Improving Coalition Performance by Exploiting Phase Transition Behavior

David W. Etherington
Matthew L. Ginsberg
Andrew J. Parkes

Computational Intelligence Research Laboratory

University of Oregon

CIRL

The Question

Why should the ANTS community care about theoretical complexity results?

The Answer

Complexity results provide key insights

necessary to make solutions that are:

Good enough

- Make solutions robust under uncertainty/dynamics
- Reduce need for coordination

Soon enough

- Increase effectiveness of limited communications
- Efficiently parallelizable solutions

Overview

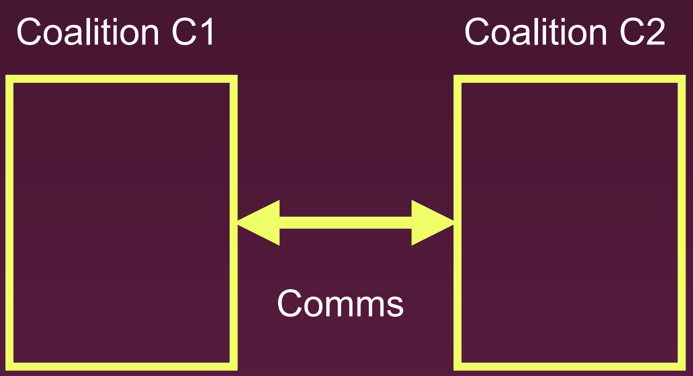
Local robustness and coalition performance

- reducing communication requirements

Achieving robustness

- lessons from phase transitions

Coupling and Robust Solutions



Good coalition structures should reduce coupling

Using locally robust solutions can reduce coupling

Maximum leverage is obtained only by considering both together

Self-Reliance and Cooperation

Self-reliance

- prefer solutions that rely less on other coalitions
- coarse-grained "least-constrain<u>ed</u>" heuristics

Cooperation

- avoid solutions that over-constrain others
- coarse-grained "least-constrain<u>ing</u>" heuristics

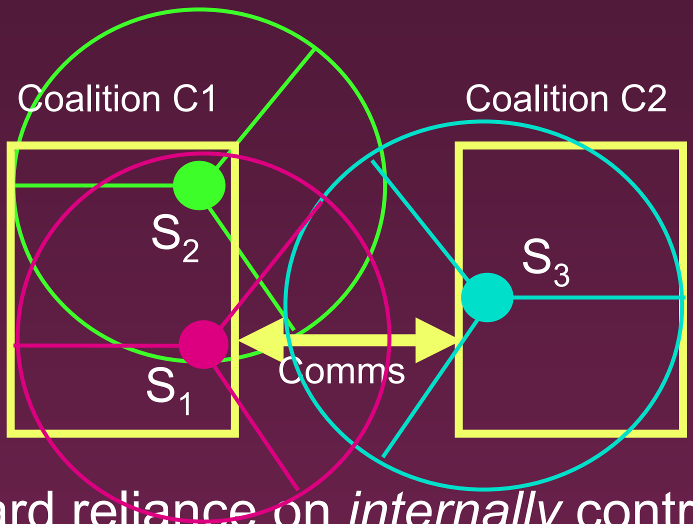
Advantages

- reduce need for renegotiation
- reduce communication with other coalitions

Soft Constraints for Self-Reliance

Each constraint involves

- internal vars: controlled by the coalition e.g. Use(S_1), Use(S_2)
- external vars: controlled by some other coalition, e.g. Use(S_3)



Bias solutions toward reliance on *internally* controlled choices.

- e.g. given C1's constraint: Use(S_1) or Use(S_2) or Use(S_3)
- add the soft auxiliary constraint: Use(S_1) or Use(S_2)

