Supplementary Material for: Co-Optimization of Design and Fabrication Plans for Carpentry

HAISEN ZHAO, University of Washington and Shandong University and IST Austria
MAX WILLSEY, AMY ZHU, CHANDRAKANA NANDI, and ZACHARY TATLOCK, University of Washington
JUSTIN SOLOMON, Massachusetts Institute of Technology
ADRIANA SCHULZ, University of Washington

1 TECHNIQUE DETAILS

This section provides additional details of our ICEE algorithm for extracting a Pareto front where each solution s represents a (design, fabrication) pair. We also include pseudocode (Algorithm 1) of our ICEE algorithm and a table (Table S1) with all parameters used in the algorithm.

1.1 Generating Design Variants

Design variants can be generated manually by a user or automatically. In our article, we implement an automatic method, which includes three steps: (1) detecting all the assembly connectors between two neighboring parts; (2) enumerating all candidate connector variants for each connect; and (3) generating design variants by selecting different connector variations of each connector, as shown in Figure 4 of the main article.

1.2 Fabrication Arrangement Generation

Given a specific design variation, we use a fabrication arrangement generation algorithm to update the BOP E-graph such that the E-graph encodes more fabrication arrangements. This algorithm uses two heuristics described in Section 4.3.3.

For an input design variation d_i, which consists of parts $p_j, 0 \ldots n$, this algorithm first groups a library of stock lumbers by their dimensions, e.g., lumber $2'' \times 4'' \times 24''$, lumber $2'' \times 4'' \times 48''$, and lumber $2'' \times 4'' \times 96''$ are grouped together. For the parts assigned to this group, this algorithm starts by packing all the parts on the largest stock lumber of the current group. The packing process will be terminated when (1) the current stock lumber is maximally packed or (2) all parts are packed. In the first case, the packing process can continue by switching to another stock lumber (also the largest one) until the second termination condition is reached. Such a layout process is called a full *Traversal*, which traverses all of the parts in a specific order. Prior work [Wu et al. 2019] has used a similar approach but used the number of *Traversals* as the termination criterion, which prevents them to control the number of fabrication arrangements to generate with this heuristic-driven method.

1.3 Cost Metrics

This section describes how we compute the three costs: material usage (f_c), cutting precision (f_p), and fabrication time (f_t). Our formulae are updated versions of the ones used by Wu et al. [2019]. Our key improvements are (1) we include a quantitative evaluation method. Each cost metric is associated with a meaningful unit: material usage (f_c) is in dollars, cutting precision (f_p) is in inches, and fabrication time (f_t) is in minutes; (2) we evaluate stock load and unload time to make the result fabrication time much more reasonable.

Material Cost. We compute material cost as

$$f_c = \sum_{i=1}^{n} p_i,$$

where p_i, the price of the ith piece of stock is estimated based on costs from standard US vendors [McMASTER-CARR 2021] as shown in Table S2, n is the total number of pieces of lumber used.

Time. We asked an expert carpenter to assign a fabrication time to each tool (Table S3) and estimate a stock loading and unloading time for each piece of wood stock (Table S5). They reported the time taken for (1) full setup of the tool, (2) partial setup when applicable (a partial setup in one where only some parameters are changed), (3) full stock load (unload) time (setting up a piece of wood stock to the tool workspace) (4) partial stock load (unload) time (setting up a piece of wood stock to the tool workspace where stocks are stacked), and (5) performing a single operation (e.g., cut). Out of all the used tools, tracksaw has the most elaborate setup process, which is indicated by its long setup times (both full and partial). Bandsaw and jigsaw are set to different full setup times while cutting lumber and plywood sheet. Only chopsaw and tracksaw allow partial setups, the remaining tools do not. The operation time for a chopsaw is constant — all cuts take a second. For all other tools, the operation time is based on the length cut per second. For example, the operation time for a tracksaw cut of length l'' is $l/4.5$ seconds.

We refer to a cut as "partial" if it requires only a partial setup. For example, if the ith cut is a partial cut on a chopsaw, the time taken for this cut would be $15s$ for partial setup and $1s$ for the cutting operation leading to a total of $16s$ for the cut to be completed. The fabrication time, f_t is therefore computed as

$$f_t = \sum_{i=1}^{k} (s_i + w_i + o_i),$$

where s_i, w_i, and o_i are the setup time, stock load and unload time, and operation time for the ith cut, respectively, and k is the total number of cuts. s_i and o_i are computed based on Table S3. w_i is computed based on Table S5. For a cut with stacking, w_i is measured as the sum of a full load and unload time for the first piece

0730-0301/2022/03-ART32 $15.00
https://doi.org/10.1145/3508499

ALGORITHM 1: Iterative Contraction and Extension on E-graphs

Input: a carpentry design d;
  a discrete set $\mathcal{D}$ of possible design variations;
Output: a Pareto front of (design, fabrication) pairs
Runtime t; Timeout $T \leftarrow 4$ hours;
A solution set S to store all solutions s during optimization;
Pareto front PF of current solution set S; Hypervolume HV of PF;
Number of iterations $n_d \leftarrow 0$, $n_p \leftarrow 0$, $I_d \leftarrow 0$, $I_p \leftarrow 0$;
Variables HV_{old} and $\overline{HV_{old}}$ save hypervolumes from the last iteration;
BOP Egraph Initialization
  Randomly select up to 10^5 designs from $\mathcal{D}$ (Sec 4.3.2);
  Select the top K_d designs from this set (Sec 4.3.2);
  Generate K_f arrangements for each design (Sec 4.3.3);
  Initialize the BOP Egraph $\mathcal{E}$ (Sec 4.2);
while $n_d < t_d$ and $I_d < mt_d$ **do**
  $HV_{old} \leftarrow HV$
  Pareto Front Extract (Sec 4.3.4)
    for an atomic e-node e in $\mathcal{E}$ **do**
      if e has not been optimized **do**
        Try a maximum, P, different orders of cuts;
        Evaluate each order with the cost metrics;
        Select the order with minimum f_p for e;
        Select the order with minimum f_t for e;
    // Use a genetic algorithm to extract Pareto front from $\mathcal{E}$
    $n_p \leftarrow 0$, $I_p \leftarrow 0$;
    while $n_p < t_p$ and $I_p < mt_p$ **do**
      $\overline{HV_{old}} \leftarrow HV$;
      Randomly select N_{pop} terms ($\mathcal{T}$) from $\mathcal{E}$;
      Define the lower and upper bound of f_p and f_t for each $\mathcal{T}$;
      if a $\mathcal{T}$'s lower bound is not dominated by PF **do**
        Run optimization with upper bound as start points;
        Randomly flip the order of some cuts until 20 iterations;
      Add current terms ($\mathcal{T}$) to S and update PF and HV;
      Update terms ($\mathcal{T}$) with the crossover and mutation operations;
      $n_p \leftarrow \overline{HV_{old}} == HV \ ? \ n_p + 1 : 0$;
      $I_p \leftarrow I_p + 1$;
  if $t < T$ **do**
    BOP Egraph Contraction (Sec 4.3.5)
      Update I_{score}, E_{score} and P_{score} for each e-class in $\mathcal{E}$;
      Prune an e-class if its P_{score} is smaller than P_{rate};
    BOP Egraph Expansion (Sec 4.3.6)
      // Generate new designs using a single step genetic algorithm
      Generate $K_m \cdot K_d$ design variations;
      Select the top K_{nd} design variations;
      Include the result K_{nd} design variations in $\mathcal{E}$;
      Generate K_f arrangements for each new design (Sec 4.3.3);
      // Generate arrangements for existing design variations
      Select top K_d root e-classes to expand based on I_{score};
      Generate an adaptive number of arrangements for each e-class;
      Include the result arrangements to $\mathcal{E}$;
  $n_d \leftarrow HV_{old} == HV \ ? \ n_d + 1 : 0$;
  $I_d \leftarrow I_d + 1$; Update t;

of stock, and multiple partial loads and unload times for the following stocks.

Precision. To measure the precision of a fabrication plan, we compute (1) measurement error, ϵ, and (2) operation error, p. We use $m = 1/16''$ as the minimum measurement that can be made in any of the currently supported tools. The measurement error for length, m' is computed as

$$\epsilon = min(m'\%m, m - (m'\%m)),$$

which, intuitively, is the residual length that cannot be measured by our tools. For operation error, we again asked an expert to estimate the error per operation for each tool (Table S4). The error for the i^{th} cut is, therefore, $\epsilon_i + p_i$. f_p is then computed as

$$f_p = \sum_{i=1}^{k} (\epsilon_i + p_i),$$

Table S1. This Table Lists All Parameters Used in Our ICEE Algorithm

Param	Significance	Usage	Value
α	tuning parameter to trade-off between exploring designs and fabrications	N/A	0.75
β	tuning parameter defined based on α	N/A	$\lfloor 44 \cdot \alpha^7 + 2 \rfloor$
t_d	terminal condition of ICEE, number of iterations without hypervolume improvement	4.3.1	10
mt_d	terminal condition of ICEE, maximum number of iterations	4.3.1	200
T	terminal condition of ICEE, timeout	4.3.1	4 hours
K_d	number of initialized design variations	4.3.2	$2^{\lceil \log_{10} \lvert\mathcal{D}\rvert \rceil}$
K_f	number of fabrication arrangements of each initialized design variation	4.3.3	$\beta \cdot n_p$
N_{pop}	population size in the Pareto front extraction step	4.3.4	$4 \cdot K_d$
P	maximum different orders of cuts of [Wu et al. 2019] method in the Pareto front extraction step	4.3.4	$2 \cdot (\beta - 2)$
t_p	terminal condition of [Wu et al. 2019] method in the Pareto front extraction step, number of iterations without hypervolume improvement	4.3.4	20
mt_p	terminal condition of [Wu et al. 2019] method in the Pareto front extraction step, maximum number of iterations	4.3.4	200
mc_p	crossover probability of [Wu et al. 2019] method in the Pareto front extraction step	4.3.4	0.95
mm_p	mutation probability of [Wu et al. 2019] method in the Pareto front extraction step	4.3.4	0.80
I_{score}	impact of an e-class, measured based on how often it is used in the set of solutions in the current Pareto front	4.3.5	N/A
E_{score}	measure how much an e-class has been explored	4.3.5	N/A
w	the weight to trade-off between exploration and impact	4.3.5	0.70
P_{rate}	If the P_{score} is smaller that the pruning rate, P_{rate}, the e-class is removed along with any e-nodes pointing to this e-class (i.e. parent e-nodes)	4.3.5	0.30
mc_d	crossover probability of the expansion step to generate new design variations using a single genetic algorithm	4.3.6	0.95
mm_d	mutation probability of the expansion step to generate new design variations using a single genetic algorithm	4.3.6	0.80
K_m	get $K_m \cdot K_d$ design variations by applying K_m times of the single step genetic algorithm of the expansion step	4.3.6	10
K_{nd}	selecting the top K_{nd}, $K_{nd} \in [0, k_d]$ of the expansion step.	4.3.6	$\lfloor (1.0 - \alpha) \cdot K_d \rfloor$

Each raw indicates the parameter (**Param**), the meaning of the parameter (**Significance**), the related section where the parameter appears for the first time (**Usage**) and the parameter value (**Value**) in our implementation.

