The Coastal Seaspace Sector Design and Allocation Problem

Brian J. Lunday[1], Hanif D. Sherali[2], Kevin E. Lunday[3]

[1]Department of Mathematical Sciences, United States Military Academy
[2]Grado Department of Industrial and Systems Engineering, Virginia Tech
[3]Headquarters, U.S. Coast Guard*

November 16, 2011

**The views expressed herein are those of the authors and do not represent the views of the Commandant nor any unit of the U.S. Coast Guard.*

Problem Statement

Given:

- A set of maritime vessels, $v \in V$, for patrolling a coastal seaspace
- A region of seaspace within 200 nmi of the a coastline
  - A discretization of the seaspace into areas, $j \in N$
  - A set of historical demands for each area
- A number of areas n_v to be assigned to each vessel

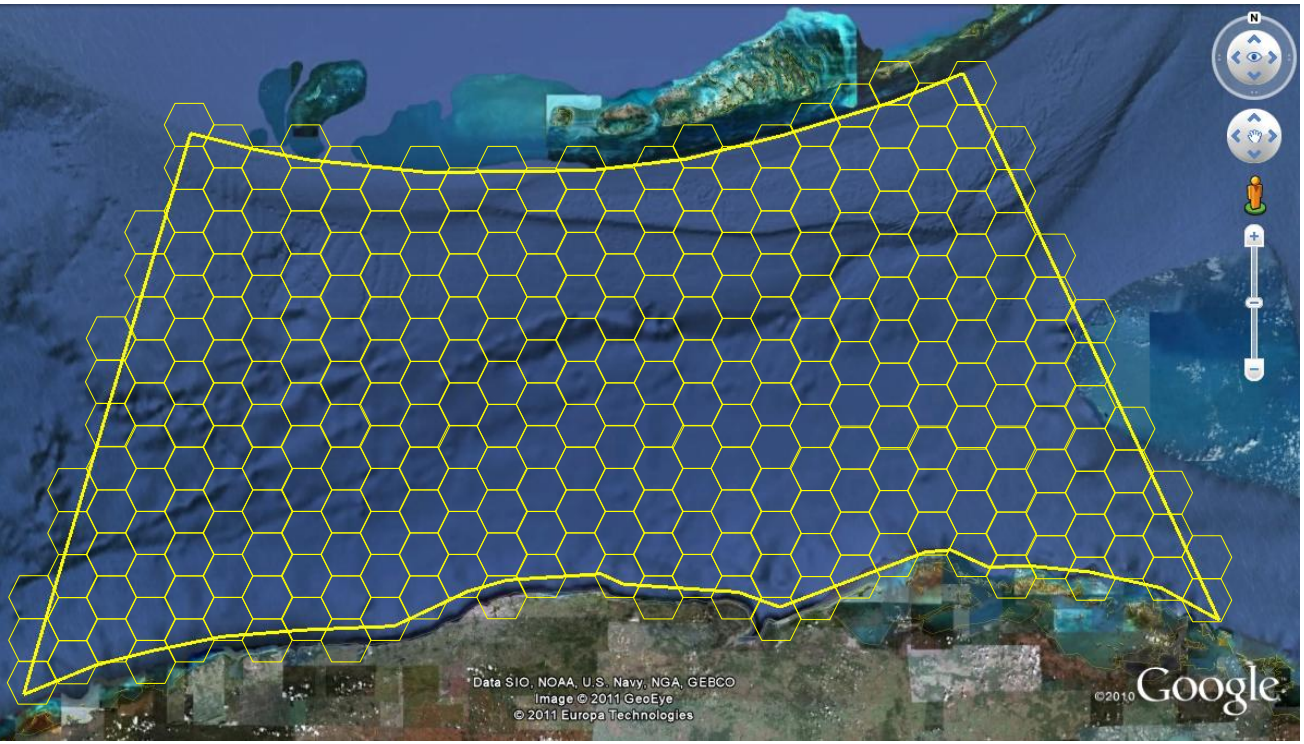
Find:

- A model that assigns n_v areas to a sector for each vessel $v \in V$ such that
  1. The areas form contiguous sectors
  2. We minimize a convex combination of the:
     - Maximum deviation of vessel's assigned historical demand from its proportionally expected demand
     - Maximum time from a vessel's home port to the farthest area in its sector
     - Maximum time-based span of the vessels' sectors

(Some related* problems: police sector design, sales territory design, school districting, political districting, snow removal sector design)

What does this problem look like?

