Evolutionary Design and Assembly Planning for Stochastic Modular Robots

Michael T. Tolley, *Student Member, IEEE*, Jonathan D. Hiller, and Hod Lipson, *Member, IEEE*

***Abstract*—A persistent challenge in evolutionary robotics is the transfer of evolved morphologies from simulation to reality, especially when these morphologies comprise complex geometry with embedded active elements. Here we present a process that automatically evolves target structures based on functional requirements and plans the error-free assembly of these structures from a large number of active components. Evolution is conducted by minimizing the strain energy in a structure due to prescribed loading conditions. Thereafter, assembly is planned by sampling the space of all possible paths to the target structure and following those that leave the most options open. Each sample begins with the final completed structure and removes one accessible component at a time until the existing substructure is recovered. Thus, at least one path to a complete target structure is guaranteed at every stage of assembly. Automating the entire process represents a step towards an interactive evolutionary design and fabrication paradigm, similar to that seen in nature.**

I. Introduction

While the use of evolutionary algorithms has gone a long way towards automating the design process, the automated fabrication of evolved designs remains a major challenge. This is especially true for designs with complex morphologies or active components. Some approaches use rapid prototyping [1],[2], with a growing effort to 3D-print active components such as actuators [3] and batteries [4], [5]. Alternatively, other approaches explore the construction of robots from many small modular active units [6]-[9]. As the number of modules increases and their size decreases, manufacturing methods need to scale appropriately. Here we explore a combined evolutionary and assembly process geared towards the design and construction of robots from such components.

Our previously-proposed fabrication concept is to assemble active components that move about stochastically in a fluidic environment (Fig. 1) [10]-[13]. This reduces the power, computation, and actuation demands on the individual modules, making it easier to scale down their size. The components are attracted to a growth substrate and to previously-assembled components to fabricate a target structure. This approach allows for the assembly of the complex shapes that arise out of evolution.

The process we present in this paper designs a target structure to fulfill prescribed functional requirements using an evolutionary algorithm, and then uses a second algorithm to plan the error-free assembly of this structure in a stochastic environment. Structures are evolved for prescribed loading conditions using a frequency-based representation [14]. The goal of the evolutionary process is to minimize the strain energy in the structures while keeping their overall mass near a prescribed value.

The assembly algorithm then plans the assembly of the best evolved structure without knowing the times or locations of component availability. At each stage of assembly this algorithm must determine the next set of locations to attract modules from among those available. This is accomplished by sampling the graph of all possible paths to the target structure and following those that leave the most options open. For each sample, the assembly problem is solved by beginning with the final structure and working backwards, removing one valid module at a time. Thus each sample is a valid path to a perfect assembly. The potential locations for the next assembly stage are then selected from among those most frequently encountered while sampling. Using this approach, at least one path to a perfectly complete final assembly is guaranteed at every stage of assembly.

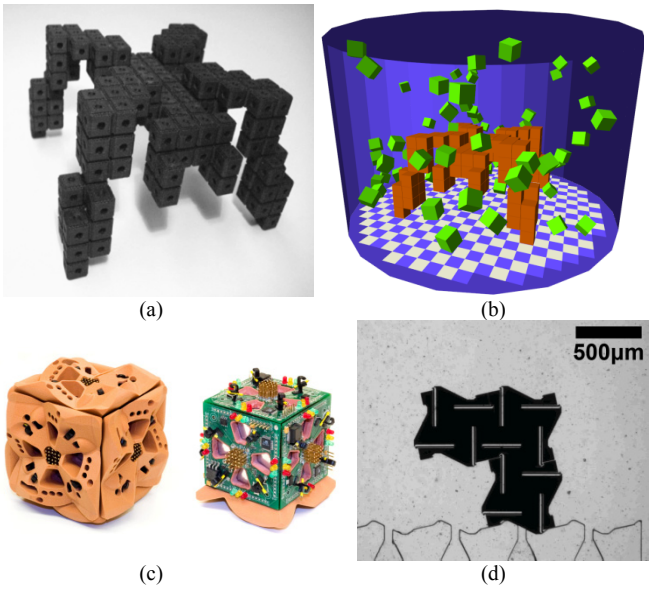


Fig. 1. Stochastic Fluidic Assembly System. Robotic components are assembled by taking advantage of their stochastic motions within a fluidic system for transportation. A target shape (a) is assembled in simulation (b), where free and attached cubes are displayed in green and red, respectively. The assembly substrate is displayed as a checkered floor. (c) Centimeter [15], and (d) micrometer [16] scaled versions of stochastic fluidic assembly experimental modules.

