Constraint Programming for Path Planning with Uncertainty
Solving the Optimal Search Path Problem

Michael Morin[1] Anika-Pascale Papillon[2] Irène Abi-Zeid[3] François Laviolette[1] Claude-Guy Quimper[1]
[1]Department of Computer Science and Software Engineering [2]Department of Mathematics and Statistics [3]Department of Operations and Decision Systems
Université Laval, Québec, Qc, Canada

UNIVERSITÉ LAVAL

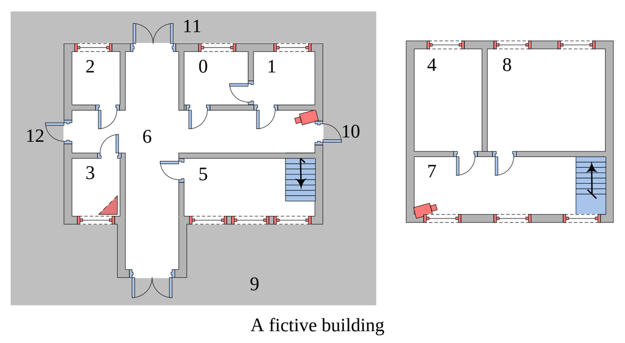
Abstract

The optimal search path (OSP) problem is a single-sided detection search problem from search theory [1] where the location and the detectability of a moving object are uncertain. A solution to this $\mathcal{NP}$-hard problem [2] is a path on a graph that maximizes the probability of finding an object that moves according to a known motion model. We developed constraint programming models to solve this probabilistic path planning problem for a single indivisible searcher. These models include a simple but powerful branching heuristic as well as strong filtering constraints. The OSP problem is particularly interesting in that it generalizes to various probabilistic search problems such as intruder detection, malicious code identification, search and rescue, and surveillance.

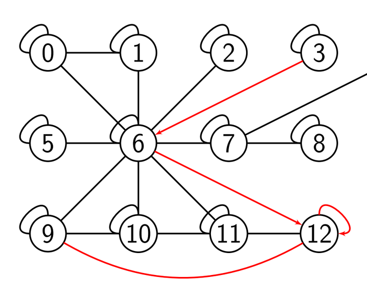
The OSP Problem

- Find an optimal search plan $P = [y_0, y_1, \dots, y_T]$ on a graph $G_A = (\mathcal{V}(G_A), \mathcal{E}(G_A))$ maximizing the cumulative overall probability of success (COS) defined as:

$$COS(P) = \sum_{t \in \{1,\dots,T\}} \sum_{r \in \mathcal{V}(G_A)} pos_t(r).$$



A fictive building



Assumptions

- A positive detection of the object stops the search.
- The object's movements are independent of the searcher's actions.

Definitions

- $\mathbf{M}$ is the Markovian motion model matrix, and $\mathbf{M}(s,r)$ is the probability of the object moving from vertex s to vertex r within one time step:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}(0,0) & \mathbf{M}(0,1) & \cdots & \mathbf{M}(0,|\mathcal{V}(G_A)|-1) \\ \mathbf{M}(1,0) & \mathbf{M}(1,1) & \cdots & \mathbf{M}(1,|\mathcal{V}(G_A)|-1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}(|\mathcal{V}(G_A)|-1,0) & \mathbf{M}(|\mathcal{V}(G_A)|-1,1) & \cdots & \mathbf{M}(|\mathcal{V}(G_A)|-1,|\mathcal{V}(G_A)|-1) \end{pmatrix}.$$

- The initial probability of containment distribution: poc_1.
- The local probability of success ($\forall t \in \{1,\dots,T\}$):

$$pos_t(r) = poc_t(r) \times pod(r).$$

- The probability of detection (conditional to the presence of the object):

$$\begin{aligned} pod(r) &\in (0,1], & \text{if } y_t = r; \\ pod(r) &= 0, & \text{otherwise.} \end{aligned}$$

- The local probability of containment ($\forall t \in \{2,\dots,T\}$):

$$poc_t(r) = \sum_{s \in \mathcal{V}(G_A)} \mathbf{M}(s,r)\left[poc_{t-1}(s) - pos_{t-1}(s)\right].$$

Blue terms are a priori known probabilities.

