
Crystalline Robots: Self-reconfiguration with Compressible Unit Modules

Daniela Rus, Marsette Vona
Department of Computer Science
Dartmouth College
Hanover, NH

Presented by Matthew Aldridge

Introduction

- Goal: To design “versatile and extensible massively-parallel distributed robot systems”
 - Crystalline Modules
 - Algorithm for general self-reconfiguration
-

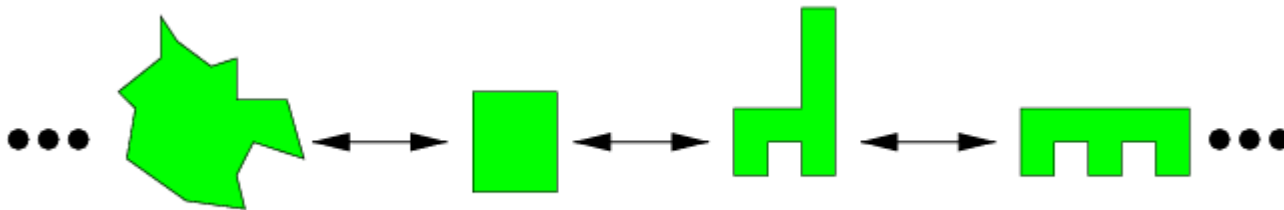
Versatile and Extensible

(One approach, taken here)

- Achieved with self-reconfigurable system
 - Minimalist approach
 - Multiple modalities of locomotion, manipulation, and sensing
 - May be either homogeneous or heterogeneous
 - Well-suited for adaptability to environment with incomplete a priori task knowledge
 - Fault tolerance
-

Requirements for Self-reconfiguration

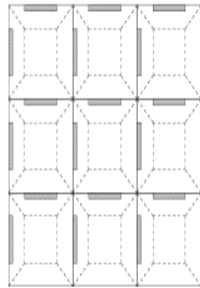
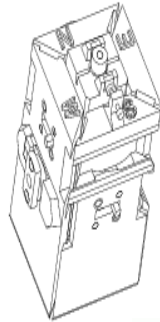
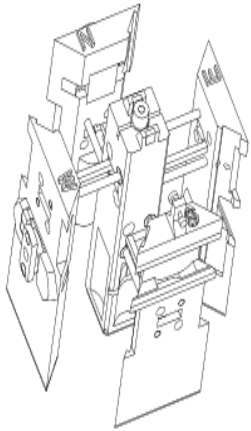
- A unit modular robot system is self-reconfigurable if:
 - 1) Its unit module can be used to assemble arbitrary shapes
 - 2) In any structure composed of unit modules, some module can be relocated arbitrarily



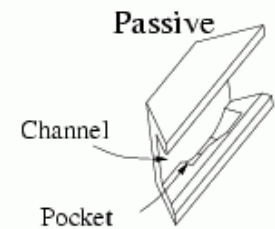
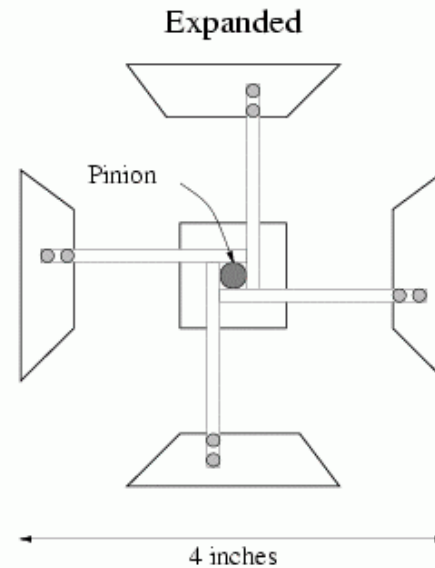
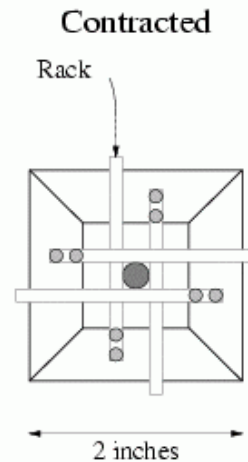
Crystalline Atoms

