System Improvement Using Structural Conservation

Stephen M. Robinson
Industrial and Systems Engineering
University of Wisconsin-Madison

ACNW Workshop, Chicago, June 2005

Sponsored by ARO and AFOSR

Outline

- Two examples of a modeling strategy
- Stochastic network improvement via 2-moment approximations
  - Current work of Julien Granger
  - Uses deterministic approximation to estimate an optimizer
  - Validates “betterness” with simulation
- Generalized equations
  - Developed area, payoffs now better known
- Can we push this farther? If so, how?

Network improvement

- Stochastic networks with parameters
  - Service rates, # of servers, etc.
  - Can (pay to) change parameters
  - Try to improve a performance measure
    - E.g. steady-state mean throughput
- Idea: use 2-moment approximations
  - Replace nodes by renewal processes with infinite buffers
    - Can model hard constraints with fork/join, kanban
  - Characterize process by mean, SCV
  - Form system of nonlinear equations

Network improvement 2

- Nonlinear system: parameters plus other performance measures
- Use deterministic methods to adjust parameters for good performance
  - 1 simulation to validate improvement (not necessarily optimization) at end
- Previous work: Whitt, Sanders, Suri, Kamath, Krishnamurthy, others
  - Current work is thesis of J. Granger
  - More info: granger@cae.wisc.edu

What's the abstract approach?

- We have a system involving
  - Nodes (servers) of various forms
  - A topology that connects them and enforces certain relationships
- Model this for improvement by:
  - *Simplifying* the nodes (perhaps greatly)
  - *Retaining* the structure to connect the simplified nodes
- This we've seen before

Generalized equations

- This same idea occurs in solving generalized equations
  - Original: $0 \in f(x) + N_C(x)$, where C is closed convex, N_C is normal cone to C
  - Approximation: $0 \in a(x) + N_C(x)$, with a being a function much simpler than f
  - Here N_C represents "system topology," e.g. introducing corners into problem
- Contrasts with direct approximation

Empirical insight, and question

- This approach, conserving system *structure* while simplifying system *elements*, has produced good tools
  - These have performed very well vs. attempts at direct approximation of the system output
- Is it more generally applicable?
- If so, what guides does it provide to where we might concentrate effort?

Implications for investigation

- If we wanted to learn more about exploiting this idea, we could ask:
  - For common systems, *what kinds of simplified elements* would make it easy to estimate the system performance?
  - *What elements of system structure* are particularly critical to defining the important output characteristics?
  - *How good* is resulting approximation, & why?
- Answers could pay off in better methods

Questions?