CARPENTRY COMPILER

Chenming Wu (Tsinghua, University of Washington)
Haisen Zhao (Speaker, University of Washington)
Chandrakana Nandi (University of Washington)
Jeffrey I. Lipton (University of Washington)
Zachary Tatlock (University of Washington)
Adriana Schulz (University of Washington)

Manufacturing Advances

Fabrication-Oriented Design

[Bächer et al. 2014]

[Wang and Whiting 2016]

[Bickel et al. 2010]

```
Diff (
  Scale (2.5, 2.5, 1) (
    Cylinder(6)
  )
  Scale (1, 1, 0.9) (
    Translate (0, 0, 0.5) (
      Cylinder(50)
    )
  )
)
```

[Nandi et al. 2018]

A design completely determines a fabrication process

[Schüller et al. 2018]

[Skouras et al. 2015]

[Bern et al. 2017]

[Konaković et al. 2016]

From Design to Fabrication Plan

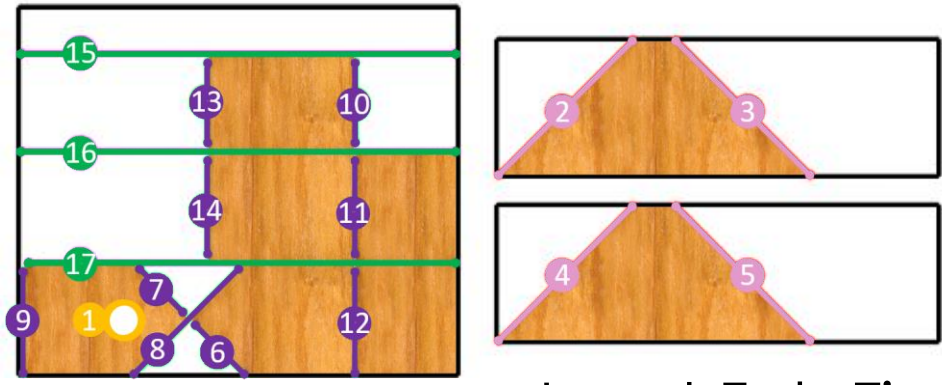
Design

Band saw

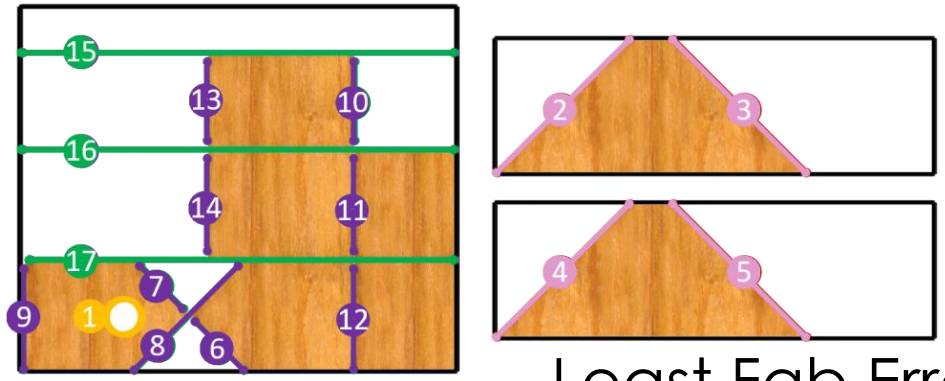
Jig saw

Table saw

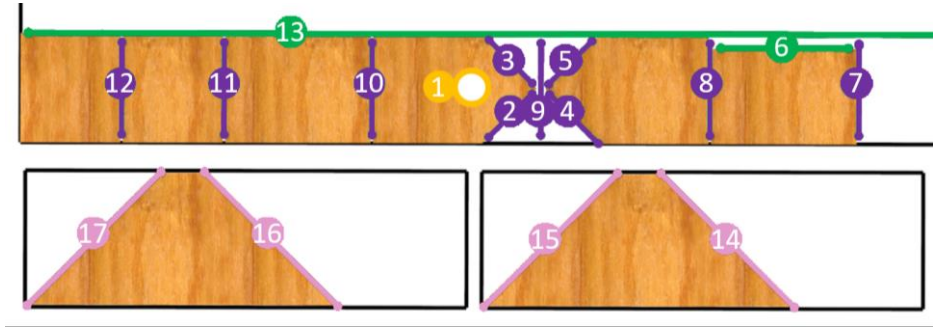
Chop saw



Least Fab Time



Least Fab Error

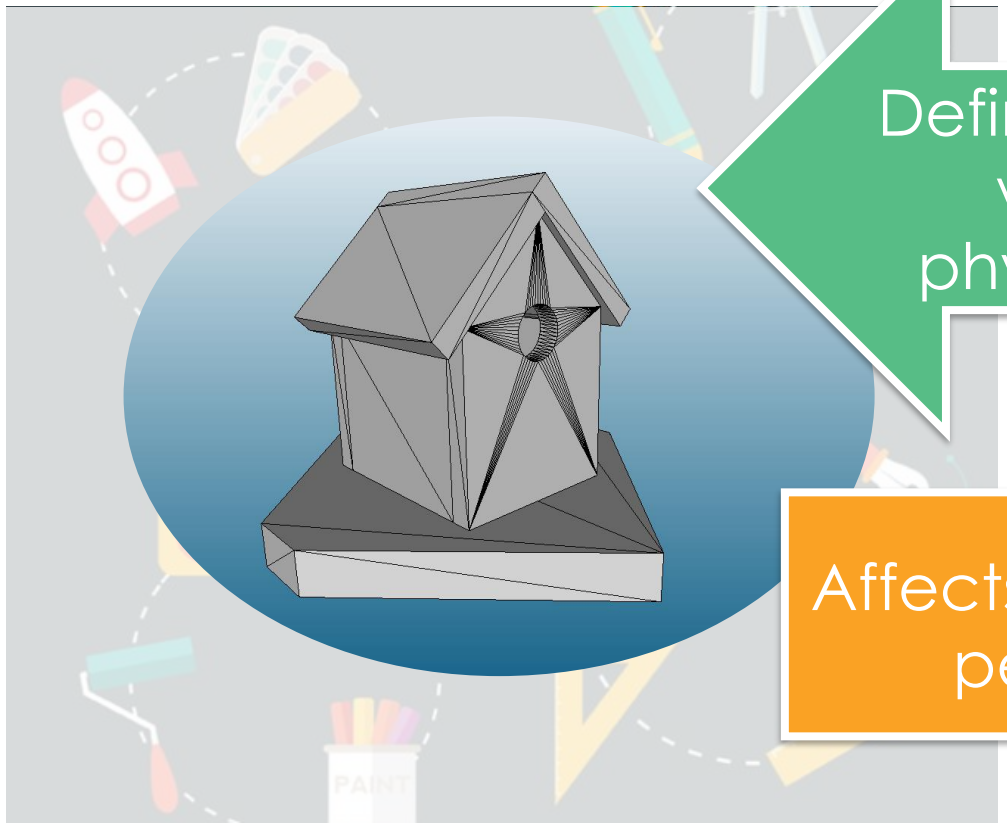


Least Material Cost

Fabrication and Design

Fabrication and design are inherently related.

Design



Defines the space of what can be physically realized

Affects the fabrication performance

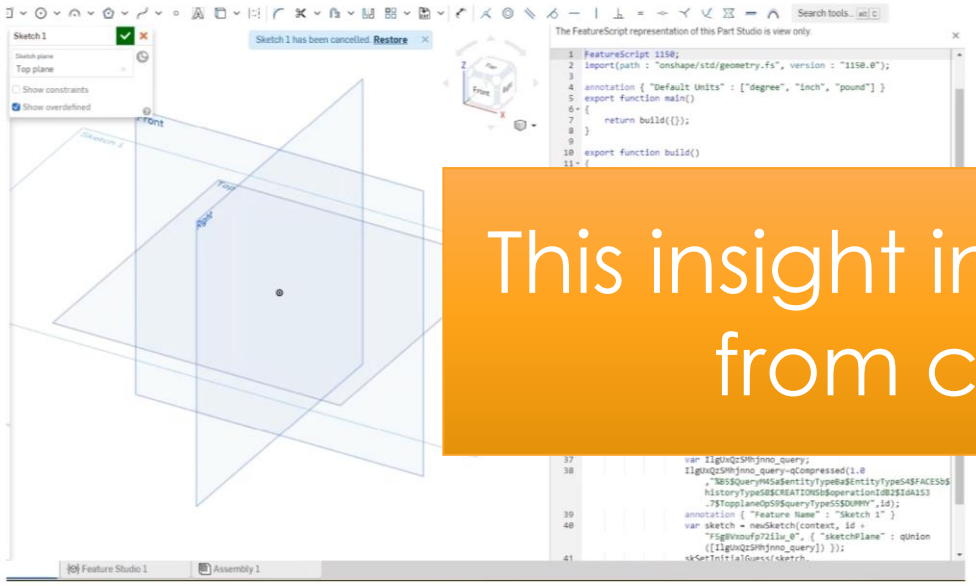
Fabrication

Optimization: physical behavior

Optimization: production cost

Insight: Designs and Fabrication Plans are Programs!

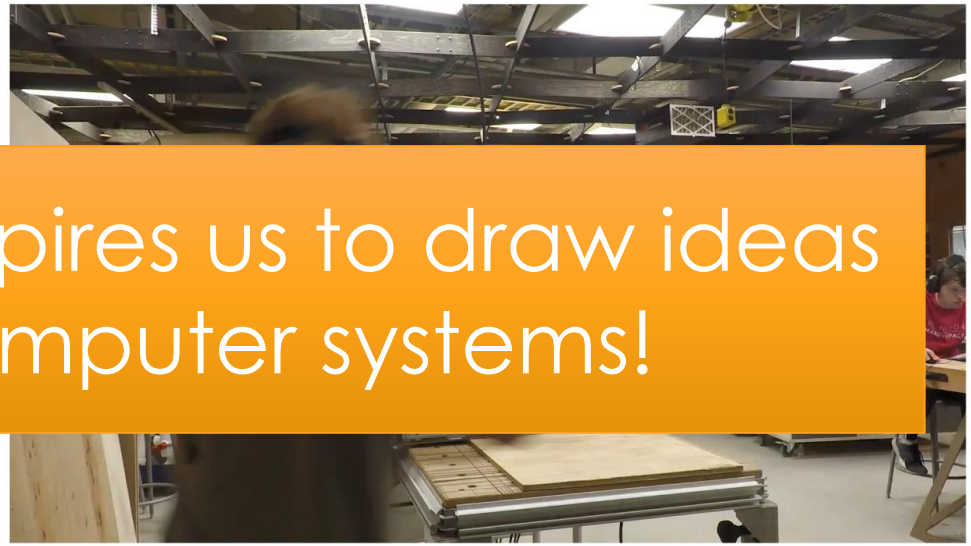
Design



[Onshape]

A sequence of geometric construction operations
(Code)