- Identical robot modules (homogeneous model)
 - Actuated by expansion and contraction
 - Design based on cubes (or squares in 2D)
 - Each module is an *atom*, each connector a *bond*, and groups of atoms are *crystals*
 - Physical prototype is 2D, but approach can be generalized to 3D
-

The 2D Physical Prototype



- Each module has two active and two passive connectors
- Expansion and contraction controlled by rack and pinion mechanism



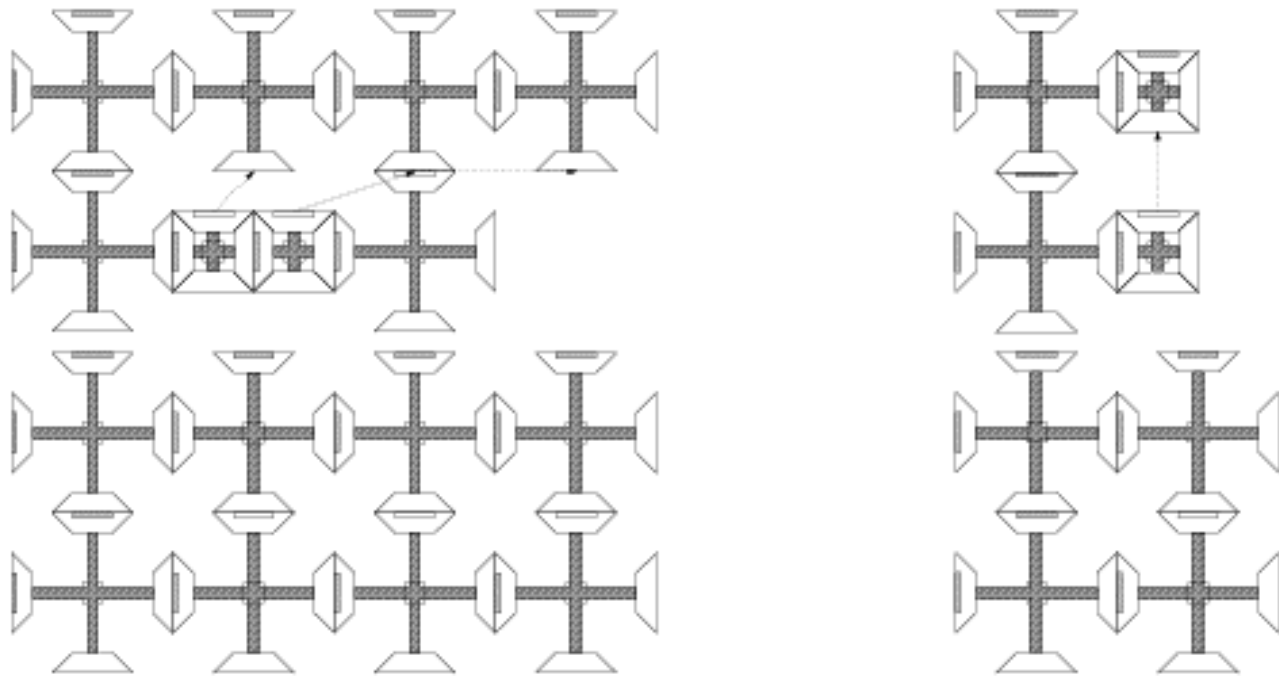
Primitive Operations for Crystal Modules

- Expand `<atom> <dimension>`
 - Expand a compressed Atom
- Contract `<atom> <dimension>`
 - Compress an expanded Atom
- Bond `<atom> <dimension>`
 - Activate one of the Atom's connectors to bond with a neighboring Atom in the structure
- Free `<atom> <dimension>`
 - Deactivate one of the Atom's connectors to break a bond with a neighboring Atom in the structure