The Seaspace
(Straits of Florida – 7520 nmi^2)



The Vessels

v	Home Port	Cruising Speed (knots)	Rel. Cap.*
1-2	Fort Pierce, FL	15	1
3	Dania Beach, FL	15	1
4-5	Miami, FL	15	1
6	Key West, FL	15	1
7	Fort Myers Beach, FL	15	1
8	San Juan, PR	15	1
9-11	Miami, FL	18	2
12-17	San Juan, PR	18	2
18-21	Key West, FL	18	2
22	Miami, FL	21	3

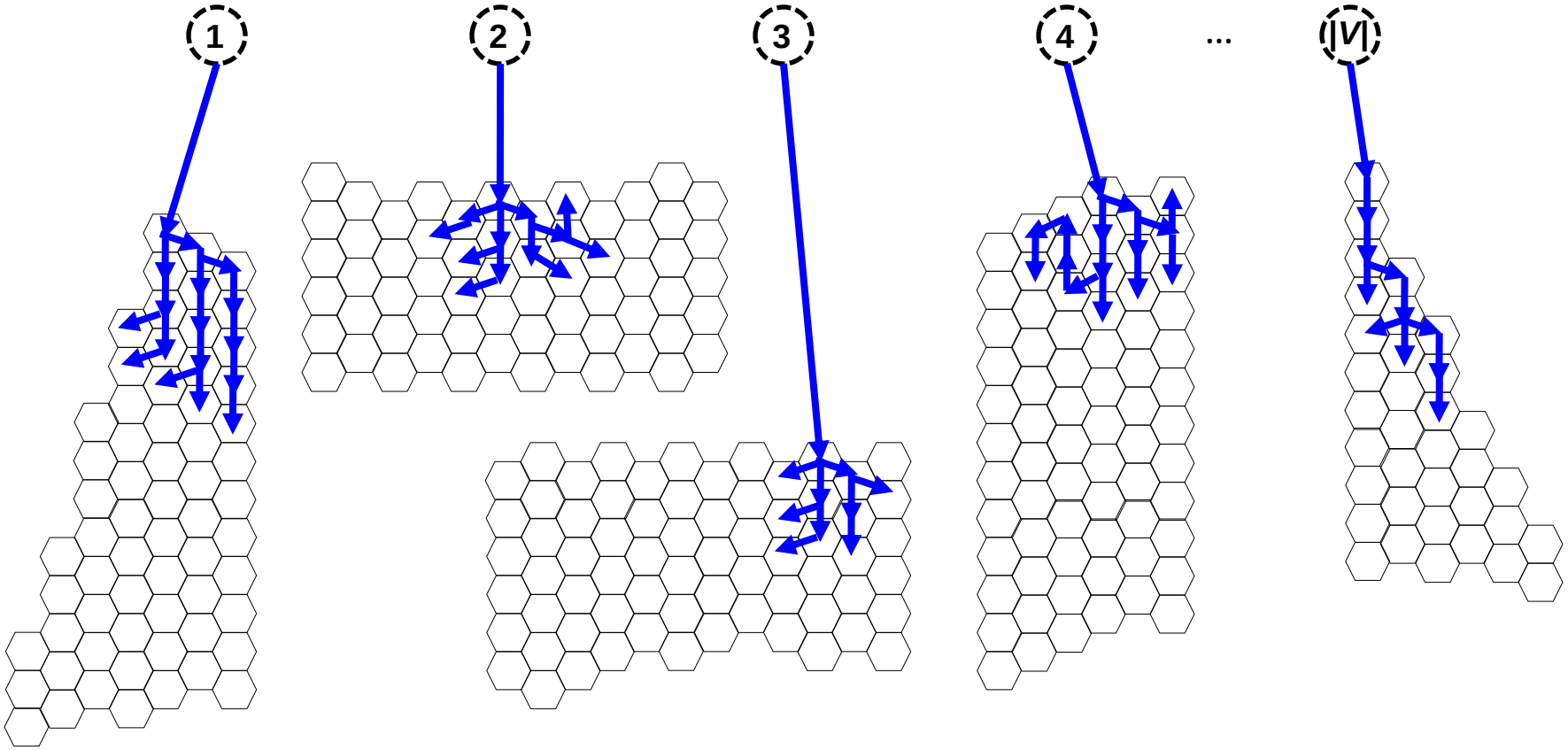
* Determined via AHP

- Discretize the seaspace for assignment

(Saaty, 1990; Yousefi and Donohue, 2004)

Our modeling approach: Construct a spanning arborescence

Create artificial nodes: $o_v \equiv \text{root node for each vessel}, v \in V$



Decision variables:

$s_j^v \equiv 1 \text{ if area } j \text{ assigned to vessel } v, 0 \text{ o/w}, \forall\ j \in N, v \in V$

$r_{ij}^v \equiv 1 \text{ if arc } (i,j) \text{ is in the arborescence constructed for vessel } v, 0 \text{ o/w}, \forall\ (i, j) \in A, v \in V$

Programming Formulation to Construct a Spanning Arborescence

$$\min \quad \text{(objective function)}$$

