

Interdisciplinary Undergraduate Research in DNA Computing

Lydia Sinapova
Computer Science Department
Simpson College
Indianola, IA, 50125
lydia.sinapova@simpson.edu

Derek Lyons
Department of Chemistry and Physics
Simpson College
Indianola, IA, 50125
derek.lyons@simpson.edu

Abstract

This paper describes the authors' experience in supervising interdisciplinary, undergraduate research in DNA computing. Our interdisciplinary research projects were implemented by allowing the Computer Science students to begin their projects in small groups under a term assignment in the sophomore-level Algorithms course. The projects are co-advised by the Computer Science and Chemistry faculty. The students have developed and implemented several algorithms that support the design, optimization, and performance of DNA strands for DNA origami construction. Chemistry students perform experiments with DNA strands, utilizing the programs developed by the Computer Science students. This paper describes the structure and organization of the projects undertaken by the Computer Science students, as well as the challenges faced by the students and their advisors when working on the interdisciplinary research projects.

1 Introduction

This paper describes the authors' experience in supervising interdisciplinary, undergraduate research in DNA computing. The value of providing students with undergraduate research experiences is widely acknowledged. The Council on Undergraduate Research has numerous publications on various aspects of undergraduate research [3]. The issue is not whether we should involve students in undergraduate research, but rather how to choose the appropriate format, how to select topics, how to ensure administrative and financial support of the faculty that supervise undergraduate research projects, and how to use the research experiences to equip students with skills and knowledge that will be useful in their endeavors beyond college.

Simpson College is a small liberal arts college with about 1,300 students. It does not offer graduate programs in science and mathematics. Since 2011, Simpson College has changed its curriculum unifying the number of courses required per major: 10 – 11 courses. This limits the exposure of our students to advanced high-level courses. We strongly rely on undergraduate research to enhance the quality of the learning experiences offered to our students. The Computer Science curriculum at Simpson provides undergraduate research experiences for students primarily as part of the course work in our CMSC 250 “Design and Analysis of Algorithms” course, and in our capstone classes. This paper discusses the research projects in CMSC 250 course. CMSC 250 is a required course for all computer science majors. It is usually taken by students in their sophomore year. Prerequisites are “Intro to programming”, “Discrete Mathematics” and “Data Structures and OOP”. It is a prerequisite for our “Introduction to Artificial Intelligence” and “Theory of Computation” courses. The research assignments in CMSC 250 have evolved throughout the years from type “survey” to open-ended research problems. For many years students were asked to do review/survey papers. Students had to examine and describe a class of algorithms/problems chosen from a list provided by the instructor. Examples are: “Radix sorting algorithms”, “Algorithms for maze generation”, “Huffman code”. While the research project assignments met their goal to expose students to topics not included in the course material, they did not involve problem-solving and programming. We found the lack of problem-solving to weaken the research projects, and since 2003 students have been assigned projects where they have to solve a significant problem and implement an algorithm. Most of those assignments involved solving an NP-complete problem, such as the knapsack problem, the bin packing problem, or problems for which there were no exact algorithms. Many projects were based on using genetic algorithms for finding approximate solutions. Overall the projects were successful and several were published in the MICS proceedings [1, 4]. However with a few exceptions, the projects were individual work, and thus students did not have teamwork experiences. Moreover, the projects lacked interdisciplinary and collaborative aspects. This changed in 2011, with assigning open-ended projects in the domain of DNA computing in collaboration with the Chemistry department at Simpson. The projects, the methods of supervising, and the outcomes are described in the next sections.

2 DNA Nanotechnology at Simpson College

DNA computing, known also as “molecular programming”, is a very young interdisciplinary area of research, bringing together scientists with expertise in Computer Science, Chemistry, Biology, and Physics. DNA computing utilizes the dynamic interactions of biological macromolecules like DNA and proteins to implement logic functions. DNA computing is a computational paradigm where the information is represented through DNA molecular structures and the processing is based on the chemical properties of the DNA molecules. A computing device has a program encoded in its memory. It receives input fed to that program, and executes operations over the input as prescribed in the program. The program transforms the input data to some output, delivered by the computing device. Very similar processes take place in every living cell. DNA encodes information using a 4-letter alphabet. This is the program that controls the life of each cell. It receives input from its environment and conducts the production of the elements necessary for the cell to live. The basic idea here is that DNA can store information which can be manipulated by applying simple operations (copying, splicing, etc.) Because of its simple, sequence-dependent self-assembly rules, DNA allows for predictable behavior of a dynamic system. However, the complexity of DNA computing is due to the requirement of a dynamic system with multiple DNA strands interacting together. This complexity of scale phenomenon can be overcome by an algorithm-based approach that is validated by physical experiments.

