# Algorithmic circular design with reused structural elements: Method and Tool

*Yijiang Huang, MIT, Cambridge, MA 02139, USA, yijiangh@mit.edu*
*Latifa Alkhayat, MIT, Cambridge, MA 02139, USA, latifa@mit.edu*
*Catherine De Wolf, Delft University of Technology, 2628 BL Delft, Netherlands & ETH Zürich, Institute of Construction & Infrastructure Management, 8093 Zurich, Switzerland, cdewolf@ethz.ch*
*Caitlin Mueller, MIT, Cambridge, MA 02139, USA, caitlinm@mit.edu*

## Abstract
Structural systems are responsible for a significant portion of embodied carbon emissions in buildings. A potential path to increase sustainability is to integrate circular economy principles in structural design, which advocate for prioritizing the reuse of structural materials to extend their service life, limiting their physical transformation to locational and functional changes. In this way, structural projects of the past can not only serve as an inspiration for the future, but the material itself can also be reappropriated. Recently, computational approaches for material reuse have gained traction. This paper extends previous work by comparing several algorithmic formulations for reuse-driven design, introducing a new Grasshopper-based tool that implements them, and demonstrating their application on a case study.

## 1 Introduction

The Intergovernmental Panel on Climate Change (IPCC) states that the building sector needs to be "zero-carbon" by 2050 to meet the targets set by the Climate Agreement and avoid extreme climate catastrophes. The construction sector, accounting for 13% of the world's GDP [1], uses 50% of all materials [2], [3], generates 36% of the waste [4], and emits up to 12% of global greenhouse gas emissions for building material extraction, manufacturing, and construction, counting only Europe [5]. This is due to the sector's linear model, which extracts, produces, uses, and disposes of building materials and resources. To remediate this detrimental condition, worldwide, a transition to a circular repair-reuse-recycling model is urgently needed in today's construction sector [6], [7]. A circular model would extract maximum value from building materials by extending their service life or reusing them at the end of their service life as new resources, while minimizing their environmental impact. Due to rapid urbanization, it has become more attractive to demolish buildings and rebuild new ones rather than deconstructing them and reusing their materials. Global implementation on a large scale of a circular model in the construction sector has not yet been successful. The fragmented supply chain in the Architecture, Engineering and Construction (AEC) sector prevents both the broad application of circular strategies in construction practice. This could be addressed through the uptake of digital tools such as computational design tools and databanks of materials. This paper proposes new algorithms and design methods that enable the use of buildings instead of the earth as material mines and depots.

Building on recent developments in computational approaches for helping designers reuse materials [8], [9], this paper considers the design of structures built with reused materials. The proposed method assesses the capacity of a newly generated design to use materials from a stock of available materials from reuse. To test and validate the algorithm and method, the linear timber structure elements of a conventional house are inventoried and reused in the design of geodesic domes (Fig. 1), which are clad and used as greenhouses. This illustrates how algorithmic matching of reused materials can be integrated into the design workflow of architects and engineers through quick computational feedback. The case study in this paper uses timber elements as working with this material is accessible to most constructors with relatively simple tools.

## 2 Material reuse approaches

Many inspiring projects have illustrated the feasibility of material reuse in various contexts: the reuse of 180 pieces of bent glass from the Centre Pompidou's façade in Maximum's architectural project in Paris, France [10]; the use of waste material as lost formwork in filler slabs such as the 2000 Wall House project by Anupama Kundoo in India [11]; to name a few.

Fig. 1 A geodesic dome example built with renewable materials (Credit: N. Petit-Barreau, Anku).

### 2.1 Computational methods for material reuse

Despite these contemporary examples, material reuse is not the norm in today’s design and construction practices. This is partly due to the added challenges that working with existing, often irregular material resources bring. Instead of designing in an unconstrained manner with an assumption of infinite material supply, architects and engineers who wish to reuse existing material must devote time, creativity, and flexibility to devising form and space with a geometrically and structurally constrained kit of parts of limited size. These challenges may be surmountable in boutique projects, but are hard to address at scale in everyday construction.

One response to these challenges is the use of computational methods, which can use automation to assist in designing with a fixed material inventory. Already common in architectural design for non-reuse cases, computational design methods such as parametric design space exploration and rule- or grammar-based design approaches can be productive generative tools for material reuse. While there are a variety of methods used in previous literature and discussed in this section, there are two fundamental design philosophies.

The first, bottom-up design, starts with available material objects and algorithmically aggregates them into architectural assemblies. Computationally, it has its origins in shape grammars [12] and more recent work in making grammars [13], and uses predefined rules to automate and control the process of aggregation. This approach has the advantage of a guaranteed geometric fit of the existing material into the new design, and more naturally follows a non-computational physical workflow, e.g. building with blocks. The challenge of this approach is that it can be very hard to control the resulting design, and to meet any additional design intentions, such as overall formal, spatial, and structural goals. This approach is also more easily adapted to inventories of self-similar parts, e.g. dimensional lumber, than truly diverse material stocks that are of interest in reuse. Examples of this method include [14]–[16].