Table S2. Prices of Stocks

Stock	Dimension	Material Cost ($)
$2'' \times 2''$	$24''$	3.0
$2'' \times 2''$	$48''$	5.5
$2'' \times 2''$	$96''$	10.0
$2'' \times 4''$	$24''$	3.0
$2'' \times 4''$	$48''$	5.5
$2'' \times 4''$	$96''$	10.0
$4'' \times 4''$	$24''$	7.5
$4'' \times 4''$	$48''$	13.75
$4'' \times 4''$	$96''$	25.0
$2'' \times 8''$	$24''$	7.5
$2'' \times 8''$	$48''$	13.75
$2'' \times 8''$	$96''$	25.0
$1/2''$	$12'' \times 20''$	5.5
$1/2''$	$24'' \times 20''$	10.0
$1/2''$	$48'' \times 36''$	30.0
$3/4\,''$	$12'' \times 20''$	7.0
$3/4\,''$	$24'' \times 20''$	12.0
$3/4\,''$	$48'' \times 36''$	32.0

where k is the total number of cuts. Lower values of f_p indicate higher precision.

1.3.1 Mixed-material Implementation. By introducing a new material, we must accommodate the cost of using this new material in our metrics. Based on an expert's input, we set the material price of metal to be 20 times that of wood. All of the same tools can be used to cut metal sheets, using a specific blade. Setup time is the same, but the execution time is set to 10 times slower, and stock loading and loading time are five times slower. Finally, the jigsaw's precision error is set to two times that of wood. Modulo the cost difference, we apply the same process to evaluate the fabrication cost.

1.4 Pareto Front Extraction

In Section 4.3.4 of the main article, we propose two methods to speed up the Pareto front extraction. This section provides more details of the two methods. For an atomic e-node that has not been previously optimized, we simply try a maximum, P, different orders of cuts (each cutting order is evaluated using the cost metrics in Section 1.3), then select the cutting order with minimum precision error (f_p) and the one with minimum fabrication time (f_t).

To apply the branch and bound technique, we need to define the lower and upper bound for the precision error (f_p) and the time cost (f_t). A term $\mathcal{T}$'s cutting order can be initialized by taking the cutting order used in its contained atomic e-node. We first select the cutting order with minimized precision of each atomic e-node, then evaluate its precision error defined as the precision upper bound of the term. The time upper bound is set by taking the cutting order with minimized time of each atomic e-node. The precision (time) lower bound is defined as the evaluated precision (time) cost, ignoring the dependency relationship between cuts belonging to the same stock. The dependency relationship indicates the order of a cut with respect to other cuts. In other words, the time lower bound is computed by the assumption that each cut is independent of the other.

If the precision (or time) lower bound is not dominated by the Pareto front of all computed solutions $\mathcal{S}$, we run an optimization that uses the upper bound as a starting point. For the optimization, we randomly flip the order of some cuts until t iterations. We set t as 20 in our experiments.

2 RESULTS AND DISCUSSION

In this section, we provide additional results of our ICEE pipeline for exploring the space of design variations and fabrication plans of Figure 7. Table S6 shows the detailed comparison between our pipeline and the no design exploration pipeline. At the end of this supplemental material, for each model, we demonstrate some representative design variations and fabrication plans extracted with our ICEE pipeline and the baseline method.

2.1 Carpentry Compiler Parameters

During the comparison between our pipeline and the Carpentry Compiler pipeline [Wu et al. 2019], we use the default parameter setting used in their experiments, the number of *Traversal* $T = 50$, the number of top e-nodes $n = 10$, the maximum different orders of cuts $P = 25$, the population size during E-graph extraction as 120, the probability of crossover and mutation $p_c = 0.95, p_m = 0.1$.

2.2 Running Time Analysis

On average, our pipeline spends 4.2% of its runtime in design extraction, 5.8% of its time generating packings, 64.5% of its time assigning cutting orders, and 21.6% of its time in applying genetic algorithms to extract Pareto front of terms. This translates to 10.0% of its runtime in the expansion and contraction phase, 86.1% in the extraction phase, and the other 3.9% everywhere else.

2.3 Comparison with Baseline

As shown in Figure S1, we cannot find exactly the same front due to convergence and stochasticity. In order to make this comparison, we have tuned the parameters to be as close as possible. This means we do not find exactly the same hypervolume indicated in Table S7. However, hypervolume is unintuitive to compare, so we believe presenting plots for comparison is far more compelling.

2.4 Convergence

Due to the combinatorial nature of the problem, we cannot guarantee that we discover the true Pareto front. In what follows we discuss how this limitation can affect the results of our evaluation.

2.4.1 Increasing the Design Space. The intractable search implies that we have a finite amount of resources to spend searching the space. Realistically, this means we can only explore a subset of the design space. Consider the design of the "Window" in Figure 7. If we permit this model to have angled joints instead of just 90 degree joints, the space of design variants becomes much larger. Ideally, the result for this new window variant should be at least as optimal as the result for the original window. However, our approach was able to explore only a portion of the vast design space ($|\mathcal{D}| = 1520052$, compared to $|\mathcal{D}| = 10463$ for the simpler window model) and found a worse landscape of solutions (Figure S3). For models with a large design space, if the initial

Table S3. Fabrication Times for Different Tools

Tool	Full setup (s)		Partial setup (s)	Op Time
	Lumber	Plywood		
Chopsaw	60	60	15	1 s
Bandsaw	20	90	N/A	1 inch/s
Jigsaw	30	60	N/A	1 inch/s
Tracksaw	180	180	75	4.5 inch/s
Drill	20	20	N/A	0.1 inch (depth)/s

All times are converted to minutes in our results.

Table S4. Error Per Cut for Each Tool

Tool	Operation Error
Chopsaw	$1/64^{th}$ of an inch
Bandsaw	$1/16^{th}$ of an inch
Jigsaw	$3/16^{th}$ of an inch
Tracksaw	$1/32^{nd}$ of an inch
Drill	$1/32^{nd}$ of an inch

Table S5. Loading and Unloading Time of Stocks

Stock	F-Load (s)	P-Load (s)	F-Unload (s)	P-Unload (s)
$2'' \times 2'' \times 4''$	10	1	5	1
$2'' \times 2'' \times 48''$	20	2	8	2
$2'' \times 2'' \times 96''$	40	3	15	2
$2'' \times 4'' \times 24''$	10	1	5	1
$2'' \times 4'' \times 48''$	20	2	8	2
$2'' \times 4'' \times 96''$	40	4	15	2
$4'' \times 4'' \times 24''$	15	2	5	1
$4'' \times 4'' \times 48''$	30	4	10	2
$4'' \times 4'' \times 96''$	60	6	20	3
$2'' \times 8'' \times 24''$	15	2	5	1
$2'' \times 8'' \times 48''$	30	4	10	2
$2'' \times 8'' \times 96''$	60	6	20	3
$1/2'' \times 12'' \times 20''$	30	3	10	2
$1/2'' \times 24'' \times 20''$	50	5	15	2
$1/2'' \times 48'' \times 36''$	100	10	20	2
$3/4'' \times 12'' \times 20''$	30	3	10	2
$3/4'' \times 24'' \times 20''$	50	5	15	2
$3/4'' \times 48'' \times 36''$	100	10	20	2

All times are converted to minutes in our results.

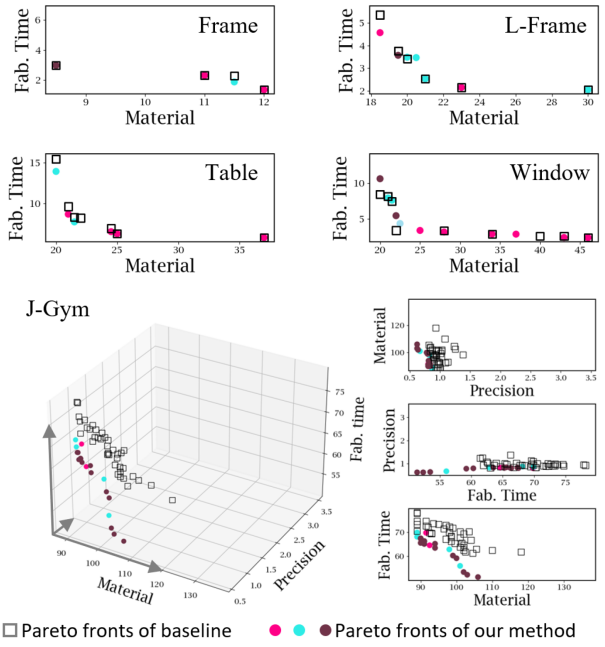

Fig. S1. Pareto fronts generated from our pipeline and baseline method.

design variants picked by our algorithm are too far from the optimal ones, then our algorithm may never reach those optimal points.

To avoid similar potential pitfalls, we might be able to allocate our resources to effectively direct the search. In particular, one thing we can do to navigate this space more precisely is to expose a parameter, alpha.

α is a parameter that roughly defines the tradeoff between breadth (exploring many designs) and depth (exploring variations within a small number of designs). For our experiments, we set α to a default value of 0.75. However, in cases where the initial design is believed to be optimal or close to optimal, the user may increase α to encourage deeper exploration of the initial and similar designs. In cases where our baseline of optimizing the fabrication for a single design outperforms exploring many designs in the default configuration, we have found that increasing α to 0.95 makes the performance at least as good again.

This note about parameter tuning reveals interesting implications. In particular, our new algorithm does not always fully dominate fabrication-plan-only exploration with default parameters. There are models where the input design is already fairly optimized and does not offer many opportunity for improvement. Take a model that is simple or is composed of many instances of the same shape. The Adirondack chair is such an example. It would be clear that an optimal packing arrangement would minimize material, cuts, and error by stacking cuts; design variations would only deviate from the simplicity of the fabrication plan and there are no improvements possible. The baseline method has the advantage with these models, because it spends its search time deeply exploring fabrication plans, while ICEE must also spend time searching across design variations. Tuning the parameter alpha can help us perform as well as the baseline method, as seen in Figure ?? (a), (b), and (c).

From these examples, we see that comparing approaches while using parameter tuning is trickier and subtler.

