University of Utah
UNDERGRADUATE RESEARCH JOURNAL

TURBO-CHARGING ODD CYCLE TRANSVERSAL

Jing Guo (Blair D. Sullivan, Ph.D., Brian Lavallee, Ph.D. Student)
School of Computing

Historically, important approaches to overcoming the computational challenges posed by NP-hard problems have been the design of polynomial-time approximation algorithms or the use of heuristics, which trade accuracy for speed. A more recent complementary approach has been the introduction of parameterized complexity, which transfers the exponential dependence of an algorithm's runtime from the input size to a "parameter" which is bounded (constant value) on relevant instances."

One may formally define a problem as fixed parameter tractable (FPT) if it can be solved in time $O(f(k) * n^c)$, where k is the parameter, n is the input size, and c is a constant. A classical FPT problem is k-Odd Cycle Transversal (k-OCT), which asks whether a simple, undirected graph G contains a subset of vertices of size k, whose removal leaves a bipartite subgraph.The parameterized complexity of k-OCT was a long-standing open problem, only resolved in 2003 when Reed, Smith, and Vetta showed it was FPT using iterative compression, which has become a fundamental technique in the FPT literature.

While the result of Reed et al. established the theoretical tractability of k-OCT, practically it remains a challenge since most real-world networks have inherently large parameter values. Our work focuses on solving k-OCT by "turbo-charging," -- using an FPT subroutine to locally improve solutions within heuristic or approximation algorithms. Such a technique was proposed by Downey et al. in 2014 to solve a dynamic version of Dominating Set problem, and can offer the high solution quality of FPT algorithms with the reduced run time of an approximation algorithm.

Theoretically, we prove that "turbo-charging" can be applied to both OCT and Vertex Cover, a related NP-hard problem. We then implemented both new algorithms, integrating common techniques from FPT algorithms, such as iterative compression and kernelization, with highly tuned approximations and heuristics. We found that computing an LP-based kernel optimized with crown decomposition as a preprocessing step improved both the solution quality and the runtime of our turbo-charged algorithms. Then, by experimenting with different approximation algorithms and heuristics, we were able to demonstrate the practicality of our approach on a collection of real-world networks as well as a large synthetic corpus.