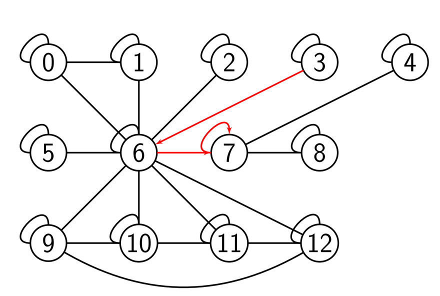
An Optimal Search Plan Example

- Let $T = 5$, $y_0 = 3$, $poc_1(4) = 1.0$, $pod(y_t) = 0.9$ ($\forall t \in \{1,\dots,T\}$), and assume a uniform Markovian motion model between accessible vertices.

	Probability of containment in vertex r at time t ($poc_t(r)$)													
t \ r	0	1	2	3	4	5	6	7	8	9	10	11	12	$COS_t(P^*)$
1	-	-	-	-	1	-	-	-	-	-	-	-	-	0
2	-	-	-	-	.500	-	-	.050	-	-	-	-	-	.450
3	-	-	-	-	.263	-	.012	.026	.012	-	-	-	-	.686
4	.001	.001	.001	.001	.138	.001	.008	.015	.013	.001	.001	.001	.001	.817
5	.001	.001	.001	.001	.073	.001	.008	.008	.01	.002	.002	.002	.002	.889

- P^* is the optimal search plan:

$$P^* = [y_0, y_1, \dots, y_5] = [3, 6, 7, 7, 7, 7].$$



A CP Model for the OSP

The Variables and the Constraints

- The variables and the constraints are given by the problem definition.

The Objective Function

- Two equivalent objective functions with a different performance:
  - First choice: The double sum definition

    max COS,
    $$COS = \sum_{t \in \{1,\dots,T\}} \sum_{r \in \mathcal{V}(G_A)} POS_t(r).$$
  - Poor filtering = poor bound:
    $$\lceil COS \rceil = \sum_{t \in \{1,\dots,T\}} \sum_{r \in \mathcal{V}(G_A)} \lceil POS_t(r) \rceil.$$
  - Second choice: The sum and max definition

    max COS,
    $$COS = \sum_{t \in \{1,\dots,T\}} \max_{r \in \mathcal{V}(G_A)} POS_t(r).$$
  - Better filtering = better bound:
    $$\lceil COS \rceil = \sum_{t \in \{1,\dots,T\}} \max_{r \in \mathcal{V}(G_A)} \lceil POS_t(r) \rceil.$$

VARIABLES are displayed in UPPER case and constants are displayed in lower case.

The Total Detection Heuristic

The intuition

- Ignore negative information when searching.
- What is the most promising vertex?
  - The one with the highest *total probability* of detecting the object in the remaining time.

The heuristic

- First, we order the decision variables: $Y_0, Y_1, \dots, Y_T$.
- Second, we order the domains of the variables:

$$\operatorname*{argmax}_{y' \in \mathrm{dom}(Y_t)} \sum_{o \in \mathcal{V}(G_A)} w_t(y', o) POC_t(o), \qquad \forall t \in \{1,\dots,T\}.$$

- $w_t(y', o)$ is computed using dynamic programming and the following data:
  - the Markovian motion model matrix $\mathbf{M}$;
  - the probability of detection pod.