2.4.2 Comparisons with [Wu et al. 2019]. Notably, comparisons with the approaches not employing design exploration are also susceptible to the imprecision in convergence. In previous sections, we ran each tool with its default parameters. Tuning the parameters for each approach would change the results and make the difference in Pareto fronts difficult to assess. Neither approach discovers the true Pareto front, so an absolute comparison depends on the input parameters. Fortunately, we avoid the brunt of this pitfall: The selling point of our design space exploration approach

Table S6. Performance Comparison between Our Method and the Baseline Method (No Design Exploration Pipeline)

Model	HV_{base}	HV_{our}	Ref. Point	$M_{base}(\$)$	$P_{base}('')$	$T_{base}(m)$	$M_{our}(\$)$	$P_{our}('')$	$T_{our}(m)$
Frame	881570	**902017**	(100.0,100.0)	10.00	N/A	1.98	**8.50**	N/A	**1.37**
L-Frame	796637	**797655**	(100.0,100.0)	18.50	N/A	2.15	18.50	N/A	**2.05**
A-bookcase	698414	**702256**	(100.0,100.0,100.0)	23.00	0.16	8.82	23.00	0.13	**8.53**
S-Chair	683887	**698016**	(100.0,100.0)	25.50	N/A	7.92	25.50	N/A	**5.92**
Table	833919	**849705**	(100.0,100.0)	20.00	N/A	7.18	20.00	N/A	**5.78**
F-Cube	817648	**825935**	(100.0,100.0)	15.50	N/A	3.08	15.50	N/A	**2.12**
Window	754863	**777711**	(100.0,100.0)	20.00	N/A	4.93	20.00	N/A	**2.38**
Bench	355824	**369260**	(100.0,100.0)	55.50	N/A	17.03	**53.00**	N/A	20.38
A-Chair	596346	578014	(100.0,100.0)	35.50	N/A	6.25	35.50	N/A	9.52
F-Pot	24155184	**24264553**	(300.0,300.0,300.0)	13.00	0.29	19.13	13.00	0.26	**17.91**
Z-Table	19716012	**20386788**	(300.0,300.0,300.0)	55.50	0.35	30.69	**53.00**	0.28	**24.28**
Loom	20469248	**21040283**	(300.0,300.0,300.0)	27.50	1.47	48.26	**25.50**	0.58	**43.86**
J-Gym	13925991	**15657249**	(300.0,300.0,300.0)	96.00	1.82	72.02	89.00	0.62	**51.38**
D-Chair	543771	539241	(100.0,100.0)	35.50	N/A	15.00	35.50	N/A	16.20
Bookcase	20266354	**21957242**	(300.0,300.0,300.0)	40.00	0.86	37.53	**30.00**	0.27	**24.71**
Dresser	21237296	**22866170**	(300.0,300.0,300.0)	30.00	0.61	36.48	30.00	0.14	**15.42**

In this table, we first report the hypervolume value of our method (HV_{our}) and the baseline method (HV_{base}). The reference points are also listed used for the hyper-volume computation. We also report the minimal material usage, the minimal cutting precision and the minimal fabrication time of the Pareto fronts of baseline method (M_{base}, P_{base}, T_{base}) and the Pareto fronts of our pipeline (M_{our}, P_{our}, T_{our}). The cutting precision of some models is not reported for that precision cost has not been taken into the objective metrics. We use bold font to indicate the hypervolume and these fabrication costs where we are better than the baseline method.

Table S7. Hypervolume Result of the Performance Validation Experiment

Model	HV_{base}	HV_{our}	Ref. Point
Frame	901997	902017	(100.0,100.0)
J-Gym	15036794	15657249	(300.0,300.0,300.0)
L-Frame	797573	797655	(100.0,100.0)
Table	751206	849705	(100.0,100.0)
Window	778158	777711	(100.0,100.0)

In this table, we report the hypervolume of the Pareto fronts computed from our method (HV_{our}) and the "baseline" method (HV_{base}) . "Baseline" indicates extracting the Pareto front fabrication plans fro each design variation explored by our method independently with the Carpentry Compiler pipeline [Wu et al. 2019]. The reference points used for hypervolume computation are also reported.

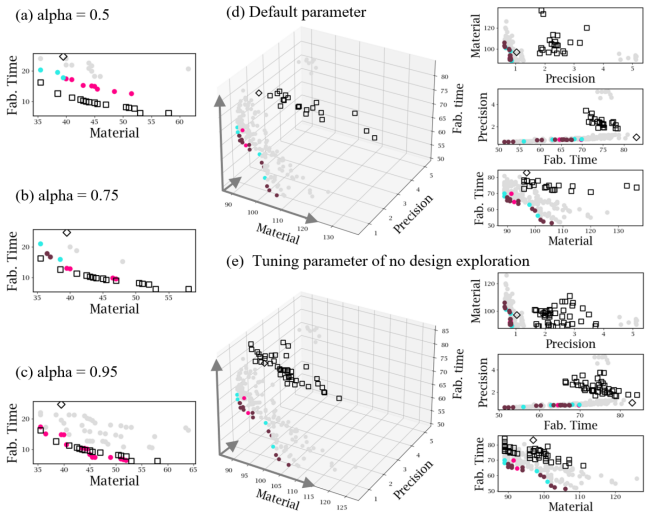

Fig. S2. Pareto fronts of the A-Chair model with different alpha, (a) shows the result of a small alpha (0.5), (b) shows the result of the default alpha (0.75), (c) shows the result of a bigger alpha (0.95). The Pareto fronts generated from no design exploration pipeline stay the same in (a, b, c). Pareto fronts of the J-Gym model with tuning parameters of the no design exploration pipeline, (d) show results of the default parameters, and (e) show the Pareto fronts with tuning parameters of the no design exploration pipeline. The Pareto fronts generated from our pipeline stay the same in (d, e).

is not about eking out marginal improvements over the baseline, but about exploring a completely different space.

Though we may not search anyone design as thoroughly when using default parameters, our approach is able to do something the baseline cannot: discover optimizations to the input design when they exist. Consider elaborate models where there are many ways to pack and fabricate any design, and due to the complexity and number of parts, the designer has diminishing intuition about how to achieve some desired point in the space of tradeoffs.

An example that illustrates this complexity is the Jungle Gym model. The model uses a mix of 1D and 2D wood and has a number of components. From Figure S2(d) and (e), we see that the initial design was suboptimal, and just by exploring several different design variations, we find a number of plans that save greatly on our three objectives. Even the expert plan is completely dominated by the pareto front, indicating that good designs are not always readily apparent.

Our tool takes much longer to finish running as it explores the design space jointly with the fabrication space. Because this is a synthesis problem, one might ask if allowing the baseline to run for longer would result in better fabrication plans for the initial design. We chose not to attempt a comparison of the two algorithms for two reasons. Although it is possible that allowing the

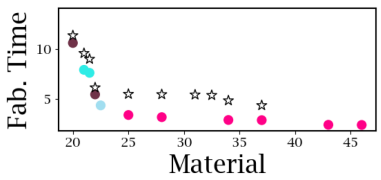

Fig. S3. Example showing a limitation of our approach: The star dots indicate the Pareto fronts with allowing the Window model to have arbitrarily angled connectors; the colored dots indicate ones without arbitrarily angled connectors. Both results are generated with the default parameter of our ICEE pipeline.

fabrication-space-only algorithm to run longer will help it find lower-cost plans, there is no parameter that directly controls how long the algorithm runs. The second reason is that we are supplying a new approach that explores a different solution space, and targets a different problem. Controlling for the running time of the algorithm would still not create an apples-to-apples comparison.

2.5 Scalarization Results

Table S8 shows the scalarization of some of the tradeoffs found on the Pareto front with our method. Note that these results are all for the wood models with our default cost metric. When time is not worth a lot, plans with low material cost dominate. When time is worth more much than materials, low time become cheapest. We observe a variety of fabrication plans being effective at different points.

Table S8. The Percent Improvement of Minimum-cost (After Scalarization) Plans of the Baseline Compared to the Minimum-cost Plans of Design Space Exploration, When Scalarized at Different Prices

Percentage improvement (%)								
Model	Carpenter price ($/hour)							
	0	10	20	40	80	160	240	400
Frame	15	19	21	20	15	10	12	16
L-Frame	0	1	2	2	1	3	1	0
A-bookcase	0	1	2	6	7	6	6	5
S-Chair	0	1	2	2	6	9	11	14
Table	0	1	2	3	4	7	10	12
F-Cube	0	4	5	3	3	5	6	8
Window	0	2	4	12	20	26	28	31
Bench	5	4	4	4	4	5	1	−4
A-Chair	0	−2	−4	−4	−3	−4	−9	−16
F-Pot	0	3	4	5	7	9	9	8
Z-Table	5	5	6	8	11	11	11	12
Loom	7	3	1	0	3	3	4	5
J-Gym	7	7	7	8	11	17	20	23
D-Chair	0	1	1	2	3	2	0	−2
Bookcase	25	16	10	7	8	12	14	18
Dresser	0	2	4	6	8	25	33	42

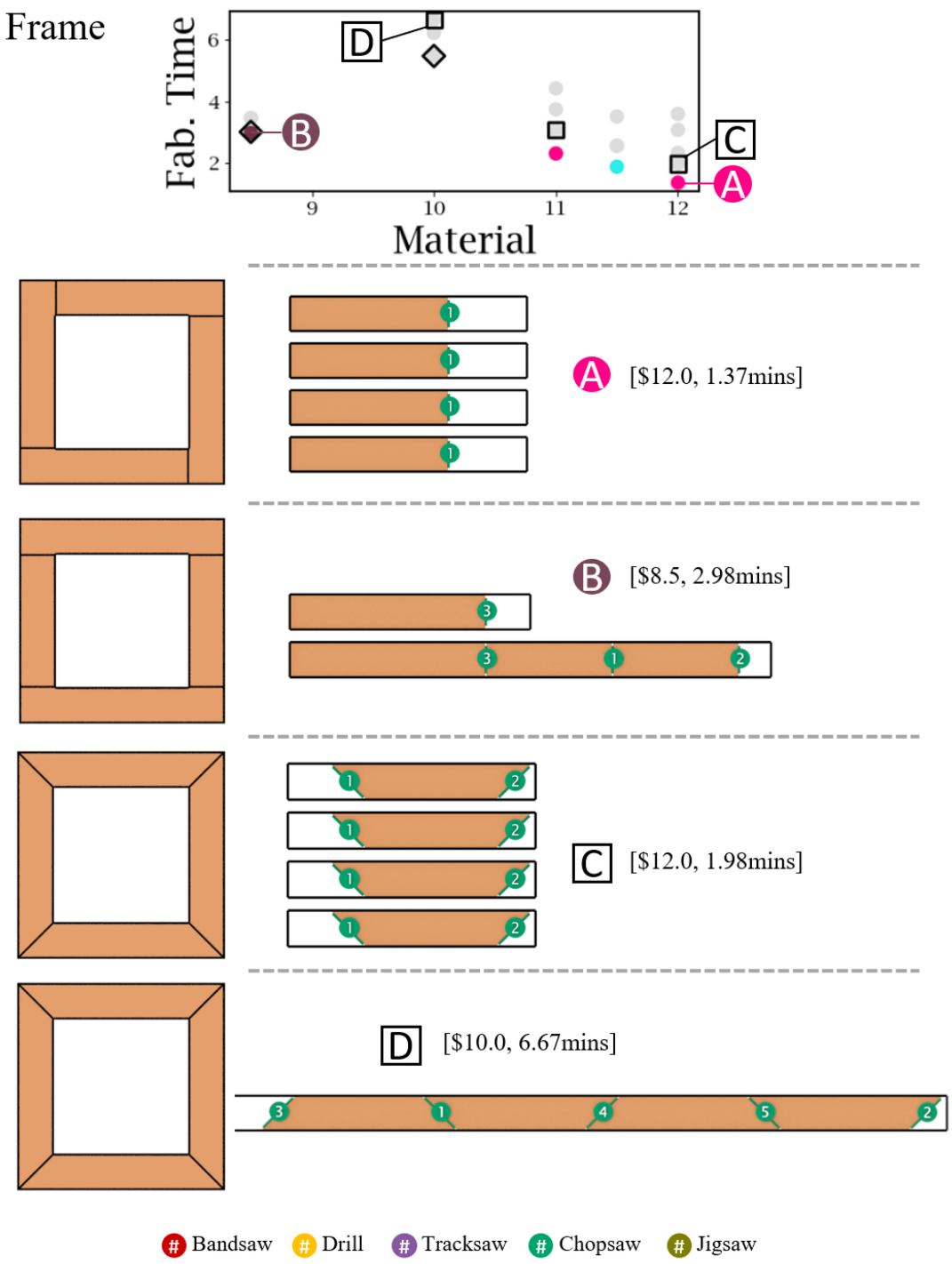

Fig. S4. Design variants and fabrication plans extracted with our ICEE algorithm (Frame).