The second philosophy, followed in this paper and others before it, is top-down design. This approach has its roots in conventional parametric design and optimization, and starts with a “target” design concept model. An inventory of available construction elements is algorithmically searched, and parts are selected and matched to the target, ideally in terms of both geometrical fit and structural capacity. Typically, the matching is not perfect, and the inventory elements must be processed in some way to be used in the final construction. Various algorithms have been used to conduct and optimize this matching process (and minimize processing and waste), as shown in Table 1. Heuristic algorithms such as Greedy Search are simpler to implement but not guaranteed to result in the best match. More rigorously formulated optimization algorithms can be slow.

If the matching process is fast enough, it can be used within or in addition to other design considerations, such as overall form. Then, the overall design can be modified or optimized to most closely fit the available material inventory. This has both practical and conceptual appeal as it has a similar philosophy to traditional form-finding, which attempts to minimize material mass; rather than imposing their abstract formal ideas on architectural problems, designers can discover geometries that meet important performance goals. In previous work, this has been demonstrated in [9], [17], [18].

Table 1 Summary of previous work in computational assignment optimization of reclaimed structural elements. *N/A means that structural capacities of the inventory elements are not considered in the matching.

| Reference | Structural application | Problem | Geometric and structural matching | Algorithm |
|---|---|---|---|---|
| Fujitani and Fujii 2000 [19] | Frames with linear elements | Inventory matching with structural mechanics | Separated | Genetic Algorithm |
| Mollica and Self 2017 [20] | Arched truss with tree fork connectors | Inventory matching | *N/A | Greedy Search |
| Bukauskas et al. 2017 [22] | Trusses with linear elements | Inventory matching with structural mechanics and cutting stock optimization | Separated | Greedy Search |
| Brütting et al. 2018[17] | Trusses with linear elements | Inventory matching with structural mechanics | Simultaneous | Mixed Integer Linear Programming (MILP) |
| Lokhandwala et al. 2018 [24] | Funicular shells with planar polygonal panels | Inventory matching (2D polygonal packing) | N/A | Dynamic Relaxation |
| Larsson et al. 2019 [21] | Interlocking grid shells with linear elements | Inventory matching | Separated | Hungarian Algorithm |
| Allner et al. 2020 [14] | Grid shells with tree fork connectors | Inventory matching | Separated | Greedy Search |
| Brütting et al. 2020 [8], [23] | Trusses/frames with linear elements | Inventory matching with structural mechanics and cutting stock optimization | Simultaneous | Mixed Integer Linear Programming (MILP) |
| Amtsberg et al. 2021 [9] | Grid shells with tree fork connectors | Inventory matching with implicit structural mechanics | N/A | Hungarian Algorithm |
| *this paper* | Grid shells with linear elements | Inventory matching with structural mechanics | Separated | Hungarian Algorithm |

## 2.2 Research gap

This paper extends previous work by introducing a new computational approach to design for material reuse in a flexible, interactive, designer-driven workflow. Central to this approach is the use of the classical Hungarian Algorithm, first introduced in 1955 [25], which has only been applied to the material reuse problem twice before in previous literature [21], [9]. This paper is the first to demonstrate the Hungarian Algorithm as implemented in a free, open-source tool for Grasshopper, which is used together with recently introduced tools for design space exploration, including sampling and single- and multi-objective optimization. Compared to previous work, which has formulated the reuse problem in mathematically rigorous but rigid ways, this paper proposes a modular approach that can be adjusted iteratively and quickly based on the evolving priorities of the design team. This workflow is tested and analyzed in a case study design problem to further understanding of algorithmic material reuse for sustainable architecture.

## 3 Methodology and case study

The reuse approach proposed by this paper is summarized in Fig. 2. Source material is identified and digitally processed into an inventory, and a parameterized design model is created and linked to structural analysis to generate possible target designs with computed axial forces. A cost matrix is assembled that includes all possible pairwise matches between the inventory and target, considering both geometry (member length) and structural mechanics (tensile and compressive demand and capacity). The Hungarian Algorithm takes the cost matrix as an input and returns the optimal global match and a matching cost, which can be used with the parameterized design model in shape optimization and related approaches. More details of each of these steps are given below.

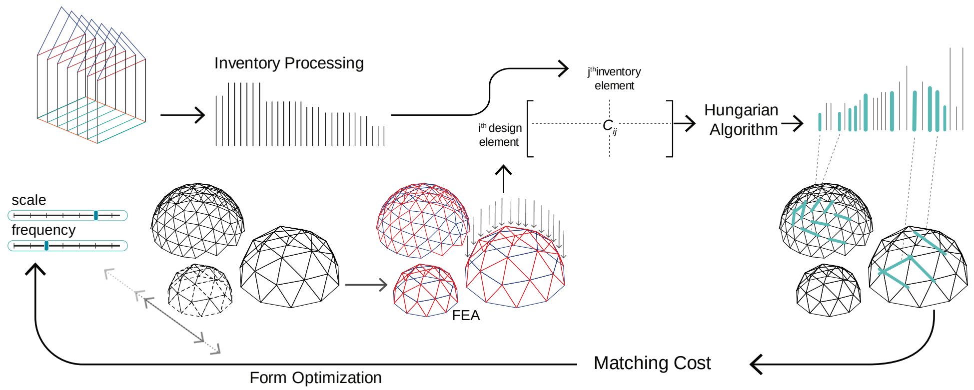


Fig. 2 Material reuse workflow conceptual overview.