We started our work in DNA computing in 2010 in collaboration with the Laboratory for Molecular Programming at Iowa State University. In 2011 we received an NSF grant CCF-1143839 “Modeling and Analysis of Molecular Programming and Nanoscale Self-Assembly”. The role of Simpson College in the project was primarily educational. The main goal from Simpson's perspective was to create relevant learning resources and expose our undergraduate students to this exciting field. Four faculty members and six students with the departments of Chemistry, Biology and Computer Science participated in the project. We supervised three undergraduate research projects, one for computer science majors, and two for chemistry majors. After the grant expired, we continued to involve our students in research projects on DNA computing. The projects were carried on in three formats: as part of the course work in our CMSC 250 Algorithms course for computer science majors, as independent research for chemistry and biochemistry majors, and as research topics in our capstone courses. This paper focuses on the projects conducted in CMSC 250.

3 Overview of Students' Projects on DNA Computing

The purpose of the research projects assigned in CMSC 250 course is to extend students' knowledge beyond the scope of the material discussed in class. Students gain experience in exploring new topics and learning about new methods and approaches in algorithm design. The work involves researching resources, studying some new theory, and

developing a program that illustrates a specific approach or a method to solve problems within the scope of the chosen topic.

In the Fall of 2011, five students worked on a project in the area of tile self-assembly. Four of those students were enrolled in CMSC 250, and one student joined the team voluntarily. Tile self-assembly refers to combining small structures of DNA strands into large 2D and 3D lattices. The research question was “given a target shape, what is the tile set that will self-assemble in that shape?” Students designed and implemented a genetic algorithm to evolve a random set of tiles into a set with the desired properties. The project was presented at MICS’2012 ([2]) and at NCUR’2012.

While the project conducted in 2011 exposed students to interdisciplinary research, it did not involve collaboration with professionals in other disciplines. To enhance the quality of the research experiences in 2013 and 2014, the projects were set up so that the students had to interact closely with a Chemistry professor to complete their projects.

In 2013 we conducted three projects related to the design of DNA origami. The projects were intended to produce results that can be used in the Chemistry lab where the DNA origami were being constructed. The first two projects, “A Genetic Algorithm for Evolving DNA Sequences” and “Calculating the Melting Temperature of Linker DNA” had the goal to design a DNA single stranded sequence with specified melting temperature. The first project used a genetic algorithm. Multiple tests were ran to determine optimal parameters for the algorithm, including the length of the sequence, percentage of mutations, percentage of tournament luck, and the effect of the roulette wheel and tournament selection methods. The second one used a mathematical model. The mathematical model worked better than the genetic algorithm. When the specified length of the sequence is small, e.g. 12 nucleotides, the best results given by the genetic algorithm had about 20% deviation from the specified melting temperature. The mathematical model also did not work well on short sequences. The third project “Designing Nanostructures with DNA Sequences” had the task to determine whether within a set of DNA sequences there are matching pairs. If such pairs were found, they had to be modified so that the match is eliminated. It used string matching algorithms and hashing techniques to search for matching pairs.

In 2014, students enrolled in CMSC 250 worked on three related projects, all in support of the Chemistry research on DNA origami. The first project was titled “Algorithmic Elimination of Unwanted DNA Hybridization in Complex DNA Mixtures”. Single strands of DNA bond together based on the simple Watson-Crick basepairing rules to make a double-stranded DNA. An optimally designed DNA sequence for assembly of DNA nanostructures matches exactly to a complementary DNA strand, forming very strong bonds. Importantly, DNA nanostructures can contain hundreds of DNA strands, and because the Watson-Crick basepairing rules use only four unique bases, A, T, C, and G, many partial matches can occur between many DNA strands when assembling into a large DNA nanostructure. Partial matches bind weaker than complete matches. Nevertheless, some partial matches may be strong enough to occur within the nanostructure, potentially leading to a failure of the nanostructure to form correctly. The

students tackled the problem of identifying DNA sequences that can form partial matches with strong enough bonding to interfere with the construction of the DNA nanostructures. Their program was built around the nearest-neighbor algorithm, which computes the bonding strength of a pair of sequences that partially match. The user is notified if a partial match with bonding strength above a user defined threshold is found. Based on this information, the user would alter the pool of DNA sequences until no partial matches were found.