Manuscript received July 27, 2009. This work was funded by DARPA grant number W911NF-08-1-0140 (Programmable Matter Program), and by NSF Creative-IT grant 0757478. M. Tolley would also like to thank the Canadian NSERC for their support through the PGS Program.

M. T. Tolley, J. D. Hiller, and H. Lipson are with the Computational Synthesis Laboratory, Cornell University, Ithaca, NY, 14853 USA (e-mail: mtt33@cornell.edu; jdh74@cornell.edu; hod.lipson@cornell.edu).

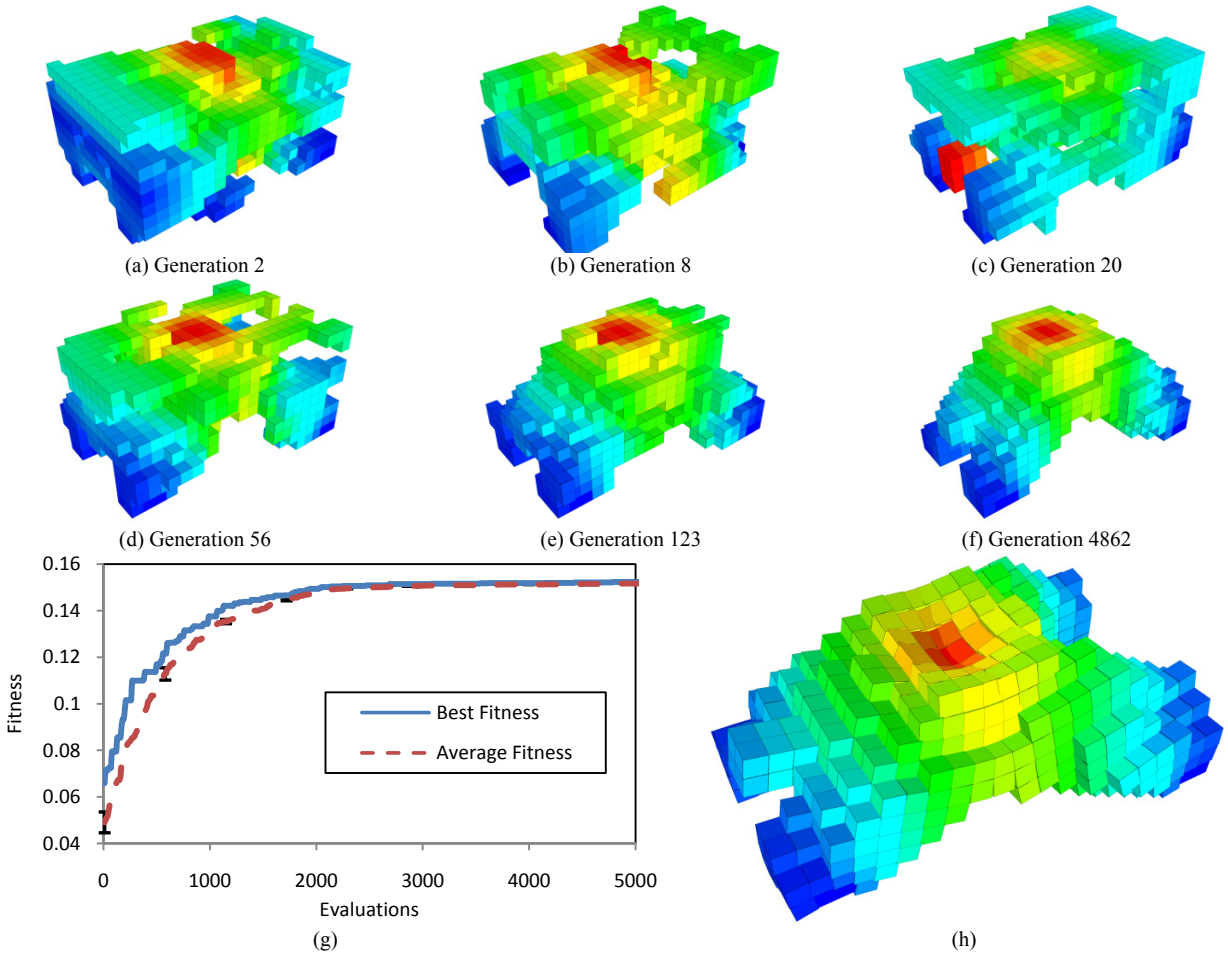


Fig. 2. Digital structure evolution. (a)-(f) Most fit individuals from indicated generations evolved to support a load on top while contacting the ground only at four perimeter locations. The colors indicate the global displacement of each component under the given loading conditions. (g) Plot of the best and average fitness versus the number of evaluations for the first 500 generations. (h) Exaggerated deformation of the final best individual in 5700 generations that was chosen as the target structure for the assembly algorithm.

