Search and Surveillance in Emergency situations – A GIS based approach to construct near-optimal visibility graphs

CERMID (Centre de recherche en modélisation, information et décision)

Université Laval
Québec, Canada

M. Morin

Irène Abi-Zeid

T.T. Nguyen
L. Lamontagne
P. Maupin

ISCRAM 2013
May 13th

UNIVERSITÉ LAVAL

Summary of contributions

- **Integration** of GIS, computational geometry, and integer linear programming
  - to design optimal **visibility** graphs in real time
  - for surveillance **coverage** of an area
  - from **structured** and **unstructured** outdoor environments
  - using **vector** or **raster** data.

Presentation Outline

- Project background
- Methodology
- Experimental results
- Conclusion

Project background

- In an emergency situation, the ability to **observe** an environment, completely or partially, is **crucial** when searching an area for survivors, missing persons, intruders or anomalies

- Where should the **observers** be placed?

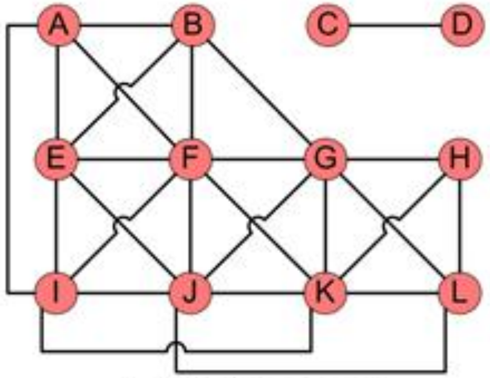
- Project funded by Department of National Defence Canada (DRDC – Valcartier) and the Network of Centers of Excellence MITACS

Project background

- Activities are part of a project for optimal **detection search planning**:
  - Where to deploy search efforts in order to maximize probabilities of detection
    - Search and Rescue
    - Surveillance
- **Input** to search planning:
  - An abstract representation of a terrain in the form of a **visibility graph**

Project background

- General objective : Construct **optimal** visibility graphs with the smallest number of observers
- A visibility graph consists of a set of vertices in an environment such that two vertices are connected by an edge if they are **inter-visible**



A visibility graph

Project background

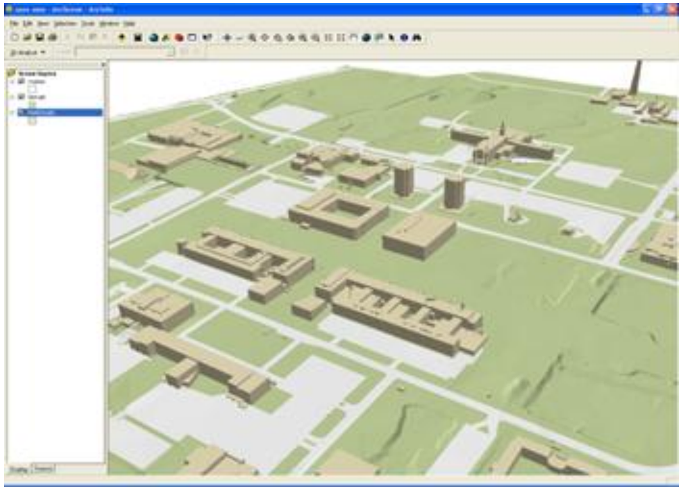
- Specific objectives
  - Find the **smallest number of observers** necessary, whether they are human spotters, sensors or cameras, and their positions in order to cover an area
  - Given a fixed number of observers, position the observers in such way to **maximize the visibility coverage** of the vertices

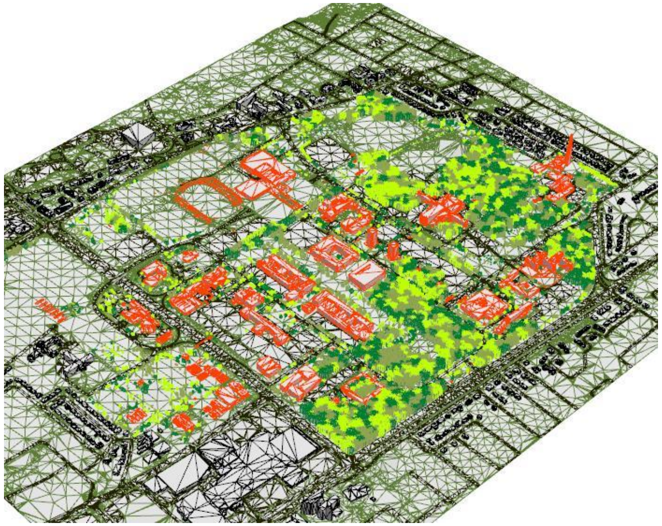
Methodology

I. **Processing** terrain data and construct a visibility graph
   - Vector data: computational geometry algorithm
   - Raster data: viewshed analysis in ArcGIS

