

Research Statement

Ho-Lin Chen

My primary research interest lies in understanding how to control complicated collective behaviors via simple rules in application to the fields of molecular programming, algorithmic game theory, robotics and sensor networks. One does not need to look far to find systems that behave in this way: colonies of bees that construct a hive cooperatively, ants that use each others bodies as bridges to reach a food source, and man-made communal systems that are built by selfish individuals (like the Internet). The general problem is to control a large number of individual components that possess limited local information to achieve a specific task cooperatively. An understanding of how to program such behavior may enable the construction of nanostructures and machines, efficient networks, reliable robotic systems and so on. What kind of tasks can these systems perform that are not currently achievable? How robust or efficient can these systems be?

One major difficulty in providing a general solution to the above questions is that rules specified by different applications may be very different from each other. For example, in molecular programming, we are restricted by the amount of information or functions that can be coded into each molecule. Therefore, the control rules must be very simple and rely only on local information. In algorithmic game theory, on the other hand, we are allowed to specify complicated rules but may want to satisfy some given design criteria such as stability or budget-balance to ensure that the system does not have unwanted failures. Furthermore, even if we just consider one particular application, the power of control rules can change drastically even with slight modifications [12, 5]. Therefore, most prior work only focused on one specific set of rules and one must start from scratch every time when analyzing or designing a new system.

The goal of my research is to provide easy-to-verify characterizations for collaborative schemes that are robust, efficient or have some other desired property and to use these characterizations to guide the construction of such systems. Previously, I have established robustness criteria that guarantee the target behavior of molecular systems. I have proven the validity of existing self assembly designs and designed and built new molecular systems that satisfy these criteria [10, 7, 8, 15]. In network formation games, I have fully characterized the set of possible cost-sharing schemes and identified the schemes with the best network efficiency guarantees [13]. My work demonstrates that these characterization theorems can provide more understanding about existing systems that lead to better designs.

Algorithmic self-assembly – Design robust system with error prone units

Self-assembly is the process by which small objects assemble into large, complex structures by themselves using simple local primitives. Although this is a prevalent process in nature, people have not yet been able to exploit its full power in molecular engineering. Currently, external control of each component is required to solve most engineering tasks. However, as scale decreases, precise control of structures by applying external signals becomes increasingly difficult, and thus self-assembly is a very appealing way to control molecules at the nano scale. Algorithmic self-assembly has been proposed as a general method for directing self-assembly. In this framework, we are allowed to specify arbitrary binding

strengths between each component. This freedom of specification leads to a lot of interesting and useful designs [18, 31, 29, 28, 1, 20, 2, 6, 17, 8, 22, 26, 27, 16] and therefore has received significant attention over the last decade. Several self-assembly systems designed under this framework have been implemented in practice [30, 19, 21, 23, 14, 4, 32] establishing the promise of this approach.

Although many powerful self-assembly systems have been proposed, only small systems have been successfully built in practice. One major problem of self-assembly is its high error rate. However, it is hard to apply previous error correction techniques to self-assembly systems because there is no centralized computational unit. Each component must contain the information to locally detect and correct errors. We have designed an error correction scheme called the snaked proofreading tile system [8]. In this construction, we design groups of redundant tiles that must be paired with each other. Whenever an error happens, the process will halt locally until the erroneous tiles fall off. Given any self-assembly system, we can produce an error-free self-assembly system that performs the same function with only a poly-logarithmic increase on assembly time and the number of tiles required in the original system. This is the first provable solution to error correction problem for self-assembly. Experimentally, we also implemented a small example of the snaked proofreading tile system using DNA tiles [15]. Statistics indicate that there is a four-fold improvement in error rates under experimental conditions. This demonstration indicates that the methods designed theoretically can improve the dynamics of self-assembly systems in the direction predicted.

