Ghrist-Peterson configuration spaces, cubings, and Boolean queries

Dan P. Guralnik , joint with Robert Ghrist

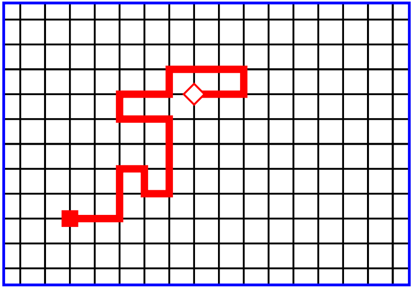
University of Pennsylvania
ESE and Mathematics

Funded by AFRL-DARPA contract No. FA8650-18-2-7840

November 3, 2019

Ghrist–Peterson Reconfigurable Systems [4]

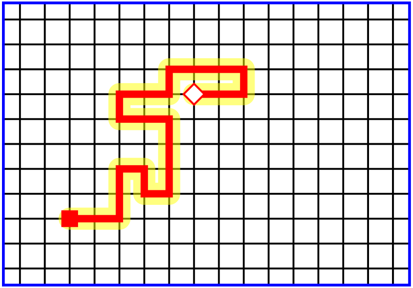
Example. Planar "snake robots":



A *snake* in the integer grid = a vertex in configuration space **X**

Ghrist–Peterson Reconfigurable Systems [4]

Example. Planar "snake robots":

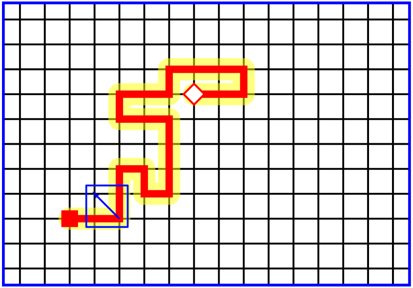


A physical snake robot has thickness, motivating the constraints:

no backtracking, no self-intersection.

Ghrist–Peterson Reconfigurable Systems [4]

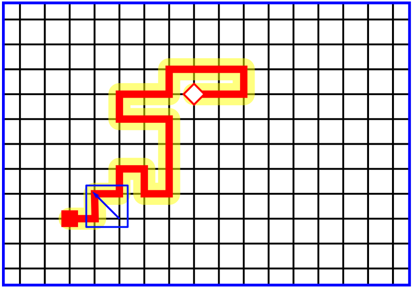
Example. Planar "snake robots":



Movement is modelled by local reconfigurations, inducing edges in **X**: a 'diagonal wiggle'...

Ghrist–Peterson Reconfigurable Systems [4]

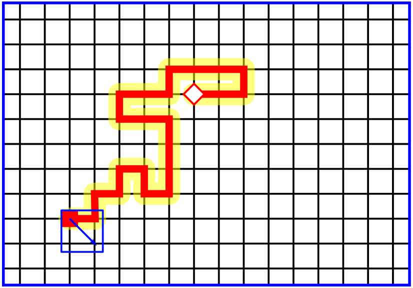
Example. Planar "snake robots":



Movement is modelled by local reconfigurations, inducing edges in **X**: a 'diagonal wiggle'…

Ghrist–Peterson Reconfigurable Systems [4]

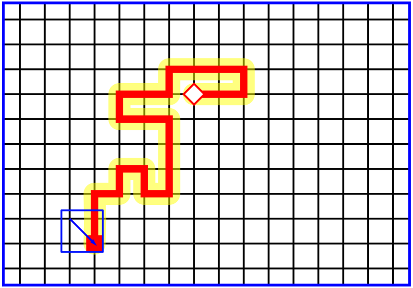
Example. Planar “snake robots”:



. . . a turn of the tail (or head).

Ghrist–Peterson Reconfigurable Systems [4]

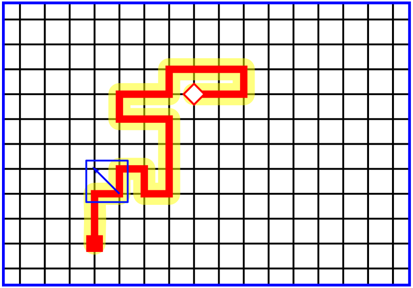
Example. Planar "snake robots":



. . . a turn of the tail (or head).