The recurrence relation

- Let $w_t(y, o)$ be the conditional probability that the searcher detects the object before the end of the search given that, at time t, the searcher is in y and the object in o:

$$w_t(y,o) \stackrel{\text{def}}{=} \begin{cases} pod(o), & \text{if } o = y \text{ and } t = T, \\ 0, & \text{if } o \neq y \text{ and } t = T, \\ p_t(y,o), & \text{if } o \neq y \text{ and } t < T, \\ pod(o) + (1 - pod(o))p_t(y,o), & \text{if } o = y \text{ and } t < T. \end{cases}$$

where

$$p_t(y,o) = \sum_{o' \in \mathcal{N}(o)} \mathbf{M}(o,o') \max_{y' \in \mathcal{N}(y)} w_{t+1}(y', o').$$

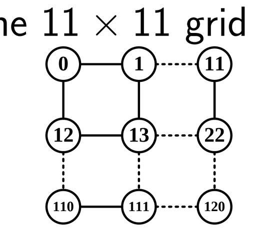
Experimentation

- Java implementation:
  - Choco solver [3]
  - Java Universal Network/Graph 2.0.1 framework [4]
- All experiments are run within a 20 minutes time limit and a maximum of 5,000,000 backtracks is allowed.
- Three different probabilities of detection: $pod(r) \in \{0.3, 0.6, 0.9\}$ ($\forall r \in \mathcal{V}(G_A)$)
- Three different graph structures:
- Three different motion models:

$$\mathbf{M}(s,r) = \begin{cases} \frac{1-\rho}{\deg(s)-1}, & \text{if } (s,r) \in \mathcal{E}(G_A), \\ \rho, & \text{if } s = r, \end{cases}$$

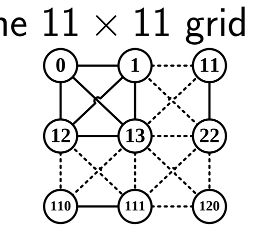
where $\deg(s)$ is the degree of s and $\rho \in \{0.3, 0.6, 0.9\}$ is the probability that the object stays in its current location.
- Six different allowed time values: $T \in \{9, 11, 13, 15, 17, 19\}$.

The 11×11 grid G^+



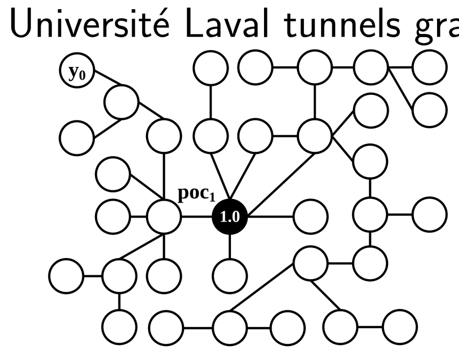
$poc_1(60) = 1$
$y_0 = 0$

The 11×11 grid G^*



$poc_1(60) = 1$
$y_0 = 0$

The Université Laval tunnels graph G^L



Results and Discussion

Phase 1: Comparing the CP Models

Table: The COS value of the last incumbent solution on a 11×11 G^+ grid with $T = 17$.

		CpMax		CpSum	
$pod(r)$	ρ	Time (s)	COS value	Time (s)	COS value
0.3	0.3	**1197.15**	**0.0837**	1045.66	0.0831
	0.6	**1198.56**	**0.1276**	990.61	0.1267
	0.9	**1026.02**	**0.3379**	1165.88	0.3379
0.6	0.3	**959.18**	**0.1532**	999.45	0.1532
	0.6	**1168.98**	**0.2202**	1015.64	0.2172
	0.9	**1166.29**	**0.5122**	942.36	0.5014
0.9	0.3	**1161.59**	**0.2162**	1184.86	0.2162
	0.6	**692.16**	**0.3151**	727.57	0.3151
	0.9	**1169.91**	**0.6283**	879.59	0.6252

Table: The COS value of the last incumbent solution on a 11×11 G^* grid with $T = 17$.

Heuristic:		CpMax		CpSum	
$pod(r)$	ρ	Time (s)	COS value	Time (s)	COS value
0.3	0.3	**1026.26**	**0.1016**	634.10	0.0969
	0.6	**1100.28**	**0.1764**	915.02	0.1634
	0.9	**960.89**	**0.5142**	376.27	0.4804
0.6	0.3	**787.87**	**0.1805**	1112.99	0.1751
	0.6	**1167.45**	**0.2959**	1042.35	0.2815
	0.9	**850.21**	**0.6606**	389.75	0.6420
0.9	0.3	**942.42**	**0.2464**	811.48	0.2459
	0.6	**796.64**	**0.4081**	680.55	0.3961
	0.9	**214.96**	**0.7491**	1004.03	0.7491

Phase 2: Evaluating the TD Value Selection Heuristic

Table: The COS value of the last incumbent solution on a 11×11 G^+ grid with $T = 17$.