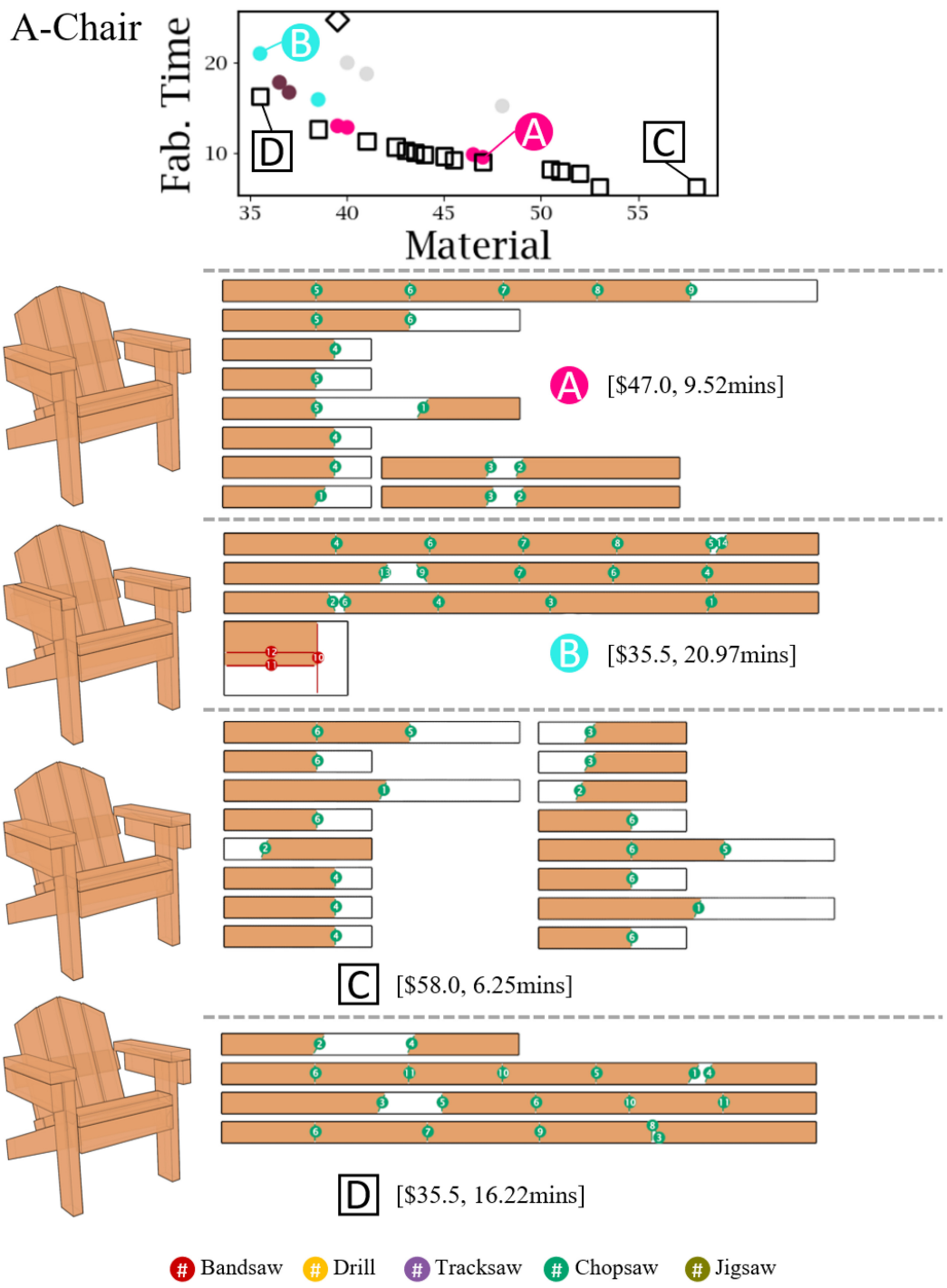

Fig. S5. Design variants and fabrication plans extracted with our ICEE algorithm (A-Chair).

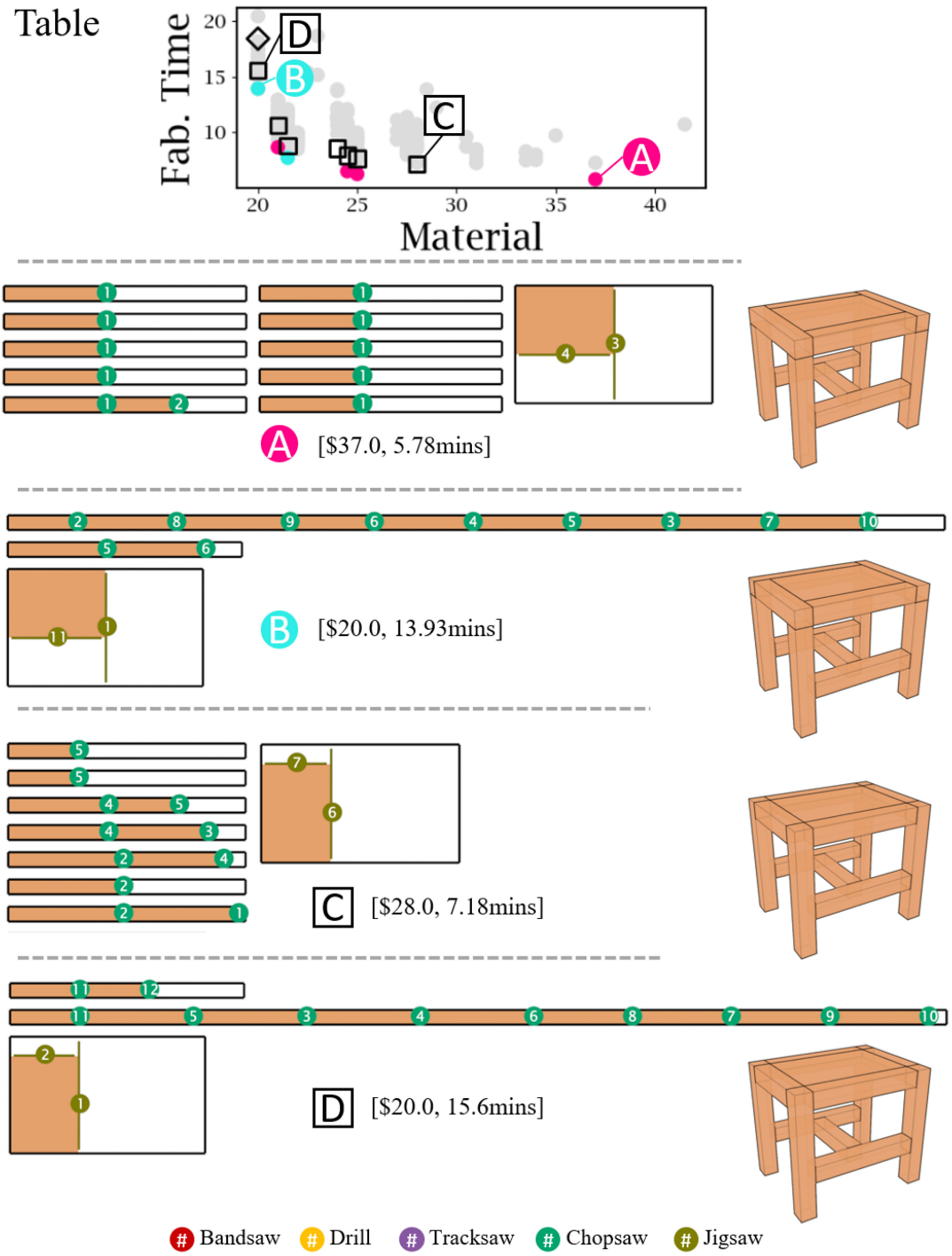

Fig. S6. Design variants and fabrication plans extracted with our ICEE algorithm (Table).

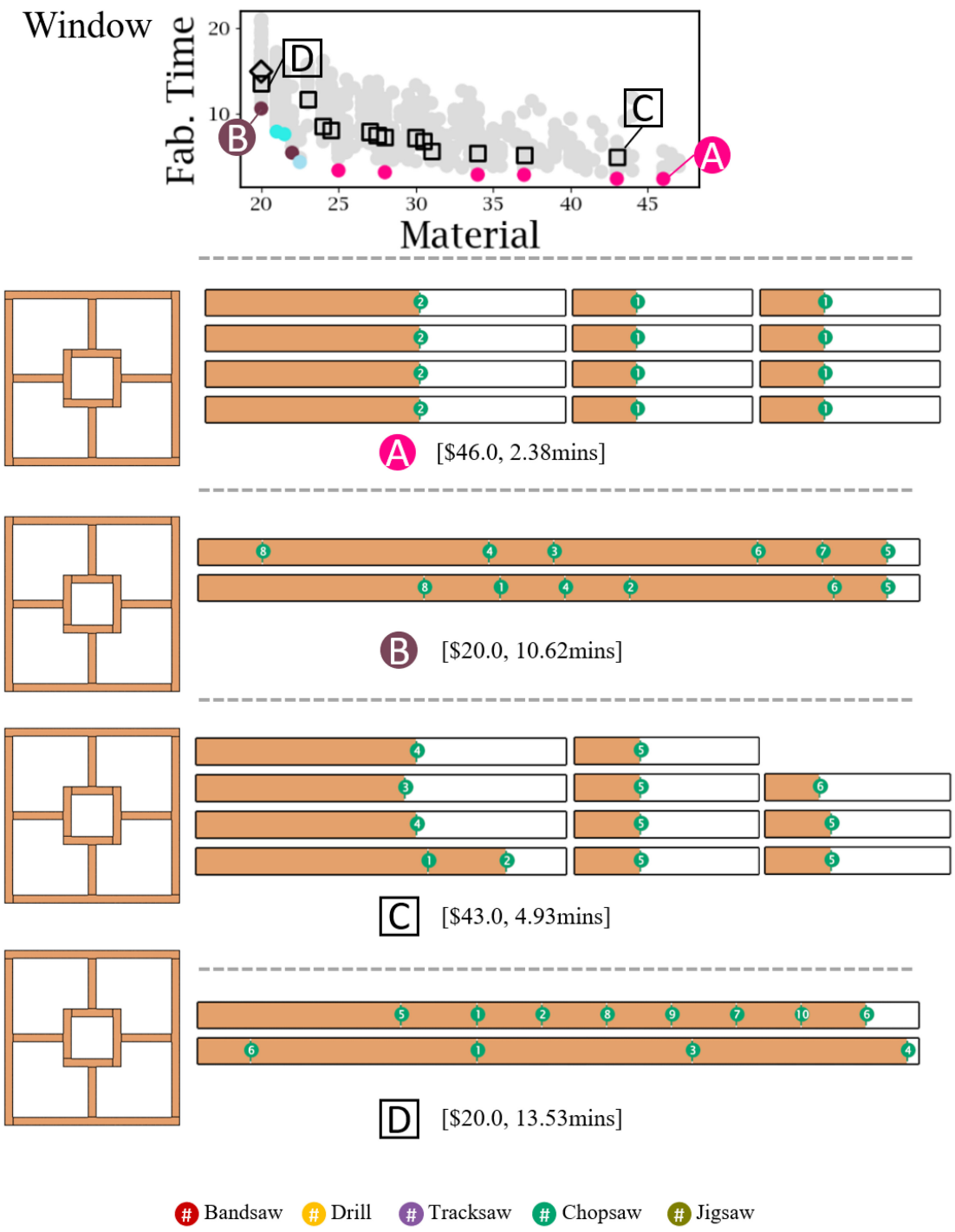

Fig. S7. Design variants and fabrication plans extracted with our ICEE algorithm (Window).