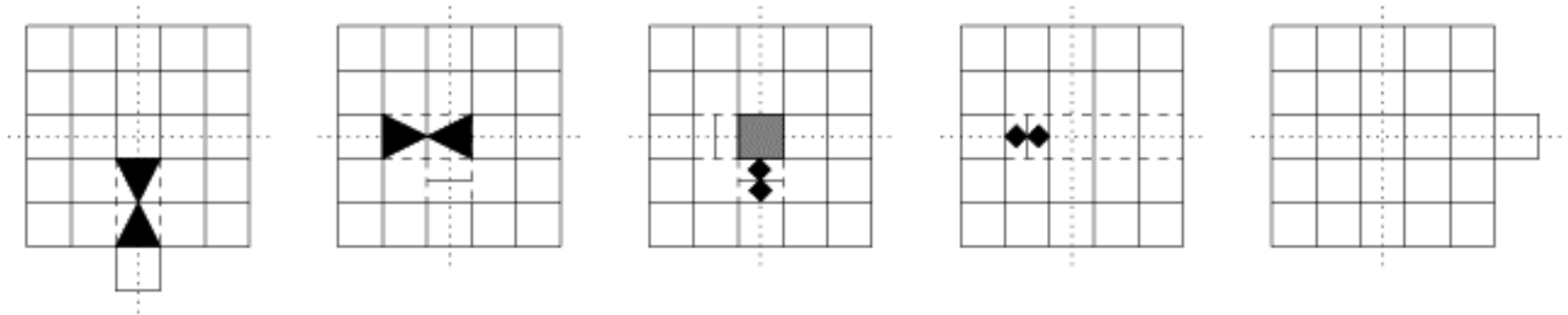
Relocating a Module on a Crystal

- Achieved by attaching one Atom to a neighboring Atom and actuating the compression or expansion mechanism
 - Unlike some other systems, Crystalline Atoms can move *through* the volume of structure, instead of on surface
 - Allows for relocation algorithm in constant time, independent of structure size
-

Using Expansion and Contraction for Locomotion



Moving a Module through the Crystal



Through a series of contractions and expansions, a module is moved from the bottom of the structure to the right side.

Note: Since each module is identical, the newly placed module on the right does not need to be the same module as on the bottom.

A Planner for Shape Metamorphosis

- Given a pair of Crystals - or structures - (S , G), each composed of n Atoms
 - A planner finds a feasible plan P that transforms S into G
 - Sequence of Atom primitive operations
-

Melt-Grow Planner

Notation: S is the starting Crystal.