		TDValSel+CpMax		CpMax	
$pod(r)$	ρ	Time (s)	COS value	Time (s)	COS value
0.3	0.3	**5.31**	**0.1055**	1197.15	0.0837
	0.6	**19.42**	**0.1645**	1198.56	0.1276
	0.9	**3.70**	**0.4418**	1026.02	0.3379
0.6	0.3	**159.42**	**0.1893**	959.18	0.1532
	0.6	**31.02**	**0.2978**	1168.98	0.2202
	0.9	**225.66**	**0.6559**	1166.29	0.5122
0.9	0.3	**54.94**	**0.2595**	1161.59	0.2162
	0.6	**37.73**	**0.4119**	692.16	0.3151
	0.9	**467.34**	**0.8194**	1169.91	0.6283

Table: The COS value of the last incumbent solution on a 11×11 G^* grid with $T = 17$.

		TDValSel+CpMax		CpMax	
$pod(r)$	ρ	Time (s)	COS value	Time (s)	COS value
0.3	0.3	**1071.56**	**0.1309**	1026.26	0.1016
	0.6	**1007.22**	**0.2218**	1100.28	0.1764
	0.9	**437.64**	**0.6046**	960.89	0.5142
0.6	0.3	**517.75**	**0.2267**	787.87	0.1805
	0.6	**366.96**	**0.3478**	1167.45	0.2959
	0.9	**830.30**	**0.7535**	850.21	0.6606
0.9	0.3	**838.03**	**0.3015**	942.42	0.2464
	0.6	**555.09**	**0.4368**	796.64	0.4081
	0.9	**1155.64**	**0.8280**	214.96	0.7491

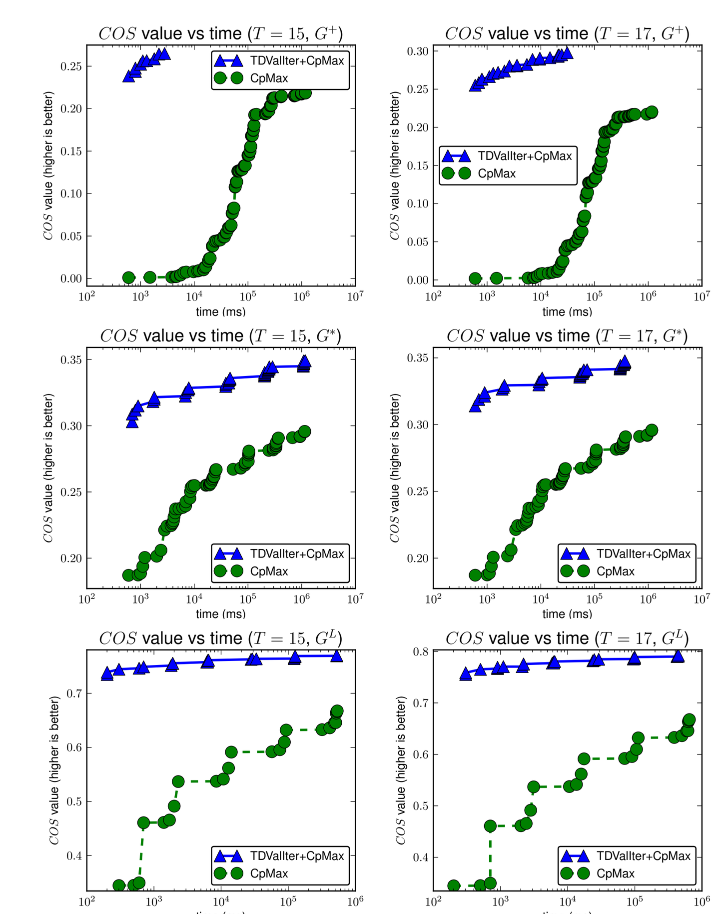


Figure: The COS value as a function of time (ms) (log scale) on G^+, G^*, and G^L with $T = 15$ (left) and $T = 17$ (right). The $pod(y_t) = 0.6$ ($\forall t \in \{1,\dots,T\}$), and the $\rho = 0.6$.