### 3.1 Inventory processing and cost matrix computation

The first input is a BIM-like 3D model of existing material to be reused. This paper focuses on reuse of linear timber elements, and assumes the stock material comes from a timber-framed house (discussed specifically in Section 4.1). Elements in the house model are digitally catalogued and processed so that they can be matched to a target design, using the process discussed in the following section.

A cost matrix $D$ of dimension $a$ and $b$ is computed by comparing each of the $a$ elements in the target design to each of the $b$ elements in the inventory. Each pairwise cost in the matrix is computed in 2D Euclidean space using the $L_2$ norm; the two dimensions of this space correspond to member length and axial load capacity/demand. As shown in Fig. 3, the axial load capacity is computed specifically for each pair, dependent on the target element's length and axial load sign (tension or compression). The load demands are computed by performing a 3D frame finite element analysis on the dome, with a fixed, pre-specified cross section and material property. A large penalty value is added to the cost measure when the inventory element is insufficient in length or capacity, as a way to ensure that matched elements are geometrically and structurally feasible. Other types of distance function, e.g. a weighted $L_2$ norm can also be used here to address specific preference.

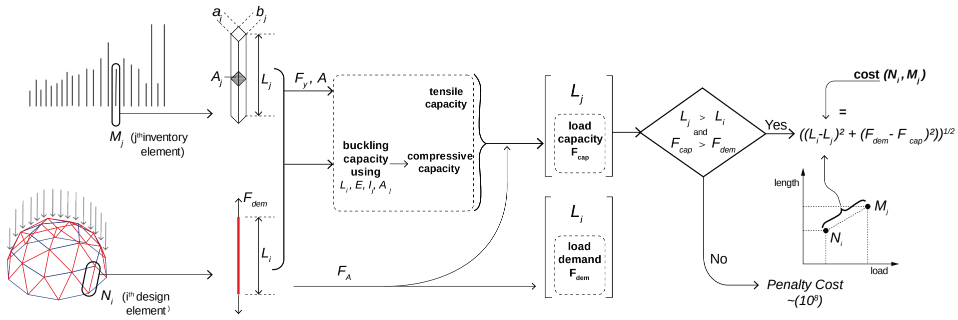


Fig. 3 Assembly of cost matrix for all pairwise combinations of inventory and target elements.

## 3.2 Optimal matching algorithm

Given a cost matrix $D$ defined in the previous section, with a target elements and b inventory elements, the optimal matching problem can be formulated as follows:

$$\begin{aligned} min_T\ c(T)\ &= \sum_{i,j=1}^{a,b} D_{ij}\, T_{ij} \\ s.t. \textstyle\sum_{i=1}^{a}\ T_{ij} &\leq 1, \forall\ j\ \in \{1, \dots, b\} \\ \textstyle\sum_{j=1}^{b}\ T_{ij} &= 1, \forall\ i\ \in \{1, \dots, a\} \\ T_{ij} &\in \{0, 1\}, \forall\ i\ \in \{1, \dots, a\},\ j\ \in \{1, \dots, b\} \end{aligned} \tag{1}$$

where the design variables are the entries $T_{ij}$ of the assignment matrix $T$, where $T_{ij} = 1$ means inventory element $j$ is assigned to target element $i$, and 0 otherwise. The first inequality constraint ensures that each inventory element $j$ can be assigned to at most one target element. The second equality constraint enforces that exactly one inventory element is used at target element $i$. The third constraint enforces that the assignment matrix is binary. The cost matrix $D$ encodes the cost of assigning element $j$ to $i$, computed as described in Section 3.1. An optimal matching $T^*$ is the assignment that minimizes the total matching cost $c(T)$. In this paper, the *matching cost* of a given design is defined to be $c(T^*)$.

The problem described in equation (1) is an unbalanced linear assignment problem, a well-studied combinatorial optimization problem in the literature and widely used in various practical contexts [26]. Because of the discrete nature of the problem, Greedy Search algorithms (also called best-first search algorithms) have been shown to be practically effective [22]. However, despite their simplicity of implementation, the assignments obtained from the greedy search algorithms are generally not globally optimal. In this paper, the Hungarian Algorithm, a combinatorial optimization algorithm specifically developed for solving the linear assignment problem, is used to solve it to the global optimality in polynomial time [25].

The formulation in eq. (1) is an instance of integer linear programming (ILP) problems, and one can use more generic ILP machinery (e.g. the branch-and-bound algorithm [27]) to solve it. Moreover, eq. (1)'s constraint matrix is totally unimodular, a mathematical property that guarantees that the optimal solution is integral even if the binary constraint is relaxed [27]. Thus, linear programming algorithms designed for continuous problems (e.g. the simplex algorithm [26]) can be directly applied, relaxing the binary constraint, but still leads to integral solutions. In summary, the linear assignment problem is an integral linear programming problem with a very special mathematical structure, and algorithms with very different inner workings can be applied to solve it. In Table 2, empirical runtime data for these algorithmic options is presented. The Hungarian algorithm is about 60 times faster than the other two options.