Fabrication

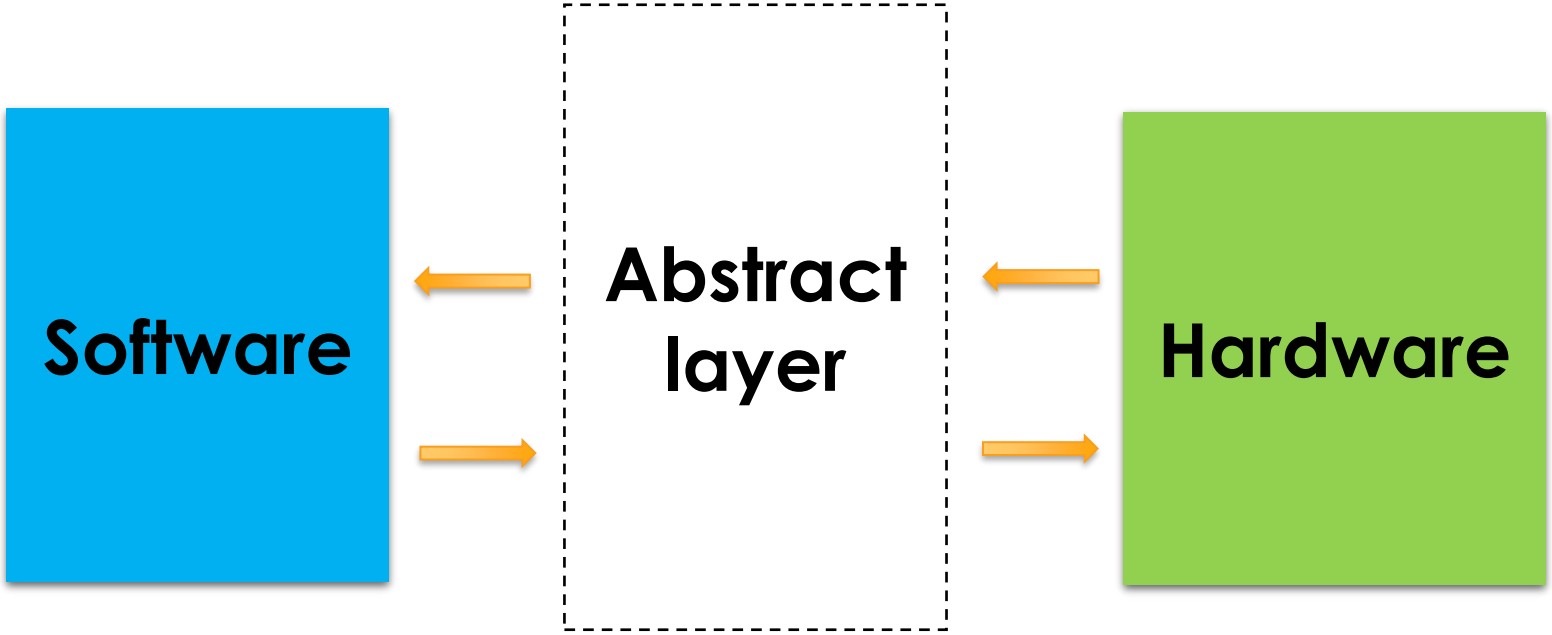


A sequence of physical instructions
(Code)

This insight inspires us to draw ideas from computer systems!

Inspiration: Instruction Set Architectures (ISA)

- An interface between software and hardware
- Enable the independent development



David A. Patterson and Carlo H. Sequin. 1981. RISC I: A Reduced Instruction Set VLSI Computer. In Proceedings of the 8th Annual Symposium on Computer Architecture (ISCA '81). IEEE Computer Society Press, Los Alamitos, CA, USA, 443–457. http://dl.acm.org/citation.cfm?id=800052.801895

Carpentry Design and Fabrication

- Vast application scope
- Appropriate level of complexity for initiating research





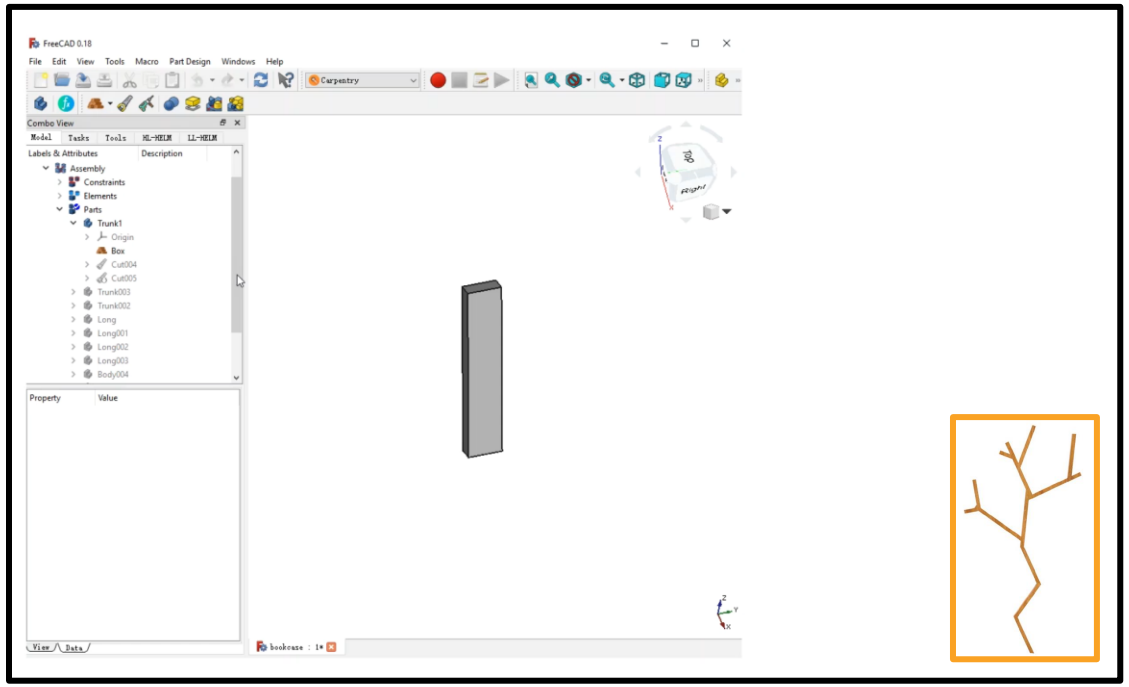

HELM: Hardware Extensible Languages for Manufacturing

Goal of our system:

- Ensure design is driven by available fabrication processes
- Decouple design and fabrication

HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages



High-Level Code

```
1  Box001 = Make_Stock(457.2, 38.1, 88.9);
2  MyLine000 = Line(457.2, 0, 435.203, 38.1);
3  Sketch = Make_Sketch(
4             Query_Face_By_Closest_Point(Box001, 228.6, 19.05, 88.9),
5             Geometry(MyLine000),
6             Constraint(Coincident(Start(MyLine000), End(
                    Query_Edge_By_Closest_Point(Box001, 228.6, 0, 88.9))),
                    PointOnObject(End(MyLine000), Query_Edge_By_Closest_Point(
                    Box001, 228.6, 38.1, 88.9)), Angle(Start(
                    Query_Edge_By_Closest_Point(Box001, 457.2, 19.05, 88.9)), Start
                    (MyLine000), 30)));
7  Cut = Make_Cut(Box001, Sketch, 0);
8  MyLine001 = Line(0, 38.1, 21.997, 0);
9  Sketch001 = Make_Sketch(
10              Query_Face_By_Closest_Point(Cut, 228.6, 19.05, 88.9),
11              Geometry(MyLine001),
12              Constraint(Coincident(Start(MyLine001), End(
                      Query_Edge_By_Closest_Point(Cut, 0, 19.05, 88.9))),
                      PointOnObject(End(MyLine001), Query_Edge_By_Closest_Point(
                      Cut, 228.6, 0, 88.9)), Angle(End(Query_Edge_By_Closest_Point
                      (Cut, 0, 19.05, 88.9)), Start(MyLine001), 30)));
```

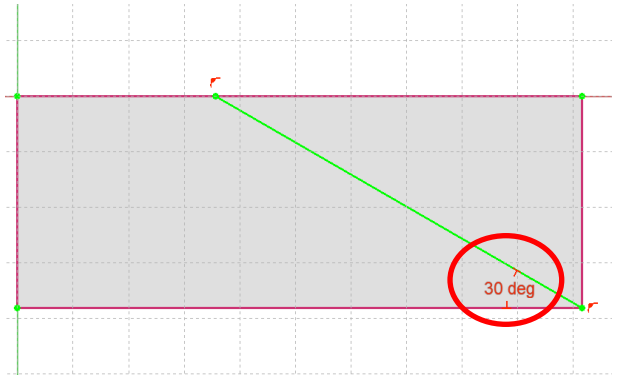
Design
High-Level HELM

Compiler

Fabrication
Low-Level HELM

HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages
- Subtractive: map to woodworking
- Parametric : design optimization



Make_Stock (6, 2, 4)
Make_Cut (6, 30^o)

Make_Stock (6, 2, 4)
Make_Cut (6, 45^o)

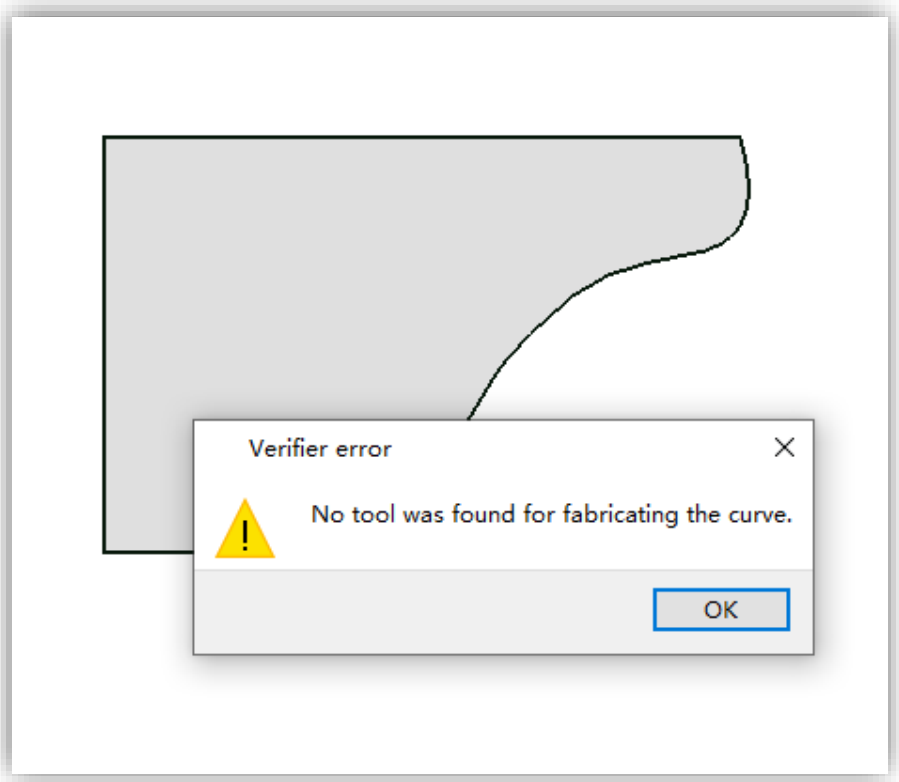
Design
High-Level HELM

Compiler

Fabrication
Low-Level HELM

HELM: Hardware Extensible Languages for Manufacturing

- Inspiration: CAD languages
- Subtractive: map to woodworking
- Parametric : design optimization
- Verifier: ensures manufacturability



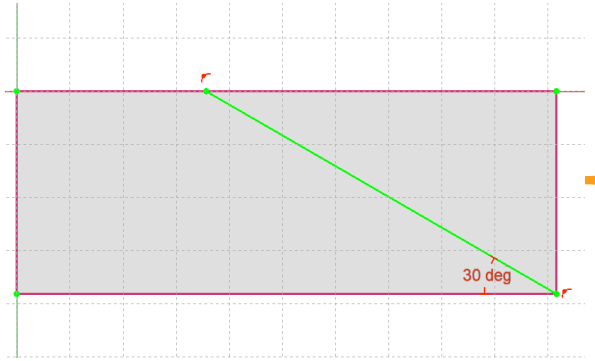
Design
High-Level HELM

Compiler

Fabrication
Low-Level HELM

HELM: Hardware Extensible Languages for Manufacturing

- Process specific: be followed to generate one design
- Extensible to more fabrication hardware



2x4_lumber 30^o

Setup_Chopsaw (30^o, 0 , 6)
Chopsaw (2x4_lumber)

Setup_Jigsaw (30^o, 0 , 6)
Jigsaw (2x4_lumber)