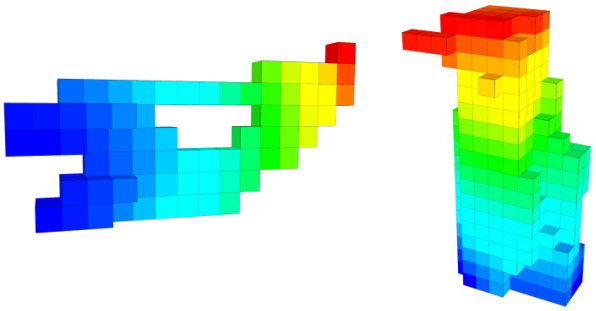
II. Target Structure Evolution

We evolved structures to be assembled from our stochastic modular robots to fulfill a particular task. In this case the task was to support a load on top with ground reaction forces applied only at four locations in the corner of a 20x14x10 cube workspace. The goal was to minimize the strain energy in the structure with 20% of the workspace volume filled. Thus, the resulting structures could be used as a type of customized bridge or a load-bearing legged robot (if the individual modules are capable of some form of actuation).

Smooth, freeform 3D structures were evolved using a frequency-based encoding. The objects were represented as a series of frequency amplitudes at harmonic multiples, and the phenotype was generated by applying an inverse discrete Fourier transform to these values. This approach is similar to that used in image compression, such as the jpeg format. In this case, the frequency representation was chosen for its evolvability, scalability in terms of the number of parameters, and ease of implementation. The fitness of the evolved structures was evaluated by calculating the total strain energy of the structure under the specified loading conditions. The structure's strain energy was calculated using a custom finite element analysis (FEA) solver employing the direct stiffness method.

Using a frequency-based representation, we were able to evolve structures in our 2800-cube workspace using an encoding of only 120 parameters. We used a population of 10 individuals which were selected for crossover using deterministic crowding. Children were produced by selecting each frequency component randomly from one of the two parents. Children were also mutated at a rate of 0.2, with 20% of the frequency components mutated by up to 10% with each mutation. If a generated child was more fit than its most similar parent, the child replaced the parent in the population, otherwise it is discarded. More details on the encoding, evolution parameters, and fitness evaluation can be found in [14].

The results of the structure evolution can be seen in Fig. 2. A representative sample of the best individuals from a number of different generations is shown with colors indicating the displacement of each module under the prescribed loading conditions (dark red and dark blue indicate maximum and minimum displacement, respectively). A plot of the best and average fitnesses vs. the



(a) (b)

Fig. 3. Evolved applications for stochastic modular robots. (a) Custom wrench evolved to withstand the applied forces necessary to manipulate an irregular mechanical component. (b) Custom prosthetic leg replacement evolved to fit prescribed kinematic and force constraints.

total number of evaluations shows the convergence of the GA on a successful structure (Fig. 2 (g)). The deformation of the most fit individual is also shown in Fig. 2 (h).

Even if we restrict ourselves to FEA-solvable loading conditions, many other goal tasks for a stochastic modular robotic system are possible. For example, a shape could be defined that corresponds to a non-standard bolt or other mechanical component that needs to be removed (Fig. 3(a)). A custom wrench shape could then be evolved to withstand loads required to remove the component. Alternatively, the kinematic and loading conditions could be defined for a damaged robot or human limb that needs to be replaced and an optimal replacement shape could be then be evolved to act as a prosthetic (Fig. 3(b)).

III. Stochastic Fluidic Assembly System Model

The structure evolved as described in Section II is to be assembled in a fluidic modular robotic system (Fig. 1). In this system, the target structure is "grown" by adding components first to a planar substrate, then to the exposed surfaces of previously-assembled components. The components are homogenous, and their availability for assembly at any particular location at any point in time is determined by the stochastic motion of the fluid in the assembly chamber. An attraction can be activated by redirecting fluid flow through the internal structure to open a sink at any surface of the structure, which pulls in nearby components until one comes close enough to attach. However, there is a limit to the number of sinks that can be opened simultaneously (due to the increased flow rates required to maintain additional sinks). In addition, the attraction probability is affected by the local and/or global assembled structure (which affects the fluid flow in the chamber), and the number of sinks attracting to a given location. Kinematic constraints – such as the inability to insert a module directly between two opposing modules – also limit the viable assembly paths.