$$s.t. \quad \sum_{j \in N} s_j^v = n_v, \quad \forall\, v \in V,$$

Assign n_v areas to vessel v

$$\sum_{v \in V} s_j^v = 1, \quad \forall\, j \in N,$$

Assign each area to one-and-only-one vessel

$$\sum_{j \in N} r_{o_v j}^v = 1, \quad \forall\, v \in V,$$

Connect the root node to only one area in its tree

$$\sum_{i \in \delta^-(j) \cup \{o_v\}} r_{ij}^v = s_j^v, \quad \forall\, j \in N, v \in V,$$

Indegree=1 if area is assigned to vessel v

$$r_{ij}^v + r_{ji}^v \le s_j^v, \quad \forall\, (i, j) \in A, v \in V,$$

Arc requires both nodes to be assigned to v

$$\text{(enforce contiguity)}, \quad \forall\, v \in V,$$

The tricky part!

$$\text{(objective function related calculations)},$$

Workload, span, distance…

$$\left.\begin{aligned} & s_j^v \in \{0,1\}, \quad \forall\, j \in N, v \in V, \\ & r_{ij}^v \in \{0,1\}, \quad \forall\, (i, j) \in A^v, v \in V. \end{aligned}\right\}$$

Binary restrictions

So… how shall we enforce contiguity…

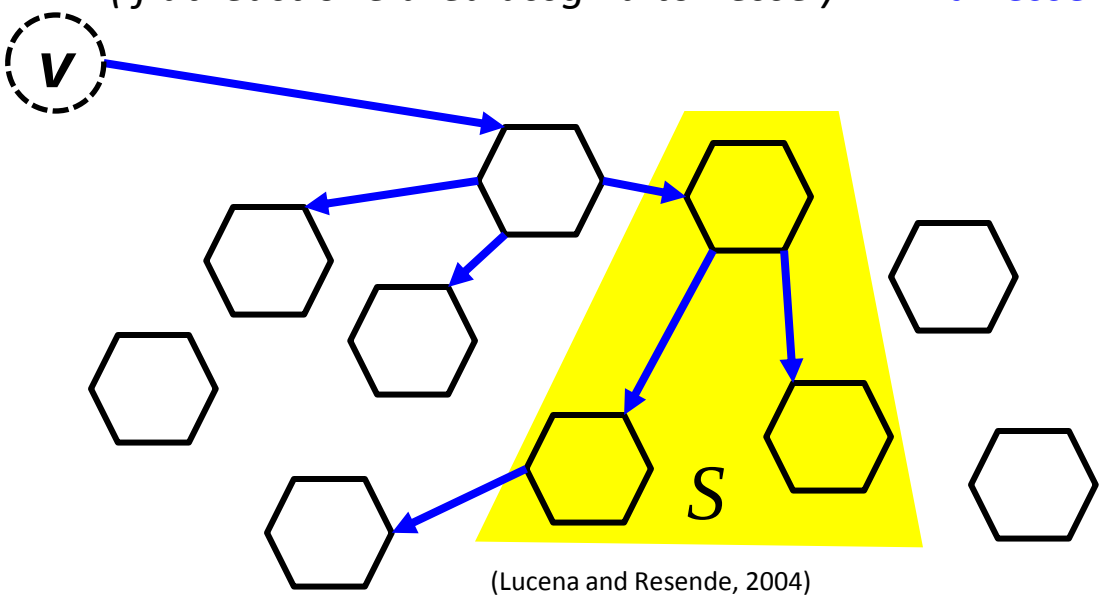
Contiguity Constraint Version #1
Subtour Elimination Constraint