Table 2 Runtime for solving a linear assignment problem with randomly generated 100*500 cost matrix. ILP and LP solved using Gurobi [28] via JuMP.jl [29], Hungarian using Hungarian.jl [30] with the Julia programming language [31]. All algorithms converge to the same optimal solution.

| Integer Linear Programming (ILP) | Linear Programming (LP) | Hungarian Algorithm |
|---|---|---|

| | | |
|---|---|---|
| 312.5 ms | 234.1 ms | 5.2 ms |

Prior work in assignment optimization for material reuse uses a mixed-integer linear programming (MILP) formulation, additionally introducing continuous design variables for the structural nodal displacements and member forces so that structural analysis and assignment are treated simultaneously [17], [23].

In contrast, the formulation presented in this paper relaxes the MILP formulation by removing the structural analysis and stock length constraint. This relaxation has its advantages and disadvantages. The relaxation turns a MILP problem into a linear assignment problem and thus enables very efficient algorithmic treatment (see Table 2). However, since the structural capacity and stock length constraints are encoded as penalties in the objective function, instead of enforced as hard constraints, global optimality in terms of structural capacity is not guaranteed. Since the FEA is performed before the material matching with pre-assigned values, the load demand calculation is conservative, compared to the accurate load demand calculation used in the simultaneous design and analysis framework enabled by the MILP formulation [23]. However, the lower-bound theory of plasticity (e.g. as discussed in [32]) guarantees that the results are structurally safe. This approach also trades in hard constraint enforcement. This is a classic trade-off in constrained optimization formulation, and its unconstrained counterpart with penalty. Furthermore, while it is easy to extend the MILP formulation to solve the cutting-stock problem where one inventory element can be partitioned into more than one structure members, the Hungarian algorithm can only compute a one-to-one assignment. Such constraints can limit the solution of equation (1) to be sub-optimal compared to the cutting-stock MILP formulation when reducing cut-off waste [23]. Despite these disadvantages, this paper values the flexibility and independence of proprietary MILP solvers to enable a rapid and interactive computational design experience.

In order to facilitate the use of this research in both academia and industry, a Grasshopper script is made freely accessible online [33], released under the MIT license. For the matching part, a C# backend is provided that works out-of-the-box, using an existing open-source implementation [34]. To further improve the computational efficiency, an alternative matching backend is provided, written in the Julia programming language [31] using the Hungarian.jl package [30], which requires slightly more installation overhead but is at least an order of magnitude faster.

### 3.3 Parametric design model and optimization

In order to explore the full potential of the material inventory, a dome collection is parameterized in a flexible way that allows the number of domes, the radius of each dome, and the subdivision (called density in this paper) of each dome to vary, illustrated in Fig. 4. Geodesic dome designs are generated individually using the RhinoPolyhedra [35] Grasshopper plugin. Two NURBS curves are parameterized by a set of control points; whose vertical positions become the main design variables. The curves are sampled $n$ times along their lengths, where $n$ is the total number of domes. The vertical coordinate of the sampled point is used to define the radius or density parameter for each dome. The total number of variables is 9 (4 control points on 2 curves, plus $n$).

Several objective functions are possible to create a performance design space to be explored. The total *matching cost*, introduced in 3.2, is one important performance metric; a minimal matching cost means that surplus material in offcuts and structural capacity are minimized. As noted in 3.1, a one-sided penalty term is used to enforce length and capacity constraints. A second related objective function is *inventory utilization*, which computes the amount of structural material used from the inventory compared to the total available. A maximal inventory utilization indicates that the inventory is being used as fully as possible, and offcuts and extra stock is minimized. If an invalid match is returned by the Hungarian Algorithm, the inventory utilization is 0. Finally, the total *floor area* of the domes is an important objective to be maximized, which reflects the functional value of the reused materials' configuration. Notice that while the first objective, the total matching cost, is *computed* by solving an *inner* optimization problem (equation (1)) using the Hungarian Algorithm, *minimizing* the matching cost or finding its trade-off with the other two objectives involves running an *outer* optimization loop using single-objective or multi-objective optimization machineries.

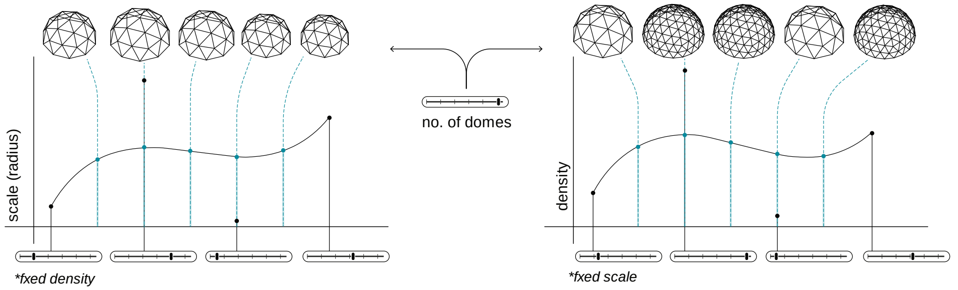


Fig. 4 Dome collection parametrization by mapping discretized points' coordinate on NURBS curves parameterized by control points to dome geometry attributes.

These objectives are used together and separately in a number of design space exploration experiments, using the free Design Space Exploration (DSE) plugin for Grasshopper [36], [37]. These include design space sampling, single-objective optimization, and multi-objective optimization.