In our model of this system we assume limit of four attractions (sinks) that can be active at one time, with only one attraction possible per grid location. We also assume that the expected cube arrival time is a function of local geometry only. Taking the kinematic constraints into account, there are three possible configurations to which a cube can be attracted (Fig. 5(a)). By running simulations of the stochastic assembly system with a volume full of randomly initialized cubes being attracted to one of these three configurations, we are able to obtain a distribution of the assembly time in each case (Fig. 5 (b)). Fig. 5 (c) shows the distributions of the assembly times in the three separate cases overlaid on the Poisson distributions for the same means, which are assumed to be the correct distributions in sampling assembly times during assembly simulations, as described below.

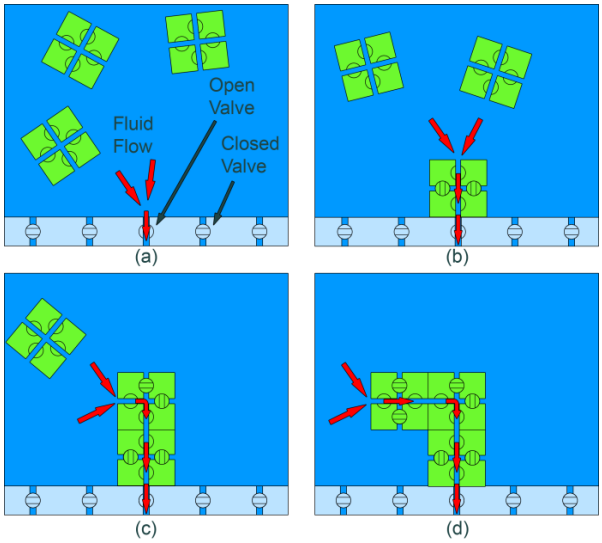


Fig. 4. Stochastic Fluidic Assembly Concept [17]. (a) Fluid flow (indicated by arrows) into a substrate attracts a nearby module moving stochastically in fluid environment. (b) Once attached, the module draws power from the substrate to activate on-board valves and redirect fluid flow through internal channels, (c) attracting new modules at desired locations. (d) This process continues layer-by-layer until the structure is complete.

When multiple sinks are opened simultaneously with different expected cube arrival times (based on the current configuration), we require a method of determining which cube arrives first. This is necessary both as a way of simulating "nature's" turn to decide which of the possible cubes arrives at a given step, and also as a method of selecting cubes in the assembly algorithm.

Drawing inspiration from chemistry, we can sample from the set of potential assembly events using a method developed to sample from the set of simultaneous reactions in a well-mixed stochastic fluidic environment with multiple chemical species [18],[19]. Gillespie's method relies on the following probability density that the next reaction is μ and it occurs at time τ:

$$P(\mu,\tau)d\tau = a_\mu exp\left(-\tau\sum_j a_j\right)d\tau \qquad (1)$$

where the a_μ's are the respective reaction rate coefficients (that can depend on chemical concentrations, environmental conditions, etc.). Integrating $P(\mu,\tau)$ from $\tau = 0$ to ∞ gives the probability that the next reaction to occur is a given reaction μ:

$$P(\mu) = \frac{a_\mu}{\sum_j a_j} \tag{2}$$

Further, summing $P(\mu, \tau)$ over all reactions μ gives the distribution of times for the next reaction to occur:

$$P(\tau)d\tau = \left(\sum_j a_j\right) exp\left(-\tau \sum_j a_j\right) d\tau \tag{3}$$

Integrating this equation and solving for the time τ as a function of a selected random number P gives an equation for randomly sampling the time taken for the next equation to occur:

$$\tau = -\frac{\ln(1-P)}{\sum_j a_j} \tag{4}$$

If we view the various parallel potential interactions that could occur at any point in time as chemical reactions in a well-mixed system, we can apply equations (2) and (4) to determine the time and location of the next cube arrival in our stochastic assembly system. In this case, the rate constants can be determined by inverting the mean times to assembly from Fig. 5(b).