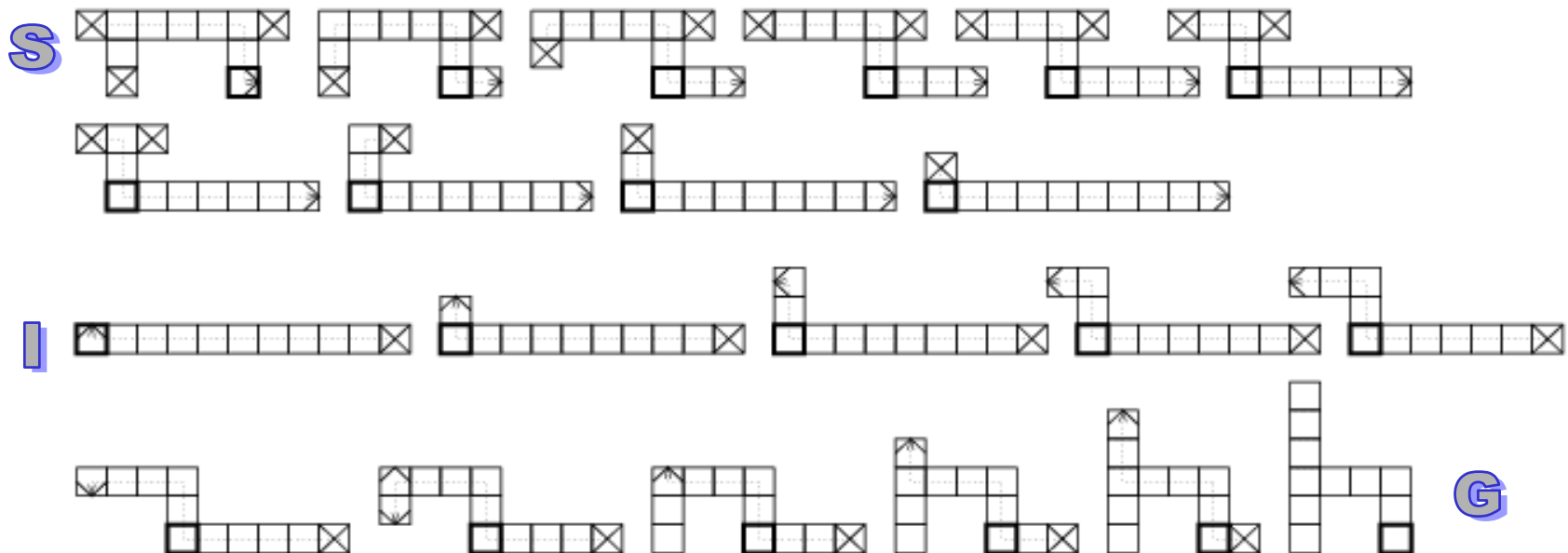
G is the goal Crystal.

I is an intermediate Crystal.

Input: S, G .

Melt-Grow: Melt S into I .

Grow G out of I .



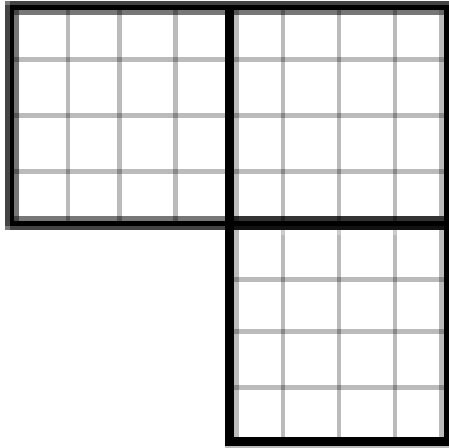
Melt-Grow Planner (cont'd)

- Centralized planning algorithm
 - Runs in $O(n^2)$ time, where n is the number of Atoms in the Crystal
 - Trades optimality for simplicity
 - Works on a useful subset *Grain(4)* of Crystals
-

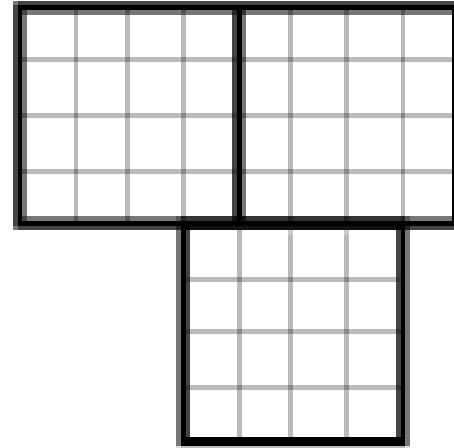
Grain(n) Class of Crystals

- Contains all Crystals that can be tiled by cubic (or square, in 2D) blocks of Atoms of side-length n
 - The set of planes (or edges, in 2D) that coincide with all sides of all blocks intersect only at block edges and corners
 - Each block of Atoms is a Grain
-

Grain(4) example



A 2D Crystal in
Grain(4)