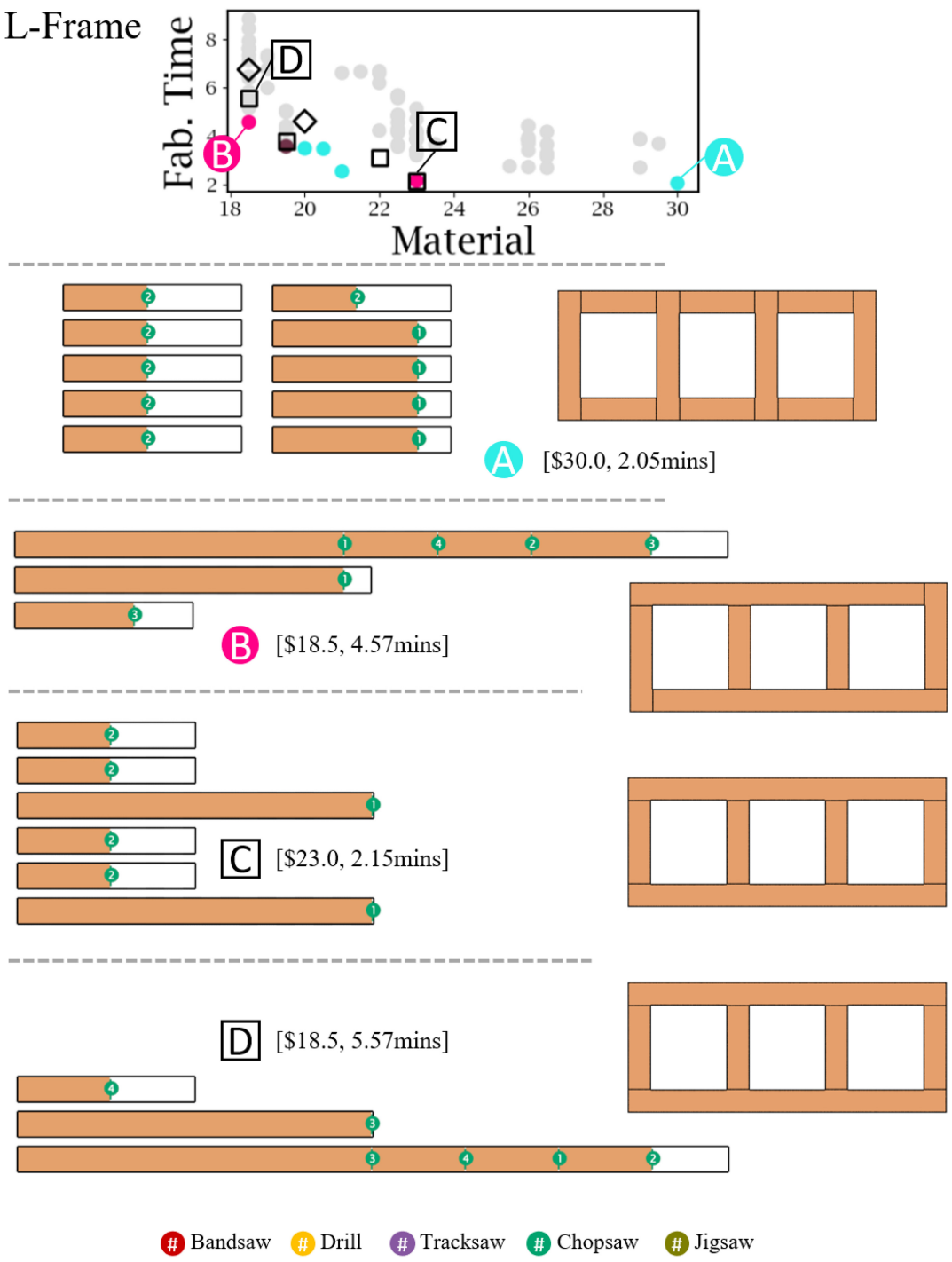

Fig. S8. Design variants and fabrication plans extracted with our ICEE algorithm (L-Frame).

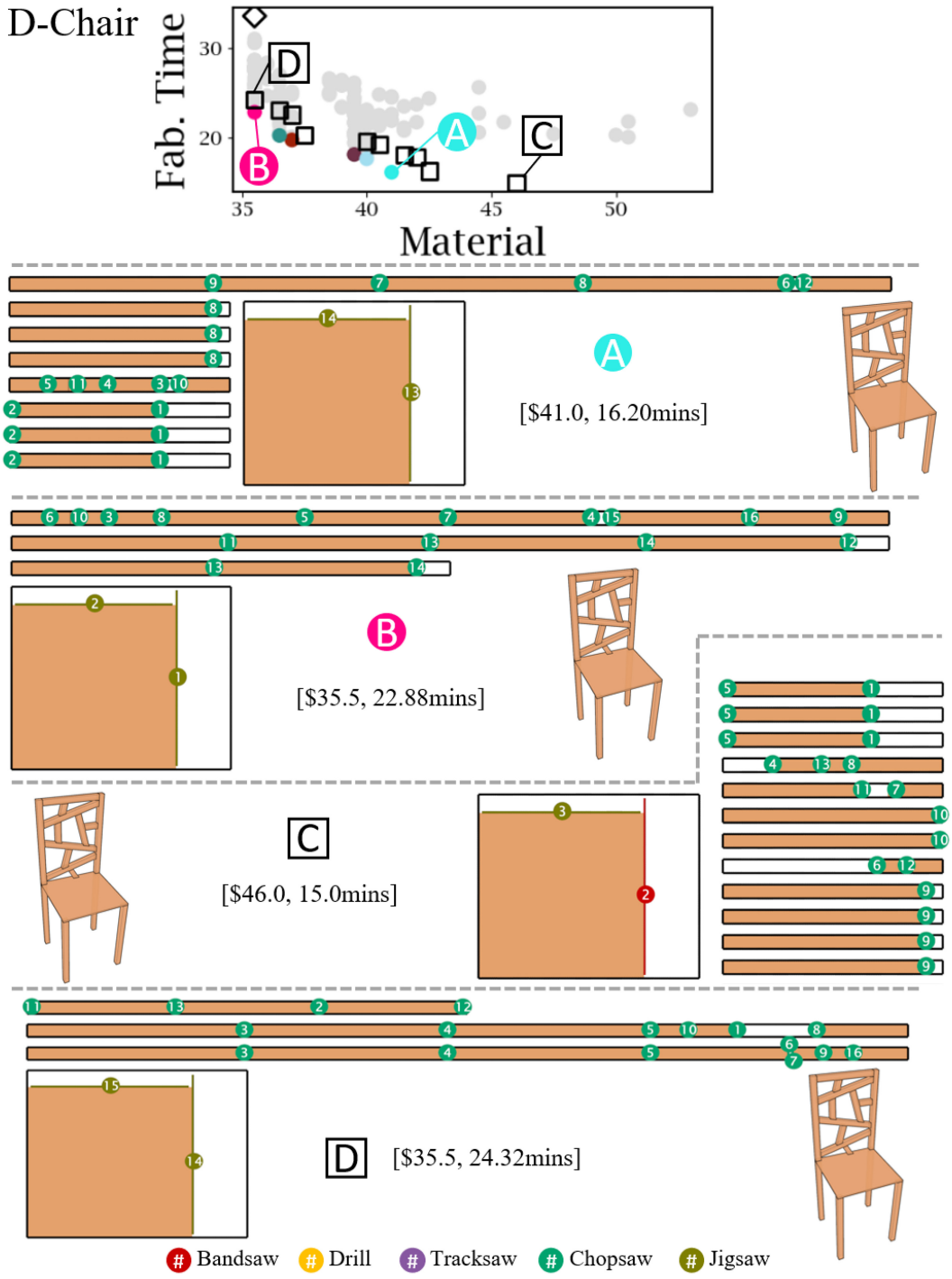

Fig. S9. Design variants and fabrication plans extracted with our ICEE algorithm (D-Chair).