Design
High-Level HELM

Compiler

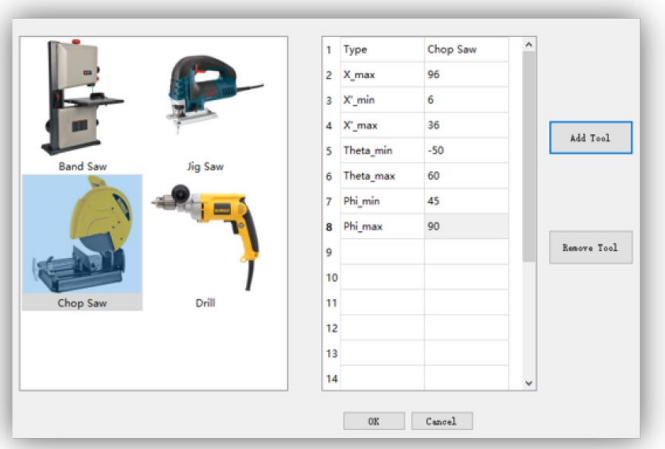
Fabrication
Low-Level HELM

HELM: Hardware Extensible Languages for Manufacturing

Make_Stock (6, 2, 4)
Make_Cut (6, 30^o)

- Design validation
- Fabrication optimization

Setup_Chopsaw (30^o, 0, 6)
Chopsaw (2x4_lumber)



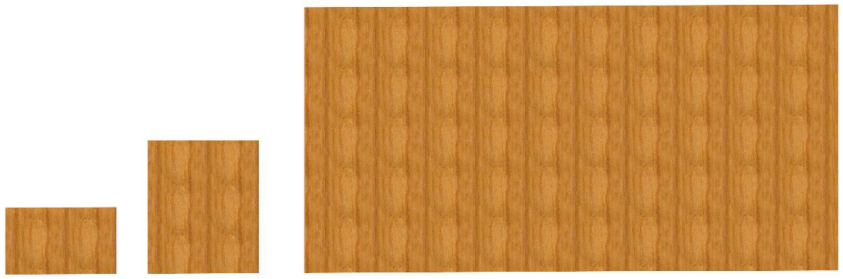
Design
High-Level HELM

Compiler

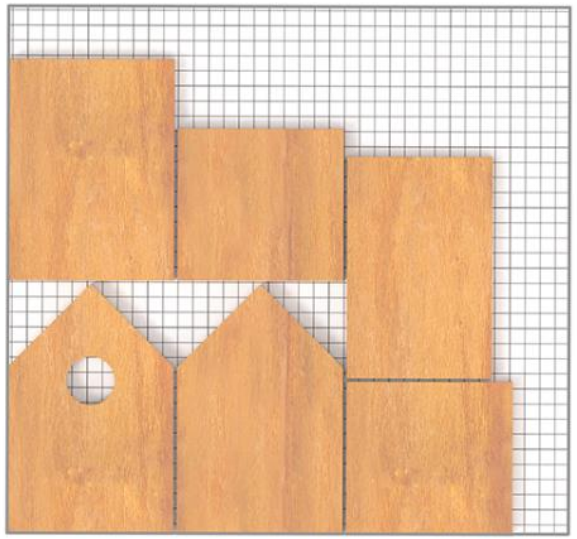
Fabrication
Low-Level HELM

Challenges of HELM Compiler

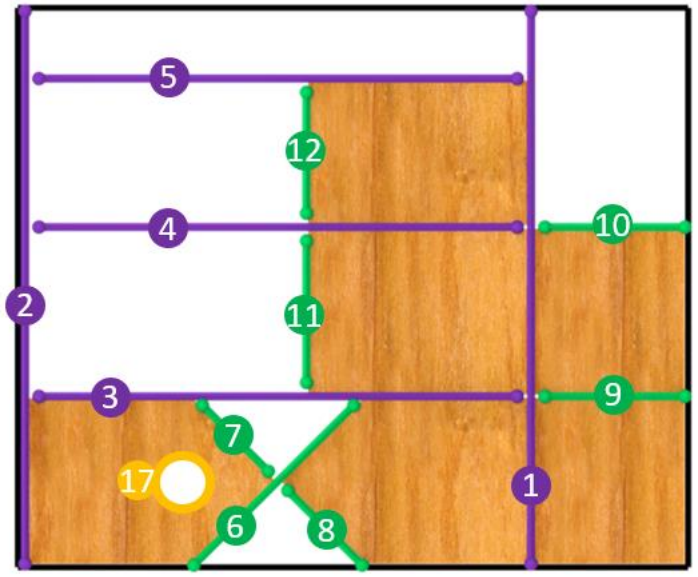
1. Long sequence of interdependent steps
   - need a data structure to represent such combinatorial space



Choose material stocks

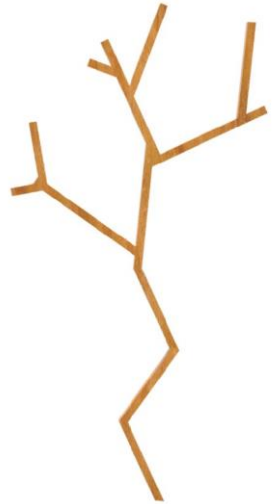


Pack the parts

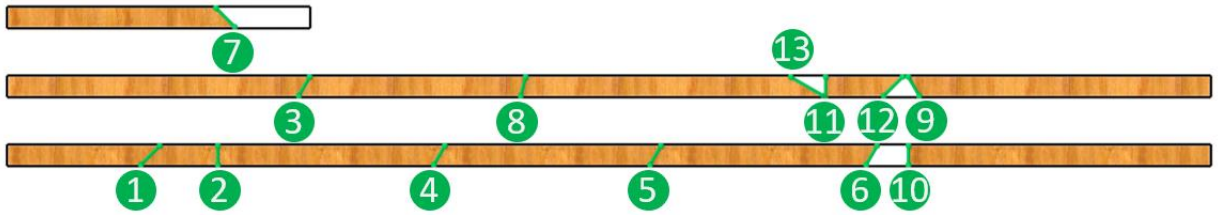


Define Cuts Tools/Orders on Stock

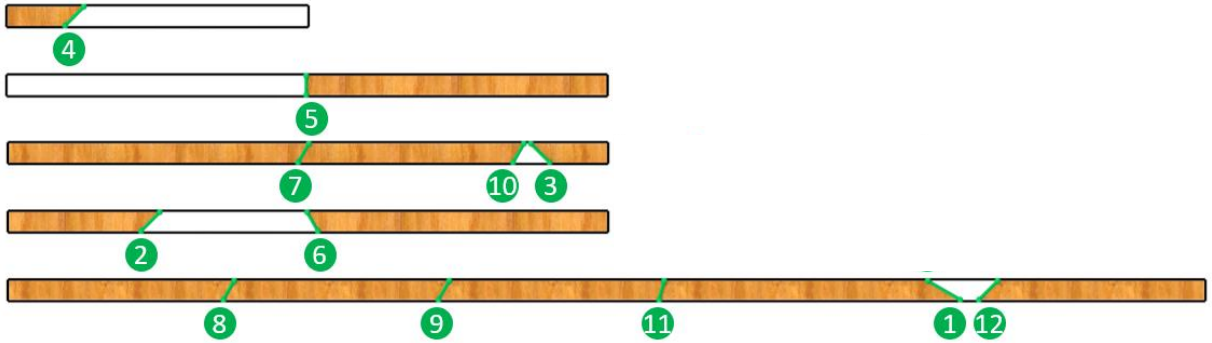
Challenges of HELM Compiler



1. Long sequence of interdependent steps
   - need a data structure
2. Multiple (conflicting) fabrication costs
   - multi-objective optimization



- Material cost: 2.3
- Fabrication time: 8.5



- Material cost: 2.95
- Fabrication time: 5

Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure
2. Multiple (conflicting) fabrication costs
   - multi-objective optimization

Learning from Programming Languages

❖ E-graphs [Joshi et al. 2002; Tate et al. 2009]

- **Compactly** and **efficiently** represent large scale **equivalence programs**
- Define a data-structure that make this search **tractable**

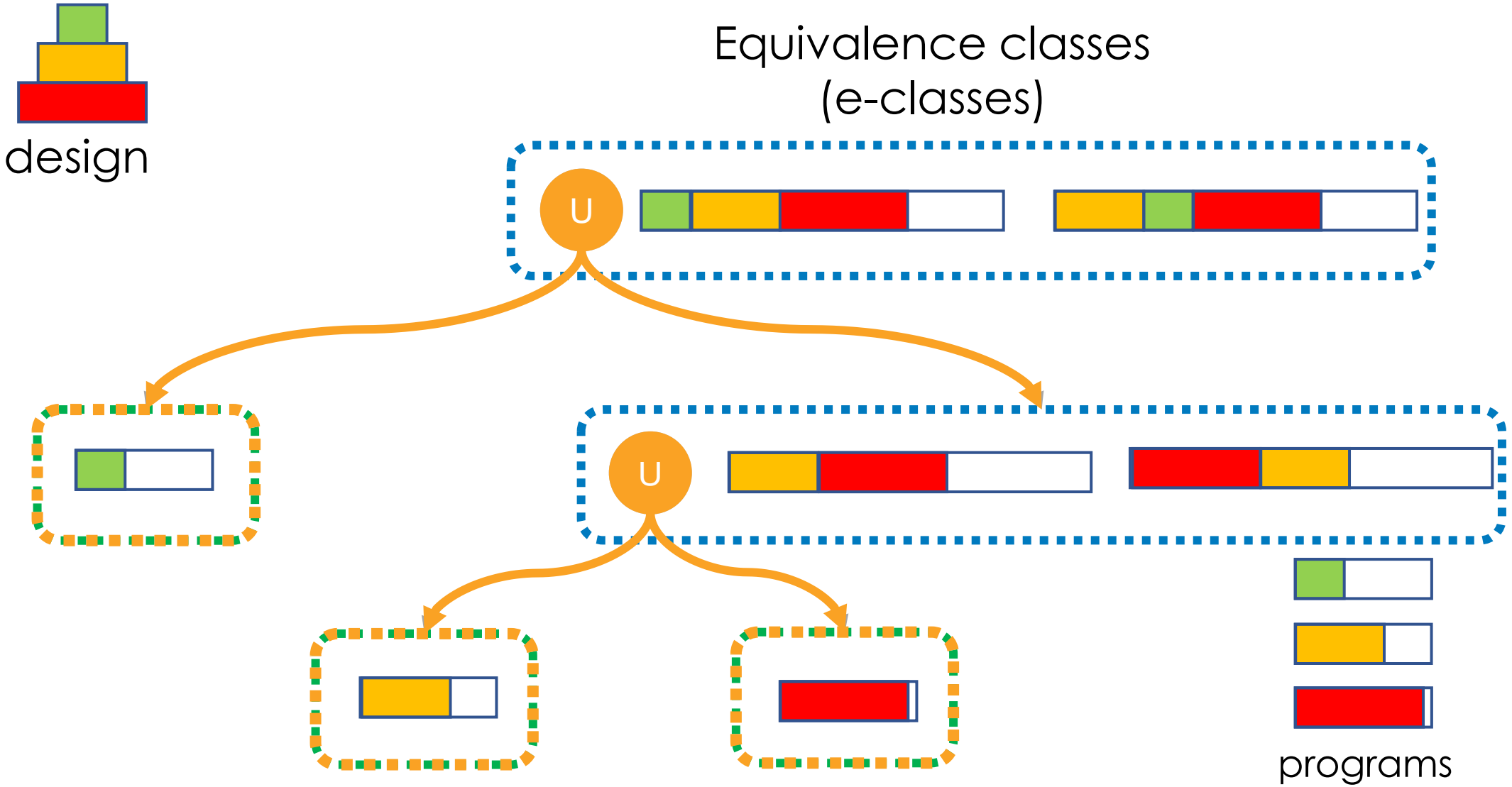
$$3x + \frac{1}{3}y - \frac{(x+2(x+y))}{3} \quad \xrightarrow{\text{simplify}} \quad 2x - \frac{1}{3}y$$

$$a + b = b + a$$
$$(a + b)/c = b/c + a/c$$
$$\ldots\ldots$$

Rewrite rules

Rajeev Joshi, Greg Nelson, and Keith Randall. 2002. Denali: A Goal-directed Superoptimizer. SIGPLAN Not. 37, 5 (May 2002), 304–314. https://doi.org/10.1145/543552. 512566
Ross Tate, Michael Stepp, Zachary Tatlock, and Sorin Lerner. 2009. Equality Saturation: A New Approach to Optimization. In Proceedings of the 36th Annual ACM SIGPLANSIGACT Symposium on Principles of Programming Languages (POPL '09). ACM, New York, NY, USA, 264–276. https://doi.org/10.1145/1480881.1480915