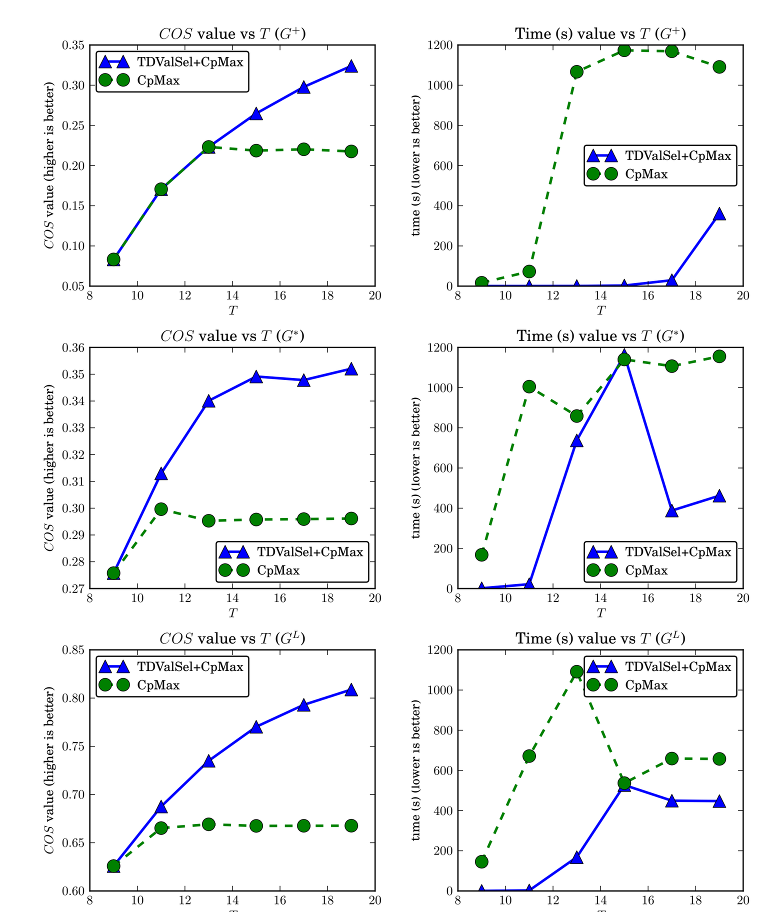


Figure: The COS value (left) and the time to last incumbent (ms) (right) as a function of T on G^+, G^*, and G^L with $pod(y_t) = 0.6$ ($\forall t \in \{1,\dots,T\}$), and $\rho = 0.6$.

Conclusion

- Contributions and novelties:
  - A new CP model to solve the OSP problem
  - A tighter bound using the max objective function encoding
  - The Total Detection value selection heuristic
- Future work:
  - Use the concept of the Total Detection heuristic to develop a bounding technique for the objective function.

References

[1] L. Stone, *Theory of Optimal Search*. New York: Academic Press, 2004.

[2] K. Trummel and J. Weisinger, "The complexity of the optimal searcher path problem," *Operations Research*, vol. 34, no. 2, pp. 324–327, 1986.

[3] F. Laburthe and N. Jussien, *Choco Solver Documentation*, 2012. http://www.emn.fr/z-info/choco-solver/.

[4] J. O'Madadhain, D. Fisher, T. Nelson, S. White, and Y. Boey, "Jung: Java universal network/graph framework." http://jung.sourceforge.net, 2010.