Multi-Agent Low Range Sensing and the Among Constraint

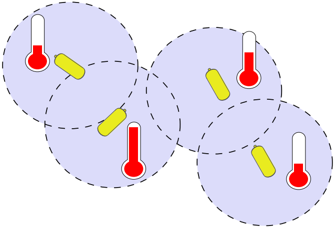
SWIM 2011

Gilles Chabert, Frédéric Boyer, Sophie Demassey

15 June 2011

Outline

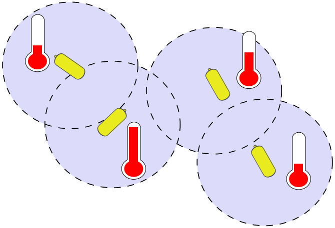
Consider N agents using emitters/receivers



Signal :

- with low range
- omnidirectional
- satisfying a superimposition principle

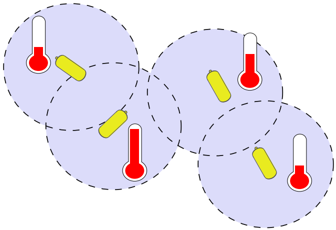
Consider N agents using emitters/receivers



Signal :

- with low range
- omnidirectional
- satisfying a superimposition principle

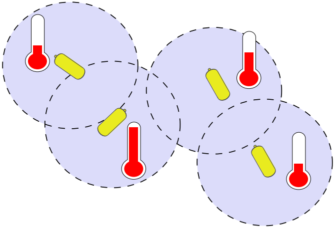
Consider N agents using emitters/receivers



Signal :

- with low range
- omnidirectional
- satisfying a superimposition principle

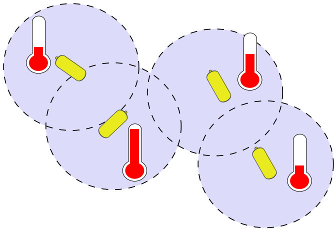
Consider N agents using emitters/receivers



Signal :

- with low range
- omnidirectional
- satisfying a superimposition principle

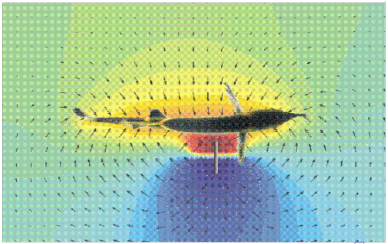
Consider N agents using emitters/receivers



Signal :

- with low range
- omnidirectional
- satisfying a superimposition principle

The example of the electric sense (*Angels project*)



Given the state (position) of n agents $x_1, \ldots, x_N$.
Evolution equation :

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = f(x, u) = \begin{pmatrix} f_1(x_1, u) \\ \vdots \\ f_n(x_n, u) \end{pmatrix}$$

Observation equation (example) :

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = g(x, u) = \begin{pmatrix} \left(\sum_{i \neq 1} \frac{1}{\|x_i - x_1\|^k}\right) \\ \vdots \\ \left(\sum_{i \neq N} \frac{1}{\|x_i - x_n\|^k}\right) \end{pmatrix}$$

SLAM is difficult since g is hard to inverse

Given the state (position) of n agents $x_1, \ldots, x_N$.
Evolution equation :

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = f(x,u) = \begin{pmatrix} f_1(x_1,u) \\ \vdots \\ f_n(x_n,u) \end{pmatrix}$$

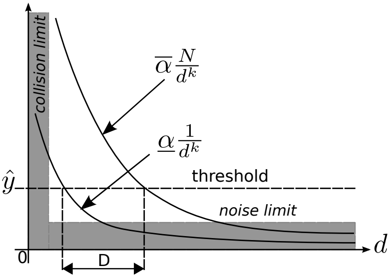
Observation equation (example) :

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = g(x,u) = \begin{pmatrix} \left(\sum_{i \neq 1} \frac{1}{\|x_i - x_1\|^k} \right) \\ \vdots \\ \left(\sum_{i \neq N} \frac{1}{\|x_i - x_n\|^k} \right) \end{pmatrix}$$

