Since parallel computing is again becoming a topic of interest in computer science (partially due to the recent popularity of multicore processors in mobile computing devices in the developed world), it is important to revisit the theoretical foundations of highly parallel computing. “Inherently sequential” computational problems see no speedup when run on highly parallel computers. Just as there are efficient approximations for intractable optimization problems, so too are there highly parallel approximations for inherently sequential optimization problems. For example, the problem of computing the optimal vector in a positive linear program, a problem relevant to distributed flow control within a network of routers, is inherently sequential, but a vector very close to the optimal one can be computed quickly by a highly parallel computer.

I intend to provide results demonstrating the structural complexity of highly parallel approximations for inherently sequential problems. This area has not been well-studied, and when it has been studied, the results focus mostly on highly parallel approximations for intractable optimization problems (that is, NC approximations for NP-hard problems), not highly parallel approximations for tractable but inherently sequential problems (that is, NC approximations for P-hard problems).

First, we show that for optimization problems, the complexity of the verification procedure matters. Specifically, for some problems, verifying a solution is an inherently sequential problem, whereas for others, verification can be done in parallel. This allows us to define a hierarchy of complexity classes consisting of optimization problems that can be approximated in parallel. This hierarchy exists between NC and NP (circumventing P). We show that this hierarchy (as well as the one resulting from its intersection with P) are likely strict.

Second, we show that LINEAR PROGRAMMING is complete for the class of maximization problems that can be solved exactly in polynomial time, thus providing an explanation as to why any highly parallel approximation for LINEAR PROGRAMMING implies NC = P (a fact known by the year 1991). In contrast, there are inherently sequential problems that admit highly parallel constant-factor approximations up to a certain fixed threshold, as well as those that admit highly parallel approximation schemes. We hope to show that one of the constant-factor approximable problems is complete for the class of all such
problems, but we have not been able to do so.

Third, we attempt to produce a probabilistically checkable proof characterization for \( P \) in order to yield inapproximability results (in the sense of “no approximation by highly parallel computer”) similar to those from the realm of \( P \) and \( NP \). Since there is no obvious characterization of \( P \) as a proof system, we use the nearby (but probably incomparable) complexity class of languages decidable by a highly parallel machine augmented with a polylogarithmic amount of nondeterministic bits. We can provide a PCP characterization of this complexity class, but we are not quite able to provide inapproximability results (we need to find an optimization problem that meets some requirements, but few potential optimization problems are known). As part of this avenue of research, we also show that the original PCP Theorem holds even if the verifier is restricted to be an \( NC \) machine.