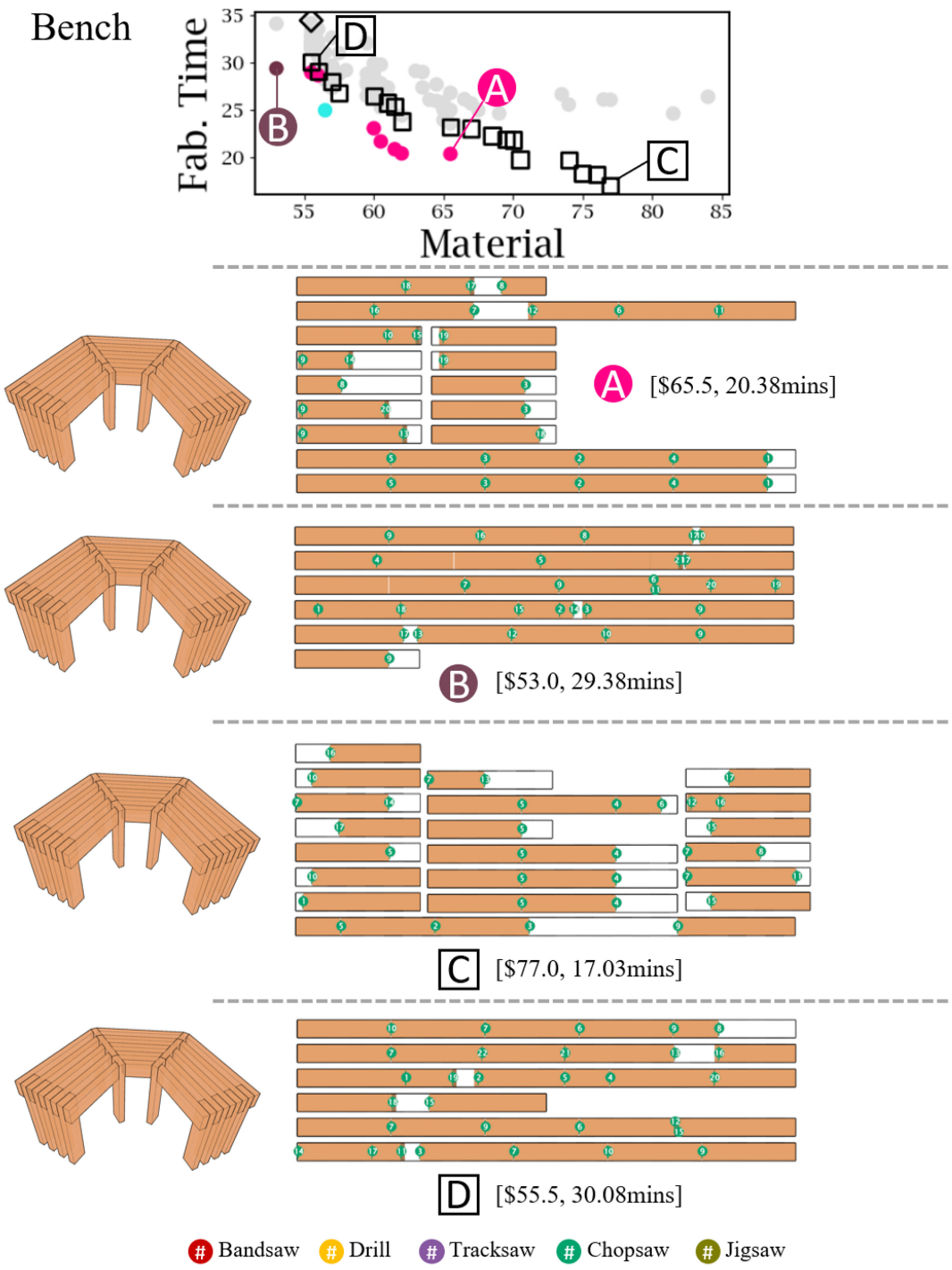

Fig. S10. Design variants and fabrication plans extracted with our ICEE algorithm (Bench).

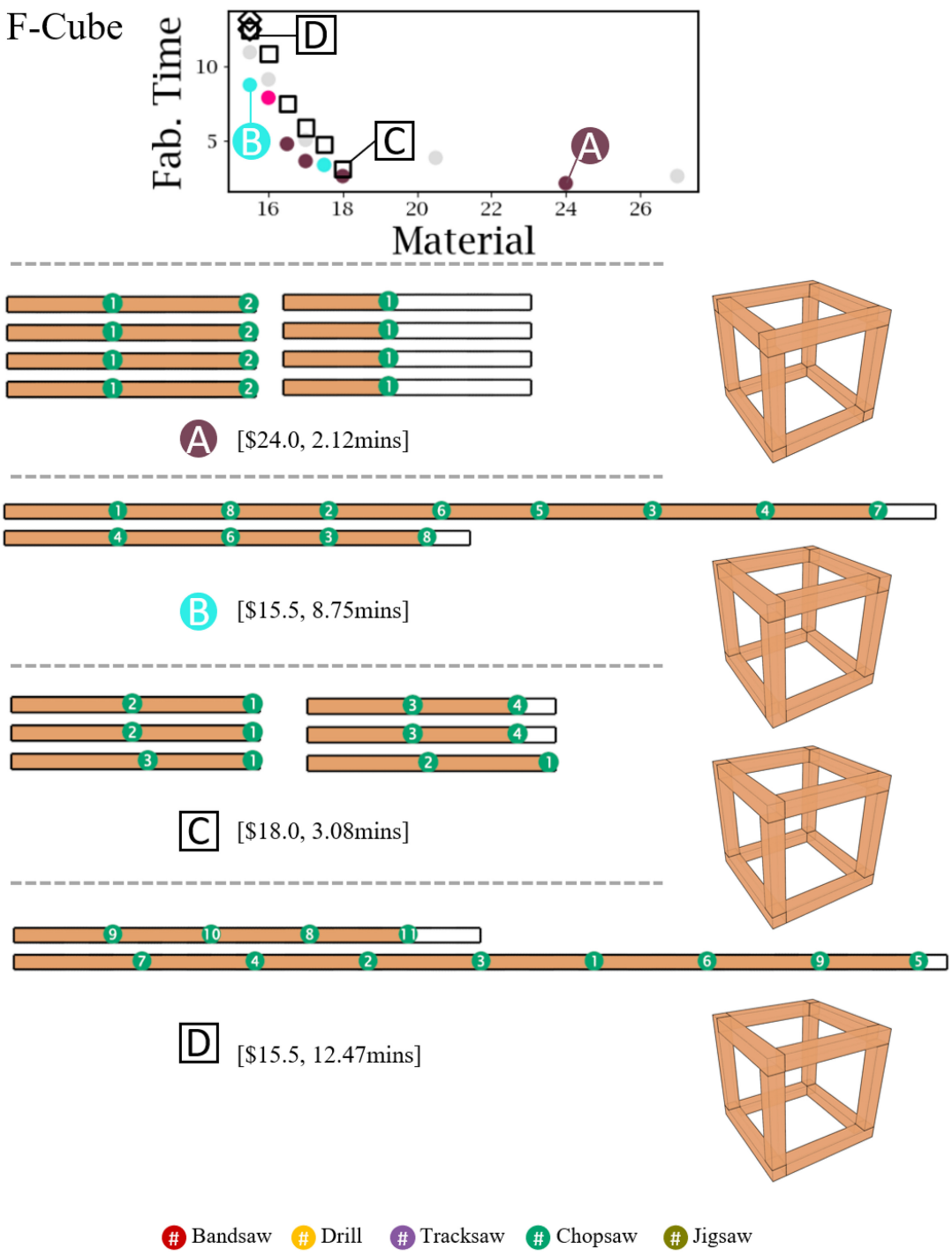

Fig. S11. Design variants and fabrication plans extracted with our ICEE algorithm (F-Cube).

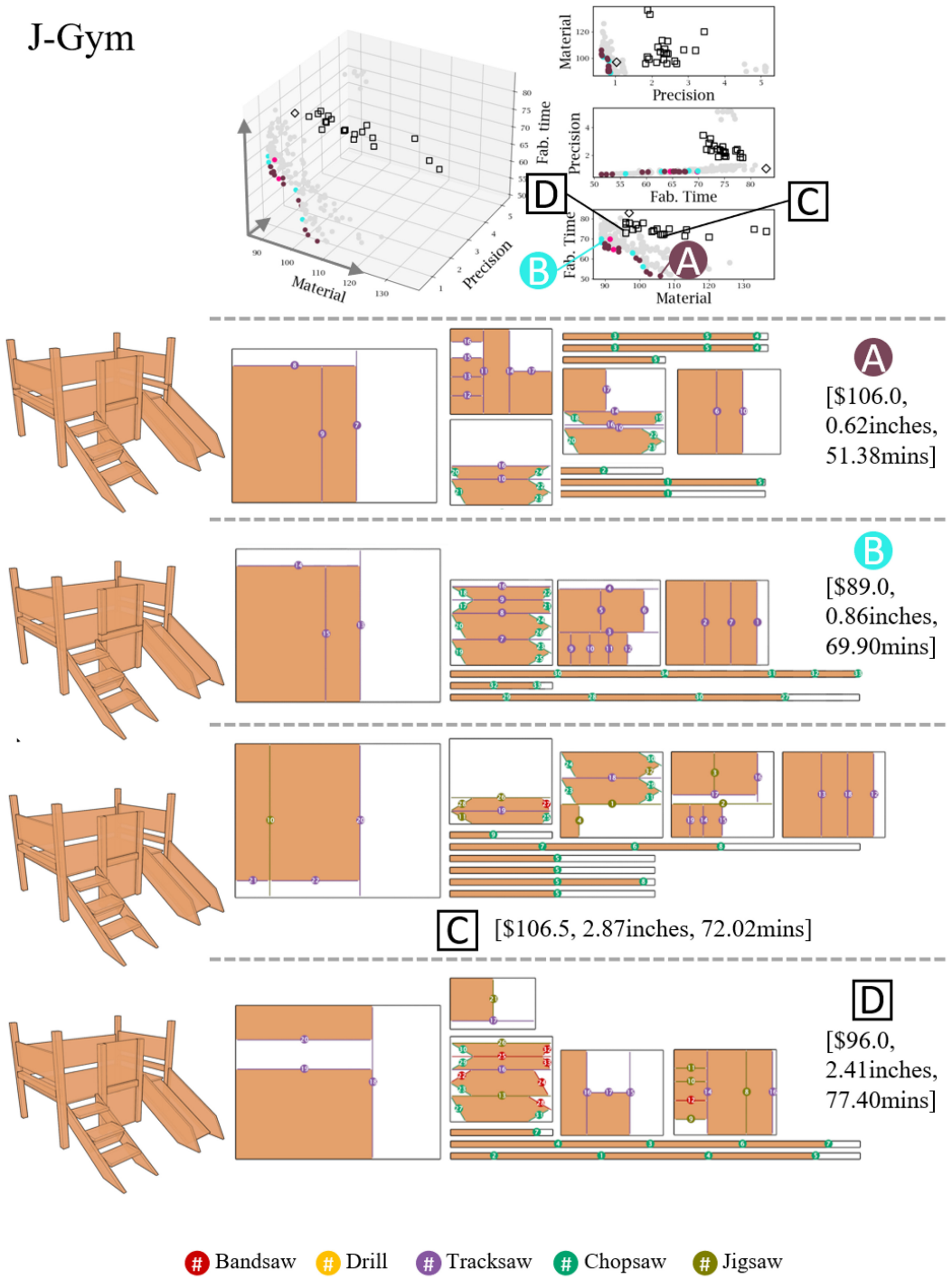

Fig. S12. Design variants and fabrication plans extracted with our ICEE algorithm (J-Gym).