The title of the second project was “Modeling of Linker Stoichiometry for Optimization of DNA Nanostructure Self-Assembly”. The result of this project was directly used to optimize the origami construction. In our Chemistry lab, complex origami constructs are built out of small triangular shapes. The triangular shapes have short sequences of nucleotides extending from the edge, called toeholds. A linker strand of nucleotides is designed to complement the toehold sequences from two different triangles. By binding to both of the toeholds, two triangles can be held together. The problem to be solved was to find the ratio of linkers to triangles so that the majority of the triangles bind in the intended configuration. Students used a mathematical model of the DNA solution kinetics and implemented a program incorporating the mathematical model.

The third project was titled “Efficient Path Generation to Maximize Data Collection of Multiple Samples using Fluorescence Spectroscopy”. The task was to write a program that controls the operation of a mechanical device used in creating DNA nanostructures. The device has reagent dispensers and optical readers attached to an arm that moves over vials with DNA solution. For this project, the team spent a lot of time in discussing the problem to be solved. They had to understand very well how the arm moves, and how the DNA solutions identifiable by the emitted color, are placed in the vials. There was no directly available algorithm to be implemented. A lot of time was spent in discussing possible paths of the arm over the vials and ways to evaluate the efficiency of the moves. The difficulty of this problem was due to having two parameters - the specific placement of the DNA solutions in the vials, and then for a given placement - the specific path of the arm over the vials. The current implemented algorithm is based on common sense heuristics and might not be the optimal solution for the problem.

4 Pedagogy

The problems in all six projects described above were real-life problems faced by the researchers in the Chemistry department. The problems were presented to the students by the Chemistry faculty at the beginning of the semester. Students had the option to choose a project to work on, and that was how the teams were formed. Weekly meeting times were scheduled - one meeting with the faculty from Computer Science and Chemistry departments of 30 min - 60 min, and one or two meetings to work on the project outside the classroom. Students were required to take notes at each meeting. These notes were used to keep track of the work progress. During the second half of the semester, based on need, students were given class time to work on the projects. The meetings with the instructors were used mostly to clarify the problem and to discuss possible algorithms to

be used. Students used textbooks when the chosen algorithms were known and described in a textbook. This was the case with three of the projects that used genetic algorithms and string matching algorithms. Some teams were given primary sources - as was the case with the team working on tile self-assembly, and the teams that computed melting temperature.

The work on the projects was done in three major phases:

- Understanding the problem
- Search for an algorithm
- Algorithm implementation

For all projects without exception, the first weeks of project work were “messy”. A lot of time was being spent on trying to understand the problem. Some teams needed 5 - 6 weeks to have a basic understanding of what they should do next. A major difficulty was the communication with a professional in another area. Our disciplines, Chemistry and Computer Science, have deferent vocabularies. Further, the topic presented to the Computer Science students was not an introductory concept in Chemistry. Rather, the topics are based on current research projects in the leading edge field of DNA nanotechnology. Due to the leading edge nature of the research topic, appropriate resources for introducing undergraduate students in disciplines outside of Chemistry to the field of DNA nanotechnology are lacking. Primary literature publications have been deemed an inappropriate vehicle for introducing the Computer Science students to the research topic. Therefore, a short document was created to accompany the Chemistry faculty member’s introductory presentation to the students. The document contained many figures to illustrate the concepts of the research project. While this document alleviated the difficulty of understanding concepts in another discipline, even then students needed considerable time to internalize the concepts.

Searching for an algorithm varied across the projects. All projects in 2013 and two of the projects conducted in 2014 were suitable for applying well-known algorithms. However, we did not have an algorithm for one of the projects, “Efficient Path Generation to Maximize Data Collection of Multiple Samples using Fluorescence Spectroscopy”. The team spent a lot of time discussing various ways to approach the problem.