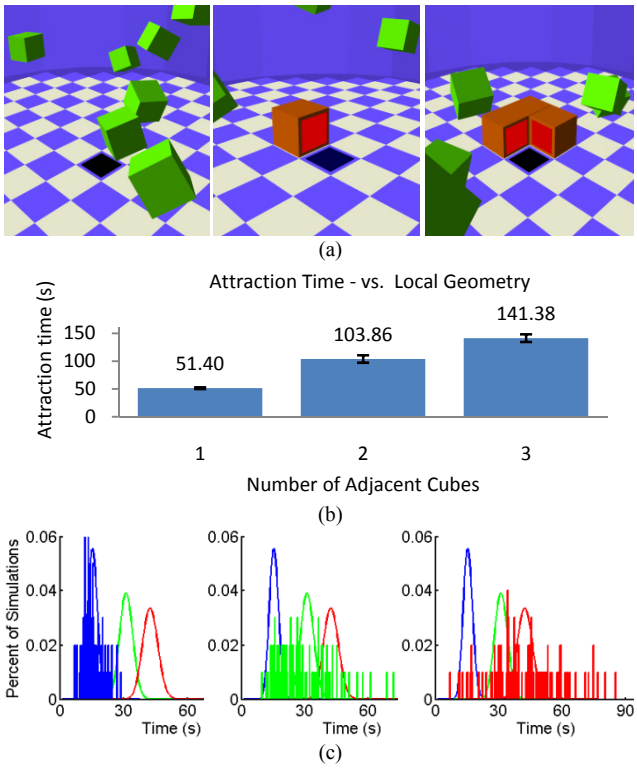


Fig. 5. Sink Configurations. (a) The three possible sink configurations to which a new cube can be attached. Green cubes represent free components floating stochastically in a fluidic environment while assembled cubes are colored red. Dark blue squares represent sinks that attract nearby cubes. (b) Plot of mean time to attraction over 100 simulation of the three configurations. (c) Plots of the distribution of assembly times grouped into one second segments overlaid on a plot of the Poisson distributions for the three mean assembly times.

IV. Assembly Algorithm

The main challenge of the assembly algorithm is to assemble error-free structures in a parallel way. Straightforward approaches, such as attracting components in a greedy manner wherever they are needed, result in porous structures with many unfillable holes ([17], Fig. 6(a)). Serial assembly approaches (e.g. filling one layer at a time) result in perfect structures but take a long time and are susceptible to single location failures. Otherwise, it is difficult to enumerate robust parallel assembly rules that result in a perfect target structure.

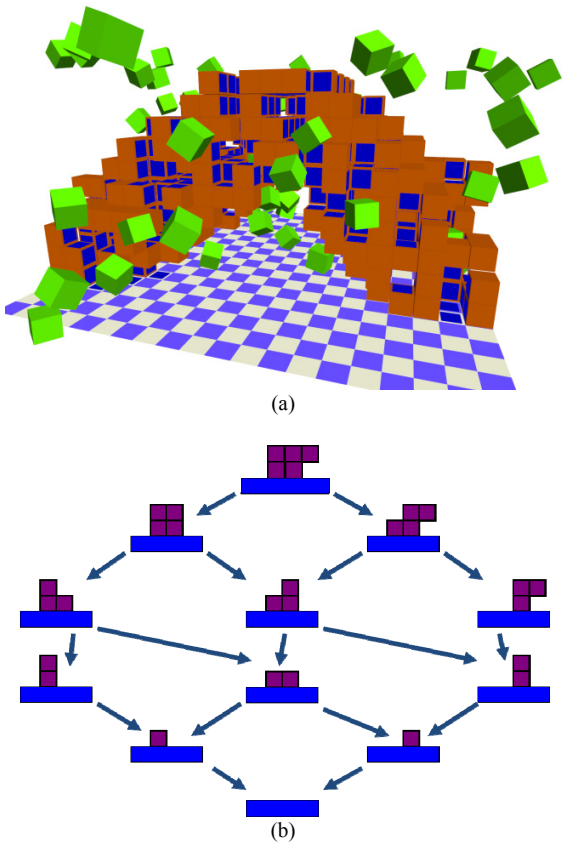


Fig. 6. Assembly by disassembly. (a) Greedy assembly algorithms result in porous structures with unfillable holes (indicated by blue squares). (b) By starting the complete structure and removing one cube at a time, one could potentially create a graph of all possible assembly sequences that result in an error-free structure.

The approach presented here is to look at the problem from the opposite direction: begin with the final structure and remove one accessible cube at a time, keeping track of the sequence. This approach efficiently generates a serial assembly sequence that results in a perfect assembly, starting with any partially-complete structure (assuming such a path exists). In order to generate a parallel assembly sequence, we would ideally want a graph of all of the possible routes to an error-free target structure (Fig. 6(b)). In practice, this graph is prohibitively expensive to compute for all but the simplest structures. However, using the serial disassembly approach, our strategy was to sample enough of the paths through this graph to determine which of the potential attraction locations were both valid (did not result in errors) and popular (many paths passed through this node). The idea was to avoid getting stuck in a part of the graph with a small number of paths to completion since this essentially degrades to serial assembly.

