Applications of Parameterized Computation in Computational Biology

X. Huang

Arkansas State University
P.O. Box 9, State University, AR, 72467
xzhuang@csm.astate.edu

ABSTRACT

This paper first gives an introduction to parameterized computation and complexity theory, a new subfield in theoretical computer science. Then it presents a summary of its applications to addressing some important NP-hard problems in computational biology. Specifically, we can design efficient parameterized algorithms and also drive computational lower bounds for the parameterized algorithms and approximation algorithms of computational biological problems.

Keywords: parameterized computation and complexity, computational biology, NP-hard, algorithms

1 INTRODUCTION TO PARAMETERIZED COMPLEXITY

According to the theory of NP-completeness, many problems that have important real-world applications in life science are NP-hard. This excludes the possibility of solving them in polynomial time unless P=NP. For example, the problems of cleaning up data, multiple sequence alignment, closest string and maximum common substructure, are all famous NP-hard problems in computational biology [6], [14], [25], [31]. A number of approaches have been proposed in dealing with these NP-hard problems. For example, the highly acclaimed approximation approach [3] tries to come up with a *good* enough solution in polynomial time instead of an optimal solution for an NP-hard optimization problem [11], [12], [24], [26].

The theory of parameterized computation [14] is a newly developed approach introduced to address NP-hard problems with *small* parameters. It tries to give exact algorithms for an NP-hard problem when its natural parameter is small (even if the problem size is big). Problems are considered fixed-parameter tractable (in the class FPT) if they can be solved in time $O(f(k)n^c)$, where n is the problem size, k is the parameter, f is a recursive function, and c is a constant. For a problem in the class FPT, researchers try to come up with more efficient parameterized algorithms. There are many effective techniques for parameterized algorithm designing, such as the methods of bounded search tree and reduction to a problem kernel. For example, the VERTEX COVER problem, a well-known NP-hard problem, is fixed-parameter tractable (in FPT).

> VERTEX COVER problem: given a graph G and an integer k, determine if G has a vertex cover C of k vertices, i.e., a subset C of k vertices in G such that every edge in G has at least one end in C. Here the parameter is k.

Given a graph of n vertices, there is a parameterized algorithm that can solve the VERTEX COVER problem in time $O(kn + 1.286^k)$ [10].

Accompanying the work on designing efficient and practical parameterized algorithms, a theory of parameter intractability is developed. In parameterized complexity, to classify fixed-parameter intractable problems, a hierarchy, *the W-hierarchy* $\bigcup_{t\geq 0} W[t]$, where $W[t] \subseteq W[t+1]$ for all $t \geq 0$, has been introduced, in which the 0-th level $W[0]$ is the class FPT. The hardness and completeness have been defined for each level $W[i]$ of the W-hierarchy for $i \geq 1$, and a large number of $W[i]$-hard parameterized problems have been identified [14]. For example, the CLIQUE problem is $W[1]$-hard.

> CLIQUE problem: given a graph G and an integer k, determine if G has a clique C of k vertices, i.e., a subset C of k vertices in G such that there is an edge in G between any two of these k vertices, i.e., the k vertices induce a complete subgraph of G. Here the parameter is k.

The CLIQUE problem is also a well-known NP-hard problem [17]. The CLIQUE problem can be solved in time $O(n^k)$, based on the enumeration of all the vertex subsets of size k for a given graph with n vertices.

Now it has become commonly accepted that no $W[1]$-hard (and $W[i]$-hard, $i > 1$) problem can be solved in time $f(k)n^{O(1)}$ for any function f (i.e., $W[1] \neq FPT$). $W[1]$-hardness has served as the hypothesis for fixed-parameter intractability. Examples include a recent result by Papadimitriou and Yannakakis [28], showing that the DATABASE QUERY EVALUATION problem is $W[1]$-hard. This provides strong evidence that the problem

cannot be solved by an algorithm whose running time is of the form $f(k)n^{O(1)}$, thus excluding the possibility of a practical algorithm for the problem even if the parameter k (the size of the query) is small as in most practical cases.

Research activities in parameterized computation have demonstrated rich complexity structures and effective algorithmic approaches. This research area has found applications in computational biology, database systems, networks, parallel computing, VLSI design and other research areas. Please refer to [7], [13]–[15], [21], [28] and the recently published special issue in Journal of Computer and System Sciences (Volume 67, No.6, 2003, Guest Editors: J. Chen and M. Fellows).