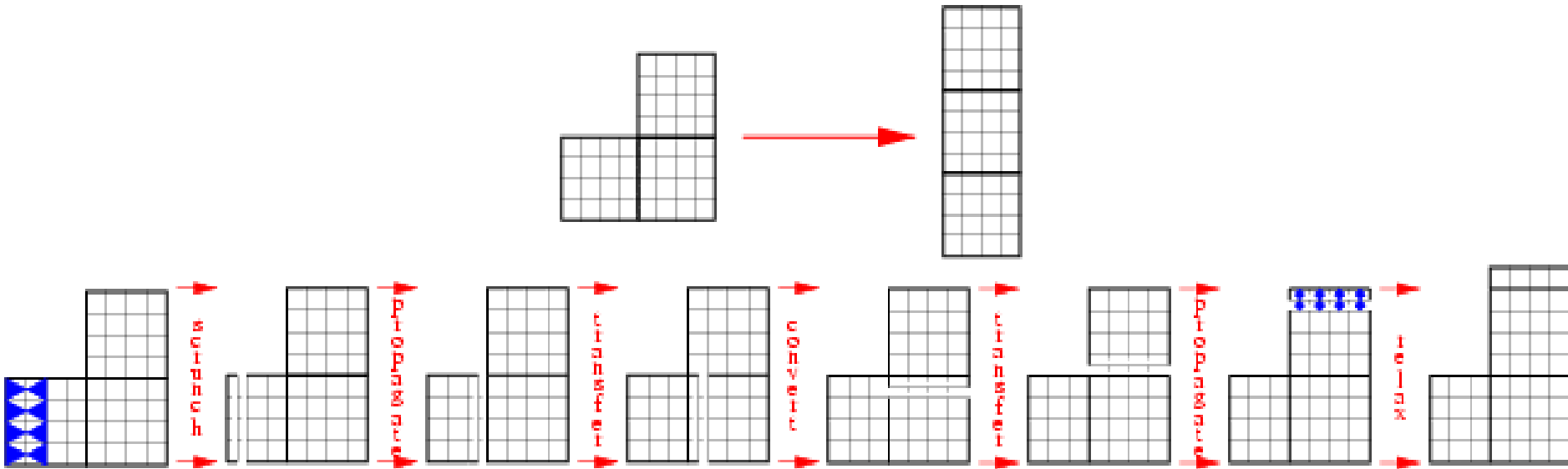


A 2D Crystal not
in *Grain(4)*. Note
the misalignment.

Why Use Grains?

- Moving Atoms requires other Atoms to act as “helpers”
 - Using grains ensures that helper Atoms are always available
 - Expansion and contraction operations now act on a whole face of the grain at one time
-

Example Reconfiguration Using Grains

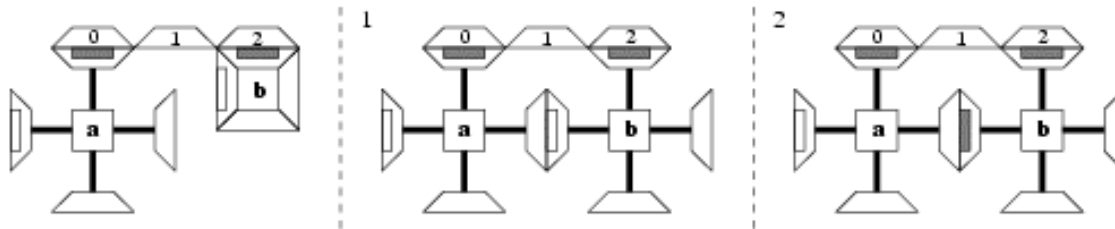


A row of Atoms is transferred from the left side of the Crystal to the top.