To make efficient error correction systems, we need to answer questions like “Is this specific system robust to errors?” or “What is the most efficient error correction technique for this system?” However, predicting the behavior of a given system requires modeling the whole system as a complicated Markov process. To prove the correctness of a given error correction system, we need to show that the corresponding Markov process will converge to a small set of states temporarily. Slight changes in the system can change the corresponding Markov process drastically. Therefore it is unclear how to generalize the proof of specific error correction schemes to a general set of criteria for robustness systems. Recently, we proved that a set of combinatorial criteria is sufficient to guarantee the robustness of self-assembly systems [9] and use this to prove that several previously designed systems are robust. Every system satisfying those criteria can self-assemble correctly with arbitrarily high probability. We also showed that there is a trade off between the complexity of the system and the time required for the system to assemble. This characterization allows us to prove the validity of error correction designs without requiring knowledge of the underlying Markov process and can lead to methods of determining whether a given system is robust or not.

Programming molecular systems – Learn from nature

There are many spectacular examples of collective behaviors in nature, ranging from the coordination of cells in a body to colonies of ants and bees to the whole ecological system. Many researchers study specific instances of a behavior without knowing if the mechanism is essential to the desired behavior or if it can be generalized. We are interested in identifying the key features that enables nature to achieve spectacular tasks. At molecular level, the set of behaviors we observe are assembly, disassembly, chemical reactions and motion. It has been theoretically proposed [24, 25] that for a given set of arbitrary chemical reactions, we can construct a set of DNA molecules that behaves according to these reactions. We also know that we can construct a large variety of self-assembly systems and molecular machines. However, natural systems are often far more efficient and robust than artificial ones. Identifying the key features in nature will both aid in engineering molecular systems and give us more insight into how nature work.

Two particular amazing phenomena in assembling structures are self-recovery from small pieces (e.g. sea stars) and exponential growth (e.g. growth from embryo). We first showed that self-recovery can be simulated by artificial tile self-assembly systems. We constructed “self-healing” tile systems that can assemble

arbitrary shapes [9]. The shape can be regenerated from any fragment that is logarithmic in the size of the desired assembly. Exponential growth, on the other hand, can not be simulated by the tile assembly model. In this model, we assume tiles can only attach one by one to the existing lattice structure. Therefore, the time required to construct any shape scales linearly with the size of the shape. However, if we allow the tiles to make relative motions after they assemble, then we can build arbitrary shapes in poly-logarithmic time. On the study of robustness, in addition to robust self-assembly, we also studied error correction schemes for chemical reaction networks [11] and molecular machines [7]. In those two works, we constructed robust and efficient systems that can do a very general set of computational tasks. These works help us understand more about several major molecular mechanisms in nature and provide good techniques for building more sophisticated artificial molecular systems.

Algorithmic game theory – Make selfish agents perform good deeds

Nowadays, many networks, including the internet, are created and used by many agents with a wide range of objectives. These networks can be very different from the centrally optimized ones that have received much more attention in previous studies. In these networks, we have to allocate scarce resources to selfish users and assign costs accordingly. Different methods of resource and cost allocation can create very different behaviors. People are interested in knowing how different allocation schemes can affect the efficiency of the network. Many prior works studied one specific resource allocation method and how selfish behavior impacts the efficiency of the system [3, 12]. We would like to identify the most efficient scheme when agents are behaving selfishly to construct and operate such networks in the future.

One major difficulty in many systems is that these schemes need to possess some good properties such as budget-balance and stability to prevent system failure. Therefore, it is hard to even determine all possible schemes we can choose from. We studied the network formation game which arises from several agents constructing a network cooperatively. In this model, we first tried to extend a previously known scheme but discovered that even the slightest generalization can make the system become unstable. Later, we were able to give a characterization of all possible cost-sharing schemes that guarantees the stability of the system [13]. We also showed that each of the possible schemes must have a potential function associated with it. The existence of potential function provides us many useful properties and allows us to describe all possible schemes easily. Using these properties, we can identify schemes that minimizes the inefficiencies caused by selfishness under different circumstances. The counting argument we presented in proving the above results can potentially become useful in other models as well.