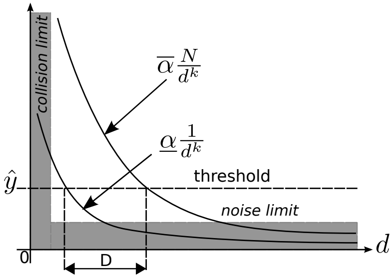
SLAM is difficult since g is hard to inverse

Outline

Obtention of a *detection* distance D for robot n˚i :



Obtention of a *detection* distance D for robot n˚i :

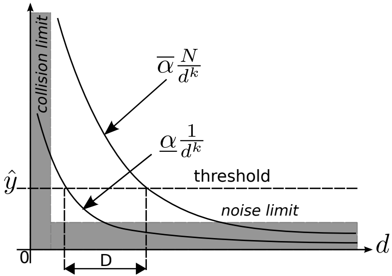


If $y_i > \hat{y}$ at least one agent is at distance $< \overline{D}$

If $y_i < \hat{y}$ no agent is at distance $< \underline{D}$

The Among constraint

Obtention of a *detection* distance D for robot n˚i :



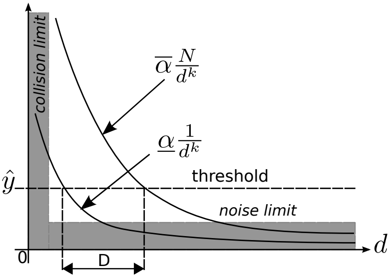
If $y_i > \hat{y}$ at least one agent is at distance $< \overline{D}$

If $y_i < \hat{y}$ no agent is at distance $< \underline{D}$

$\Rightarrow$ This is an example of a new (here : boolean-valued) **"function g"**

The Among constraint

Obtention of a *detection* distance D for robot n˚i :



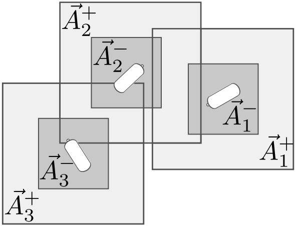
If $y_i > \hat{y}$ at least one agent is at distance $< \overline{D}$
If $y_i < \hat{y}$ no agent is at distance $< \underline{D}$

$\Rightarrow$ This is an example of a new (here : boolean-valued) **"function g"**

$\Rightarrow$ Very similar to Jaulin's approach (range only SLAM)

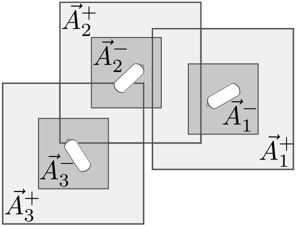
Inversion by hand (example)

Remark : for simplicity, assume 3 agents do not move. They are considered as **beacons**. Two others are "explorers".



Inversion by hand (example)

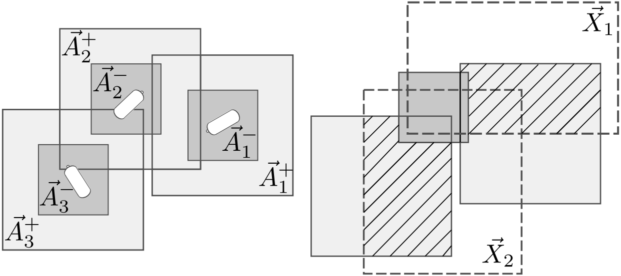
Remark : for simplicity, assume 3 agents do not move. They are considered as **beacons**. Two others are "explorers".



$y_1 > \hat{y}$ $y_2 < \hat{y}$ $y_3 > \hat{y}$

Inversion by hand (example)

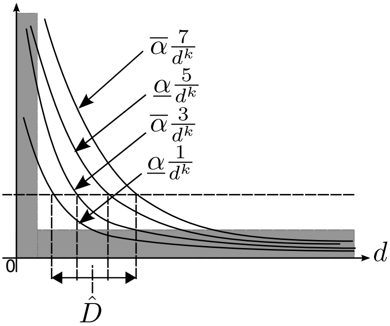
Remark : for simplicity, assume 3 agents do not move. They are considered as **beacons**. Two others are "explorers".



$y_1 > \hat{y}$ $\quad y_2 < \hat{y}$ $\quad y_3 > \hat{y}$

The Among constraint

Now, the *y* function can be further discretized.