## 4 Results

Using the methods described above, a series of design space exploration experiments can reveal a variety of opportunities for material reuse, discussed briefly in this section.

### 4.1 Inventory processing

First, the inventory of available stock material is digitally processed from a BIM model (in this case, from a publicly available model of a wood-framed house [38]). Individual linear framing elements are catalogued and digitally "cut" lengthwise to create a stock of elements with square cross sections, based on the construction and connector logic of the geodesic dome greenhouses that inspire this research (Fig. 1). In total, the processed inventory contains 2371 elements, organized into 13 types (roof rafter, wall stud, etc.). The minimum, maximum, median, and mean lengths are 0.2 m, 10.6 m, 3 m, and 3.62 m respectively. There are four different cross sections: 25 mm$^2$, 50 mm$^2$, 60 mm$^2$, and 100 mm$^2$. In this paper, all timber elements are assumed to have the same allowable strength and elastic modulus, 6.5 MPa and 10500 MPa respectively. The inventory is summarized visually in Fig. 5.

The large size of the inventory was found to make fast execution of the Hungarian algorithm challenging. Therefore, a decomposition strategy is implemented in which the original inventory is binned into two inventories of similar size and character in an automated process. The following results demonstrate possible constructions with one-half of the total inventory from the original house; the other half of the inventory could be used to construct similar arrays of domes.

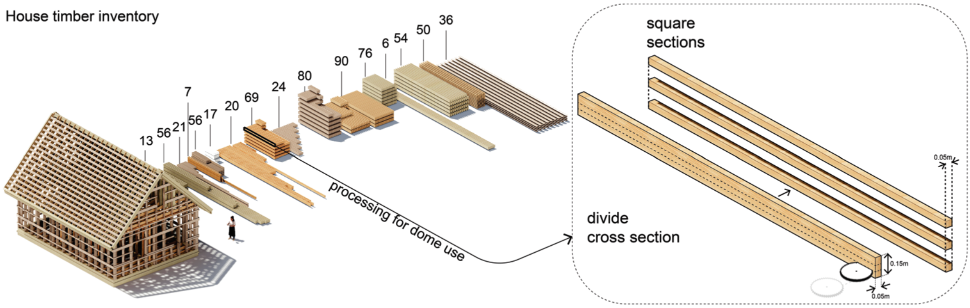


Fig. 5 House to inventory processing by element type

### 4.2 Multi-objective optimization

The digital inventory is linked to the parameterized geodesic dome models described in 3.3, with optimal assignment to elements in the domes executed by the Hungarian Algorithm and tool described in 3.2. This creates a design space with multiple objective functions: matching cost (minimize), inventory utilization (maximize), and floor area (maximize). Using the MOO tool from DSE, the bi-objective plots in Figs. 6 and 7 are created; each takes about one hour on a standard laptop. Intuitively, there are trade-offs between some of objectives: a minimal matching cost will produce fewer domes with less total inventory material, to reduce waste. Inventory utilization and floor area are intuitively correlated, but not necessarily perfectly so. Fig. 6 reveals the first trade-off, with Pareto optimal and near-Pareto optimal designs highlighted. There is an increased total matching cost when material coverage is maximized, and the best matching option uses only 2% of the inventory (Fig. 6, bottom-left). The best option to choose depends on how much the design team wants to avoid offcuts and oversizing, and whether the unused inventory can be easily stored for future use. In this paper, offcuts are not considered for additional matching, but could be re-added to the inventory to make them easier to reuse.

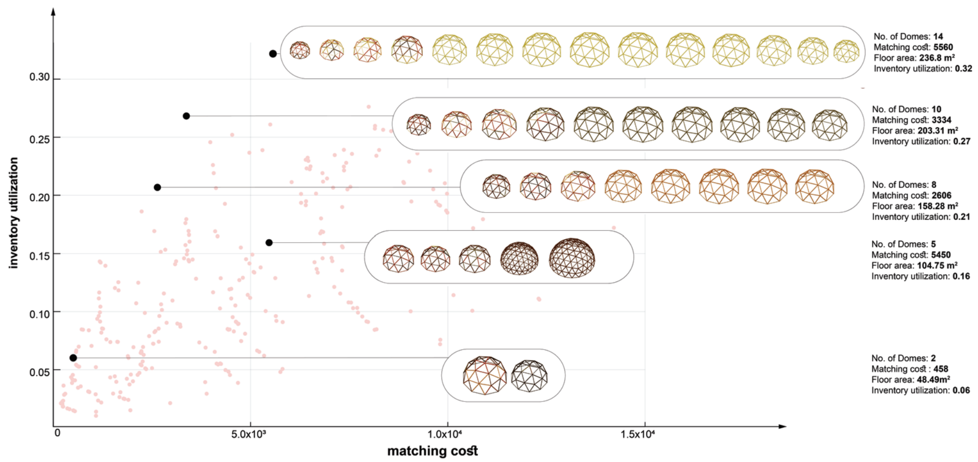


Fig. 6 Bi-objective plot for matching cost (minimize) and inventory utilization (maximize)