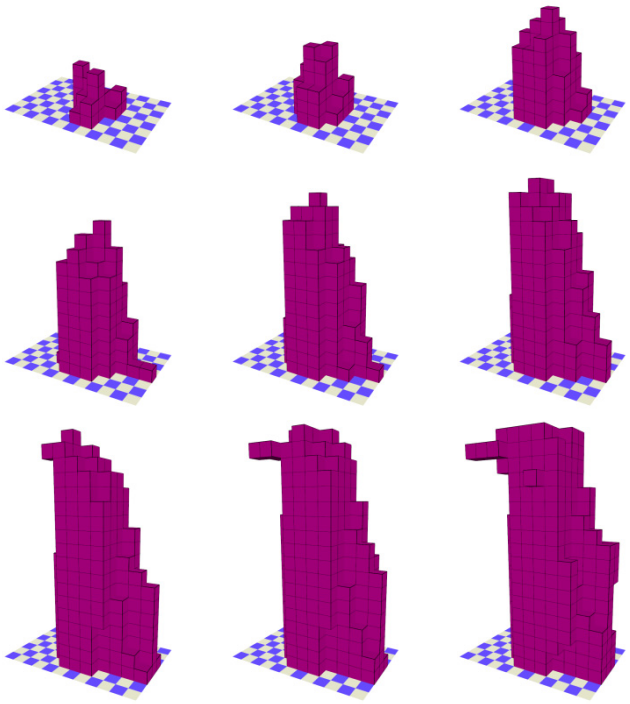


Fig. 7. Assembly Sequence. Sequence of images taken from simulated assembly of robotic prosthetic from Fig. 3(b) using the parallel assembly algorithm.

An example of applying this approach to the assembly of an evolved target structure can be seen in Fig. 7. We evaluated multiple potential selection methods for choosing the next cube to be removed at each step of each disassembly sample (Fig. 8(a)). Most of these relied on obtaining a complete list of all of the accessible cubes at each step, and choosing among them to decide which cube to remove. This selection is made either randomly, favoring the attractions that take the longest (i.e. with three adjacent cubes), or shortest (one adjacent cube) first, greedily selecting the fastest or slowest attractions first, or using Gillespie sampling (Section II) to choose from among the possible cubes. A final selection method was to randomly check cubes for availability and choose the first accessible cube found.

Surprisingly, all of these selection methods found serial assembly sequences with very similar expected assembly times. The best and worst selection methods were the greedy approaches, however neither of these are viable for sampling the overall assembly graph since they will always pick the same path through the graph. Discarding these options, simply choosing the first randomly found accessible cube turned out to be a much less computationally-intensive option since it avoided the necessity of constructing a list of all of the accessible cubes at every step of the disassembly. However, it is interesting to look at the different routes these approaches take to assembly (Fig. 8(c)-(e)). Because single neighbor attraction sites are the fastest, the Fastest First approach first assembles a skeletal structure while Slowest First assembles a solid core of cubes and builds outward. The more random approaches such as Gillespie's method seem to be a hybrid of these two extremes.

We ran the parallel assembly algorithm by taking 20 samples of the overall assembly graph at each stage, and selecting the four most commonly encountered next locations to attract cubes in parallel. Based on the results shown in Fig. 8(a), we used the first randomly found

