

## **abstract**

COMPUTER SCIENCE/DISCRETE MATH II

Topic:

Speaker:

Affiliation:

Date:

Time/Room:

---

The "Design and Analysis of Algorithms" or "Introduction to Algorithms" is a basic undergraduate course in CS. In such courses, the organizational theme is often in terms of "basic algorithmic paradigms" such as greedy algorithms, dynamic programming, divide and conquer, local search, network flows, and (in more advanced or graduate courses) LP rounding, primal dual, algebraic techniques. Maybe it is just that time doesn't permit us to illustrate many such paradigms or maybe there aren't that many to illustrate. In any case, even though we usually discuss (by examples) a few general techniques, we rarely if ever try to precisely define these algorithmic concepts and hence cannot rigorously address the question as to their power and limitations. (Within the field of Operations Research there have been some early attempts to formally model and study dynamic programming and branch and bound algorithms but this work has been largely ignored in CS.)

I have been considering these questions for a few years with a number of colleagues and we have attempted to define and study precise models corresponding to some of the above mentioned paradigms. I will discuss some models corresponding to greedy algorithms and "simple" forms of DP and primal-dual (local-ratio) algorithms and a few of the negative search and approximation results that can be derived within these models. Even the concept of a greedy (approximation) algorithm seems to be subject to debate and it is surprising how much algorithmic diversity (and complexity) can be incorporated within this conceptually "simple" algorithmic concept.