j) in different tours by 2 other arcs (inter-route)
  - **Relocate** (i, j): remove a sequence of visits (starting at i and ending at j) from a tour and insert the sequence in a tour (intra- or inter-route)
  - **Swap** (i, j): swap locations i and j in a tour or in 2 tours (intra- or inter-route)

TOP: Local Search (continued)

- Typical Metaheuristic Framework:
  - Start from routes computed by greedy (randomized) heuristic (or from empty routes)
  - For each (i, j) (in random order), try each move and move to improving solution
  - Restart by perturbing the solution (Iterated Local Search) or by starting from scratch (Multistart/GRASP)
- Other Metaheuristics are possible: Simulated Annealing (SA), Tabu Search (TS), Variable Neighborhood Search (VNS)
- Suggested data structures for a route:
  - Doubly linked list + array of pointers: move performed in $O(1)$
  - Forward and backward times: feasibility checking in $O(1)$

Achievements

- A formalization of the problem presented by the company
- A "natural" decomposition based on planning horizon: year divided into weeks
- Mathematical programming models for annual and weekly problems
- Local Search heuristics for the weekly problem
- Python codes for:
  - Preprocessing and input/output
  - Solving model for the annual problem (with Gurobi)
  - Solving the weekly problem by Local Search

Forthcoming (?)

- Solve the weekly problem by Local Search/Mathematical Programming
- Solve the annual problem by heuristic column generation + Branch-and-Bound
- Solve the annual problem by LP-based column generation + Branch-and-Price
- Implement solution with the company