II. **Optimization** using integer linear programming and the visibility graph
   - Formulate and solve the set covering problem
   - Formulate and solve the maximum coverage problem

Methodology - Processing vector data

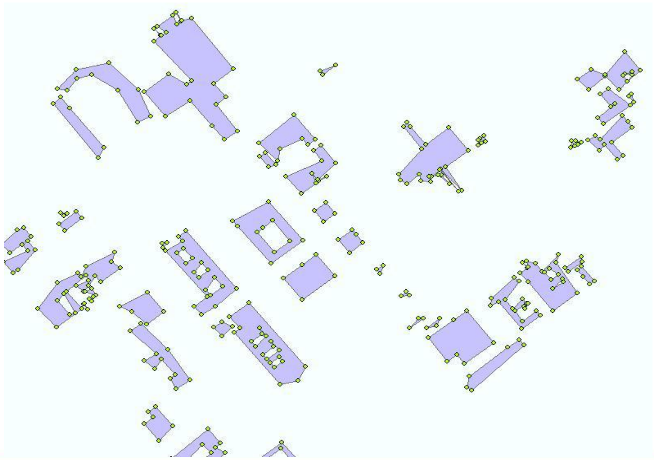




Laval University campus – structured environment

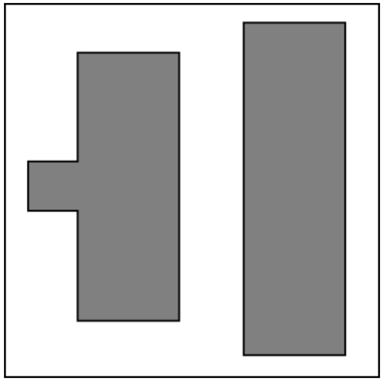
Methodology - Processing vector data

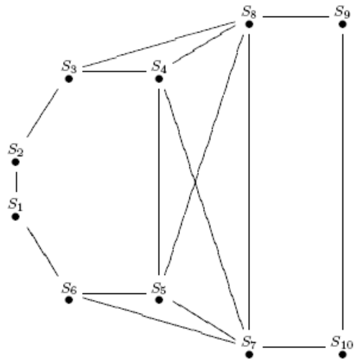
- Using ArcGIS:
    - Extract the buildings layer as polygons
    - Add points to the vertices of the polygons
    - Group the connected polygons into a single polygon



Methodology - Processing vector data

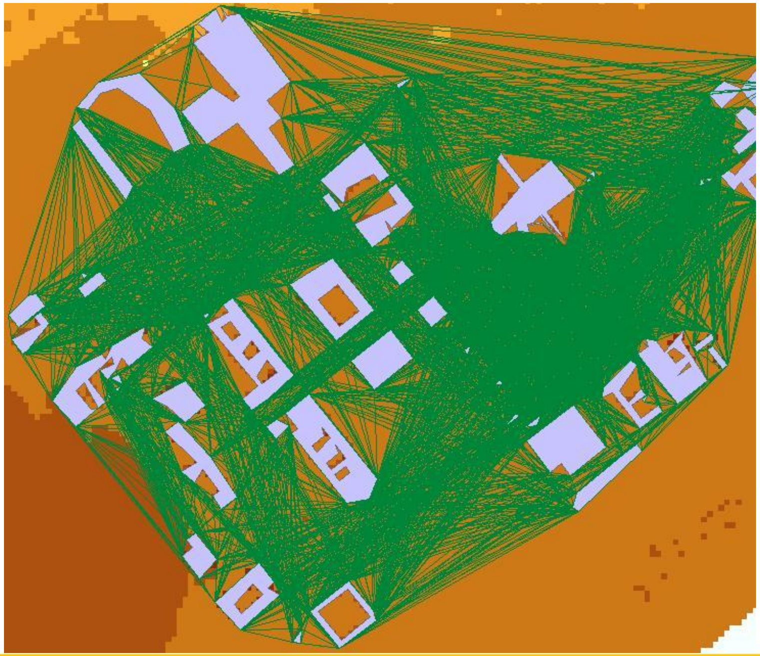
- Construct a visibility graph from a bidimensional environment defined by a set of polygons representing obstacles (VisiLibity and CGAL libraries)
- An edge connects two vertices if they are not separated by an obstacle
- Only critical vertices are included in the visibility graph: angle formed by adjacent vertices is larger than 180°