2 PARAMETERIZED ALGORITHMS FOR COMPUTATIONAL BIOLOGY PROBLEMS

In this section, we discuss efficient parameterized algorithms for computational biology problems. As an example, we illustrate the applications of the fixed-parameter tractable VERTEX COVER problem.

For parameterized algorithm design, there are two basic methods. One of the basic methods is *bounded search tree*. This method is based on the observation that many problems can be solved by algorithms of two steps: First the algorithm computes a search space which is often an exponential-sized search tree; Then the algorithm applies some efficient techniques on each branch of the search tree. Based on the bounded search tree method, an algorithm for the VERTEX COVER problem of time $O(2^k n)$ was designed, where k is the parameter and n is the number of vertices of a given graph [14]. Another method is called *reduction to a problem kernel*. The basic idea is to reduce an instance of a problem to an equivalent instance of size bounded by some function of the parameter. By applying this method, the algorithm for the VERTEX COVER problem was further improved to $O(n + k^k)$ [14].

Many researchers have worked on the parameterized algorithms for the VERTEX COVER problem. It is interesting to review the research progress for the VERTEX COVER problem [14]. In 1988, Fellows gave an algorithm of time $O(2^k n)$. In 1989, Buss described an algorithm of time $O(kn + 2^k k^{2k+2})$. In 1992, Downey et al. described an algorithm of time $O(kn + 2^k k^2)$. In 1996, Balasubramanian et al. gave an algorithm of time $O(kn + (4/3)^k k^2)$. In 2000, Niedermeier and Rossmanith presented an algorithm of time $O(kn + 1.292^k)$. After many rounds of improvement, the current best algorithm for the VERTEX COVER problem is of time $O(kn + 1.286^k)$ due to the work of Chen et al. in 2001 [10].

Now we discuss the applications of the algorithms for the VERTEX COVER problem in solving computational biology problems. First, we look at the DATA CLEANING problem [14]: Given a set of experimental data, there are some conflicts between them. The problem asks to remove the least number of data to resolve all the conflicts. This problem can be formulated as the vertex cover problem as follows. We first build a graph, in which each data is represented as a vertex and each conflict between two data is represented as an edge between the two corresponding vertices. We can see that the minimum vertex cover of the graph corresponds to a set of data, the removing of which resolves all the conflicts.

Another important computational biology problem is MULTIPLE SEQUENCE ALIGNMENT, which is usually performed to fit one of the following scopes [29]: In order to characterize protein families, identify shared regions of homology in a multiple sequence alignment; Determination of the consensus sequence of several aligned sequences; Help prediction of the secondary and tertiary structures of new sequences, and; Preliminary step in molecular evolution analysis using phylogenetic methods for constructing phylogenetic trees.

The Computational Biochemistry Research Group at the ETH Zürich has successfully applied algorithms for the VERTEX COVER problem to their research in MULTIPLE SEQUENCE ALIGNMENTS [32], [33], where the parameter value k, i.e., the number of sequences, can be bounded by 60. Here the basic idea of is the same. Based on the given multiple sequences, a graph is constructed where a vertex is built to correspond to a sequence, and an edge is built between two vertices, if there is a conflict between the two corresponding sequences, that is, if the alignment of the two sequences has a score lower than a certain threshold. Here the goal is to remove the fewest possible sequences that will eliminate all conflicts in the alignment. It can be seen that the removing of the sequences corresponding to the vertex cover of the graph will resolve all the conflicts. As we know, the current best known parameterized algorithm for the VERTEX COVER problem runs in time $O(1.286^k + kn)$ [10]. This algorithm has been implemented and is quite practical. For example, by also applying parallel processing techniques, the algorithm can solve problem instances with $k \geq 400$ (e.g. $k = 461$) in less than 1.5 hours [6].

Other research work on investigating efficient parameterized algorithm for computational biological problems, such as the CLOSEST STRING problem, the LONGEST COMMON SUBSEQUENCE problem and the DISTINGUISHING SUBSTRING SELECTION problem, can be found in [1], [2], [18]–[20].

3 PARAMETERIZED LOWER BOUNDS FOR COMPUTATIONAL BIOLOGY PROBLEMS

In the last section, we have discussed the applications of parameterized algorithms for solving computational biology problems. We will see in this section that parameterized intractability also has interesting applications in addressing computational biology problems. For some biological problems, we can show that no effort could lead to a better parameterized algorithm or approximation algorithm.