Future research

Programming many diverse components using simple rules is a topic of interest to many different fields. Currently, when a particular system is studied, first thing is to make models that can be generalized, and then study what kind of tasks this system can do. There are a few drawbacks to this approach. First, we might not be able to achieve the original tasks that we were seeking to accomplish since we start with a random set of rules. Second, most of the previous work can not be applied if the set of rules is changed. The goal of my research is to characterize the set of rules / systems that can achieve a given task with performance guarantees and use these characterizations to speed up and automate the design process. Eventually, I believe my results will change the way people analyze and construct these types of systems.

In the short term, my plan is to continue my research on each of the fields mentioned above. There are many interesting and important open problems in each of these fields. Here are a few problems that I plan to study:

Is there a better characterization of robust molecular systems?

In real applications, we often want to find the simplest system that does a certain task robustly. To find these systems, we need a simple robustness verification method. Given any molecular system, I would like to easily determine whether it is robust and be able to automate this verification process. Previously, we have established a set of criteria sufficient to show the robustness of some systems in several different molecular models. Although we have not shown that all robust system must satisfy these criteria, we have evidence that each of these criteria is necessary for the robustness of the system. Also, the verification process is deterministic and does not have to deal with the underlying complicated Markov process. Therefore, it should allow us to easily automate the verification process.

Does combining different rules give us more power?

Previously, we studied models generated by existing synthesized molecular systems. In each of these systems, we are often limited to one set of control rules. However, in order to create more complicated systems, models will inevitably require several types of rules. Before trying to construct such systems, we would like to compare the power that different combinations of rule sets. Recently, we studied the case of active self-assembly (in which molecules can make relative motions after assembling) and discovered that even just adding simple motion rules can make self-assembly much more powerful. I am planning to extend this work by studying other important combinations of rulesets.

How does the rate of molecular movement affect the behavior of molecular systems?

In previous studies on molecular systems, we often assume that the system is well mixed, meaning that the probabilities for any two molecules to collide with each other are the same. In practice, the movement of molecules rely on diffusion and are thus more likely to react with molecules that are nearby. I am interested in studying how much this phenomenon can affect the behavior of molecular systems and their ability to perform computation and error correction. More generally, I would like to know how much communication between different components is required for a system to perform specific tasks.

When can we guarantee the existence of a potential function?

In network formation games, we showed that all possible cost-sharing schemes have natural potential functions associated with them. The existence of potential function gives us many useful properties and allows us to identify the most efficient scheme. We would like to generalize this result to a wider class of models and determine the minimal set of restrictions that will give us similar characterization results. We have some preliminary ideas on generalizing this result to distributed welfare games and we have found several interesting features that might be critical to the existence of potential function.

References

- [1] L. Adleman. Towards a mathematical theory of self-assembly. Technical Report 00-722, Department of Computer Science, University of Southern California, 2000.
- [2] L. Adleman, Q. Cheng, A. Goel, M.-D. Huang, D. Kempe, P. Moisset de Espans, and P. Rothmund. Combinatorial optimization problems in self-assembly. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 23–32. ACM Press, 2002.