If no agent is less distant than $\hat{D}$ from the beacon then :
$y_i > \hat{y} \Longrightarrow$ there is between 3 and 7 agents at distance $< \overline{D}$

The Among constraint

This leads to the following generalization :

Generalization

Let :

$(\vec{x}_j)_{j \in J}$	be variables	(*agents positions*)
$\vec{V}$	be value domains	(*detection areas*)
K	a discrete inteval	(*bounds on the number of detected agents*)

The Among constraint

This leads to the following generalization :

Generalization

Let :

$(\vec{x}_j)_{j \in J}$	be variables	(*agents positions*)
$\vec{V}$	be value domains	(*detection areas*)
K	a discrete inteval	(*bounds on the number of detected agents*)

among

$$\texttt{among}(\ (\vec{x}_j)_{j \in J}, \vec{V}, K\ ) \quad \text{iff}$$

$$\underline{k} \leq \left|\{j \in J, \vec{x}_j \in \vec{V}\}\right| \leq \overline{k}.$$

The Among constraint

This leads to the following generalization :

Generalization

Let :

$(\vec{x}_j)_{j \in J}$	be variables	(*agents positions*)
$\vec{V}$	be value domains	(*detection areas*)
K	a discrete inteval	(*bounds on the number of detected agents*)

among

$$\texttt{among}(\ (\vec{x}_j)_{j \in J}, \vec{V}, K\ ) \quad \text{iff}$$

$$\underline{k} \leq \left| \{ j \in J, \vec{x}_j \in \vec{V} \} \right| \leq \overline{k}.$$

In fact, we have here a conjunction of among (*as many as agents playing the role of beacons*).

Outline

Complexity

State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?

State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?

Complexity

State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?

Complexity

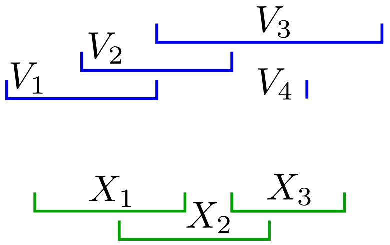
State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?

Complexity

State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?

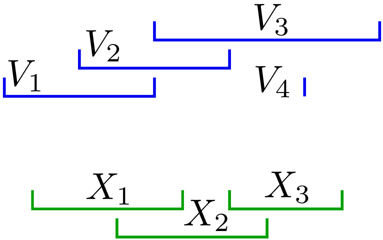


The SACAMI problem

Complexity

State of the art on the satisfiability of conjonctions of `among` :

- NP-Complete when V_i are not necessarily intervals
- But polynomial if they are all disjoints
- ... and when V_i are non-disjoints intervals ?
- ... and in dimension > 1 ?



The SACAMI problem

SACAMI is NP-complete in dimension > 1

Transformation from NVECTOR.

SACAMI is NP-complete in dimension > 1

Transformation from NVECTOR.

Input : $([x]_1, \ldots, [x]_n, k)$.

SACAMI is NP-complete in dimension > 1

Transformation from NVECTOR.

Input : $([x]_1, \ldots, [x]_n, k)$.
Question : does exist $x_i \in [x]_i$ such that

$$\texttt{nvector}((x_1, \ldots, x_n), k)?$$

SACAMI is NP-complete in dimension > 1

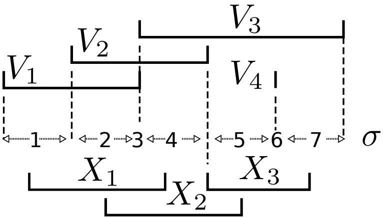
Transformation from NVECTOR.

Input : $([x]_1, \ldots, [x]_n, k)$.
Question : does exist $x_i \in [x]_i$ such that

$$\texttt{nvector}((x_1, \ldots, x_n), k)?$$

New Question : does exist $y_i \in \bigcup_{j=1}^{n} [x]_j$? such that

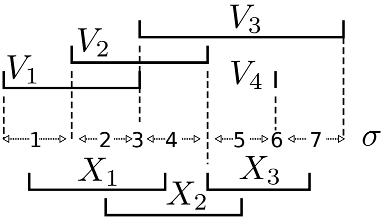
$$\begin{array}{c} \texttt{among}((y_1, \ldots, y_k), [x]_1, [1, k]) \\ \wedge \\ \texttt{among}((y_1, \ldots, y_k), [x]_n, [1, k]) \end{array} ?$$



We will count the number of variables that take their values inside the slice "s". Let $y_s = |j \in J, x_j = s|$.