Ghrist–Peterson Reconfigurable Systems [4]

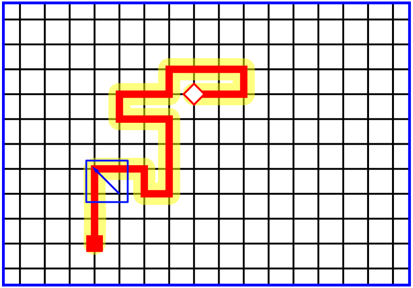
Example. Planar “snake robots”:



Here is another wiggle.

Ghrist–Peterson Reconfigurable Systems [4]

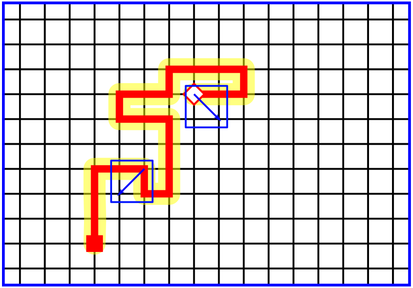
Example. Planar "snake robots":



Here is another wiggle.

Ghrist–Peterson Reconfigurable Systems [4]

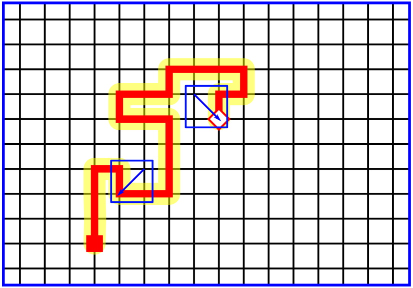
Example. Planar "snake robots":



Transitions with disjoint supports may be effected independetly and synchronously…

Ghrist–Peterson Reconfigurable Systems [4]

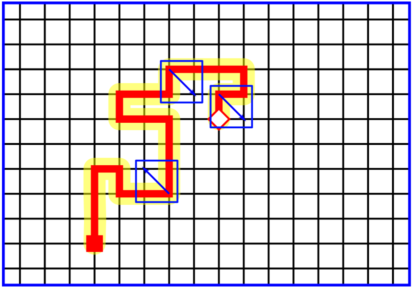
Example. Planar “snake robots”:



Transitions with disjoint supports may be effected independetly and synchronously. . .

Ghrist–Peterson Reconfigurable Systems [4]

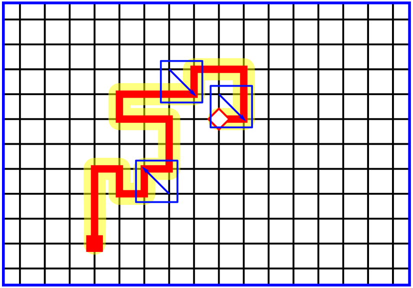
Example. Planar "snake robots":



. . . inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

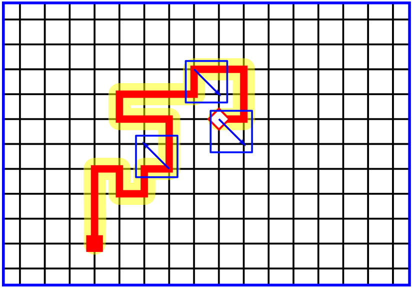
Example. Planar "snake robots":



. . . inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

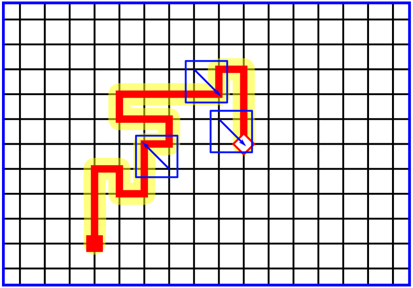
Example. Planar "snake robots":



…inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

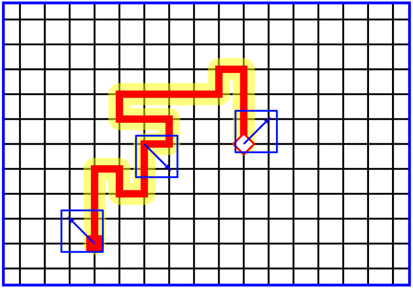
Example. Planar "snake robots":



... inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

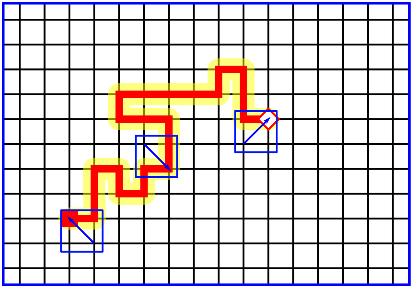
Example. Planar "snake robots":



...inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

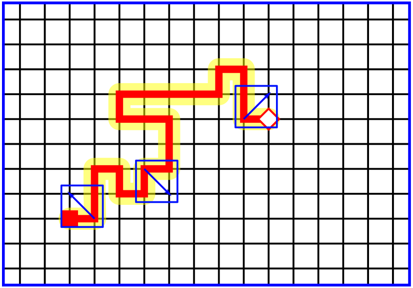
Example. Planar "snake robots":



. . . inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

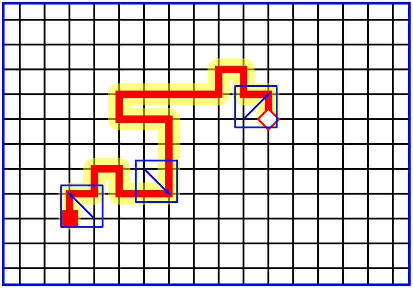
Example. Planar “snake robots”:



. . . inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

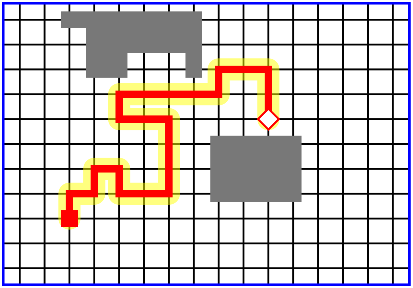
Example. Planar "snake robots":



... inducing [combinatorially] embedded cubes in **X**.

Ghrist–Peterson Reconfigurable Systems [4]

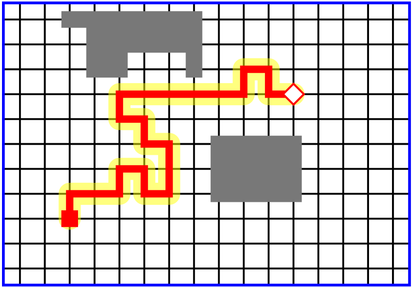
Example. Planar "snake robots":



Additional constraints may derive from obstacles…

Ghrist–Peterson Reconfigurable Systems [4]

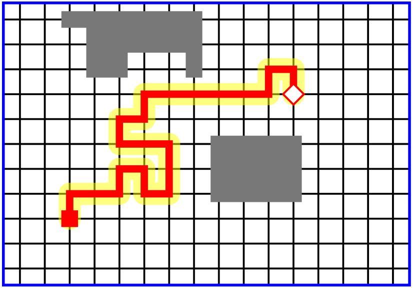
Example. Planar "snake robots":



. . . not to mention optimality considerations. . .

Ghrist–Peterson Reconfigurable Systems [4]

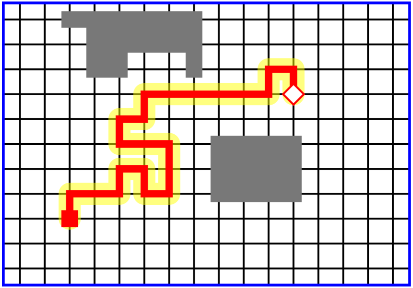
Example. Planar "snake robots":



…e.g., wanting to avoid useless alcoves (see top obstacle).

LEARNING Ghrist–Peterson Reconfigurable Systems [4]

Example. Planar "snake robots":

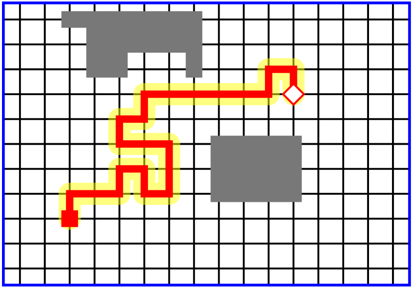


…e.g., wanting to avoid useless alcoves (see top obstacle).

Problem. Given such a robot, how might it LEARN to navigate in **X**?

LEARNING Ghrist–Peterson Reconfigurable Systems [4]

Example. Planar "snake robots":



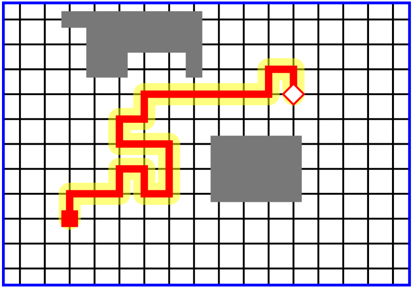
. . . e.g., wanting to avoid useless alcoves (see top obstacle).

Problem. Given such a robot, how might it LEARN to navigate in **X**?