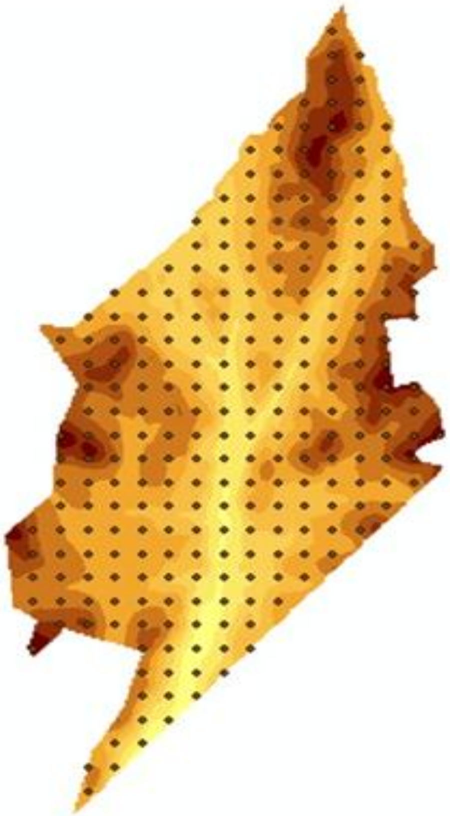
Methodology – Processing vector data

- Structured environment
- Laval University Campus
- Visibility graph: **255** vertices



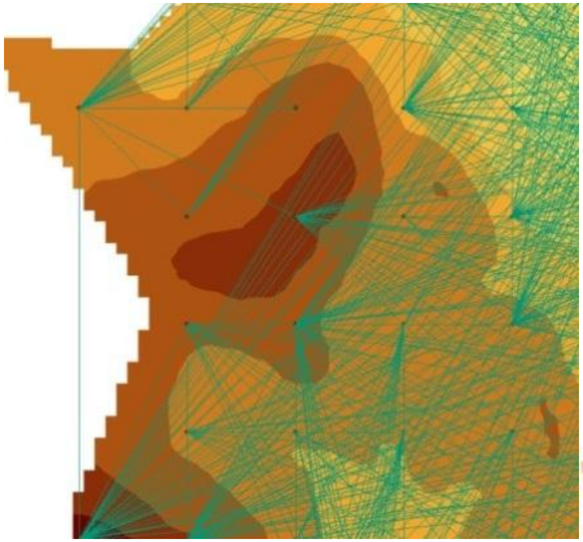
Methodology - Processing raster data

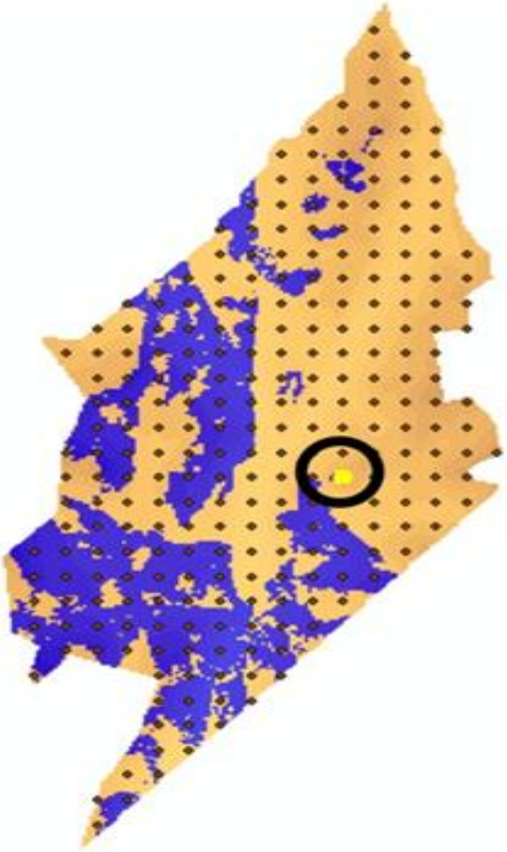
- **Unstructured** environment
- Montmorency Forest near Québec city
- Area approximately 66 km^2
- Superimpose over the digital terrain elevation model a **uniform grid** of square cells with a width of 50 m
- Assign a vertex to the **center** of each cell