- (1) is a linear program solvable in polynomial time
- (2) is polynomial (Regin's theorem)
- What about the conjunction of both ?

Complexity



We will count the number of variables that take their values inside the slice "s". Let $y_s = |j \in J, x_j = s|$.

A decomposition

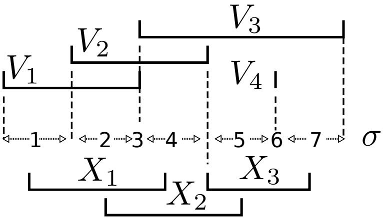
$\texttt{among}(\ (\vec{x}_j)_{j \in J}, \vec{V}, K\ ) \quad \Longleftrightarrow$

$$\underline{k}_i \leq \sum_{s \in V_i} y_s \leq \overline{k}_i, \qquad \forall i \in I \qquad (1)$$

$$\wedge \quad \texttt{among}(\ (x_j)_{j \in J}, \{s\}, y_s\ ), \qquad \forall s \in \sigma. \qquad (2)$$

- (1) is a linear program solvable in polynomial time
- (2) is polynomial (Regin's theorem)
- What about the conjunction of both ?

Complexity



We will count the number of variables that take their values inside the slice "s". Let $y_s = |j \in J, x_j = s|$.

A decomposition

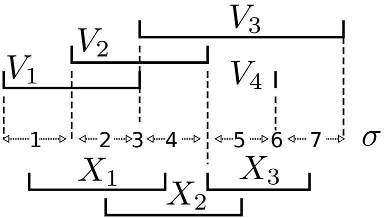
$\texttt{among}(\ (\vec{x}_j)_{j\in J}, \vec{V}, K\ ) \iff$

$$\underline{k}_i \leq \sum_{s\in V_i} y_s \leq \overline{k}_i, \qquad \forall i \in I \qquad (1)$$

$$\wedge \quad \texttt{among}(\ (x_j)_{j\in J}, \{s\}, y_s\ ), \qquad \forall s \in \sigma. \qquad (2)$$

- (1) is a linear program solvable in polynomial time
- (2) is polynomial (Regin's theorem)
- What about the conjunction of both ?

Complexity



We will count the number of variables that take their values inside the slice "s". Let $y_s = |j \in J, x_j = s|$.

A decomposition

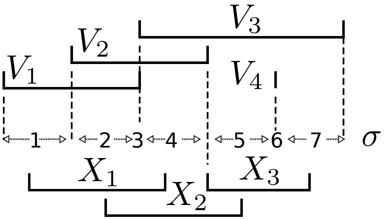
$$\texttt{among}(\ (\vec{x}_j)_{j \in J}, \vec{V}, K\ ) \iff$$

$$\underline{k}_i \leq \sum_{s \in V_i} y_s \leq \overline{k}_i, \qquad \forall i \in I \qquad (1)$$

$$\wedge \quad \texttt{among}(\ (x_j)_{j \in J}, \{s\}, y_s\ ), \qquad \forall s \in \sigma. \qquad (2)$$

- (1) is a linear program solvable in polynomial time
- (2) is polynomial (Regin's theorem)
- What about the conjunction of both ?

Complexity



We will count the number of variables that take their values inside the slice "s". Let $y_s = |j \in J, x_j = s|$.