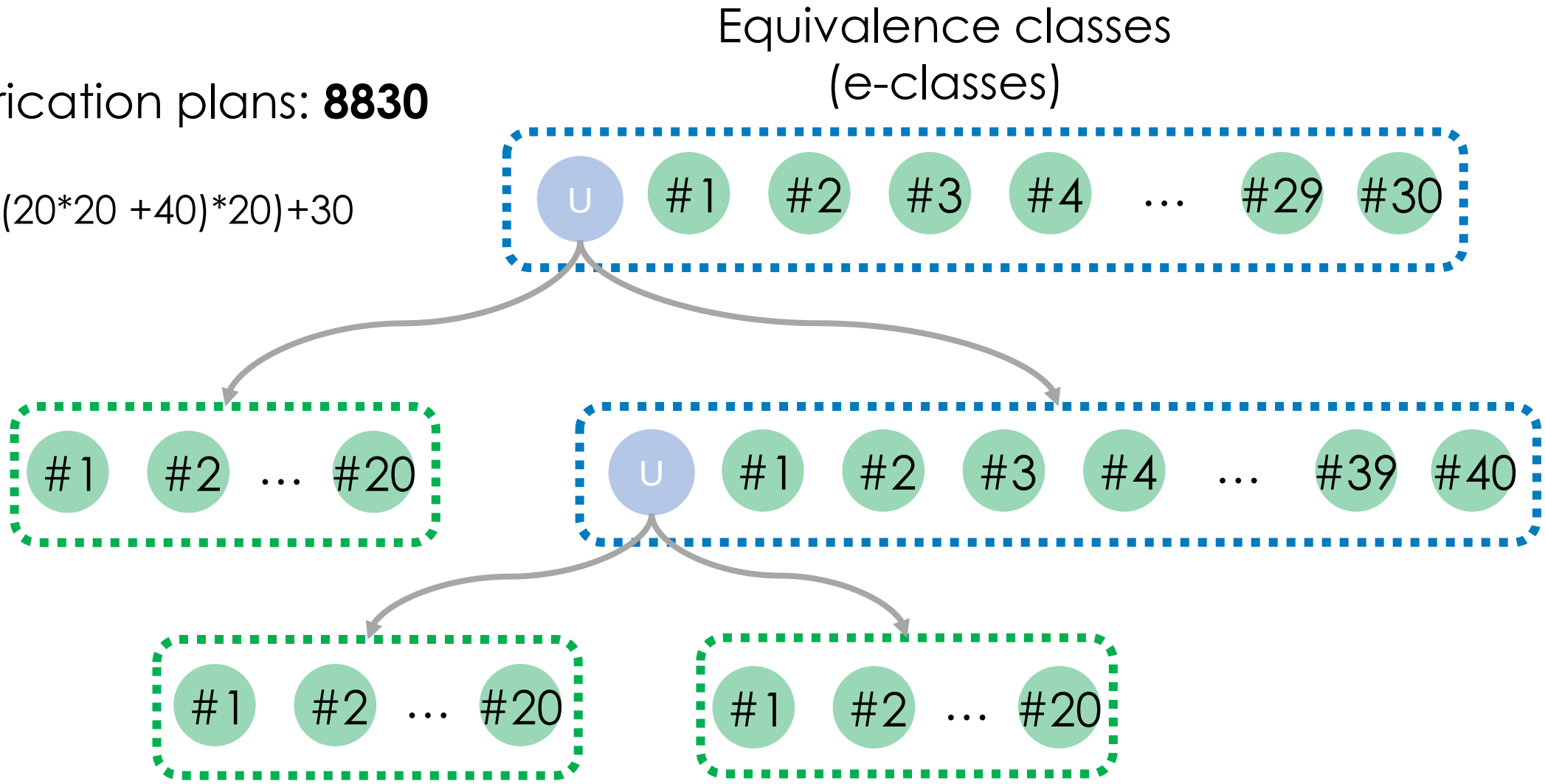
E-Graphs for Carpentry Compiler



E-Graphs for Carpentry Compiler

#Fabrication plans: **8830**

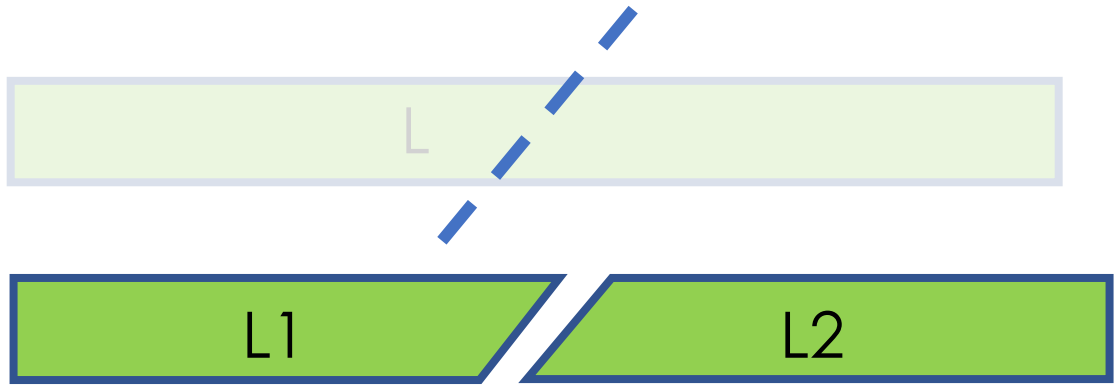
((20*20 +40)*20)+30



Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables

```
X = 10;
Y = X + 5;
Z = X + Y ;
```



No need linearity constraints

Fabrication needs linearity constraints

Common E-Graphs Applications

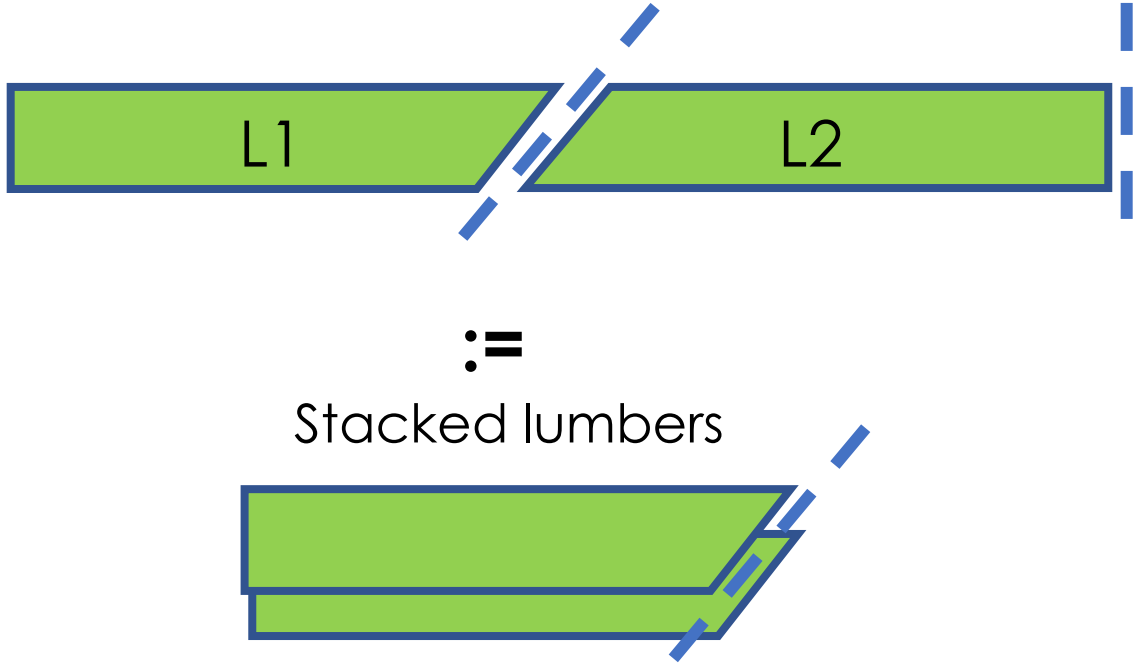
Carpentry Compiler

Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables
2. Rewrite rules

$$a + b = b + a$$
$$(a + b)/c = a/c + b/c$$
$$a - b = a + (-b)$$
$$a * b = b * a$$
$$\ldots$$

Syntactic rewrite rules



Geometric information

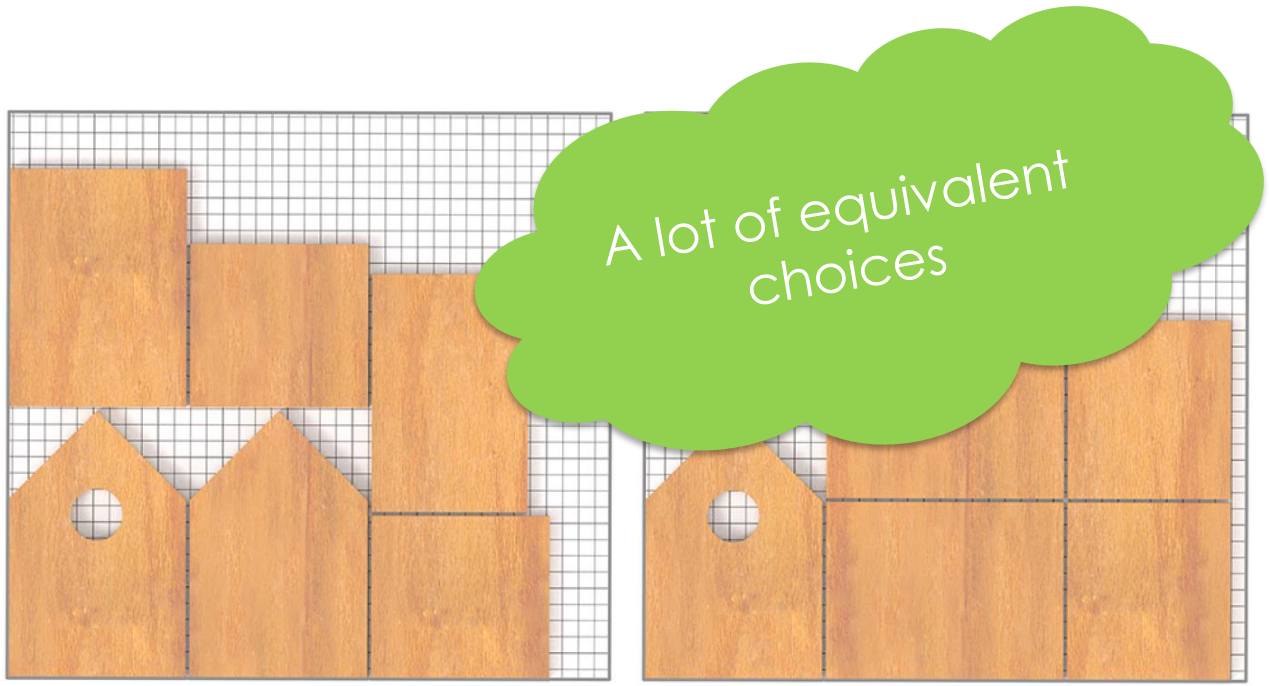
Common E-Graphs Applications

Carpentry Compiler

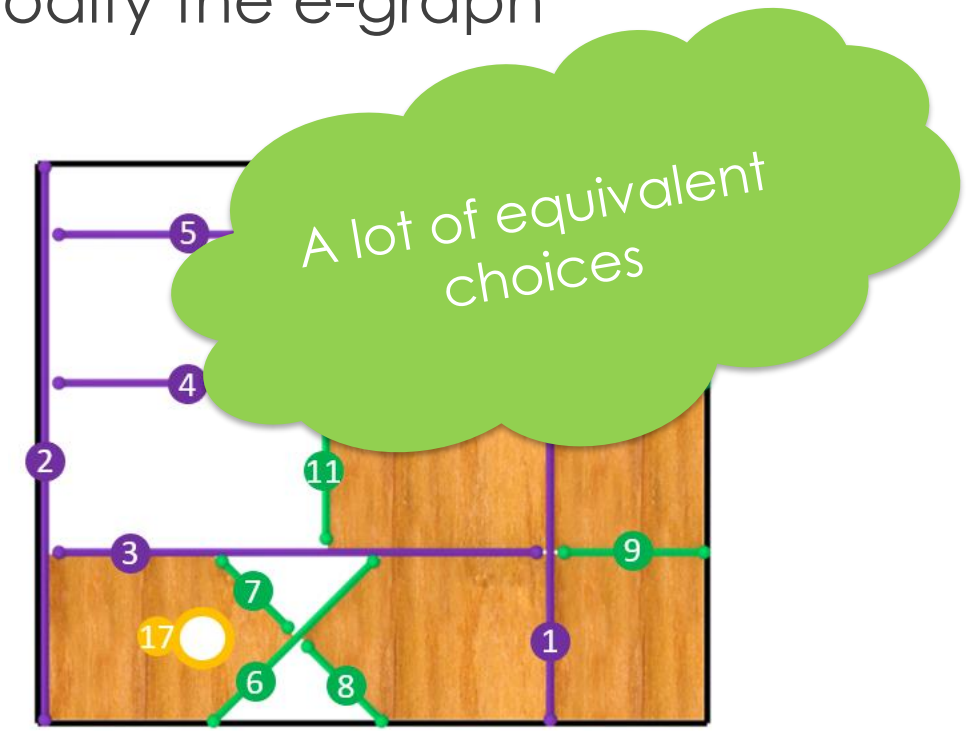
Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables
2. Rewrite rules