$$\underbrace{\sum_{(i,j)\in A(S)} r_{ij}^{v} \le \left(\sum_{j\in S} s_j^{v}\right) - s_k^{v}, \quad \forall\, k \in S,}_{} \; \underbrace{S \subseteq N, 2 \le |S| \le n_v,}_{} \; \underbrace{v \in V}_{}$$

The number of arcs in a subset $\le$ The number of areas within a subset in a vessel's tree *(if at least one area assgn'd to vessel)* -1

For every subset of areas that *could* be in a vessel's tree

For every vessel



(Lucena and Resende, 2004)

Contiguity Constraint Version #2 (a & b) STP-RLT

New decision variable:

$$u_j^v \equiv \text{ no. of arcs in directed path from root node } o_v \text{ to node } j, \forall\ j \in N, v \in V$$

Constraints:

$$u_{o_v}^v \equiv 0, \quad \forall\ v \in V,$$ Fix the arc count for the root nodes of the arborescence

$$u_j^v r_{ij}^v = \left(u_i^v + 1\right) r_{ij}^v, \quad \forall\ (i, j) \in A^v, v \in V,$$ Increment count based on constructed arc

$$1 \le u_j^v \le n_v, \quad \forall\ j \in N, v \in V.$$ Bound the arc count

Proposition. Transformation via Reformulation Linearization Technique (RLT) yields two alternative, equivalent linear representations.

(Haouari et al., 2010; Miller et al. 1960; Sherali and Adams, 1990 and 1994; Sherali and Driscoll, 2002)

Contiguity Constraint Version #3
Multicommodity Flow

New decision variable:

$$\psi_{ij}^{kv} \equiv \text{ flow on arc } (i, j) \text{ of commodity } v \text{ that reaches area } k, \forall (i, j) \in A^v, k \in N, v \in V$$

Constraints:

$$\sum_{j \in N} \psi_{o_v j}^{kv} = s_v^k, \quad \forall k \in N, v \in V,$$

Outflow of one unit from root node *v*

$$\sum_{i \in \delta^-(k) \cup \{o_v\}} \psi_{ik}^{kv} = s_v^k, \quad \forall k \in N, v \in V,$$

Inflow of one unit into node *k*

$$\psi_{kj}^{kv} \equiv 0, \quad \forall (k, j) \in A, v \in V,$$

Outflow from node *k* is 0

$$\sum_{i \in \delta^-(j) \cup \{o_v\}} \psi_{ij}^{kv} - \sum_{i \in \delta^+(j)} \psi_{ji}^{kv} = 0, \quad \forall k \in N, j \in N \setminus \{k\}, v \in V,$$

Conservation of flow at all other nodes

$$0 \le \psi_{ij}^{kv} \le r_{ij}^v, \quad \forall (i, j) \in A^v, k \in N, v \in V.$$

Flow can only be nonzero if arc *(i,j)* is in the arborescence for *v*

(Haouari et al., 2010)

Modeling Enhancements

- Symmetry Defeating Constraints

  1A Smallest indexed assigned node as the virtual root node

$$s_i^v \le \left(1 - r_{o_v j}^v\right), \quad \forall\ j \in N, i \in N : i < j\ , v \in V$$