- without prior knowledge of situational constraints;

LEARNING Ghrist–Peterson Reconfigurable Systems [4]

Example. Planar "snake robots":



. . . e.g., wanting to avoid useless alcoves (see top obstacle).

Problem. Given such a robot, how might it LEARN to navigate in $\mathbf{X}$?

- without prior knowledge of situational constraints;
- only using [sufficient] sensory information?

What does "learning to navigate in $\mathbf{X}$" mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot's sensory system*

What does "learning to navigate in $\mathbf{X}$" mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot's sensory system*

Loose Statement of the Overall Problem. Given a reconfigurable system with configuration space $\mathbf{X}$ and a split system S on $\mathbf{X}^{(0)}$, find an algorithm that generates:

What does “learning to navigate in $\mathbf{X}$” mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot's sensory system*

Loose Statement of the Overall Problem. Given a reconfigurable system with configuration space $\mathbf{X}$ and a split system S on $\mathbf{X}^{(0)}$, find an algorithm that generates:

1. a walk $w = (x_0, \ldots, x_t, \ldots)$ in $\mathbf{X}$;
2. representations R_t of $\mathbf{X}$ from the observation vectors $(\sigma(x_t))_{\sigma \in S}$;

What does “learning to navigate in $\mathbf{X}$” mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot’s sensory system*

Loose Statement of the Overall Problem. Given a reconfigurable system with configuration space $\mathbf{X}$ and a split system S on $\mathbf{X}^{(0)}$, find an algorithm that generates:

1. a walk $w = (x_0, \ldots, x_t, \ldots)$ in $\mathbf{X}$;
2. representations R_t of $\mathbf{X}$ from the observation vectors $(\sigma(x_t))_{\sigma \in S}$;

and gurantees a $T \in \mathbb{N}$ such that:

- R_t enables efficient navigation in $\mathbf{X}$ for $t \geq T$;
- R_t does not change for $t \geq T$;
- T is as small as possible (as a function of $|S|$).

What does “learning to navigate in $\mathbf{X}$” mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot’s sensory system*

A More Basic Problem. Given a contender algorithm,

- HOW DO WE QUANTIFY PROGRESS?
- HOW DO WE CERTIFY SUCCESS?

What does "learning to navigate in $\mathbf{X}$" mean?

Definition. A SPLIT on a set V is a Boolean function $\sigma\colon V \to \{0,1\}$. Its COMPLEMENT is the function $\sigma^* := 1 - \sigma$. A SPLIT SYSTEM on a set V is a point-separating multi-set of splits on V that is symmetric under complementation.

$\rightsquigarrow$ *This will be our model of the robot's sensory system*

A More Basic Problem. Given a contender algorithm,

- HOW DO WE QUANTIFY PROGRESS?
- HOW DO WE CERTIFY SUCCESS?

Today we explore one idea:

LEVERAGE THE GEOMETRY & TOPOLOGY OF $\mathbf{X}$
TO ENCODE THE PROBLEM AS AN OPTIMIZATION PROBLEM
OVER THE SPACE OF SPLIT-SYSTEMS ON $\mathbf{X}^{(0)}$.

Review and Preliminaries

Reconfigurable Systems and Non-Positive Curvature

Theorem (Ghrist–Peterson [4]). *The configuration space* **X** *of any reconfigurable system is a* NON-POSITIVELY CURVED *cubical complex (*NPC^3*).*

Examples to keep in mind:

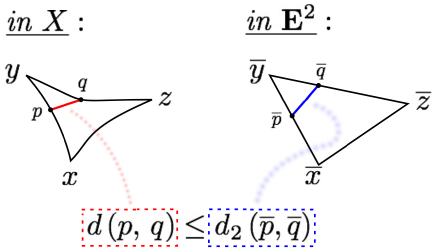
- For collision-free navigation of $N \geq 2$ particles on a graph (Abrams–Ghrist [7]), $X \in \mathrm{NPC}^3$ may be obtained.
- **X** is a cubing for a restricted class of snakes (Ardila *et. al.* [2]).

Remark. In fact, Ghrist-Peterson's result is stronger: **X** is a *special* NPC cubical complex, excluding a range of pathologies.

This is important for the overall project, but not for this talk, and we omit this discussion in the interest of time.