The implementation phase had its own problems. Some algorithms were too slow, and students had to look for ways to speed them up. Other algorithms did not work for all input. A separate issue for the genetic algorithms was how to represent the problem within the algorithm. Overall, students put a lot of work in implementing their algorithms.

There was a need to dedicate additional time to help students with the project development. Unfortunately, we did not have much available time due to our primary teaching responsibilities. Thus, while some of the teams were helped, others did the work almost entirely on their own and not all of the project goals were accomplished.

In spite of all the stumbling, we consider the projects to be successful. The projects completed in previous years were presented at NCUR and MICS. The projects conducted in the Fall of 2014 are accepted for presentation at NCUR'15 and MICS'15.

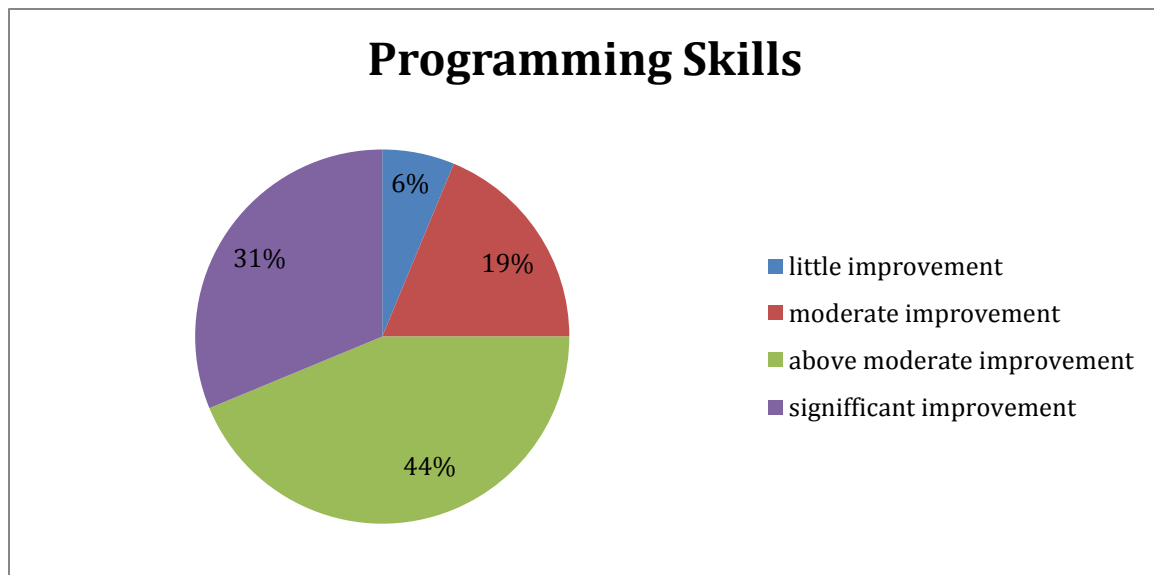
5 Assessment

We used two assessment tools to assess our pedagogy. As a direct tool we used the rate of acceptance of students' abstracts to NCUR and MICS. We are happy to say that this rate is 100%. As an indirect tool we used a short survey conducted after the completion of the projects.

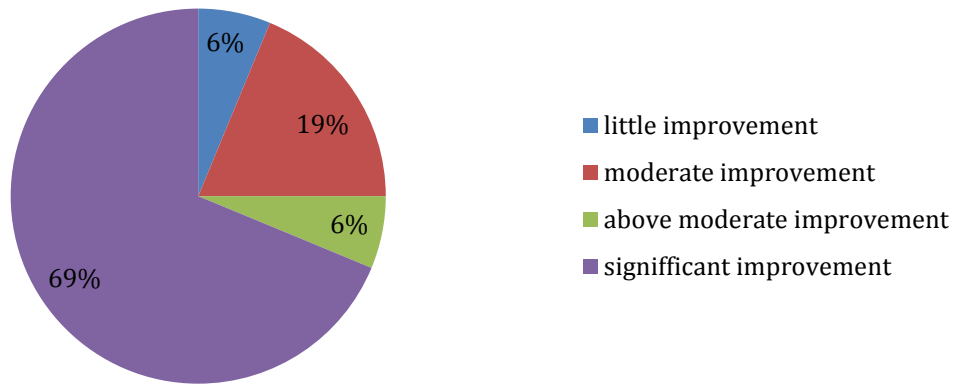
We asked the students to score on a scale of 1 to 10 the improvement of their skills in the following areas: programming, problem solving, oral communication, written communication, teamwork, and leadership. We grouped the responses in four categories based on the scores:

<i>Score</i>	<i>Category</i>
0 – 3	Little improvement
4 – 5	Moderate improvement
6 – 7	Above moderate improvement
8 – 10	Significant improvement

Out of 23 students involved in the projects, 16 responded. The results are presented below.

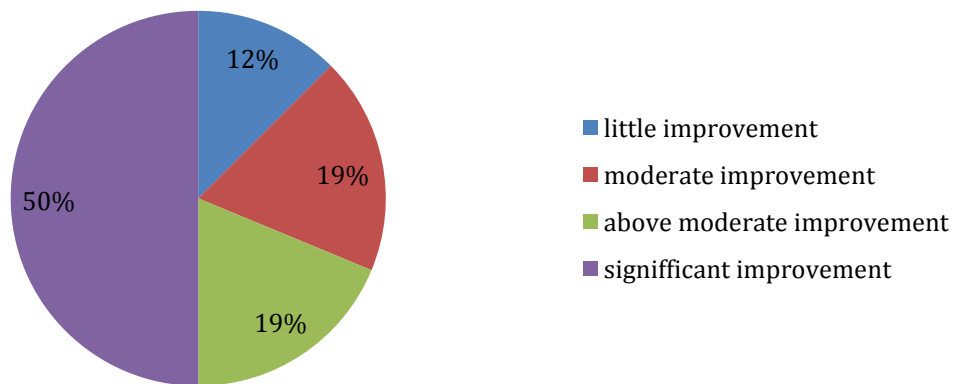


Problem Solving Skills

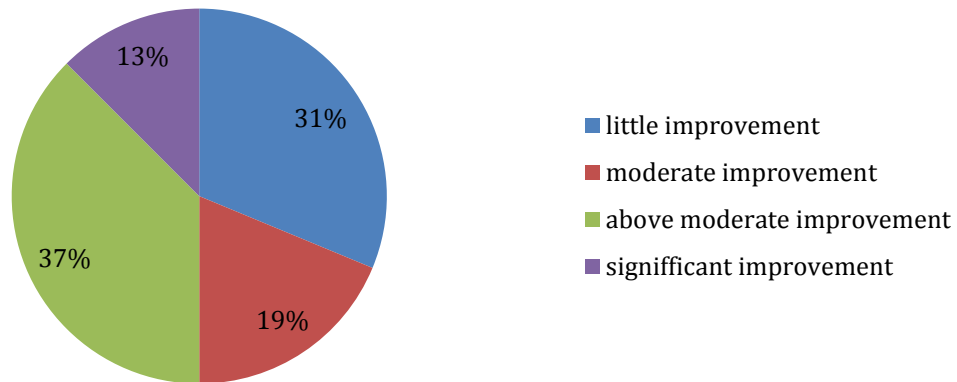


As seen in the charts, the projects have a higher impact on the problem-solving skills than on the programming skills. More than 2/3 of the students reported significant improvement of their problem-solving skills compared to 1/3 of students with significant improvement of programming skills. Due to the nature of the projects, the problem-solving component of the projects had higher emphasis than the programming component. One student wrote: "I liked that it was pretty much a from scratch experience, that we had to come up with our own ideas and try to create our own algorithm, not base our work off of something that was already solved." Another student wrote: "The fact that the problems we were presented with were unsolved and actually applicable to real things was great."