❖ A geometric method to populate and modify the e-graph



1: Pack pieces onto stocks

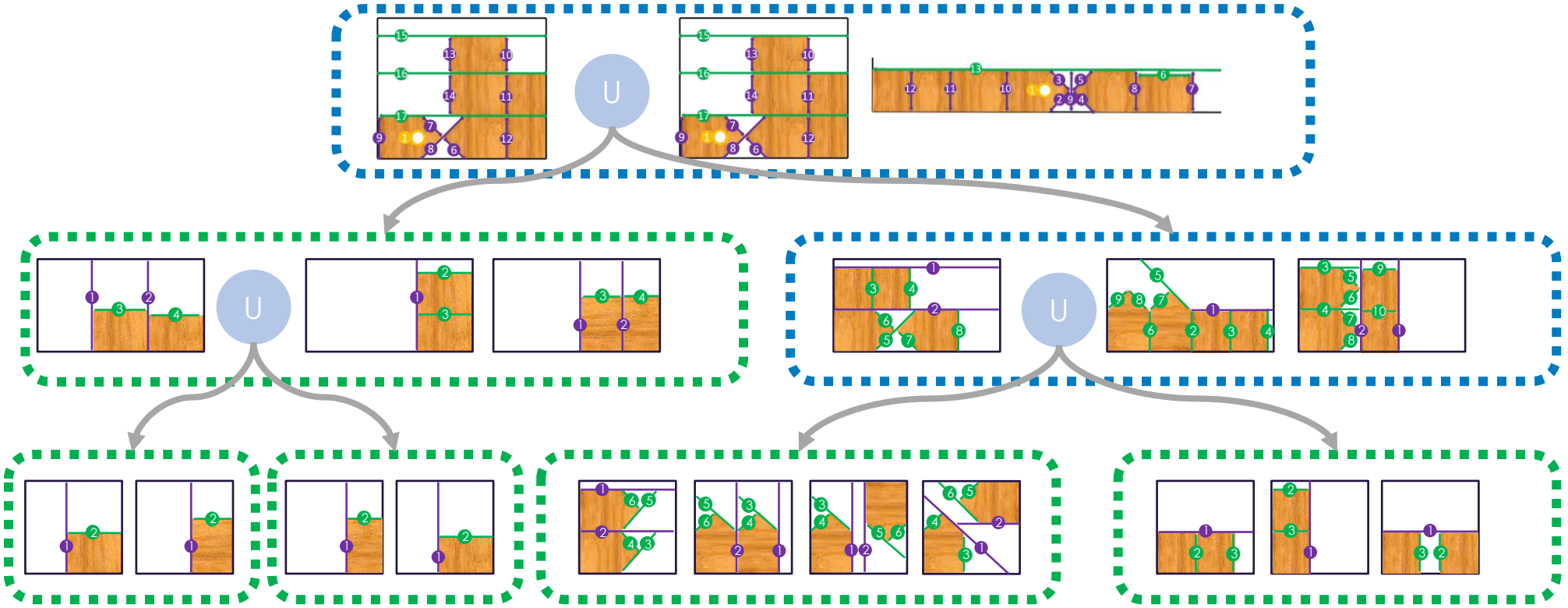


2: Define Cuts Tools/Orders on Stock

Challenges of E-Graphs for Carpentry Compiler

1. Linearity: reuse-ability of variables
2. Rewrite rules

❖ A geometric method to populate and modify the e-graph



Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure

❖ Apply **E-Graphs** to this carpentry domain. ✓

2. Multiple (conflicting) fabrication costs
   - multi-objective optimization
   - extract the optimal solution from E-graph

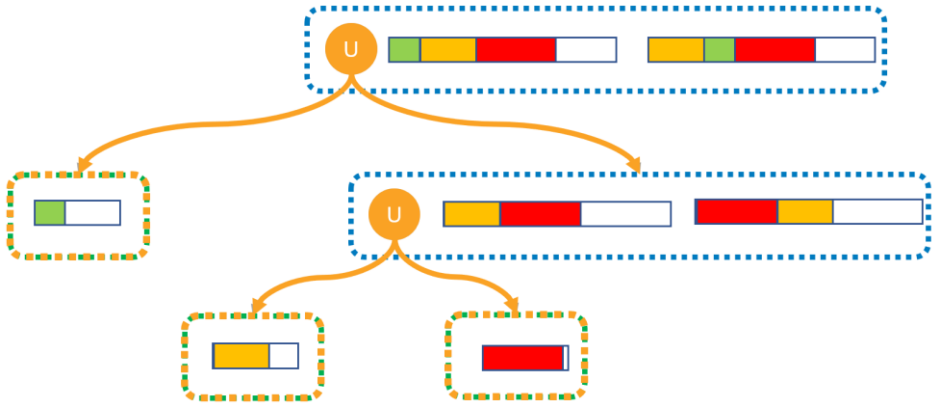
Extract the optimal solution from E-graph

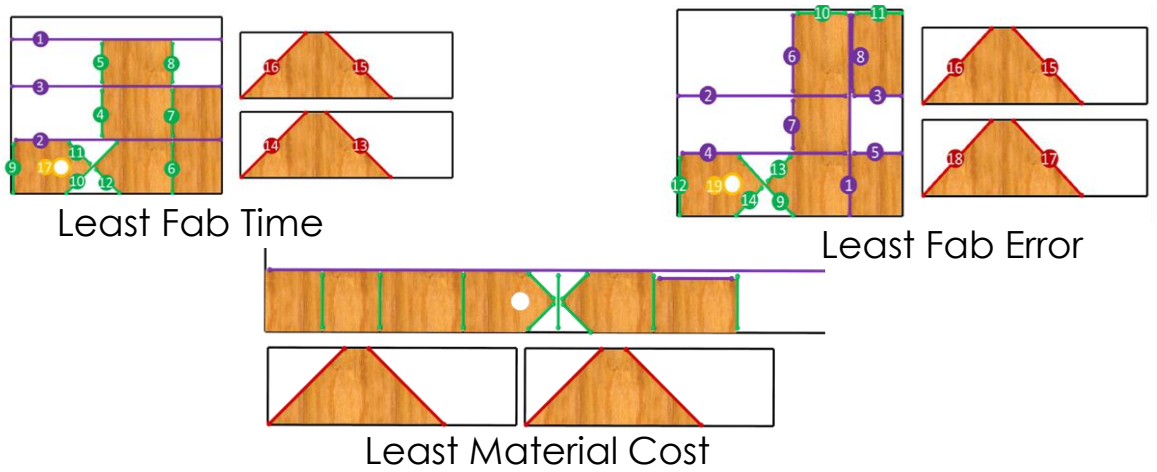
Single objective:

- Running time or Memory cost

Multiple objectives:

- Material cost
- Fabrication time
- Precision





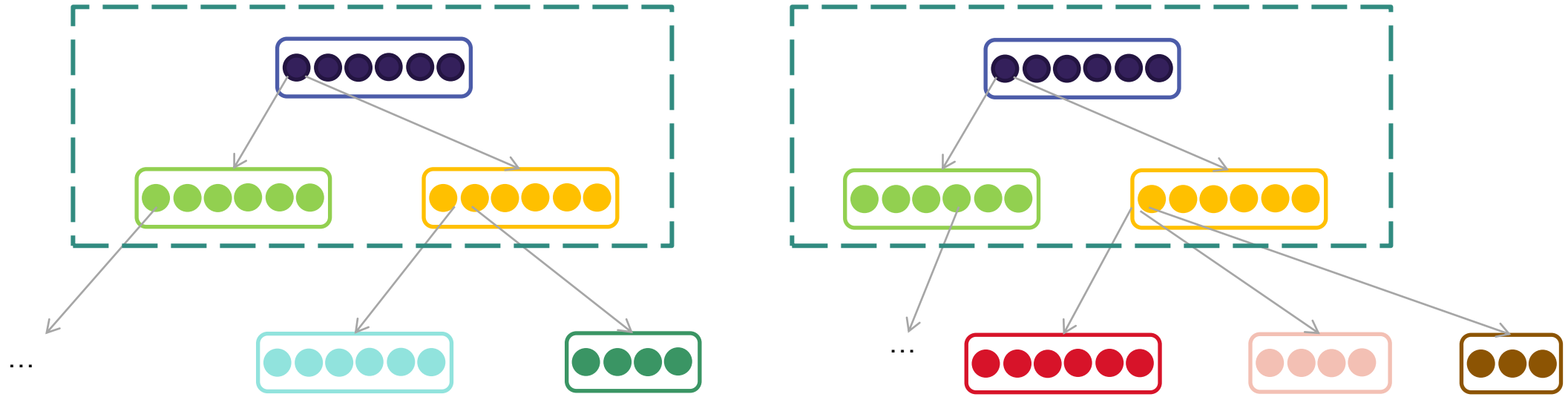
Common E-Graphs Applications

Carpentry Compiler

Extract the optimal solution from E-graph

❖ Genetic algorithm for multi-objective optimization in the E-Graphs

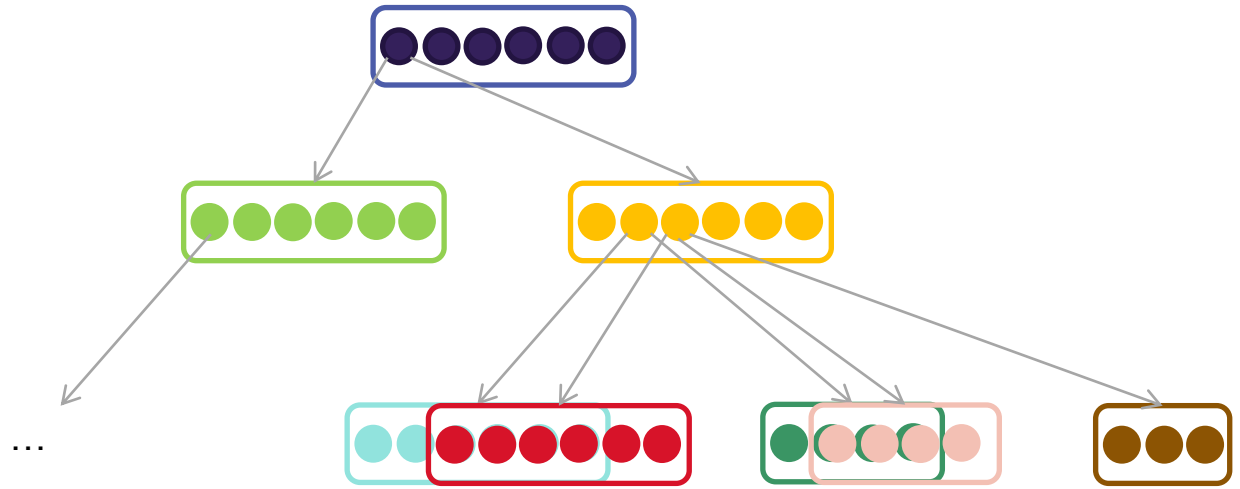
Cross-over



Extract the optimal solution from E-graph

❖ Genetic algorithm for multi-objective optimization in the E-Graphs

Mutation



Randomly expand e-graphs

Challenges of HELM Compiler

1. Long sequence of interdependent steps
   - need a data structure

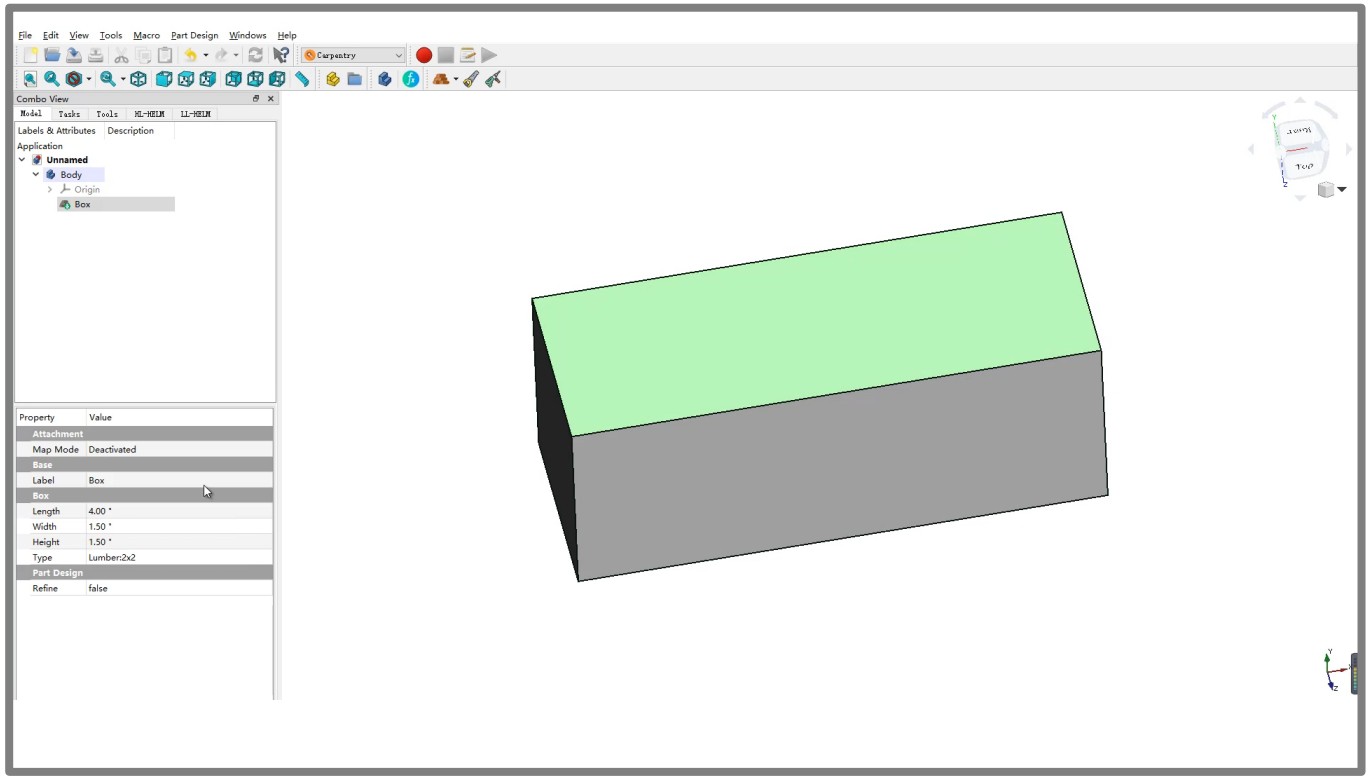
❖ Apply **E-Graphs** to this carpentry domain. ✓

2. Multiple (conflicting) fabrication costs
   - multi-objective optimization
   - extract the optimal solution from E-graph

❖ **Genetic algorithm** for multi-objective optimization ✓

Results

Design Results by Domain Experts




(A)

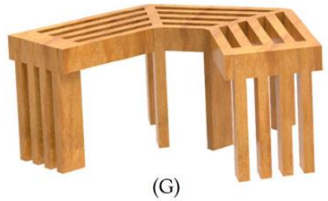

(B)


(C)


(D)

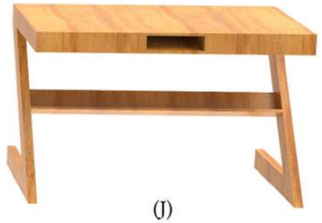

(E)


(F)


(G)


(H)


(I)


(J)

(A)

(B)

(C)

(D)

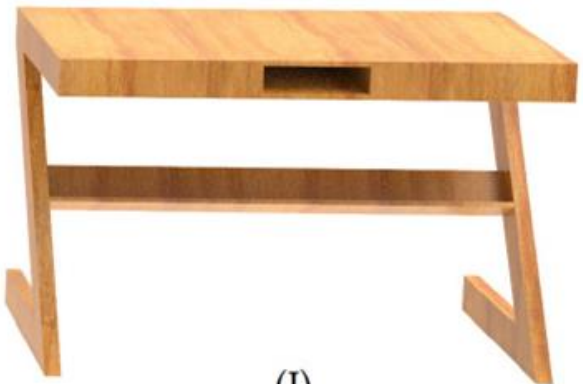
(E)

(F)

(G)

(H)

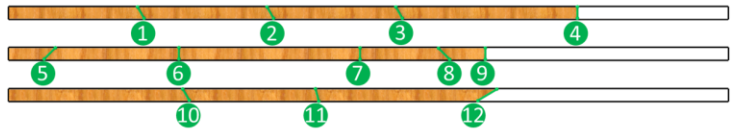
(I)



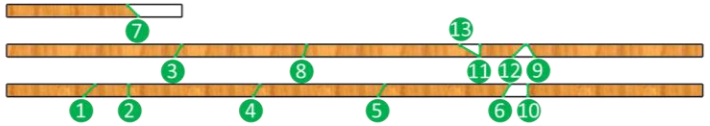
(J)

Optimization Results – Expert Comparison

9.C: Bookcase

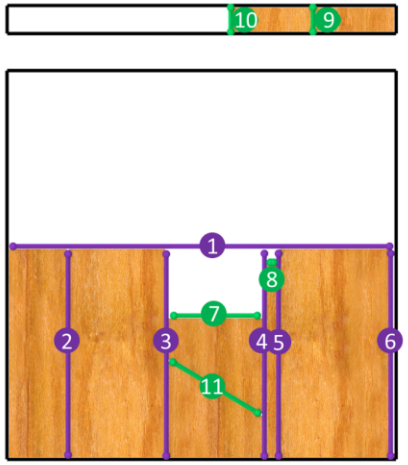


Material Cost: 3.0
Precision: 1.0
Fabrication Time: 9.5



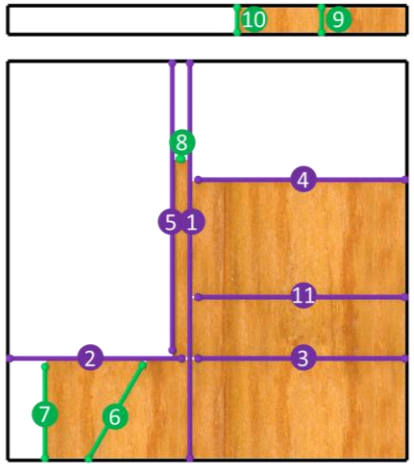
Material Cost: 2.3
Precision: 1.0
Fabrication Time: 8.5

9.I: Flowerpot



Material: 1.3
Precision: 1.06
Fabrication Time: 23.0

Experts



Material Cost: 1.3
Precision: 1.03
Fabrication Time: 22.5

Our solution

Bandsaw Drill Tracksaw Chopsaw Jigsaw

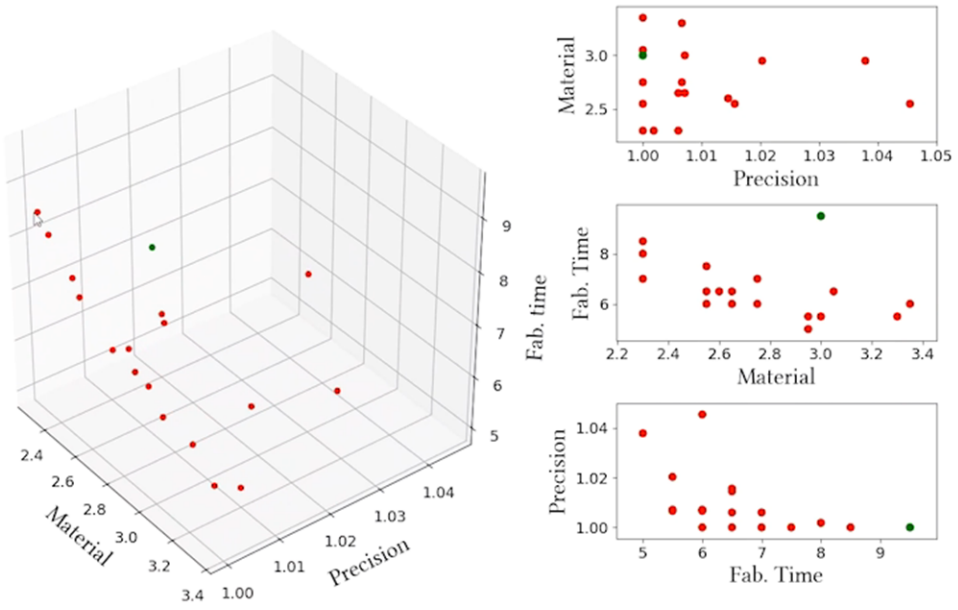
Optimization Results – Expert Comparison

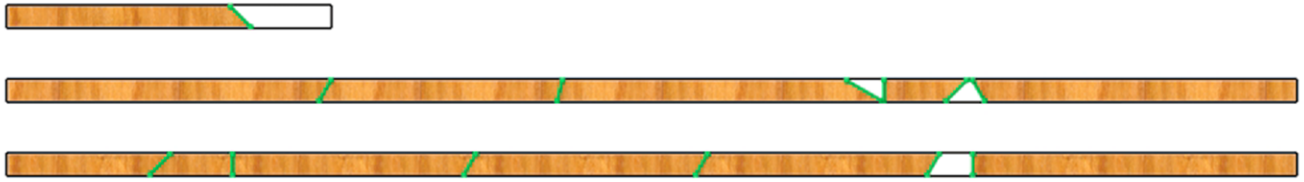
- A tool that can be used for **non-experts**
- Help **expert carpenters** search this large combinatorial space

	Experts			Ours			
	Mat. Cost	Precision	Fab. Time	Mat. Cost	Precision	Fab. Time	
 Adirondack chair	4.0	1.05	6.5	**3.85**	**1.02**	**4.5**	✓
 Drafting Table	9.8	1.29	16	**9.05**	**1.17**	**16**	✓
 Book case	3.0	1.0	9.5	**2.3**	**1.0**	**8.5**	✓
Bird house	2.7	1.57	32	**2.7**	**1.57**	**31.5**	✓
 Dining room chair	4.3	1.06	13.5	**3.75**	**1.02**	**10.5**	✓
Bench	8.0	1.06	7	**5.3**	**1.05**	10	
 Coffee table	10.48	5.72	69.5	**10.43**	**5.06**	**54.5**	✓
Flower pot	1.3	1.06	23	**1.3**	**1.03**	**22.5**	✓
 Z-table	9.15	1.14	17	**9.05**	**1.09**	**16.5**	✓