Non-Positive Curvature (NPC), see [3]

Definition. A complete geodesic metric space (X, d) is CAT(0), if all geodesic triangles $\triangle xyz$ are thinner than their Euclidean comparison triangles:



Theorem. *The metric d of a CAT(0) space (X, d) is convex. In particular, geodesics are unique, and X is contractible.*

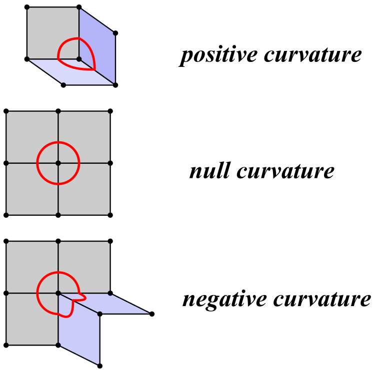
Definition. A complete geodesic metric space (X, d) is NPC, if every $x \in X$ has $r_x > 0$ such that the ball $\bar{B}_d(x, r_x)$ is CAT(0).

Alexandrov's patchwork (AP): *The universal cover $\tilde{X}$ of a NPC space (X, d) is CAT(0). In particular, it is contractible.*

NPC Cubical Complexes (NPC^3)

Theorem (Gromov [3]). *A cubical complex* **X** *is NPC if and only if the link of every cube is a flag simplicial complex.*

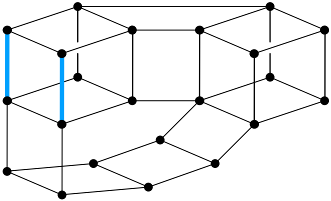
Definition (Sageev [9]). A CUBING is a simply connected NPC cubical complex.



- Cubings are precisely the CAT(0) cubical complexes, by AP.
- In particular, every cubing is contractible.
- The Cartesian product of two cubings is a cubing.

Representations for Cubings

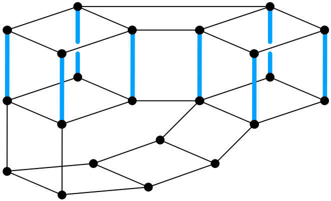
Hyperplanes in Cubical Complexes (CCs):



In a CC, edges on opposite sides of a square are said to be parallel.

Representations for Cubings

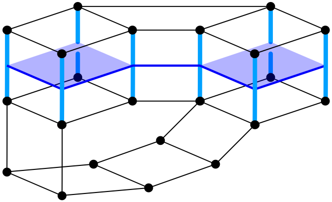
Hyperplanes in Cubical Complexes (CCs):



Each parallelism class P yields, via transitive closure…

Representations for Cubings

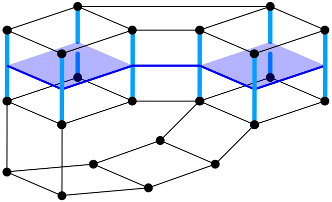
Hyperplanes in Cubical Complexes (CCs):



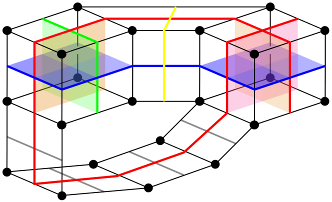
Its DUAL HYPERPLANE: the union of perpendicular bisectors of every $e \in P$.

Representations for Cubings

Hyperplanes in Cubical Complexes (CCs):



Its DUAL HYPERPLANE: the union of perpendicular bisectors of every $e \in P$.



Hyperplanes may separate (blue) or not (e.g. yellow,green). They may self-cross (red).

Representations for Cubings

Suppose $\mathbf{X}$ is a cubing. Sageev-Roller duality [9, 8] provides:

- Every hyperplane in $\mathbf{X}$ is a convexly embedded cubing.
- Every hyperplane in $\mathbf{X}$ separates $\mathbf{X}$ into two convex components—the HALFSPACES of $\mathbf{X}$ corresponding to that hyperplane.
- The graph $\mathbf{X}^{(1)}$ is simple (no multiple edges) and bipartite.
- The containment order on $\mathfrak{h}(\mathbf{X})$, the halfspace system of $\mathbf{X}$, completely encodes $\mathbf{X}$.

  $\rightsquigarrow$ *we may regard* $\mathfrak{h}(\mathbf{X})$ *as a split system on* $\mathbf{X}^{(0)}$

Representations for Cubings

Suppose X is a cubing. Sageev-Roller duality [9, 8] provides:

- Every hyperplane in $\mathbf{X}$ is a convexly embedded cubing.
- Every hyperplane in $\mathbf{X}$ separates $\mathbf{X}$ into two convex components—the HALFSPACES of $\mathbf{X}$ corresponding to that hyperplane.
- The graph $\mathbf{X}^{(1)}$ is simple (no multiple edges) and bipartite.
- The containment order on $\mathfrak{h}(\mathbf{X})$, the halfspace system of $\mathbf{X}$, completely encodes $\mathbf{X}$.