Oral Communication Skills

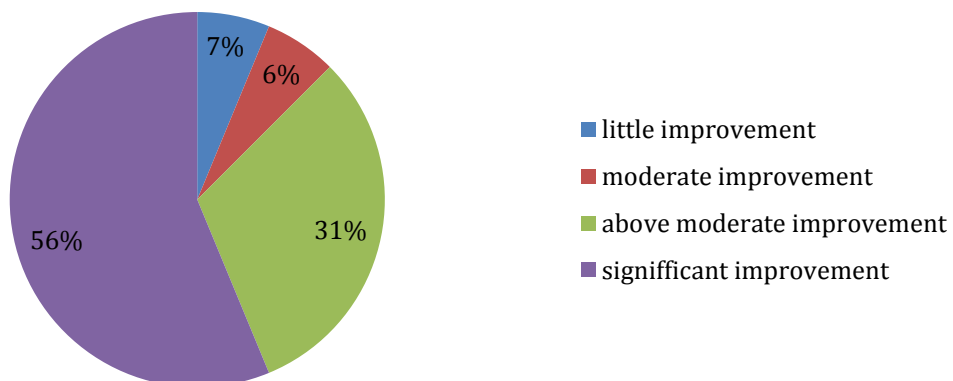


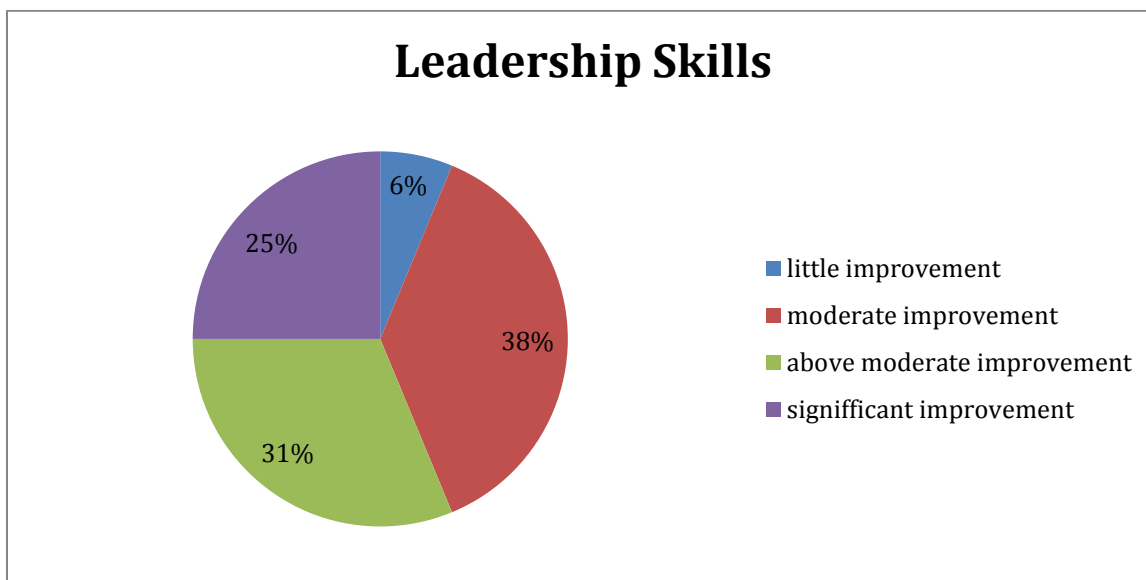
Written-Communication Skills



Due to the intensive discussions within the teams and at the meetings with the supervising instructors, the projects had a positive impact on the oral communication skills. The writing communication skills were not improved at a level that would be satisfactory. The reason lies in the fact that the students did not have time to write a paper that describes their project. Two papers were written and the writing took place in the Spring semester, after the end of the course. We see that these projects consume a lot of time in figuring out what algorithm to be used, and then in implementing that algorithm. For the next semester, we are considering reducing the number of homework assignments and lab assignments, so that students have more time to complete and describe their projects.

Teamwork Skills





More than half of the students reported significant improvement of their teamwork skills, compared to only 25% reporting significant improvement of their leadership skills. We expected that the teamwork skills will be improved, and we also expected the leadership skills to be improved. We think that a better split of the responsibilities among the team members would have a better impact on the leadership skills.

We asked what students liked about the project. Nine students responded that they liked how the project applied to a real-world problem. Seven students found most valuable the interdisciplinary collaboration. Some students valued the opportunity to come up with own ideas to solve an unknown problem.

We asked what they did not like about the project. Nine students did not like that the project was completed outside class time. Overall, students wanted more contact time with the instructors and more class time spent on the projects.