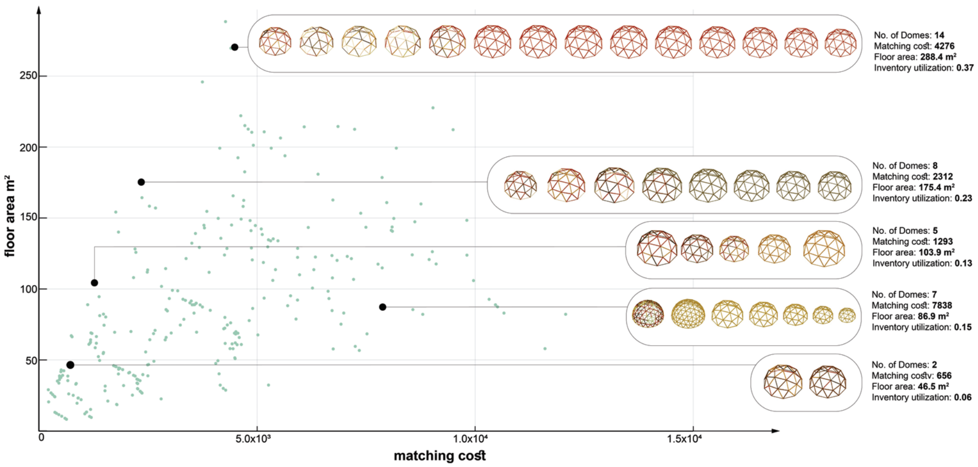


Fig. 7 Bi-objective plot for matching cost (minimize) and total floor area (maximize)

In Fig. 7, a similar trend is observed: more than 10x more floor area is achievable compared to the optimal matching cost design, with an increase in matching cost of approximately 5x. If the design team has a preferred number of domes, that could also be used to choose among the Pareto-optimal options. Interestingly, similar amounts of floors area can be achieved with a wide variety of dome counts by using fewer large domes or more small domes.

### 4.3 Design space sampling and feasible region analysis

Design space sampling techniques can also be used to assess the effects of the length and strength constraints on the matching process. Fig. 8 shows the results of a Latin hypercube sample across multiple numbers of domes of varying radii with fixed density (=1), in terms of matching cost and inventory utilization (achieved using the Sampler and Capture tools in DSE in about one hour). Infeasible designs have very high matching costs and inventory utilizations of 0. The sampling results show that 42% of the generated designs are infeasible due to elements with no possible matches; these designs typically have long or highly stressed members. This type of analysis can be used to reveal the overlap between a parametric design space and a material inventory, and the design space can be adjusted to contain more feasible options. In this case, the feasible rate is considered acceptable and still allows for many possible designs.

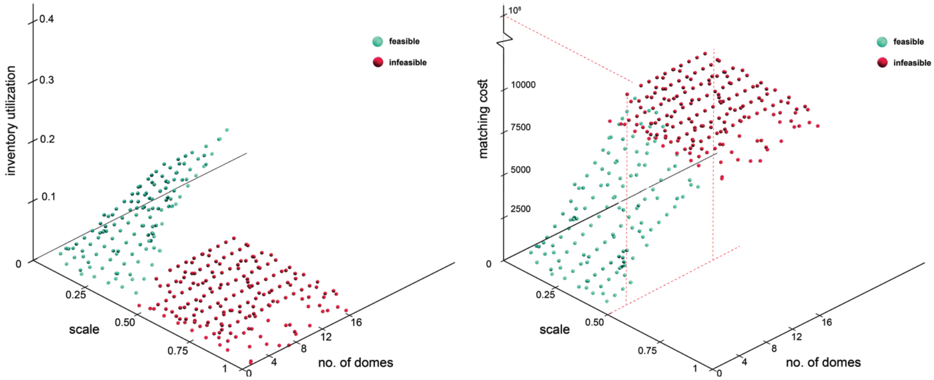


Fig. 8 Visualization of feasible regions in design subspaces for the geodesic dome example. As the number of domes increases, there is less flexibility on radius (scale).

### 4.4 Single objective optimization

Finally, conventional single-objective optimization can be used to find a preferred configuration of domes, using Improved Stochastic Ranking Evolution Strategy (ISRES) as the global optimizer and Constrained Optimization BY Linear Approximations (COBYLA) as the local optimizer via the Radical tool in the DSE suite. The stochastic, global optimization method is used to find a region of interest in the non-convex design space, the local optimization is used to fine-tune the final result. The total runtime is about 10 minutes. The objective function is a combination of two of the previous metrics discussed: matching cost divided by floor area, which balances the need for waste minimization with the desire to generate as much functional space as possible. The number of domes is held constant in this case to reduce the challenges of discrete variables; Fig. 9 shows the result for a single dome, and Fig. 10 shows the results for 15 domes. The results show that high-quality and flexible results can be generated quickly within a design workflow.

## 5 Conclusions

This paper reviews and compares algorithmic formulations for reuse-driven design in existing computational approaches, and introduces a new Grasshopper tool to implement them. Both flexible design space exploration and efficient optimization are obtained by the use of the Hungarian Algorithm in a nested loop workflow. For small problems, the material reuse efficiency is computed in real time; for larger problems in a few seconds. The potential of this approach and tool are demonstrated on a real world case study that will be explored through physical experiments in future work.