  $\rightsquigarrow$ *we may regard $\mathfrak{h}(\mathbf{X})$ as a split system on $\mathbf{X}^{(0)}$*

Our earlier work [5, 6] shows:

- Given $S = \mathfrak{h}(\mathbf{X})$, the inclusion order can be learned (e.g. from a random walk), and leveraged for efficient navigation.

An alternative representation (PIPs [1, 2]) exists, but has some computational disadvantages:

1. the PIP representation encodes a pointed cubing $(\mathbf{X}, x)$, $x \in \mathbf{X}^{(0)}$;
2. two relations (separation, incompatibility) on the collection of hyperplanes are used;
3. both relations change significantly as the observer's state (the base vertex x) varies.

Problem Statement and Results

Today's Problem Statement

Final Problem Statement when X is a Cubing. *Assume:*

- $\mathbf{X}$ *is the configuration space of an RS;*
- S *is a vertex-separating collection of Boolean queries on* $\mathbf{X}^{(0)}$.

Find an algorithm which constructs a walk in $\mathbf{X}$ *while using the observations made along this walk to deform* S *into* $\mathfrak{h}(\mathbf{X})$.

What is required to approach the problem? First and foremost, a certificate informing us when the job is done, such as:

- A functional $\Psi_{\mathbf{X}}$ on the space of split systems, minimized by $\mathfrak{h}(\mathbf{X})$.
- We want $\mathfrak{h}(\mathbf{X})$ to be the only minimum of $\Psi_{\mathbf{X}}$;
- We need an efficient test determining whether $\Psi_{\mathbf{X}}(S)$ is minimal.

Hopes for the general case?

- Learn a cover $\mathbf{X} = \bigcup_i \mathbf{X}_i$ by cubings, with convex $\mathbf{X}_i \cap \mathbf{X}_j$;
- Cellular sheaf co/homology: global certificate from local ones.

Current result: Recongnizing a Cubing

Definition. Let Γ be a simple graph and S be a split system on Γ. For a split $\sigma : V\Gamma \to \mathbf{2}$, the COBOUNDARY of σ is:

$$\delta\sigma := \{e \in E\Gamma\,,\ \sigma \text{ separates the endpoints of } e\} \tag{1}$$

We set:

$$\begin{aligned}
\Phi_\Gamma(S) &:= \sum_{x\in V\Gamma}\sum_{\sigma\in S}\left|\delta\{x\}\cap\delta\sigma\right| && (2)\\
&= \sum_{e\in E\Gamma}\sum_{\sigma\in S}(\delta\sigma)(e) && (3)\\
&= \sum_{\sigma\in S}\left|\delta\sigma\right| && (4)\\
&= \sum_{\sigma\in S}\langle\sigma,\Delta\sigma\rangle\ , && (5)
\end{aligned}$$

where Δ denotes the graph Laplacian. □

WE WILL HAVE $\Gamma = \mathbf{X}^{(1)}$, THE 1-SKELETON OF OUR CUBING $\mathbf{X}$.

Current result: Recongnizing a Cubing

Why $\Phi_{\mathbf{X}}$ may seem like a good candidate? Focus on (3):

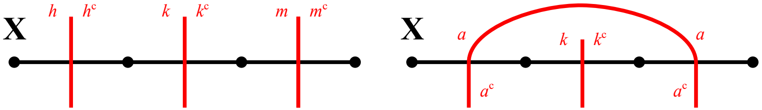
- S is vertex-separating, so every edge contributes at least 2 to $\Phi_{\mathbf{X}}(S)$.
- For $S = \mathfrak{h}(X)$, every edge contributes exactly 2 to $\Phi_{\mathbf{X}}(S)$.
- Hence $\mathfrak{h}(X)$ minimizes $\Phi_{\mathbf{X}}$.

Current result: Recongnizing a Cubing

Why $\Phi_{\mathbf{X}}$ may seem like a good candidate? Focus on (3):

- S is vertex-separating, so every edge contributes at least 2 to $\Phi_{\mathbf{X}}(S)$.
- For $S = \mathfrak{h}(X)$, every edge contributes exactly 2 to $\Phi_{\mathbf{X}}(S)$.
- Hence $\mathfrak{h}(X)$ minimizes $\Phi_{\mathbf{X}}$.