The results of the survey will help us better organize the project in the fall of 2015.

6 Conclusion

The result of the students' work has advanced the research in the Chemistry department by providing useful software tools for the researchers. Typically, the software replaced a laborious or inefficient task performed by the Chemistry researchers. Having the software developed in collaboration between the Chemistry researchers and Computer Science students, created software that directly replaced the task that one once performed manually. For example, sets of DNA sequences for use as linkers between DNA nanostructures had to be designed to meet specific criteria like length and melting

temperature. This process of tuning a DNA sequence to meet the properties desired is an iterative process that requires intelligent changes to be made to a DNA sequence, like swapping a guanine (G) nucleotide for an adenine (A) within the sequence. The process progresses until the desired criteria for the DNA sequence have been achieved. The algorithm implemented by the Computer Science students does this same iterative process and achieves the same results much faster than the manual method.

Currently, several algorithms have been combined into a software package called DNA Sequence Designer (DSD). DSD was created by several students who participated in the development of the algorithms earlier in their education and desired to continue their work on the project. DSD incorporates several algorithms into a user interface. In addition to making the algorithms interactive, packaging the algorithms into software with a user interface allows data to be input and output in formats that are familiar to Chemistry researchers like spreadsheets.

With replacement of laborious tasks done by the Chemistry researchers as the subject for the Computer Science projects, the most recent work has taken this to a new level. An algorithm was recently developed to simulate experiments done in the laboratory. This level of sophistication removes the laborious task of actually performing the experiment! Clearly, the simulations make significant assumptions, requiring that the simulations act only as a tool for guiding the actual experiments. Nevertheless, the simulations have succeeded in reducing the number of experiments needed to optimize the initial conditions for assembly of a large DNA nanostructure.

Further efforts to reduce the laborious tasks performed by the Chemistry research targeted the automation of scientific instrumentation. Specifically, the instrument used most often for analyzing the assembly of DNA nanostructures is a thermocycler containing optical equipment. Our instrument is similar to a quantitative PCR (qPCR) instrument, where liquid samples are precisely heated and cooled while the amount of fluorescence emitted from the sample is measured. Fluorescence is used as a molecular signal for attachment of DNA strands by virtue of the Forster Resonance Energy Transfer (FRET) technique. Computer Science students began developing an algorithm that optimizes the movement of the optical equipment to maximize the amount of data captured by the instrument over many samples. The students who began this work are continuing this project as part of their capstone experience. Completing the automation of the instrumentation will allow the experiments to be initiated at any time and data to be accessible from any device in real-time.

In summary, the key benefit of this collaboration for the research program in the Chemistry department is the gift of customized software programs to use as tools for designing, optimizing, and even performing experiments. The software reduces the burden of time-consuming, laborious tasks. Alleviating the laborious tasks enables the research projects to progress faster. For the Chemistry faculty member, the payment for these software tools is the time commitment of weekly meetings with the Computer Science students to maintain progress on the students' projects. Hour-long, weekly

meetings with each group are necessary for introducing and developing algorithms within one semester's time.

The faculty in the Chemistry and Computer Science departments are very excited about the impact of this pedagogy on advancing research efforts and improving problem-solving skills of our undergraduate students. We believe that open-ended projects, in a leading edge research area, opens new horizons for the students and enhances their qualifications as scientists. We are happy with the students' achievements to date and we are determined to further improve our pedagogy to ensure that the students continue to succeed.

References

[1] Davidson, C. [Identifying Gene Regulatory Networks Using Evolutionary Algorithms](#). In *CCSC Central Plains Conference*, April 9 – 10, 2010, Park University, Parkville MO.

[2] Gawarecki, J. et al. [Optimization of Tile Sets for DNA Self-Assembly](#). In *Midwest Instruction and Computing Symposium* April 13 – 14, 2012, University of Northern Iowa, Cedar Falls.

[3] Karukstis, K., N. Hensel (Eds) *Transformative Research at Predominately Undergraduate Institutions*. Council on Undergraduate Research. 2010.

[4] Pilling, S. [Using Multiple Sensors In An Obstacle Avoidance Algorithm For Lego Mindstorms Robots](#). In *Midwest Instruction and Computing Symposium*, April 7 – 8, 2006, Iowa Wesleyan College.