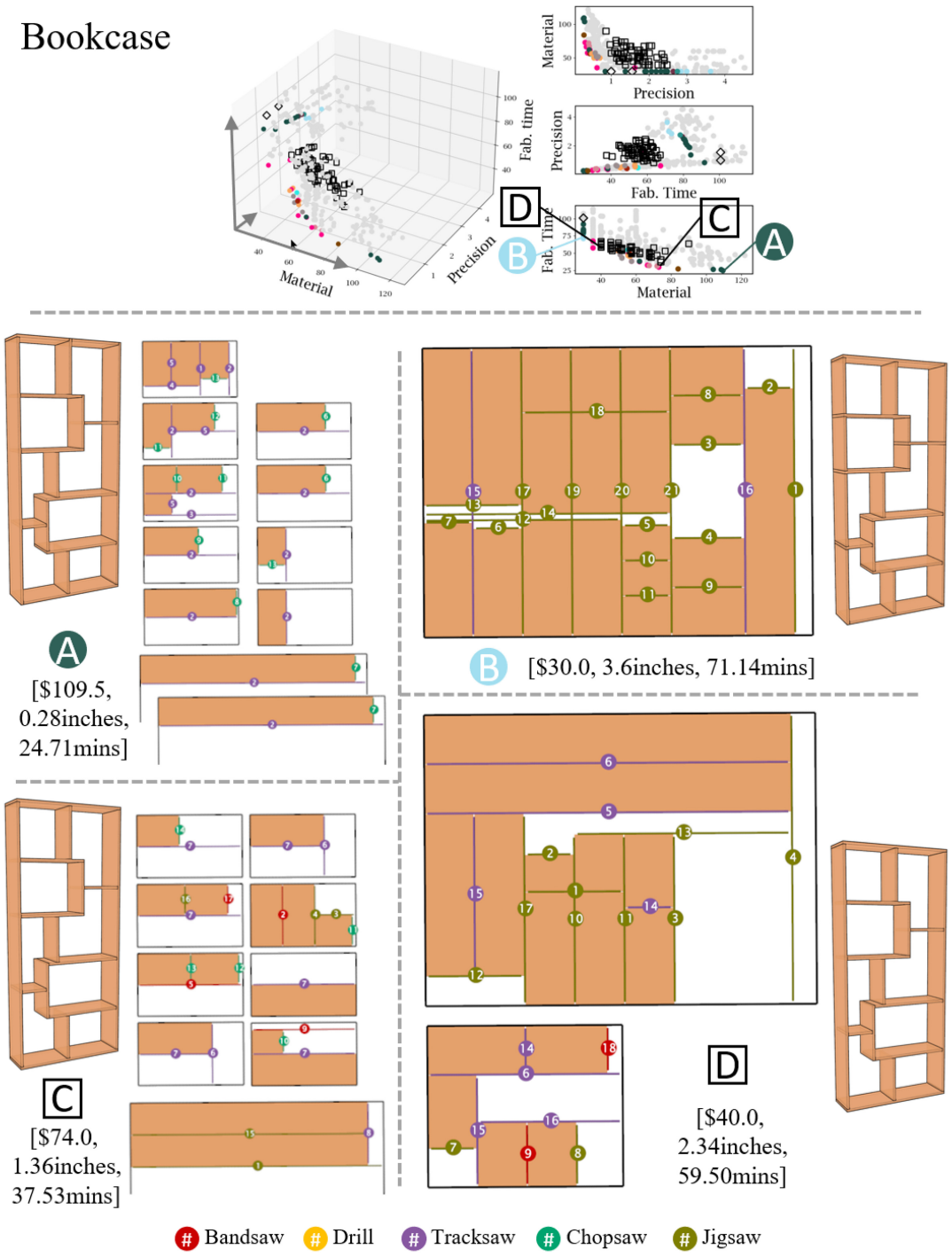

Fig. S13. Design variants and fabrication plans extracted with our ICEE algorithm (Bookcase).

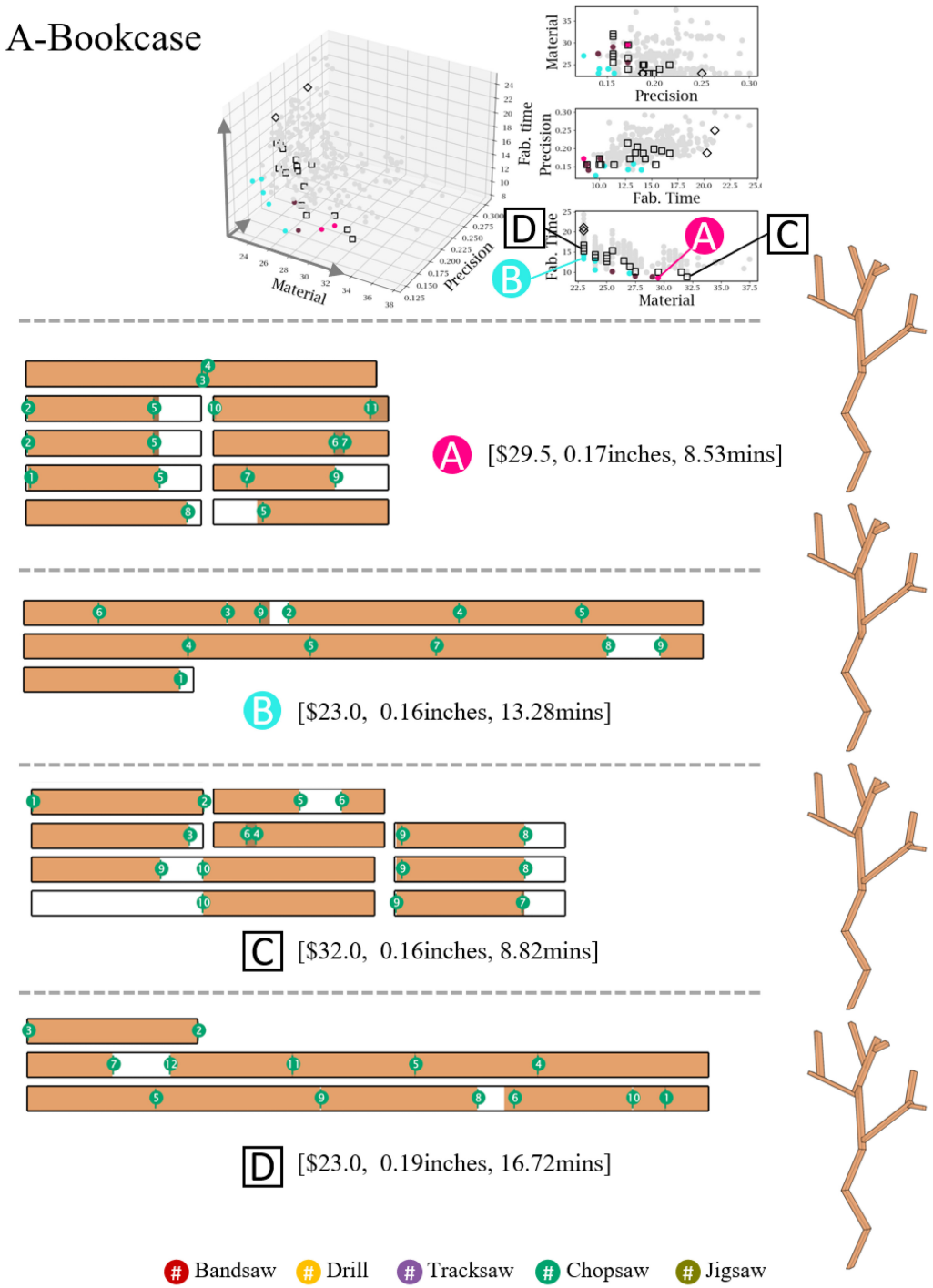

Fig. S14. Design variants and fabrication plans extracted with our ICEE algorithm (A-Bookcase).

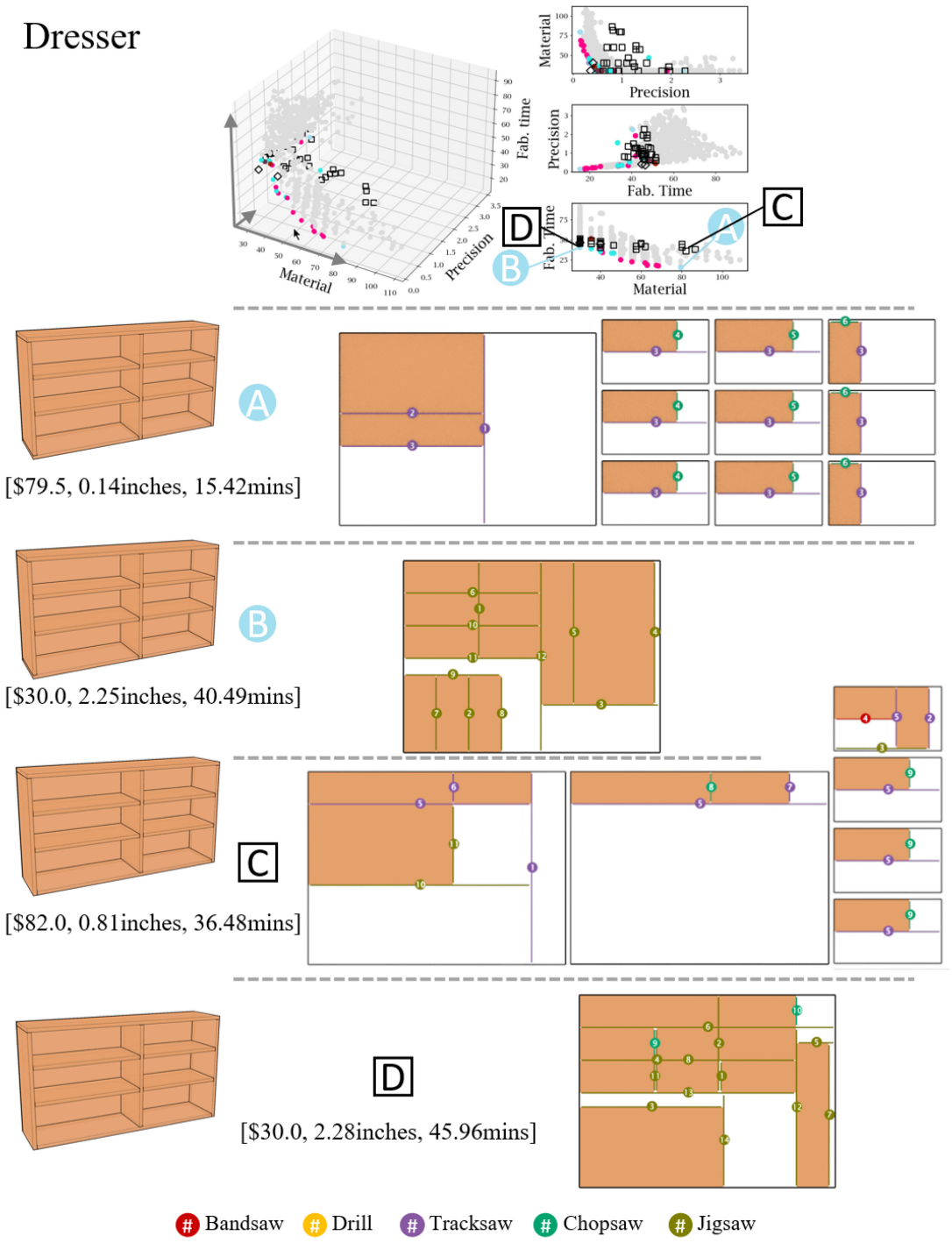

Fig. S15. Design variants and fabrication plans extracted with our ICEE algorithm (Dresser).