But $\Phi_{\mathbf{X}}$ is NOT a good candidate. $\mathfrak{h}(X)$ is NOT the only minimizer of $\Phi_{\mathbf{X}}$:



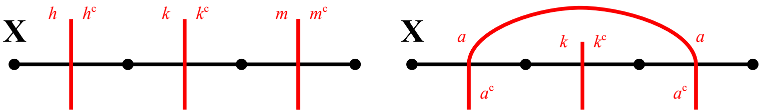
$\mathfrak{h}(\mathbf{X})$ has 3 hyperplanes (left), but there is an S with fewer elements (right), contributing to the same value of $\Phi_{\mathbf{X}}(S)$.

Current result: Recongnizing a Cubing

Why $\Phi_{\mathbf{X}}$ may seem like a good candidate? Focus on (3):

- S is vertex-separating, so every edge contributes at least 2 to $\Phi_{\mathbf{X}}(S)$.
- For $S = \mathfrak{h}(X)$, every edge contributes exactly 2 to $\Phi_{\mathbf{X}}(S)$.
- Hence $\mathfrak{h}(X)$ minimizes $\Phi_{\mathbf{X}}$.

But $\Phi_{\mathbf{X}}$ is NOT a good candidate. $\mathfrak{h}(X)$ is NOT the only minimizer of $\Phi_{\mathbf{X}}$:



$\mathfrak{h}(\mathbf{X})$ has 3 hyperplanes (left), but there is an S with fewer elements (right), contributing to the same value of $\Phi_{\mathbf{X}}(S)$.

Main Theorem. *Suppose $\mathbf{X}$ is a cubing and S is a split system of maximum cardinality among the split systems which minimize $\Phi_{\mathbf{X}}$. Then, S coincides with $\mathfrak{h}(\mathbf{X})$.* □

Current result: Recongnizing a Cubing

Main Theorem. *Suppose* $\mathbf{X}$ *is a cubing and* S *is a split system of maximum cardinality among the split systems which minimize* $\Phi_{\mathbf{X}}$. *Then,* S *coincides with* $\mathfrak{h}(\mathbf{X})$. □

Some remarks:

- For any graph Γ, maximizing $\left|S\right|$ for a given value of $\Phi_{\Gamma}(S)$ requires each $\sigma \in S$ to be connected.
- Is $\Psi_{\mathbf{X}}(S) := \Phi_{\mathbf{X}}(S) - \alpha\left|S\right|$, $\alpha > 0$ the functional we are looking for?—Some evidence:
  - Restricted to trees, it is, for $\alpha \in (0, 1)$.
  - In fact, for $\alpha \in (0, 1)$, a minimizer of Ψ_{Γ} must contain all half-spaces arising from bridges of Γ.
  - If S is a $\Psi_{\mathbf{X}}$ minimizer and $\left|S\right| \leq \left|\mathfrak{h}(\mathbf{X})\right|$ then $S = \mathfrak{h}(\mathbf{X})$.
- By (4), $\Psi_{\Gamma}(S) \geq 0$ for all S, but it is not clear anymore whether $\Psi_{\mathbf{X}}(S) \geq \Psi_{\mathbf{X}}(\mathfrak{h}(\mathbf{X}))$ when $\mathbf{X}$ is a cubing.

Current result: Recongnizing a Cubing

Main Theorem. *Suppose* $\mathbf{X}$ *is a cubing and* S *is a split system of maximum cardinality among the split systems which minimize* $\Phi_{\mathbf{X}}$. *Then,* S *coincides with* $\mathfrak{h}(\mathbf{X})$. □

Corollary: If $\mathbf{X}_1, \mathbf{X}_2$ are cubings, and S is a split system on $\mathbf{X} := \mathbf{X}_1 \times \mathbf{X}_2$ minimizing $\Phi_{\mathbf{X}}$, then $\left|S\right| \leq \left|\mathfrak{h}(\mathbf{X}_1)\right| + \left|\mathfrak{h}(\mathbf{X}_2)\right|$. □

$\rightsquigarrow$ *there is something information-theoretic about this inequality...*

Question: is it true that $\mathfrak{h}(\mathbf{X})$ is the unique minimum of the functional $\Psi_{\mathbf{X}}$?

- What is the right value for α, if at all?
- Perhaps another regularization of $\Phi_{\mathbf{X}}$ could work?
- What is the connection with graph cohomology?