Based on the W[1]-hardness of the CLIQUE algorithm, computational intractability of problems in computational biology has been derived [4], [5], [16], [22], [27], [30]. For example, in [30], the author point out that "Unless an unlikely collapse in the parameterized hierarchy occurs, this (This refers to the results proved in [30] that the problems LONGEST COMMON SUBSEQUENCE and SHORTEST COMMON SUPERSEQUENCE are $W[1]$-hard) rules out the existence of exact algorithms with running time $f(k)n^{O(1)}$ (i.e., exponential only in k) for those problems. This does not mean that there are no algorithms with much better asymptotic time-complexity than the known $O(n^k)$ algorithms based on dynamic programming, e.g., algorithms with running time $n^{\sqrt{k}}$ are not deemed impossible by our results."

Recent investigation has derived stronger computational lower bounds for well-known NP-hard parameterized problems in [8], [9]. For example, although a trivial enumeration can easily test in time $O(n^k)$ if a given graph of n vertices has a clique of size k, it is proved that unless an unlikely collapse occurs in parameterized complexity theory, the problem is not solvable in time $f(k)n^{o(k)}$ for *any* function f. Under the same assumption, it is shown that even if we restrict the parameter values k to be of the order $\Theta(\mu(n))$ for *any* reasonable function μ, no algorithm of running time $n^{o(k)}$ can test if a graph of n vertices has a clique of size k.

Based on the hardness of the CLIQUE problem, we derive lower bound results for a number of computational biology problems [23]. One example is the MOTIF FINDING problem, which has applications in finding conserved regions in molecular biology, as well as applications in coding theory. A graph theoretical formulation of the MOTIF FINDING problem was proposed in [34]. It reduces the MOTIF FINDING problem to finding a maximum clique in a k-partite graph. According to the parameterized complexity theory, we can prove that this problem formulation is $W[1]$-complete with respect to the number of strings k as the parameter. We can derive computational lower bounds of the parameterized algorithms for this problem. We are working on the parameterized complexity of the problem with respect to the maximum allowed Hamming distance d. The maximum allowed Hamming distance d is considered as the value of the objective function in designing an approximation scheme in [26]. To resolve the parameterized complexity of this problem with respect to the parameter d will answer the open problem posed in [16], [20].

Moreover, the hardness result has also offered a method for deriving lower bounds on the running time of approximation algorithms for NP-hard combinatorial optimization problems in computational biology. The NP-hard DISTINGUISHING SUBSTRING SELECTION problem was studied in [9]. This problem has important applications in the genetic drug design, where the goal is to find a gene sequence that is close to bad genes (the target) but far from all good genes (to avoid side-effects). For the DISTINGUISHING SUBSTRING SELECTION problem, a polynomial time approximation scheme has been recently developed [11], [12]. The approximation algorithm runs in time $O(mn^{O(1/\epsilon^6)})$, where m is the problem size, and n is total number of good and bad strings. It was showed that this problem has no polynomial time approximation schemes of running time $f(1/\epsilon)n^{o(1/\epsilon)}$ for any function f unless an unlikely collapse occurs in parameterized complexity theory [9]. The techniques can also be extended to derive computational lower bounds for PTAS algorithms for the LONGEST COMMON SUBSEQUENCE problem in computational biology. This seems to have opened a new direction for the study of computational lower bounds on the approximability of NP-hard optimization problems.

4 CONCLUSION

As we know, most computational biology problems are NP-hard. In considering that many of them involve small parameters, the newly developed research area, parameterized computation, is one proper approach for studying these NP-hard computational biology problems. We can design efficient parameterized algorithms for practical use and also drive computational lower bounds for the parameterized algorithms and approximation algorithms of computational biological problems. In future, we would like to further explore the applications of parameterized computation and complexity theory for other important problems in computational biology.

REFERENCES

[1] J. Alber, J. Gramm, J. Guo, and R. Niedermeier, "Towards optimally solving the longest common subsequence problem for sequences with nested arc annotations in linear time," CPM 2002, 99, 2002.