Methodology - Processing raster data

- Using ArcGIS **Viewshed Analysis** determine inter-visible points within a maximum distance of 1 km
- **6025** vertices





Methodology: Optimization – Minimize number of observers

- Minimize the number of observers on a visibility graph such that all vertices are covered: **set covering** problem

$$\text{minimize } \sum_{i=1}^{n} y_i$$

$$\text{such that: } \sum_{i=1}^{n} x_{ji} y_i \geq 1$$

$$j = 1..n$$

$$y_i \in \{0,1\}$$

$y_i = 1$ if there is an observer at vertex i
0 otherwise

$x_{ji} = 1$ if vertex j is visible from vertex i
0 otherwise

Methodology: Optimization – Maximize coverage

- Given a number of observers *p*, minimize the number of vertices uncovered: **maximum coverage** problem

$$\text{minimize} \sum_{i=1}^{n} z_i$$

$$\text{subject to} \quad \sum_{i=1}^{n} y_i = p$$

$$\sum_{i=1}^{n} x_{ji} y_i = 1 - z_j$$

$$j = 1..n$$

$$y_i,\ z_i \in \{0,1\}$$

y_i = 1 if there is an observer at vertex *i*
0 otherwise

x_{ji} = 1 if vertex *j* is visible from vertex *i*
0 otherwise

z_i = 1 if vertex *i* is not visible by any observer
0 otherwise

Experimental results

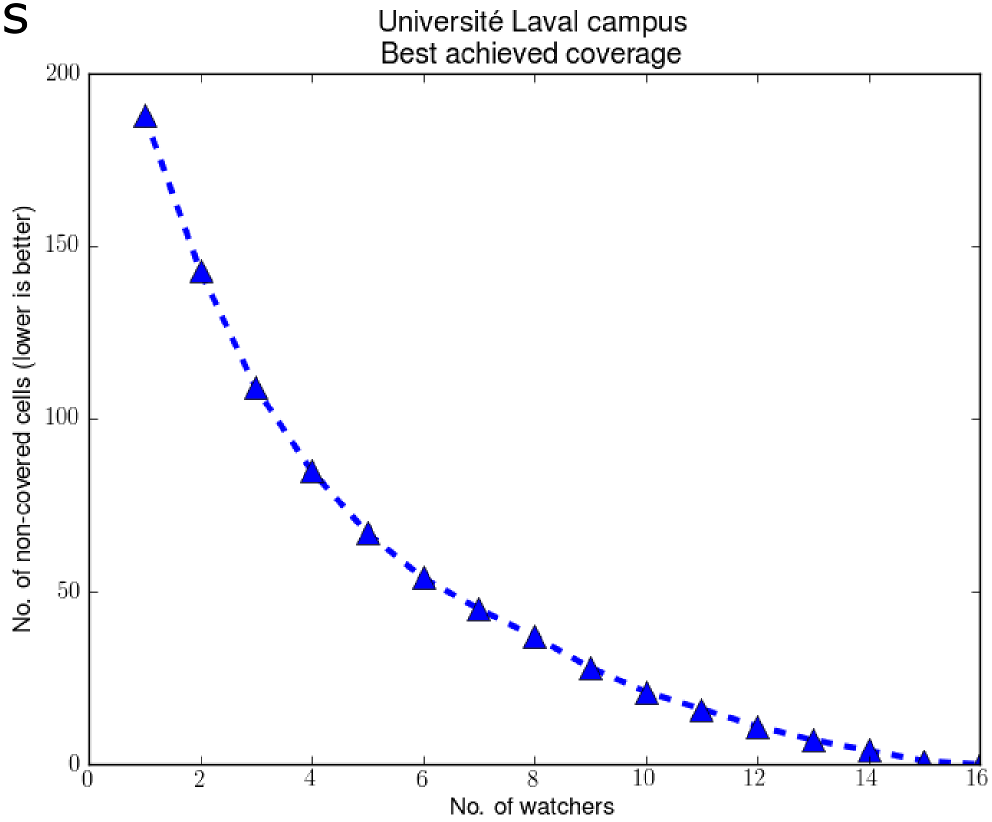
- ArcGIS 9.2 with VBA
- C++
- VisiLibity, Boost, CGAL libraries
- CPLEX 12.5, OPL
- All experiments were run on an Intel i7 Q740 processor with 8GB of RAM.
- Structured environment (vector): 255 vertices
- Unstructured environment (raster): 6025 vertices