  1B Weaker but more compact version of 1A

$$\sum_{i \in N : i < j} s_i^v \le \left(1 - r_{o_v j}^v\right) \min\left\{n_v \mid \left\{i \in N : i < j\right\}\right\}, \quad \forall\ j \in N, v \in V$$

  2 Defeat the symmetry among the like vessels with same portage

$$\sum_{j \in N} j r_{o_{v_1} j}^v \le \left(\sum_{j \in N} j r_{o_{v_2} j}^v\right) - 1, \quad \forall\ v_1 < v_2 \in V_w,\ w = 1, \ldots, W$$

- Objective Function Perturbation
  - Non-preemptively weight the symmetry defeating constraint(s)

(Ghoniem and Sherali, 2011; Sherali and Soyster, 1983)

Comparative Testing

- 4 alternative formulations
- 3 alternatives for solution enhancement techniques
  - Direct CPLEX application (i.e., ‘none’)
  - CPLEX symmetry breaking technique
  - Non-hierarchical objective perturbation
- 6 combinations of 3 symmetry defeating constraints
  - 2 methods regarding the area node connected to root node
  - 1 method regarding like-vessel symmetry
- Implementation Environment
  - C++ invoking CPLEX 12.1
  - Intel Model T7100 Processor with 4GB RAM

Comparative Testing Results (1 of 3)

- Modeling Strategies (6 hrs CPU time, ε=5%)

Strategy	Feasible Solution	ε-optimal Solution	Best Solution
Subtour Elimination	100%	82.4%	5.9%
STP-RLT #1	100%	82.4%	47.1%
STP-RLT #2	100%	82.4%	41.1%
Multicommodity Flow*	31.3%	6.0%	5.9%

MC Flow reformulations only tractable when:

- *Invoking symmetry defeating constraint(s)*
- *Non-preemptively perturbing the objective function*

- Objective function perturbation (6 hrs CPU time, ε=5%)

Technique	Feasible Solution	ε-optimal Solution	Best Solution
None; CPLEX default settings	75%	70.8%	10%
Non-preemptive	100%	45.0%	60%
CPLEX symmetry breaking option	75%	70.8%	30%

Comparative Testing Results (2 of 3)

- Symmetry Defeating Constraints (6 hrs CPU time, ε=5%)

Modeling	Feasible Solution	ε-optimal Solution	Best Solution
Root Node #1	83.3%*	83.3%	8.3%
Root Node #2	83.3%*	58.3%	33.3%
Like Vessels	83.3%*	33.3%	33.3%
Root Node #1 & Like Vessels	83.3%*	75.0%	8.3%
Root Node #2 & Like Vessels	83.3%*	58.3%	0%

** MC Flow reformulations only tractable when non-preemptively perturbing the objective function*

- Same performance trends confirmed over variants of the instance based on altering
  - The relative optimality gap ε
  - Time computational time limit
  - The vessel set (higher demand load per vessel)
  - The historical demand set (as distributed over the seaspace)

Comparative Testing Results (3 of 3)

- A closer look at the top four combinations (24 hrs CPU time, ε=1%)

Contiguity Modeling	Symmetry Defeat Mech.	Obj. Pert	CPU Time (hrs)	Assured Opt Gap
STP-RLT #1	Root Node #2	Non-pre.	24.0	13.14%*
STP-RLT #2	Root Node #2	Non-pre.	24.0	12.28%*
STP-RLT #2	Root Node #2 & Like Vessels	Non-pre.	24.0	13.98%
Subtour Elimin.	None	n/a	8.29	1%

** Within 0.0196% of optimal, based on assured optimality gap of final solution*

- Same performance trends confirmed over variants of the instance based on altering
  - The vessel set (higher demand load per vessel)
  - The historical demand set (as distributed over the seaspace)

Conclusions & Future Research

Conclusions

- STP-RLT reformulations for implementation to obtain a near-optimal solution
- STP-RLT reformulations, combined with symmetry defeat constraints and objective perturbation, are promising as heuristics

Future Research

- Larger instances (and/or tighter discretizations)
- Alternative methods to discretize the seaspace
- Modification of AHP vessel ratings to defeat symmetry
- Allow for alternative portage of vessels

Questions