[2] J. Alber, J. Gramm, J. Guo, and R. Niedermeier, "Computing the similarity of two sequences with nested arc annotation," Theor. Comput. Sci., 312, 337, 2004.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, "Complexity and Approximation, Combinatorial Optimization Problems and Their Approximability Properties," Springer-Verlag, 1999.
[4] H. Bodlaender, R. Downey, M. Fellows, M. Hallett, and H. Wareham, "Parameterized complexity analysis in computational biology," Computer Applications in Biosciences, 11, 49, 1995.
[5] H. Bodlaender, R. Downey, M. Fellows, and H. Wareham, "The parameterized complexity of sequence alignment and consensus," Theoretical Computer Science, 147, 31, 1995.
[6] J. Cheetham, F. Dehne, A. Rau-Chaplin, U. Stege, and P. J. Taillon, "Solving large FPT problems on coarse-grained parallel machines," JCSS, 67, 691, 2003.
[7] J. Chen, "Parameterized computation and complexity: a new approach dealing with NP-hardness," Survey, 2004.
[8] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia, "Tight lower bounds for parameterized NP-hard problems," Proc. of the 19th Annual IEEE Conference on Computational Complexity, 150, 2004.
[9] J. Chen, X. Huang, I. Kanj, and G. Xia, "Linear FPT reductions and computational lower bounds," Proc. of the 36th ACM Symposium on Theory of Computing, 212, 2004.
[10] J. Chen, I. Kanj, and W. Jia, "Vertex cover: further observations and further improvements," Journal of Algorithms, 41, 280, 2001.
[11] X. Deng, G. Li, Z. Li, B. Ma, and L. Wang, "A PTAS for distinguishing (sub)string selection," LNCS, 2380, 740, 2002.
[12] X. Deng, G. Li, Z. Li, B. Ma, and L. Wang, "Genetic design of drugs without side-effects," SIAM Journal on Computing, 32, 1073, 2003.
[13] R. Downey, "Parameterized complexity for the skeptic," Proc. 18th IEEE Annual Conference on Computational Complexity, 132, 2003.
[14] R. Downey and M. Fellows, "Parameterized Complexity," Springer, New York, 1999.
[15] M. Fellows, "Parameterized complexity: the main ideas and some research frontiers," Lecture Notes in Computer Science, 2223, 291, 2001.
[16] M. Fellows, J. Gramm, and R. Niedermeier, "Parameterized intractability of motif search problems," LNCS, 2285, 262, 2002.
[17] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman, New York, 1979.
[18] J. Gramm, R. Niedermeier, and P. Rossmanith, "Exact solutions for closest string and related problems," LNCS, 2223, 441, 2001.
[19] J. Gramm, R. Niedermeier, and P. Rossmanith, "Fixed-parameter algorithms for CLOSEST STRING and related problems," Algorithmica, 37, 25, 2003.
[20] J. Gramm, J. Guo, and R. Niedermeier, "On exact and approximation algorithms for distinguishing substring selection," LNCS, 2751, 195, 2003.
[21] M. Grohe, "Parameterized complexity for the database theorist," SIGMOD Record, 31, 86, 2002.
[22] M. Hallett, "An Integrated Complexity Analysi of Problems for Computational Biology", Ph.D. Thesis, University of Victoria, 1996.
[23] X. Huang, "Parameterized Complexity and Polynomial-time Approximation Schemes," Ph.D. Dissertation, Texas A&M University, 2004.
[24] T. Jiang and M. Li, "On the Approximation of Shortest Common Supersequences and Longest Common Subsequences," SIAM J. Comput., 24, 1122, 1995.
[25] J. K. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang, "Distinguishing string selection problems," Inf. Comput., 185, 41, 2003.
[26] M. Li, B. Ma, and L. Wang, "On the closest string and substring problems," Jounal of the ACM, 49, 157, 2002.
[27] C. Papadimitriou and M. Yannakakis, "On limited nondeterminism and the complexity of VC dimension," JCSS, 53, 161, 1996.
[28] C. Papadimitriou and M. Yannakakis, "On the complexity of database queries," JCSS, 58, 407, 1999.
[29] Online information at http://www.infobiogen.fr/doc/MAcours/multalign.html.
[30] K. Pietrzak, "On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems," JCSS, 67, 757, 2003.
[31] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," Journal of Computer-aided Molecular Design, 16, 521, 2002.
[32] C. Roth-Korostensky, "Algorithms for building multiple sequence alignments and evolutionary trees," Ph.D. Thesis, No. 13550, ETH Zürich, 2000.
[33] U. Stege, "Resolving conflicts from problems in computational biology," Ph.D. Thesis, No. 13364, ETH Zürich, 2000.
[34] P. A. Pevzner and S.-H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences," Proc. of 8th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'2000), 269, 2000.