Experimental results – Structured environment

- Minimum number of observers solved to optimality: 16 to cover 255 vertices
- Obtained in less than 1 second
- No feasible solution if multiple coverage is not allowed

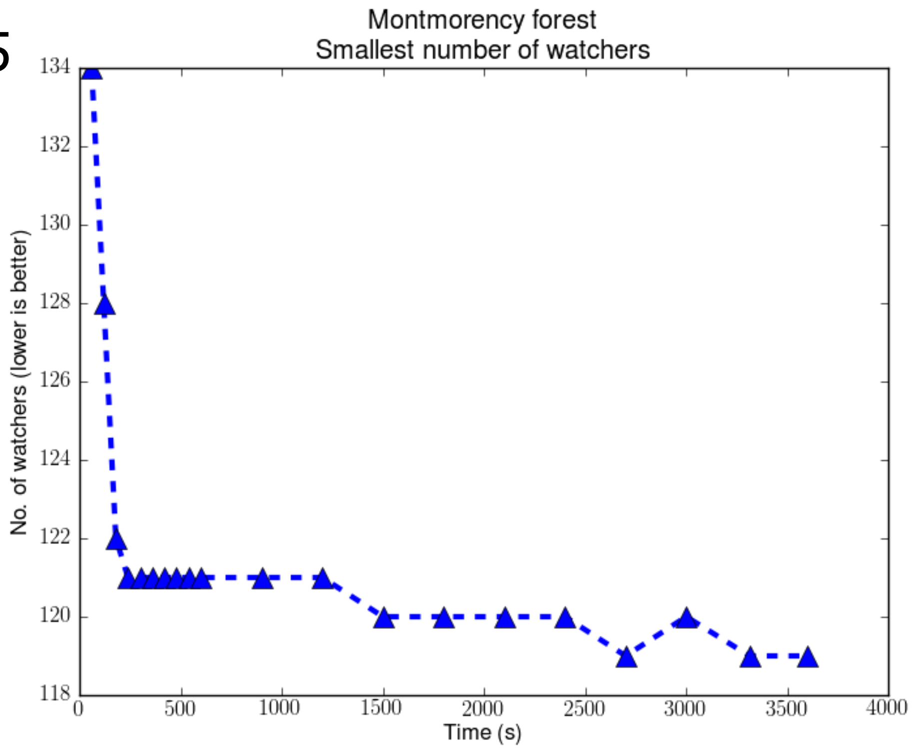
Experimental results – Structured environment

- Minimise number of non-covered vertices



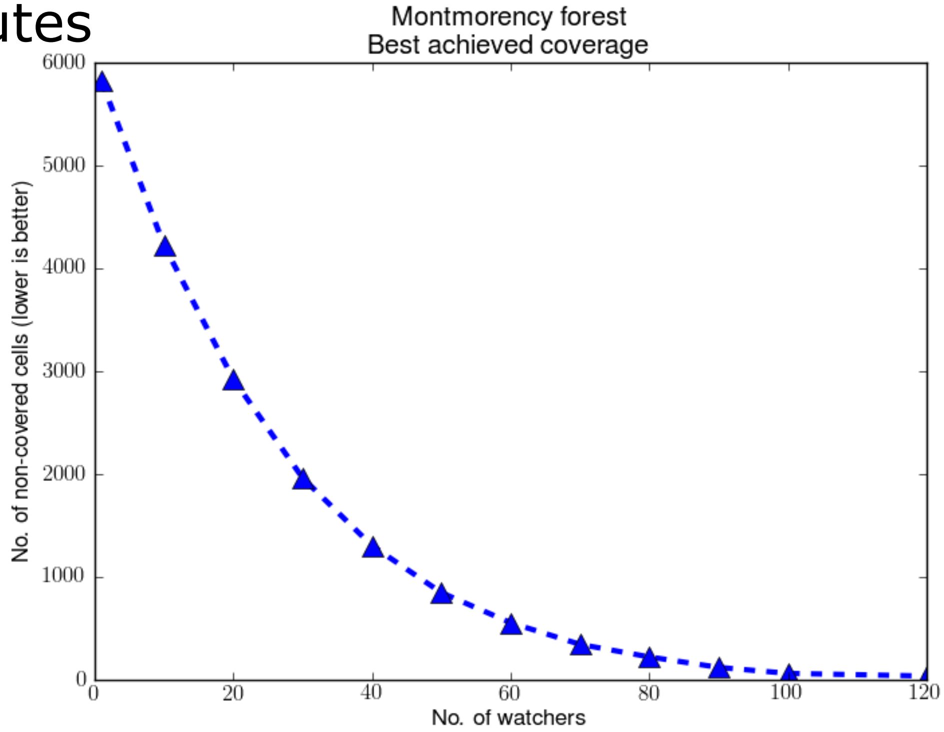
Experimental results – Untructured environment