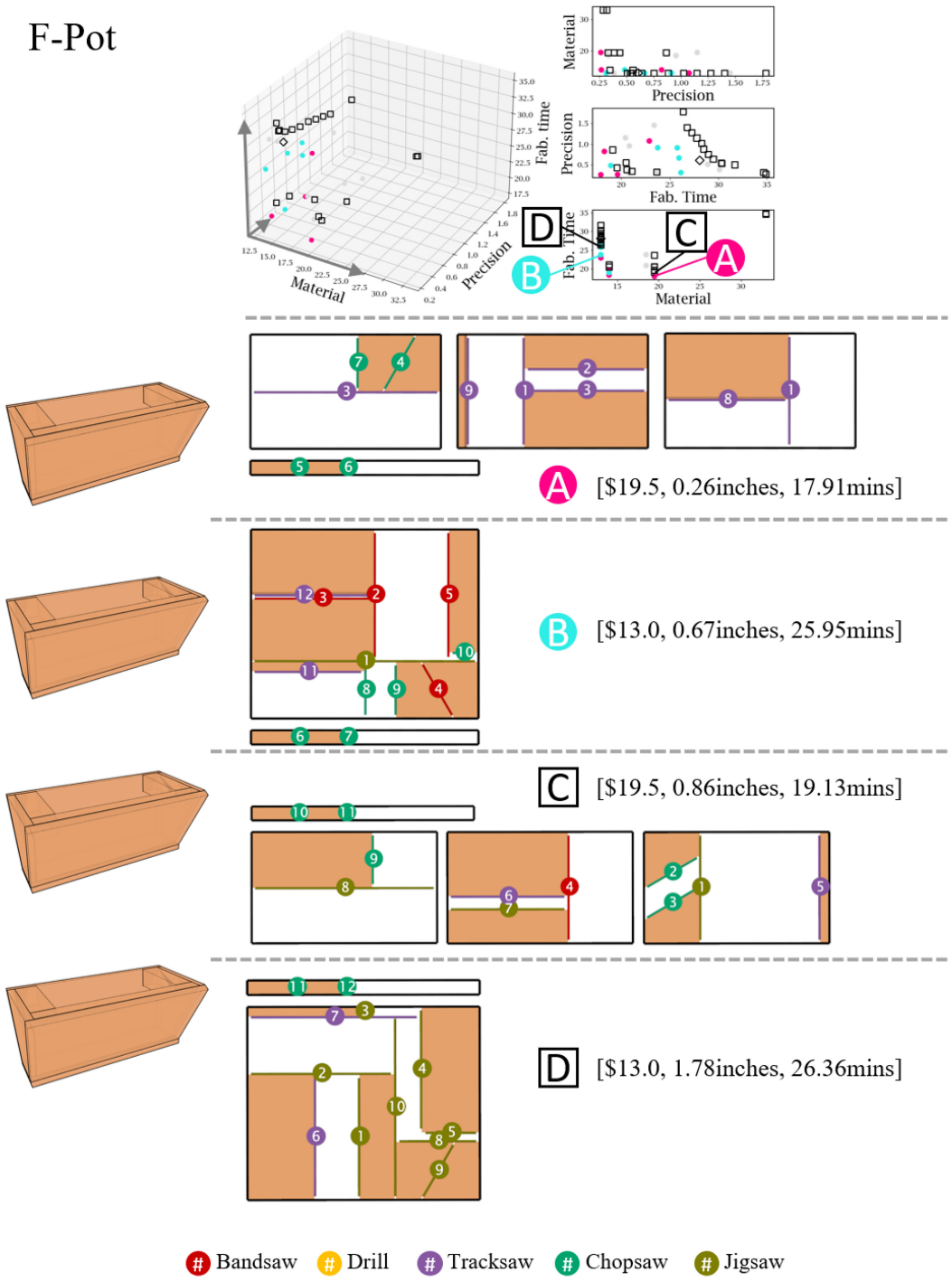

Fig. S16. Design variants and fabrication plans extracted with our ICEE algorithm (F-Pot).

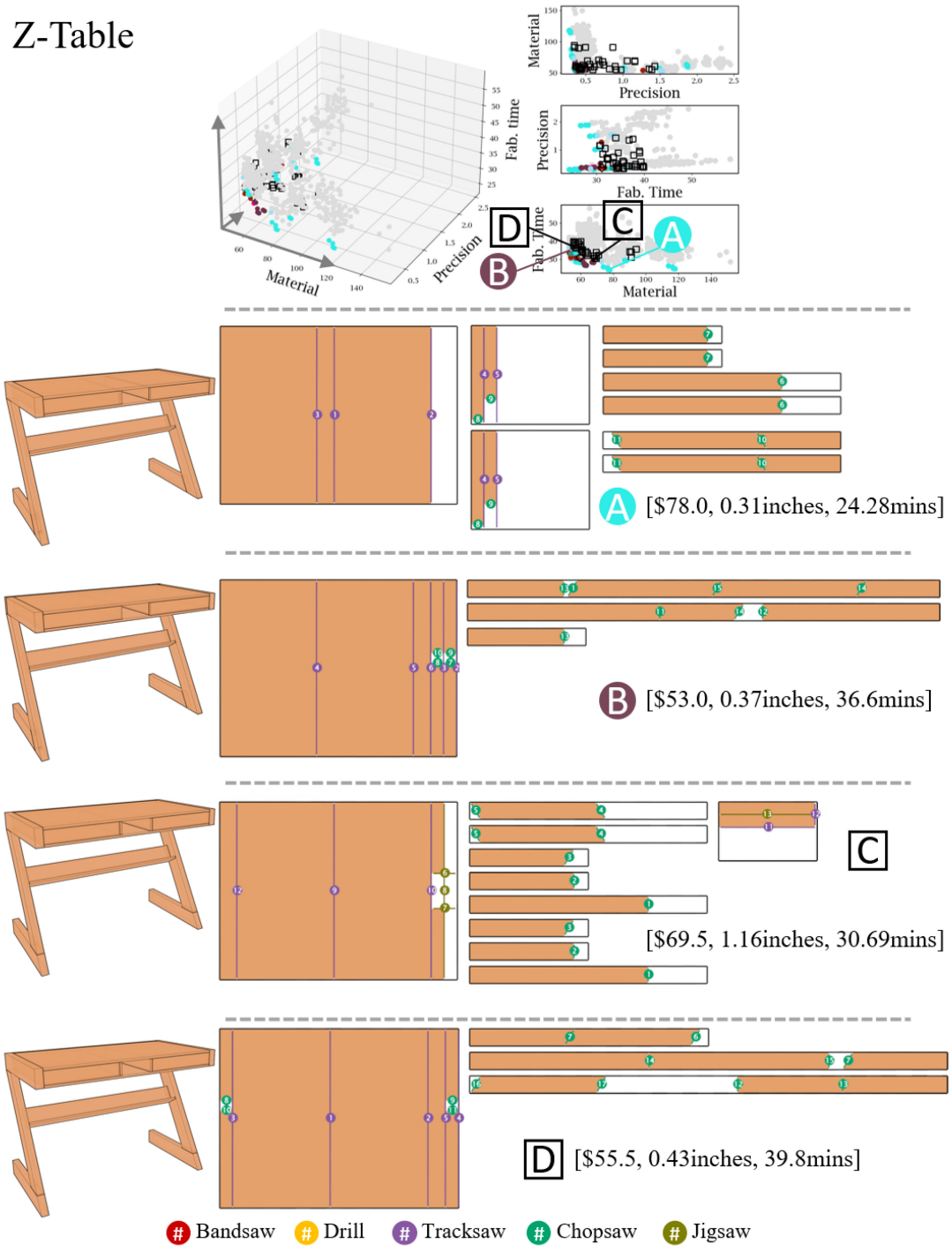

Fig. S17. Design variants and fabrication plans extracted with our ICEE algorithm (Z-Table).

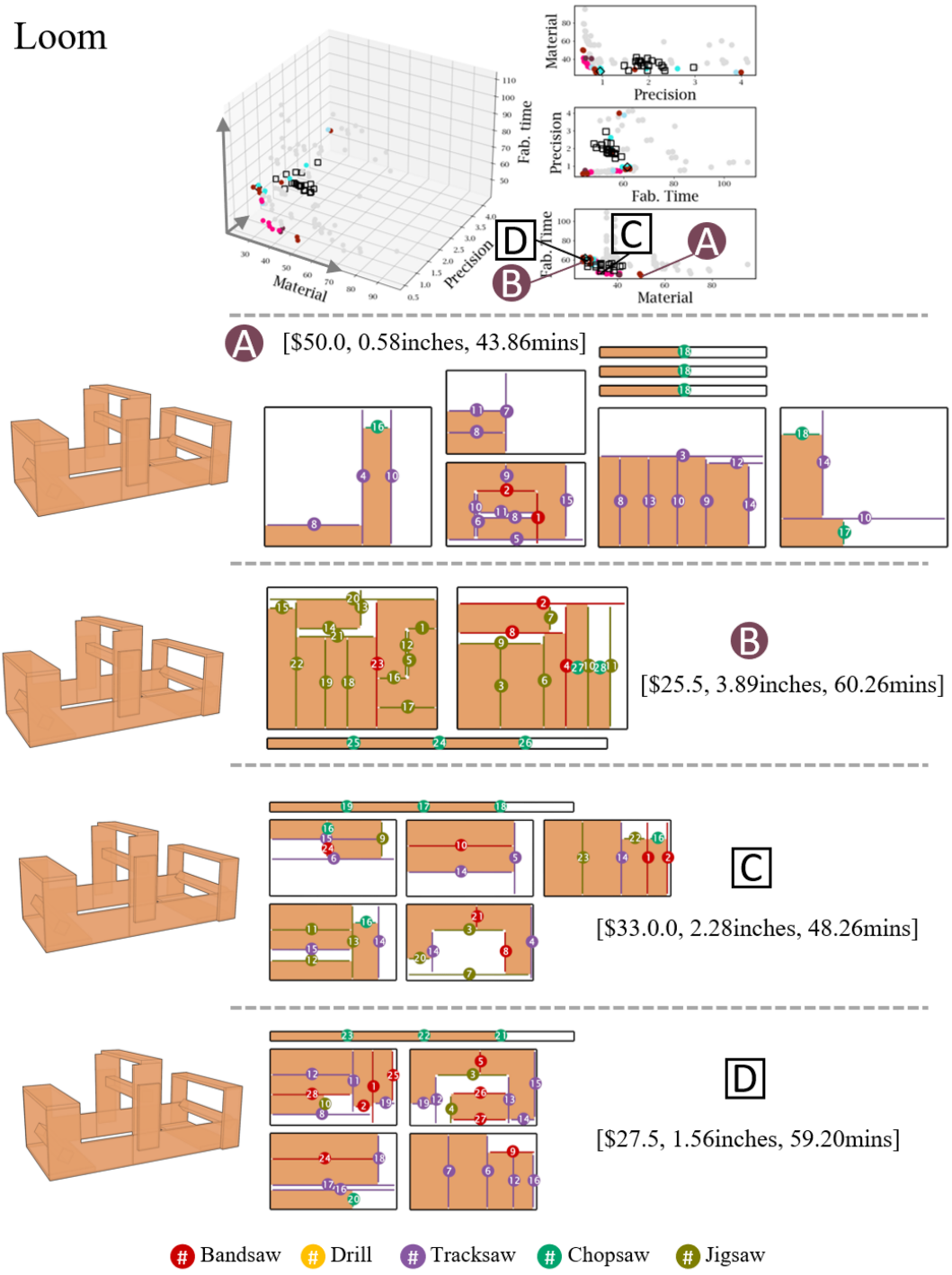

Fig. S18. Design variants and fabrication plans extracted with our ICEE algorithm (Loom).

REFERENCES

McMASTER-CARR. 2021. McMASTER-CARR. Retreived 05 May 2021 from https://www.mcmaster.com/. (2021).

Chenming Wu, Haisen Zhao, Chandrakana Nandi, Jeffrey I. Lipton, Zachary Tatlock, and Adriana Schulz. 2019. Carpentry compiler. *ACM Transactions on Graphics* 38, 6 (2019), 1–14.