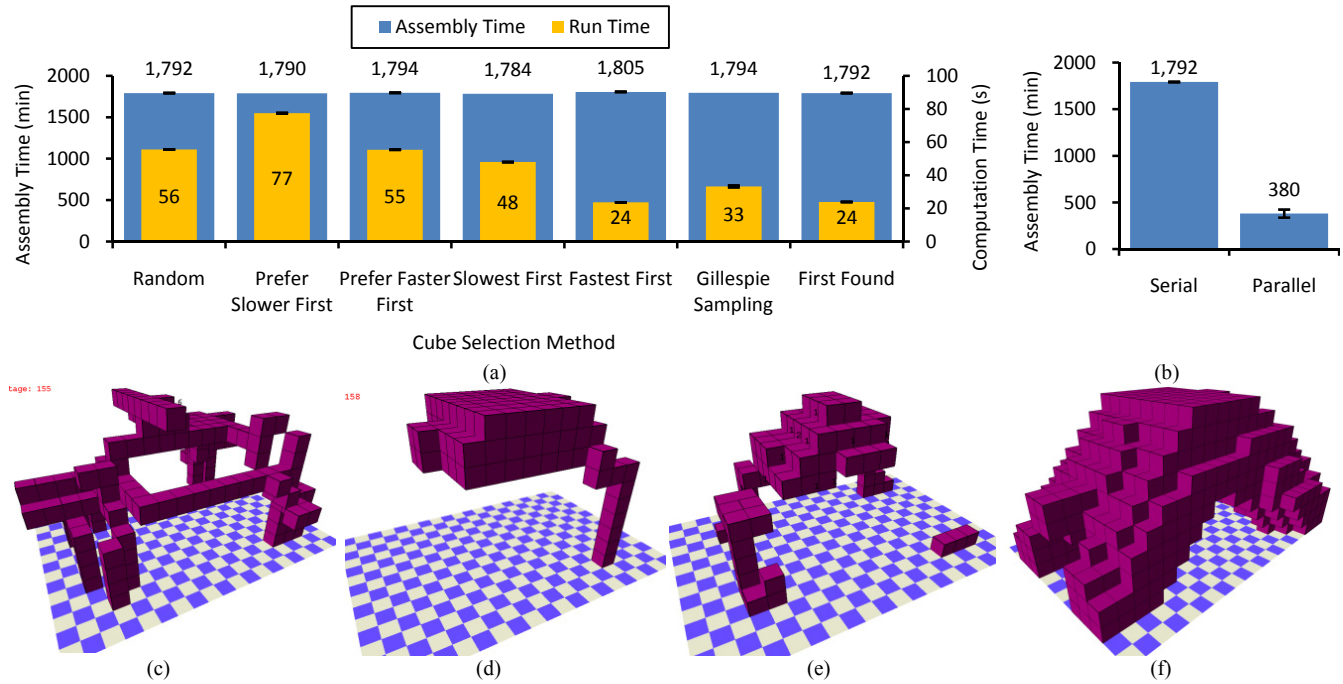


Fig. 8. Effect of cube selection method. (a) Total expected serial assembly time (blue), and computation time (red) averaged over 100 runs, for various selection methods. Error bars indicate standard error. (b) Comparison of average expected assembly time for serial and parallel assembly using the First Found selection method. (c)-(e) Partial structures assembled for Greedy - Slowest First, Greedy - Fastest First, and Gillespie Sampling selection methods (respectively). (e) Completely assembled target structure.

accessible cube as the selection method for each disassembly step of each sample. Using the Gillespie method, we simulated which of the four locations attracted a cube at each assembly stage. The total assembly time for the structure was reduced from 1792 to 380 minutes (Fig. 8(b)), while still maintaining the guarantee of error-free assembly.

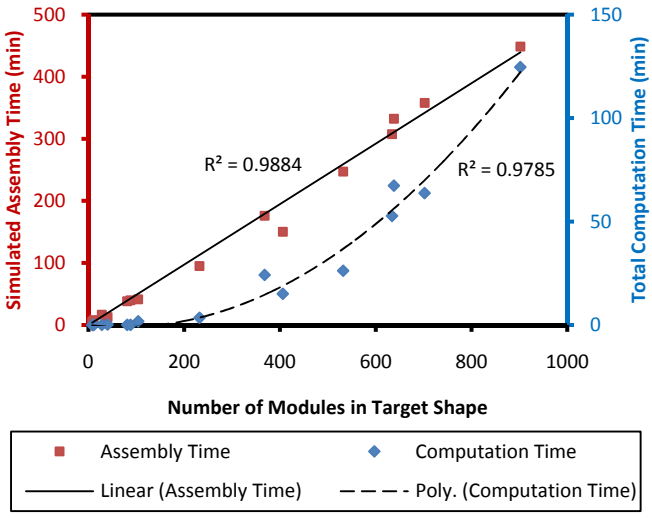


Fig. 9. Assembly algorithm scaling. The time taken to assemble a structure in simulation scales linearly with the number of modules while the total time taken to compute the locations to next attract modules at each stage scales with the square of the number of modules.

Finally, we studied the scaling of the simulated assembly times and the total assembly algorithm computation times as a function of the number of modules in the target structure (Fig. 9). Interestingly, while the predicted structure assembly time scales linearly with the number of modules, the total time required to compute which locations to attract modules to at each stage of assembly increases with the square of the number of modules. Thus, after a certain structure size the assembly rate would be limited by algorithm computation as opposed to physical assembly. However, since the assembly algorithm is re-sampling the same overall assembly graph over and over again, we believe it would be possible to employ a dynamic programming technique to reduce the algorithmic complexity by re-using previous samples.