Trade-off Exploration

- Return **multiple** solutions with different **trade-offs**
- Allow users to **interactive explore** the solution space





Material cost: 2.3
Precision cost: 1.0
Time cost: 8.5

Physical Models

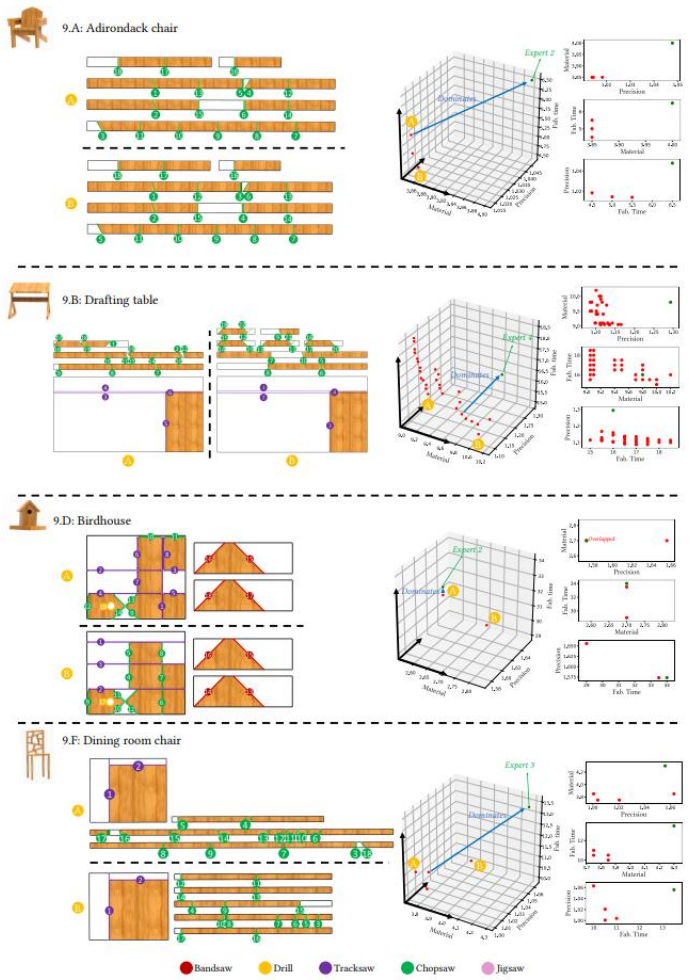
Bird house

Toy car

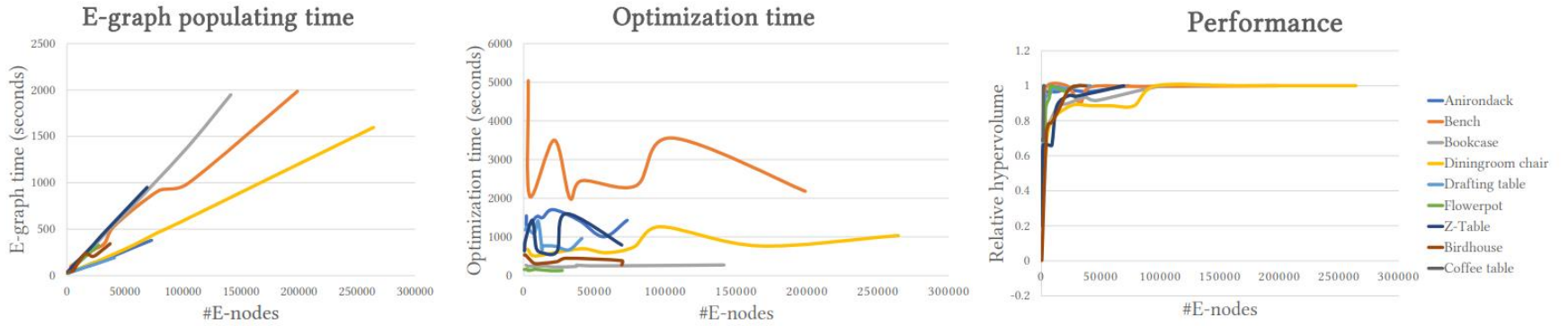
Book case

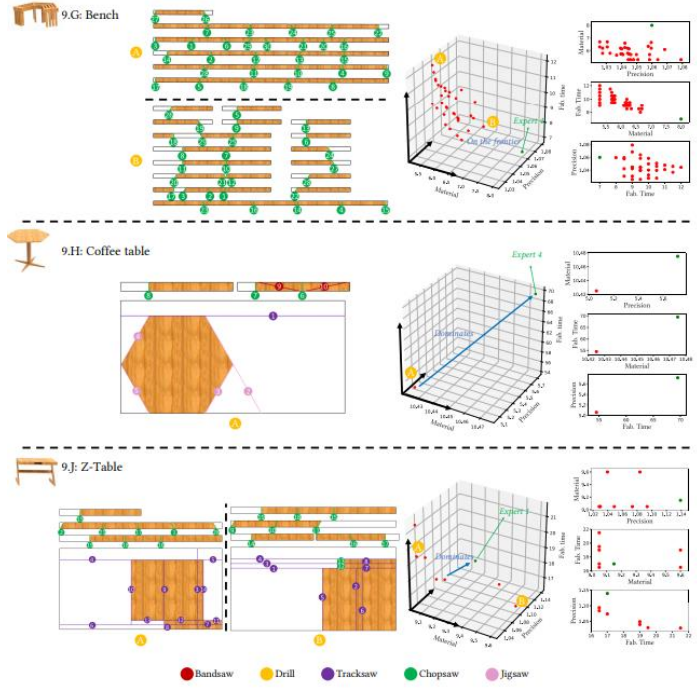
Validation

Additional results

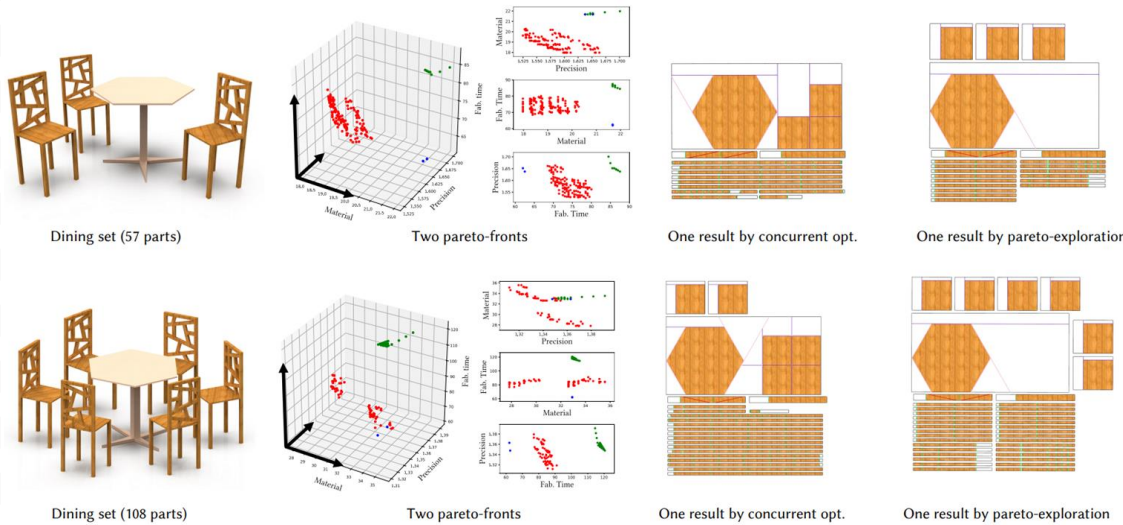


Statistics





Large scale tests



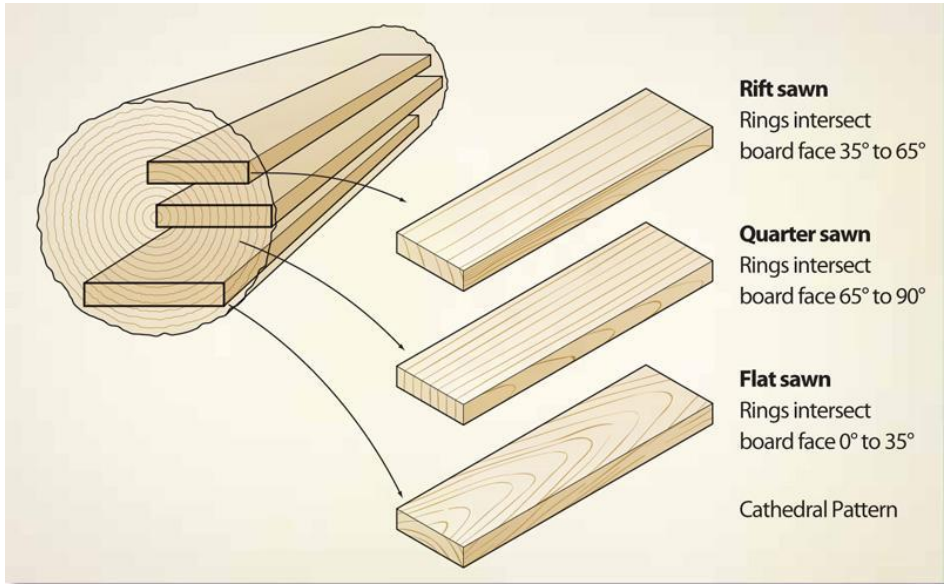
Conclusion

HELM: Hardware Extensible Language for Manufacturing

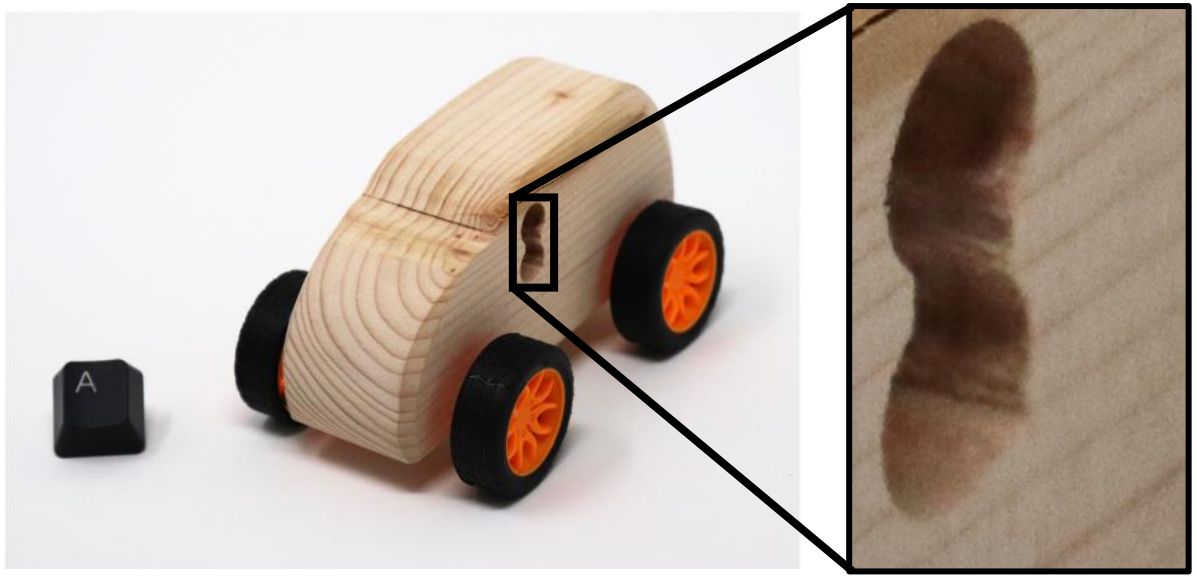
- Key insight: fabrication plans are programs
- New DSLs for HL-HELM and LL-HELM
- Applied and extended compiler technique
- Multi-objective optimization