## Acknowledgements

Authors acknowledge assistance from Nicolas Petit-Barreau on constructing geodesic domes.

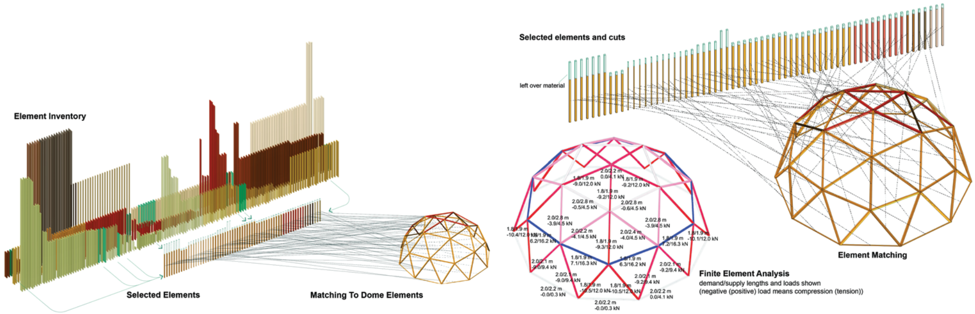


Fig. 9 Left: single dome matching; number of design elements: 65, matching cost: 197, floor area: 29.6, material coverage: 0.04, objective: 6.67. Right: Close-up view of the matching result and the FEA analysis.

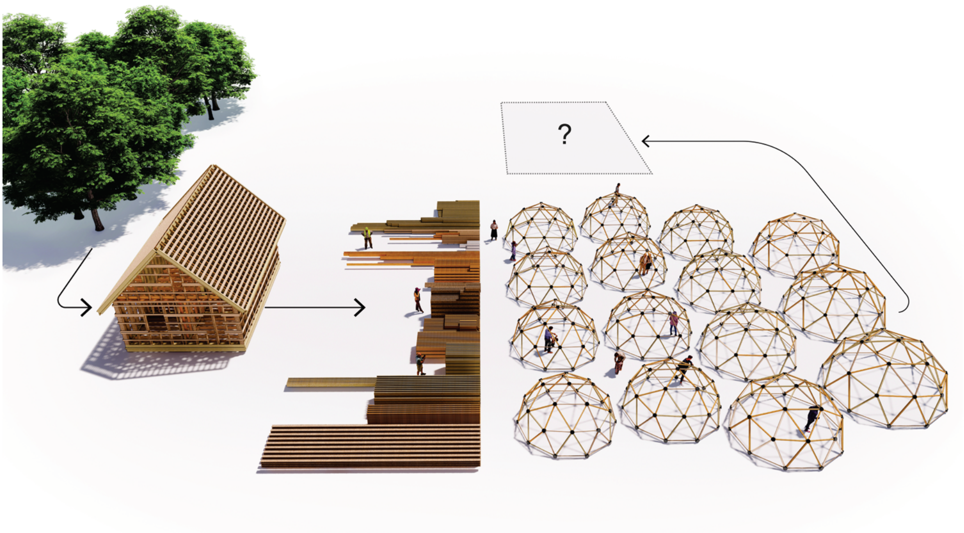


Fig. 10 Number of design elements: 975, matching cost: 4316, floor area: 317.3, material coverage: 0.39, objective: 13.71

## References


[1] M. Ribeirinho *et al.*, "The Next Normal in Construction: How Disruption is Reshaping the World's Largest Ecosystem," McKinsey & Company, 2020. [Online]. Available: https://www.mckinsey.com/business-functions/operations/our-insights/the-next-normal-in-construction-how-disruption-is-reshaping-the-worlds-largest-ecosystem#.

[2] M. Herczeg *et al.*, "Resource efficiency in the building sector.," European Commission, Rotterdam, 2014.

[3] BIO Intelligence Service, "Sectoral Resource Maps. Prepared in response to an Information Hub request," European Commission, 2013.

[4] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.

[5] European Commission, "Buildings – Energy," 2019. [Online]. Available: ec.europa.eu/energy/en/topics/energy.

[6] C. De Wolf, E. Hoxha, A. Hollberg, C. Fivet, and J. Ochsendorf, "Database of Embodied Quantity Outputs: Lowering Material Impacts Through Engineering," *Journal of Architectural Engineering*, vol. 26, no. 3, p. 04020016, 2020.