The answer to these questions is vital to understanding whether or not our approach extends to NPC^3s arising as state spaces of reconfigurable systems.

Proof Sketch

The Hamming cube $\mathbb{H}(S)$ over S. Let $n := |S|$

- $(*)$-SELECTIONS are subsets $A \subset S$ with $A^* \cap A = \varnothing$;
- VERTICES OF $\mathbb{H}(S)$ are the $(*)$-selections A with $A^* \cup A = S$;
- d-FACES OF $\mathbb{H}(S)$ correspond to $(*)$-selections B with $|B| = \frac{n-d}{2}$.

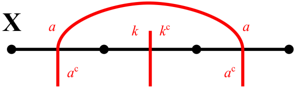
  The vertices of the B-face are: $V_S(B) := \{A \in \mathbb{H}(S)^{(0)} \mid A \supseteq B\}$

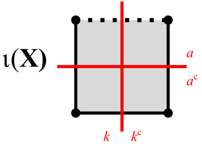
Mapping $\mathbf{X}^{(0)}$ into $\mathbb{H}(S)$.

Consider the function $\iota : \mathbf{X}^{(0)} \to \mathbb{H}(S)$ given by $\iota(x) = \{\sigma \in S \mid \sigma(x) = 1\}$.

$\rightsquigarrow$ *this is the mapping of x to its "vector of Boolean sensations"*

When S is a $\Phi_{\mathbf{X}}$ minimizer, ι extends to a cellular embedding:





Proof Sketch

We study the mapping $\iota : \mathbf{X} \to \mathbb{H}(S)$. The heart of the argument is:

Lemma. The map ι induces a surjective $(*)$-equivariant map $\varphi : \mathfrak{h}(\mathbf{X}) \to S$ such that:

1. $\iota(\delta h) \subseteq \delta\varphi(h)$ for all $h \in \mathfrak{h}(\mathbf{X})$.

   $\rightsquigarrow$ *$\mathbb{H}(S)$-hyperplanes partition the $\mathbf{X}$-hyperplanes*

2. $\iota(\partial h) \subseteq V_S(\varphi(h))$ for all $h \in \mathfrak{h}(\mathbf{X})$.

   $\rightsquigarrow$ *φ is "orientation-preserving"*

3. $\varphi^{-1}(\{\sigma, \sigma^*\}) \neq \varnothing$ and nested.

   $\rightsquigarrow$ *$\iota(\mathbf{X})$ intersects every facet of $\mathbb{H}(S)$*

   $\rightsquigarrow$ *halfspace preimages are unions of nested halfspaces*

4. φ is injective if and only if $S = \mathfrak{h}(\mathbf{X})$.

   $\rightsquigarrow$ *This seals the proof.*

$\square$

THANK YOU!

$\rightsquigarrow$ *I've recently moved to University of Florida, MAE*

$\rightsquigarrow$ *please contact me at* `danguralnik@ufl.edu`

References

[1] Federico Ardila, Tia Baker, and Rika Yatchak. Moving robots efficiently using the combinatorics of CAT (0) cubical complexes. *SIAM Journal on Discrete Mathematics*, 28(2):986–1007, 2014.

[2] Federico Ardila, Hanner Bastidas, Cesar Ceballos, and John Guo. The configuration space of a robotic arm in a tunnel. *SIAM Journal on Discrete Mathematics*, 31(4):2675–2702, 2017.

[3] Martin R. Bridson and André Haefliger. *Metric spaces of non-positive curvature*, volume 319 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1999.

[4] Robert Ghrist and Valerie Peterson. The geometry and topology of reconfiguration. *Advances in applied mathematics*, 38(3):302–323, 2007.

[5] D.P. Guralnik and D.E. Koditschek. Toward a memory model for autonomous topological mapping and navigation: The case of binary sensors and discrete actions. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 936–945, 2012.

[6] D.P. Guralnik and D.E. Koditschek. Iterated Belief Revision Under Resource Constraints: Geometry instead of Logic. *preprint*, 2018.

[7] Abrams, Aaron and Ghrist, Robert. Finding topology in a factory: configuration spaces. *The American mathematical monthly*, 109(2):140–150, 2002.

[8] M.A. Roller. Poc Sets, Median Algebras and Group Actions. *University of Southampton, Faculty of Math. stud., preprint series*, 1998.

[9] Michah Sageev. Ends of groups pairs and non-positively curved cube complexes. *Proc. London Math. Soc.*, 3(71):586–617, 1995.