Future work

- Carpentry Extensions



Grain orientation



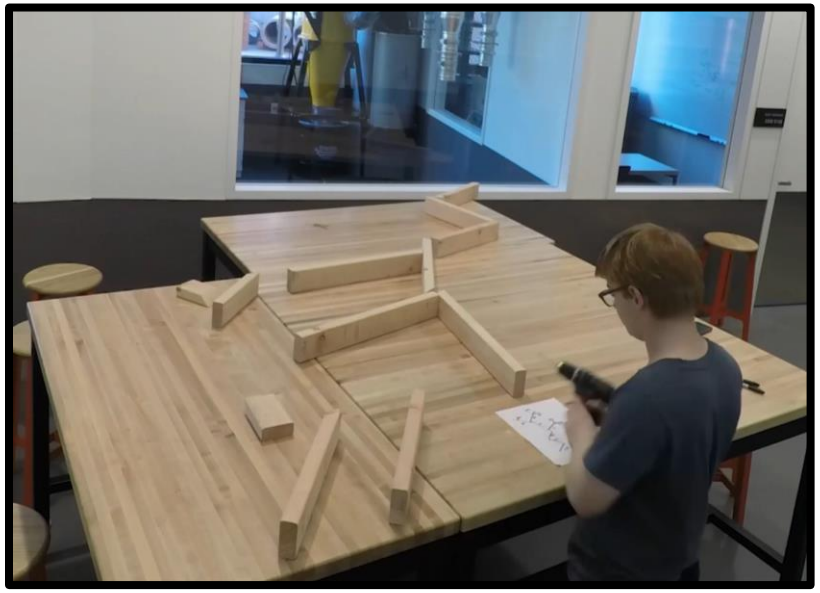
Fabrication uncertainty

Future work

- Carpentry Extensions
- Other processes

Additive Manufacturing

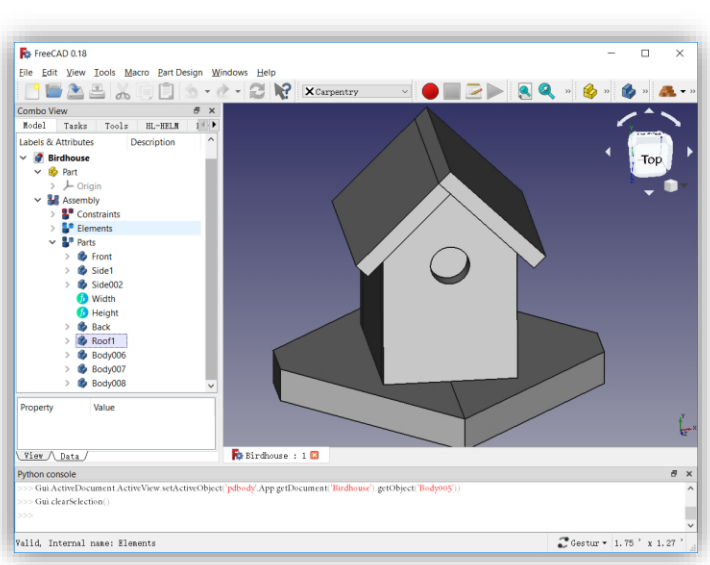
Subtractive Manufacturing



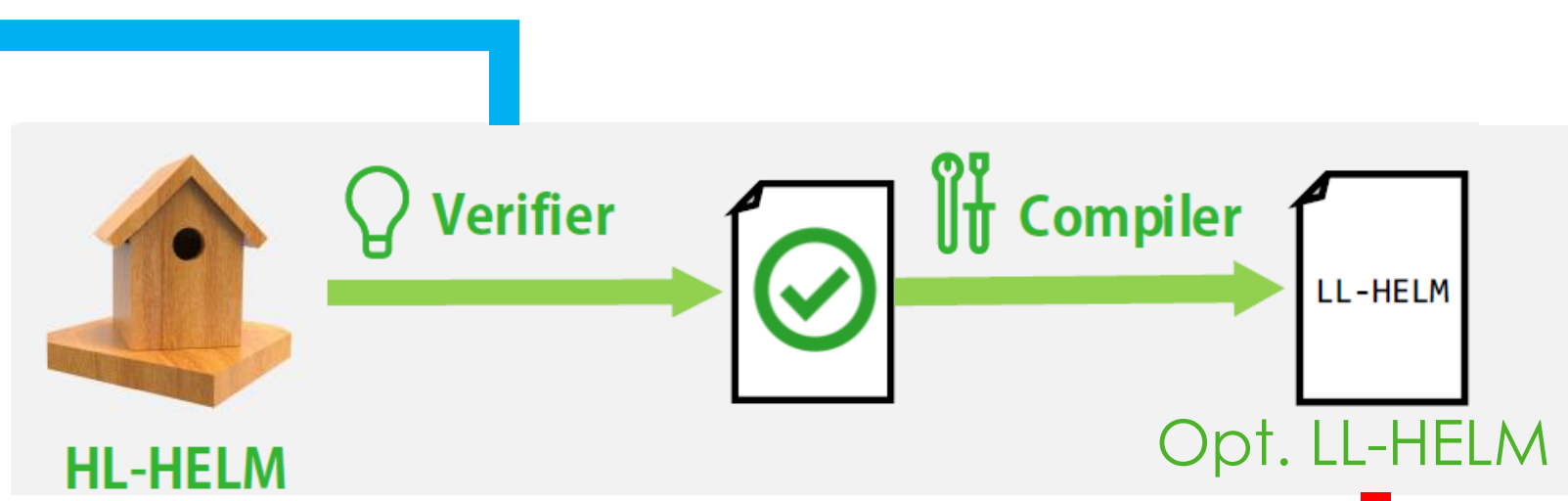
Assembly

Future work

- Carpentry Extensions
- Other processes
- Design Optimization



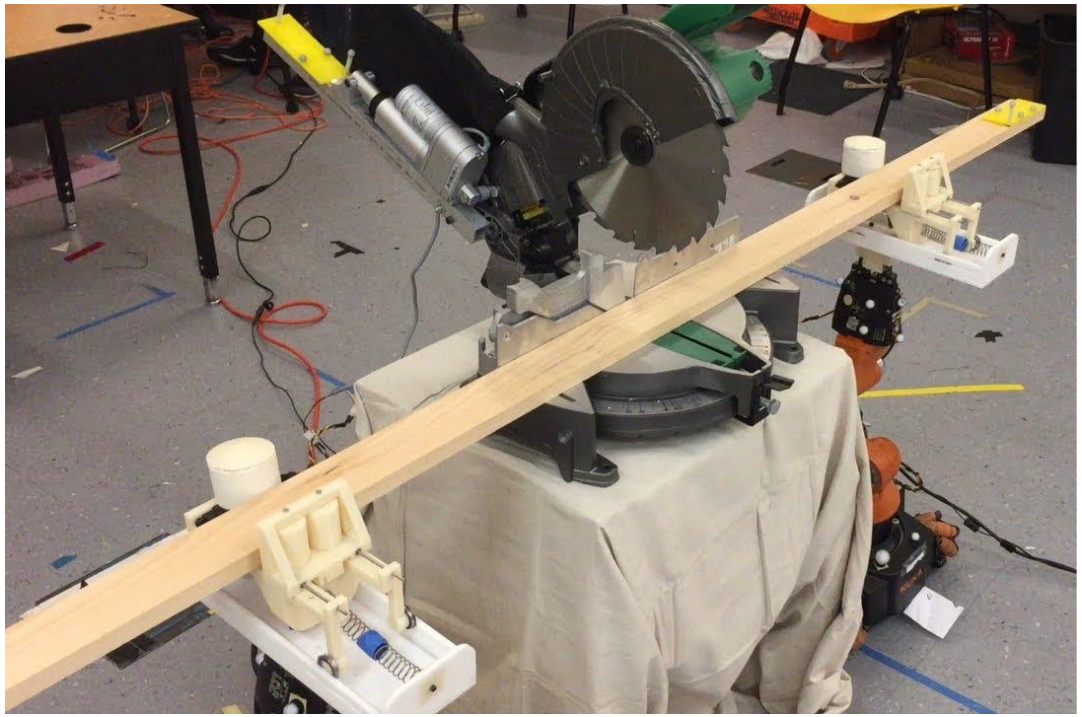
Design UI



Fabrication-cost-aware design optimization

Future work

- Carpentry Extensions
- Other processes
- Design Optimization
- Scheduling for future workshops



[1] Lipton J I, Schulz A, Spielberg A, et al. Robot Assisted Carpentry for Mass Customization[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 1-8.
[2] Lipton J I, Manchester Z, Rus D. Planning cuts for mobile robots with bladed tools[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 572-579.

Thank you!

Acknowledgement:
Anonymous reviewers, Dan Grossman, Benjamin T. Jones, Colin Topper, Victor Wu, Mary Mattsen, Guoxin Fang, Liang He, Max Willsey and Yuxuan Mei

National Science Foundation: 1813166 and 1644558

https://grail.cs.washington.edu/projects/carpentrycompiler/

Chenming Wu

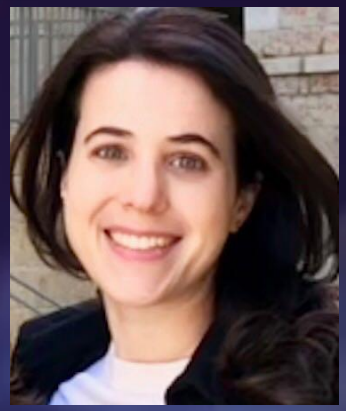
Haisen Zhao

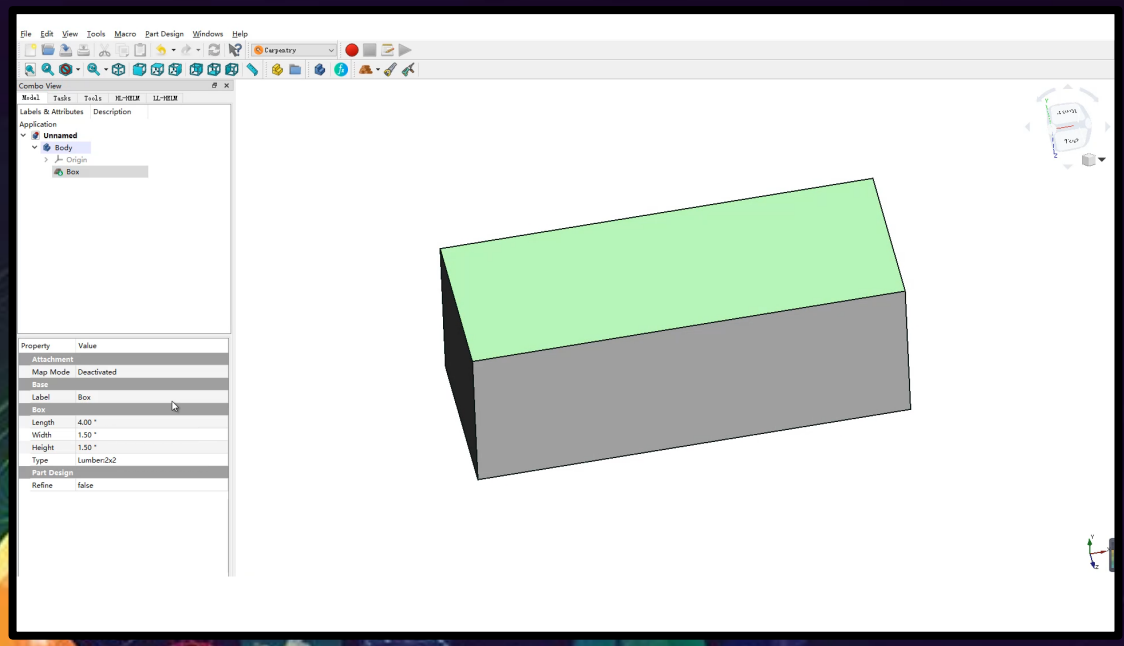
Chandra Nandi

Jeffrey Lipton

Zachary Tatlock

Adriana Schulz





Hardware Extensible Language for Manufacturing

1. Key insight: fabrication plans are programs
2. New DSLs for HL-HELM and LL-HELM
3. Applied and extended compiler technique
4. Multi-objective optimization

Future work:

1. Carpentry Extensions
2. Other processes
3. Design Optimization
4. Scheduling for future workshops