V. Conclusion

We have presented an automated approach to stochastic modular robotics that, given a target function, evolves a structure to achieve this function, and then directs the structure's assembly. Finite Element Analysis was used to determine the fitness of the evolved structures and a frequency domain representation of the structures simplifies the number of parameters and enhances evolvability. Once a suitable structure was found, its assembly was planned using a sampling approach to map out a path through the graph of all possible assembly sequences that minimizes assembly time. Each sample of this graph was obtained by starting with the target structure and peeling away one accessible module at a time until the initial structure is revealed. This simple algorithm guarantees a valid path to assembly of the target structure that does not leave behind unfillable holes.

While our approach follows the traditional paradigm of first designing, then determining how to assemble a structure, the automated nature of the entire process allows for the possibility of feedback between the two parts. Not only could assembly simulations factor in to the evolution's fitness function, but the design itself could in fact be tweaked through evolution in response to current assembly conditions.

References

[1] H. Lipson and J. B. Pollack, "Automatic design and Manufacture of Robotic Lifeforms," *Nature*, vol. 406, pp. 974-978, Aug. 2000.

[2] G. Hornby, "Functional scalability through generative representations: the evolution of table designs," *Environment and Planning B-Planning & Design*, vol. 31, pp. 569-587, 2004.

[3] E. Malone and H. Lipson, "Freeform Fabrication of Ionomeric Polymer-Metal Composite Actuators," *Rapid Prototyping Journal*, Vol. 12, No. 5, pp. 244-253, 2006.

[4] A. Piqué, C. B. Arnold, H. Kim, M. Ollinger, and T.E. Sutto, "Rapid prototyping of micropower sources by laser direct-write," *Applied Physics A: Materials Science & Processing*, Vol. 79, No. 4, pp. 783-786, Jul. 2004.

[5] E. Malone, M. Berry, and H. Lipson, "Freeform Fabrication and Characterization of Zinc-Air Batteries," *Rapid Prototyping Journal*, Vol 14, pp. 128-140, 2008.

[6] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirickjian, "Modular Self-reconfigurable robotic systems," *IEEE Robotics and Automation Magazine*, Vol. 14, pp. 43-52, Mar. 2007.

[7] E. Klavins, "Programmable Self-Assembly," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 43-56, Aug. 2007.

[8] S. C. Goldstein, J. D. Campbell, and T.C. Mowry, "Programmable matter," *IEEE Computer*, vol. 28, no. 6, pp. 99–101, May 2005.

[9] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu. "Miche: Modular Shape Formation by Self-Disassembly," *Int. J. Robotics Research*, vol. 27, pp. 345-372, Mar. 2008.

[10] M. T. Tolley, M. Kalontarov, J. Neubert, D. Erickson, and H. Lipson, "Stochastic Modular Robotic Systems: A Study of Fluidic Assembly Strategies," *IEEE T. Robotics,* submitted for publication.

[11] P. J. White, K. Kopanski, and H. Lipson, "Stochastic Self-Reconfigurable Cellular Robotics," in *2004 Proc. Robotics and Automation (ICRA)*, pp. 2888-2893.

[12] P. J. White, V. Zykov, J. Bongard, and H. Lipson "Three Dimensional Stochastic Reconfiguration of Modular Robots," in *2005 Proc. Robotics Science and Systems*.

[13] M. T. Tolley, M. Krishnan, H. Lipson, and D. Erickson, "Advances Towards Programmable Matter," in *2008 Proc. Miniaturized Systems for Chemistry and Life Sciences (MicroTAS)* pp. 653-655.

[14] J. Hiller and H. Lipson, "Multi Material Topological Optimization of Structures and Mechanism," in *2009 Proc. Genetic and Evolutionary Computation Conference*, pp. 1521-1528.

[15] V. Zykov and H. Lipson, "Experiment Design for Stochastic Three-Dimensional Reconfiguration of Modular Robots," presented at the Int. Conf. Intelligent Robots and Systems, Self-Reconfigurable Robotics Workshop, Atlanta, GA, June 27-30, 2007.

[16] M. T. Tolley, M. Krishnan, D. Erickson, and H. Lipson, "Dynamically Programmable Fluidic Assembly," *Appl. Phys. Lett.*, vol. 93, 254105, Dec. 2008.

[17] M. T. Tolley, M. Kalontarov, J. Neubert, D. Erickson, and H. Lipson, "Stochastic Modular Robotic Systems: A Study of Fluidic Assembly Strategies", *in review*.

[18] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Physical Chemistry*, vol. 81, pp. 2340–2361, Dec. 1977.

[19] M. A. Gibson and J. Bruck, "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels," *Physical Chemistry A*, vol. 104, pp 1876–1889, Feb. 2000.