[7] C. De Wolf, J. Brütting, and C. Fivet, “Embodied Carbon Benefits of Reusing Structural Components in the Built Environment: a Medium-rise Office Building Case Study,” in *Proceedings of the PLEA 2018 Conference*, 2018.
[8] J. Brütting, C. Vandervaeren, G. Senatore, N. De Temmerman, and C. Fivet, “Environmental impact minimization of reticular structures made of reused and new elements through Life Cycle Assessment and Mixed-Integer Linear Programming,” *Energy and Buildings*, vol. 215, p. 109827, May 2020.
[9] F. Amtsberg, Y. Huang, D. J. M. Marshall, K. M. Gata, and C. Mueller, “Structural upcycling: Matching digital and natural geometry,” in *Advances in Architectural Geometry*, 2020.
[10] R. Ménard, *Faire et refaire du verre*. 169 architecture, Pavillion de l’Arsenal, 2018.
[11] A. Kundoo, “Freedom from the Known: Imagining the Future Without the Baggage of the Past,” *Architectural Design*, vol. 88, no. 3, pp. 54–61, 2018, doi: https://doi.org/10.1002/ad.2301.
[12] G. Stiny, “Introduction to shape and shape grammars,” *Environment and planning B: planning and design*, vol. 7, no. 3, pp. 343–351, 1980.
[13] T. Knight and G. Stiny, “Making grammars: from computing with shapes to computing with things,” *Design Studies*, vol. 41, pp. 8–28, 2015.
[14] L. Allner, D. Kroehnert, and A. Rossi, “Mediating Irregularity: Towards a Design Method for Spatial Structures Utilizing Naturally Grown Forked Branches,” in *Impact: Design With All Senses*, C. Gengnagel, O. Baverel, J. Burry, M. Ramsgaard Thomsen, and S. Weinzierl, Eds. Cham: Springer International Publishing, 2020, pp. 433–445.
[15] A. Rossi and O. Tessmann, “From voxels to parts: hierarchical discrete modeling for design and assembly,” in *International Conference on Geometry and Graphics*, 2018, pp. 1001–1012.
[16] S. Leder, R. Weber, D. Wood, O. Bucklin, and A. Menges, “Distributed Robotic Timber Construction,” in *Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, The University of Texas at Austin School of Architecture, Austin, Texas, 2019, p. 10.
[17] J. Brütting, G. Senatore, and C. Fivet, “Optimization Formulations for the Design of Low Embodied Energy Structures Made from Reused Elements,” in *Advanced Computing Strategies for Engineering*, Cham, 2018, pp. 139–163.
[18] J. Brütting, G. Senatore, and C. Fivet, “Design and fabrication of a reusable kit of parts for diverse structures,” *Automation in Construction*, vol. 125, p. 103614, May 2021.
[19] Y. Fujitani and D. Fujii, “Optimum structural design of steel plane frame under the limited stocks of members,” in *Proceedings of the RILEM/CIB/ISO International Symposium, Integrated Life-Cycle Design of Materials and Structures*, 2000, pp. 198–202.
[20] Z. Mollica and M. Self, “Advances in Architectural Geometry 2015 - Tree Fork Truss: geometric Strategies for Exploiting Inherent Material Form,” 2016.
[21] M. Larsson, H. Yoshida, and T. Igarashi, “Human-in-the-loop fabrication of 3D surfaces with natural tree branches, ” In: *Proceedings of the ACM Symposium on Computational Fabrication*. SCF ’19.ACM, pp. 1-12, 2019. doi:10.1145/3328939.3329000
[22] A. Bukauskas, P. Shepherd, P. Walker, B. Sharma, and J. Bregulla, “Form-Fitting Strategies for Diversity-Tolerant Design,” in *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium*, Hamburg, Germany, 2017, p. 10.
[23] J. Brütting, G. Senatore, M. Schevenels and C. Fivet, “Optimum design of frame structures from a stock of reclaimed elements,” *Frontiers in Built Environment*, 2020, 6, p.57.
[24] Z. Lokhandwala, “Computational Design with Reused Materials,” Paris, France, 2018, p. 13.
[25] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[26] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*, vol. 6. Athena Scientific Belmont, MA, 1997.
[27] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*, vol. 55. John Wiley & Sons, 1999.
[28] “jump-dev/Gurobi.jl,” Mar. 25, 2021. https://github.com/jump-dev/Gurobi.jl (accessed Apr. 03, 2021).
[29] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A Modeling Language for Mathematical Optimization,” *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017, doi: 10.1137/15M1020575.

[30] "Gnimuc/Hungarian.jl: The Hungarian (Kuhn-Munkres) algorithm for Julia." https://github.com/Gnimuc/Hungarian.jl (accessed Apr. 03, 2021).

[31] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[32] P. Block and J. Ochsendorf, "Lower-bound analysis of masonry vaults," in *Structural Analysis of Historic Construction: Preserving Safety and Significance*, E. Fodde, Ed. CRC Press, 2008, pp. 593–600.

[33] Y. Huang, L. Alkhayat, and C. Mueller, "algorithmic circular design: implementation." https://github.com/yijiangh/algorithmic_circular_design.

[34] "vivet/HungarianAlgorithm: Hungarian Algorithm Implementation." https://github.com/vivet/HungarianAlgorithm (accessed Apr. 03, 2021).

[35] "RhinoPolyhedra," *Food4Rhino*, Feb. 12, 2014. https://www.food4rhino.com/app/rhinopolyhedra (accessed Apr. 11, 2021).

[36] N. C. Brown, V. Jusiega, and C. T. Mueller, "Implementing data-driven parametric building design with a flexible toolbox," *Automation in Construction*, vol. 118, p. 103252, Oct. 2020.

[37] "Design Space Exploration," *Food4Rhino*, Aug. 07, 2017. https://www.food4rhino.com/app/design-space-exploration (accessed Dec. 12, 2018).

[38] BAHC, "House Frame." https://3dwarehouse.sketchup.com/model/62aef36d-ce14-440b-be54-d20ad8b1a4e1/House-Frame (accessed Apr. 12, 2021).