A decomposition

$$\mathtt{among}(\ (\vec{x}_j)_{j \in J}, \vec{V}, K\ ) \iff$$

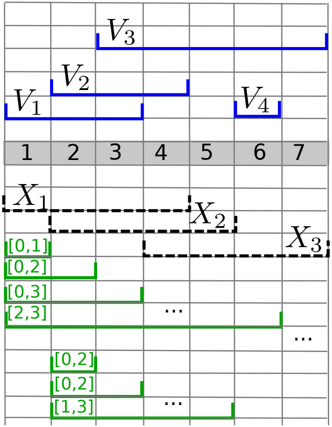
$$\underline{k}_i \leq \sum_{s \in V_i} y_s \leq \overline{k}_i, \qquad \forall i \in I \qquad (1)$$

$$\wedge \quad \mathtt{among}(\ (x_j)_{j \in J}, \{s\}, y_s\ ), \qquad \forall s \in \sigma. \qquad (2)$$

- (1) is a linear program solvable in polynomial time
- (2) is polynomial (Regin's theorem)
- What about the conjunction of both ?

Complexity

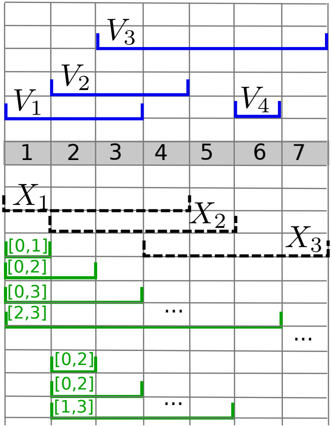
Key idea : "drop" (2) and "inflate" (1) :



- We build a flow problem from the y_s
- We show that the maximal flow is n sing Hoffman's theorem

Complexity

Key idea : "drop" (2) and "inflate" (1) :

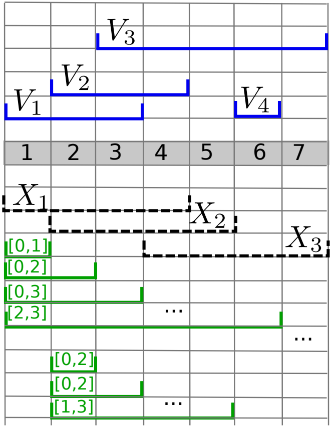


Difficulty : prove that the new problem is satifisfiable => SACAMI is satisfiable

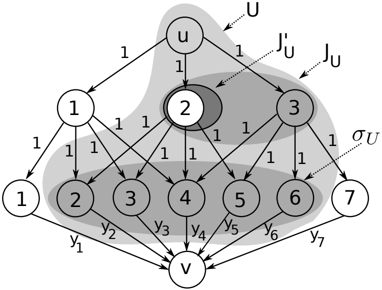
- We build a flow problem from the y_s
- We show that the maximal flow is n sing Hoffman's theorem

Complexity

Key idea : "drop" (2) and "inflate" (1) :



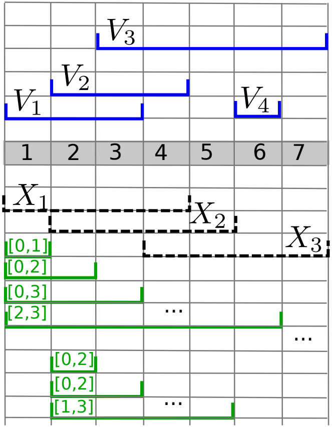
Difficulty : prove that the new problem is satififsfiable => SACAMI is satisfiable



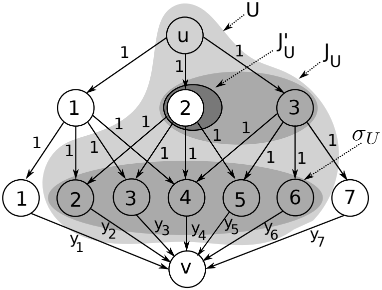
- We build a flow problem from the y_s
- We show that the maximal flow is n sing Hoffman's theorem

Complexity

Key idea : "drop" (2) and "inflate" (1) :



Difficulty : prove that the new problem is satifisfiable => SACAMI is satisfiable



- We build a flow problem from the y_s
- We show that the maximal flow is n sing Hoffman's theorem

Outline

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project

Conclusion

- We have generalized Jaulin's idea of detection/no-detection constraint introducing the `among` constraint
- This makes unexpected connections with other fields of research (logistics)
- We have studied the complexity of a conjunction of `among`
- It is NP-hard with vector domains
- And polynomial in the one-dimensional case
- A filtering algorithm (not using Simplexe !) has been set up
- This technique will be implemented with the electric sense in the Angels project