Physical Experiments with Self-reconfiguring Crystals

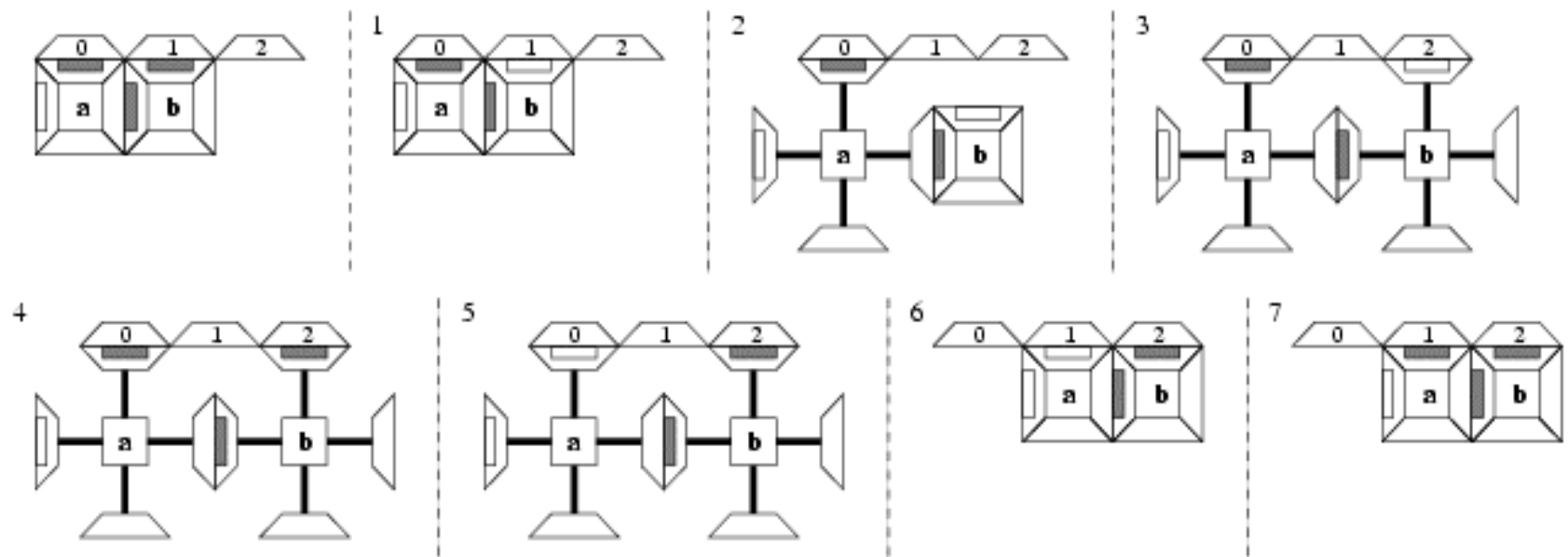
- Ten prototype modules constructed
- Row of eight fixed passive connectors used to simulate surface of a Crystal

1. expand b
2. connect b.w



First experiment involved expanding one module and attaching to an adjacent module

Physical Experiments with Self-reconfiguring Crystals (cont'd)



Second experiment tested an inchworm locomotion algorithm.

Physical Experiments with Self-reconfiguring Crystals (cont'd)

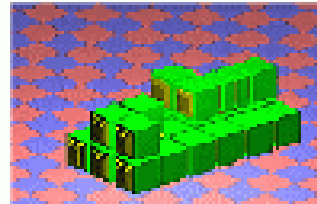
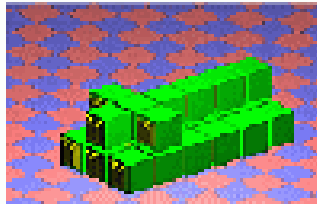
- Post-publication experimentation includes a more complex propagation in a set of ten joined modules



Simulated Experiments with Self-reconfiguring Crystals

- Facilitates the design and debugging of reconfiguration algorithms for larger scale structures
 - Both preplanned and automatic reconfiguration (such as with Melt-Grow) has been simulated
-

An Example Simulated Preplanned Reconfiguration (Dog to Couch)



Potential Applications and Drawbacks

- Ability to adapt would allow such a robot system to maneuver around, through, or over a wide degree of obstacles
 - Could navigate through narrow or awkward passageways
 - Seems suitable only for tasks that do not require rapid movement
 - The reconfiguration planning proposed is a centralized algorithm requiring a controller to know status of entire system
-