- [3] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *Annual IEEE Symposium on Foundations of Computer Science*, 2004.
- [4] R. D. Barish, P. W. K. Rothemund, and E. Winfree. Two computational primitives for algorithmic self-assembly: Copying and counting. *Nano Lett.*, 5:2586–2592, 2005.
- [5] L. Cardelli and G. Zavattaro. On the computational power of biochemistry. In *Proceedings of the third International Conference on Algebraic Biology*. Austria, July 2008.
- [6] H.-L. Chen, Q. Cheng, A. Goel, M.-D. Huang, and P. Moisset de Espans. Inadable self-assembly: Combining robustness with efficiency. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 883–892, 2004.
- [7] H.-L. Chen, A. De, and A. Goel. Towards programmable molecular machines, 2008.
- [8] H.-L. Chen and A. Goel. Error free self-assembly using error prone tiles. In *Proceedings of the Tenth International Meeting on DNA Based Computers*. Milan, Italy, June 2004.
- [9] H.-L. Chen, A. Goel, C. Luhrs, and E. Winfree. Self-assembling tile systems that heal from small fragments. In *Proceedings of the Thirteenth International Meeting on DNA Based Computers*. Memphis, TN, June 2007.
- [10] H.-L. Chen, C. Luhrs, and A. Goel. Dimension augmentation and combinatorial criteria for efficient error-resistant DNA self-assembly. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2008.
- [11] H.-L. Chen, G. Mendoza, D. Soloveichik, and E. Winfree. Fault-tolerance in asynchronous circuits with applications to chemical reaction networks. *in preparation*.
- [12] H.-L. Chen and T. Roughgarden. Network design with weighted players. *Proceedings of the Eighteenth ACM Symposium on Parallelism in Algorithms and Architectures*, 2006.
- [13] H.-L. Chen, T. Roughgarden, and G. Valiant. Design networks with good equilibria. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2008.
- [14] H.-L. Chen, R. Schulman, A. Goel, and E. Winfree. Error correction for DNA self-assembly: Preventing facet nucleation. In *Proceedings of the Eleventh International Meeting on DNA Computing*. London, Ontario, Canada, June 2005.
- [15] H.-L. Chen, R. Schulman, A. Goel, and E. Winfree. Error correction for DNA self-assembly: Preventing facet nucleation. In *NanoLetters*, 2007.
- [16] P. M. de Espanes and A. Goel. Toward minimum size self-assembled counters. In *Proceedings of the Thirteenth International Meeting on DNA Based Computers*. Memphis, TN, June 2007.
- [17] S. Sahu J. Reif and P. Yin. Compact error-resilient computational DNA tiling assemblies. In *Proceedings of the Tenth International Meeting on DNA Based Computers*. Milan, Italy, June 2004.
- [18] M. Lagoudakis and T. LaBean. 2D DNA self-assembly for satisfiability. In *Proceedings of the 5th DIMACS Workshop on DNA Based Computers in DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 54. MIT: Cambridge, 1999.

- [19] P. Rothemund. *Theory and Experiments in Algorithmic Self-Assembly*. PhD thesis, University of Southern California, 2001.
- [20] P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468. ACM Press, 2000.
- [21] P. W. K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLOS Biology*, 2:424–436, 2004.
- [22] R. Schulman and E. Winfree. Programmable control of nucleation for algorithmic self-assembly. In *Proceedings of the Tenth International Meeting on DNA Based Computers*, pages 319–328. Milan, Italy, June 2004.
- [23] R. Schulman and E. Winfree. Self-replication and evolution of DNA crystals. In *Advances in Artificial Life, 8th European Conference*, volume 3630, Berlin Heidelberg, 2005. Springer-Verlag.
- [24] D. Soloveichik, M. Cook, E. Winfree, and S. Bruck. Computation with finite stochastic chemical reaction networks, 2008.
- [25] D. Soloveichik, G. Seelig, and E. Winfree. Dna as a universal substrate for chemical kinetics, 2008.
- [26] D. Soloveichik and E. Winfree. Complexity of compact proofreading for self-assembled patterns. In *Proceedings of the Eleventh International Meeting on DNA Computing*. London, Ontario, Canada, June 2005.
- [27] D. Soloveichik and E. Winfree. Complexity of self-assembled shapes. *SIAM Journal on Computation*, 36:1544–1569, 2007. Extended abstract in LNCS 3384: 344–354 (2004); preprint is cs.CC/0412096 on arXiv.org.
- [28] E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, Pasadena, 1998.
- [29] E. Winfree and R. Bekbolatov. Proofreading tile sets: Error correction for algorithmic self-assembly. In *Proceedings of the Ninth International Meeting on DNA Based Computers*. Madison, Wisconsin, June 2003.
- [30] E. Winfree, F. Liu, L. Wenzler, and N. Seeman. Design and self-assembly of two-dimensional DNA crystals, 6 pages. *Nature*, (394):539–544, Aug 1998.
- [31] E. Winfree, X. Yang, and N. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In *Proceedings of the Second Annual Meeting on DNA Based Computers*. Princeton University, June 1996.
- [32] P. Yin, R. Hariadi, S. Sahu, H. Choi, S. H. Park, B. Walters, T. Labean, and J. Reif. Single strand DNA tiles and molecular tubes with precisely programmable circumferences, 2007.