- Minimize number of observers to cover 6025 vertices
- After 4 minutes: 121 observers
- After 45 minutes: 119 observers
- After 12 hours 118 observers (best solution)
- Not able to prove optimality on this instance



Experimental results – Unstructured environment

- Minimize number of non-covered vertices out of 6025 vertices – allowed solution time is 10 minutes



Experimental results – Unstructured environment

- Minimize non-coverage of 6025 vertices – maximum allowed time is 10 minutes
- With 100 observers after 10 minutes: 1% is left unobserved
- With 120 observers after 10 minutes: .05% is left unobserved
- For example, after 1 hour, only 2 are left unobserved by 120 observers

No. of Observers	Time (s)	No. of non-covered cells
1	2.4	5820
10	3.7	4225
20	7.5	2923
30	65.7	1962
40	420.4	1304
50	600	848
60	600	555
70	600	345
80	600	223
90	600	121
100	600	63
120	600	37

Conclusion

- We have presented an approach integrating a GIS, Integer linear programming and computational geometry to obtain optimal visibility graphs
  - Minimize number of observers for complete coverage
  - Maximize coverage with a given number of observers
  - Set covering (minimize number of observers) formulation seems more efficient
  - Both are NP-hard problems

Conclusion

- In critical situations with short response times, an optimal visibility graph, computed in a reasonable time, provides an efficient basis for real time planning of complex emergency operations
- Future work involves more experimentations and verification of the robustness of the integrated tool
  - Take into account priority area coverage

QUESTIONS?

Thank you for your attention

irene.abi-zeid@osd.ulaval.ca

References

- Dawes, B., Abrahams, D., and Rivera, R. (2009) Boost C++ libraries, *http://www.boost.org*.
- De Berg, M. (2000) Computational geometry: algorithms and applications. Springer Science & Business.
- De Floriani, L., and Magillo, P. (2003) Algorithms for visibility computation on terrains: a survey. *Environment and Planning B*, *30*(5), 709-728.
- Emrich, C. T., Cutter, S. L., and Weschler, P. J. (2011) GIS and emergency management. *The SAGE Handbook of GIS and Society,* London, Sage, 321-43.
- Fabri, A., and Pion, S. (2009) CGAL: The computational geometry algorithms library, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 538-539.
- Garey, M. R., and Johnson, D. S. (1979) Computers and intractability*: A guide to the theory of NP-Completeness*. San Francisco, CA. Freeman.
- Goodchild, M. F. and Lee, J. (1989) Coverage problems and visibility regions on topographic surfaces. *Annals of Operations Research*, 18, 175–186.
- Lamontagne L., Rouet F. H., Abi-Zeid I., in collaboration with J.-F. Potvin (2008), Étude d'algorithmes de poursuite-évasion sur graphes de visibilité, Research Report II, 40 pages.

References

- Latombe, J.C. (1990) Robot Motion Planning, Springer International Series in Engineering and Computer Science.
- Morin, M., Lamontagne, L., Abi-Zeid, I., Lang, P., and Maupin, P. (2010) The Optimal Searcher Path Problem with a Visibility Criterion in Discrete Time and Space, *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, 2217-2224.
- Morin, M., Papillon, A.P., Laviolette, F., Abi-Zeid, I., and Quimper, C.G. (2012) Constraint Programming for Path Planning with Uncertainty: Solving the Optimal Search Path problem, *Proceedings of the 18th Conference on Principles and Practice of Constraint Programming*, Québec, Canada, 2012, 988-1003.
- Murray, A. T., Kim, K., Davis, J. W., Machiraju, R. and Parent, R. (2007) Coverage Optimization to Support Security Monitoring, *Computers, Environment and Urban Systems*, 31, 133-147.
- Obermeyer, K. J. and Contributors (2008), The VisiLibity library, A C++ library for floating-point visibilitycomputations, *http://www.VisiLibity.org*.
- Pellier, D., and Fiorino, H. (2005) Coordinated exploration of unknown labyrinthine environments applied to the pursuit evasion problem, *Proceedings of the fourth international joint conference on Autonomous Agents and Multiagent Systems* (pp. 895-902).