Effectiveness of Self-Reliance

Two coalitions

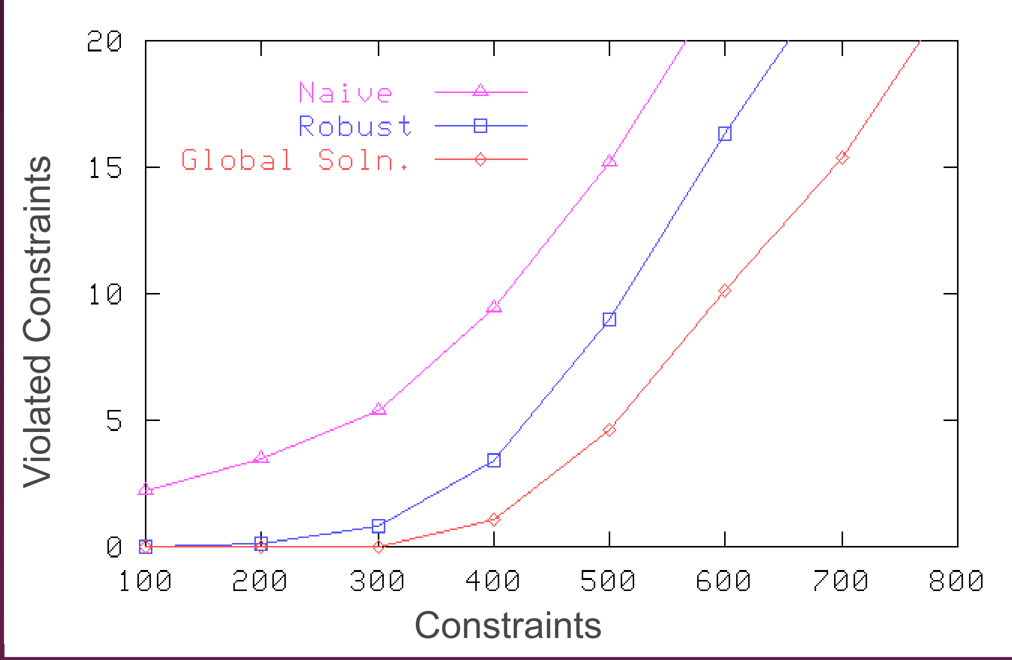
- separately find solutions
- merge in single round

Naive:

- no attempt to be robust

Robust:

- biased to self-reliance



Result: simple bias towards robust, self-reliant, solutions

- significantly improves performance
- reduces need for further negotiation rounds

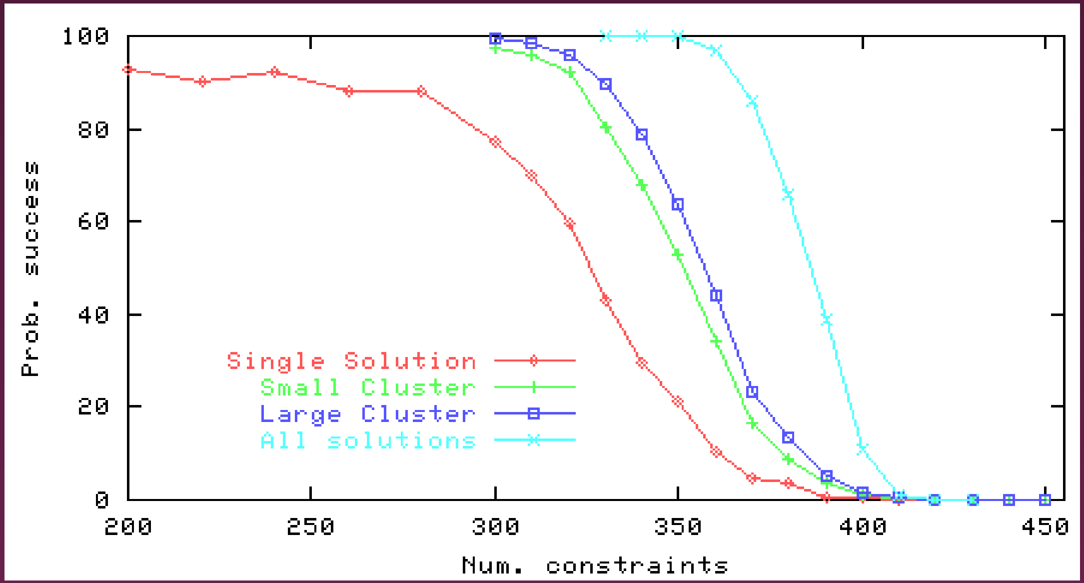
Cooperation

A solution cluster is a set of solutions with
- small set of forced variables
- other variables are relatively unconstrained

Use to reduce constraints on other members

Effectiveness of Cooperation

Measure probability of successful integration of local solutions



Result: solution clusters can markedly improve success rates

Overview

Local robustness and coalition performance

- better to exchange robust solutions than random
- reduces communication requirements

Achieving robustness

- lessons from phase transitions

Quantifying Robustness

How far can a solution can be tweaked before it breaks?

Achievable robustness: the maximum percentage of variables that can be reset in some solution

- higher percentage means more robust

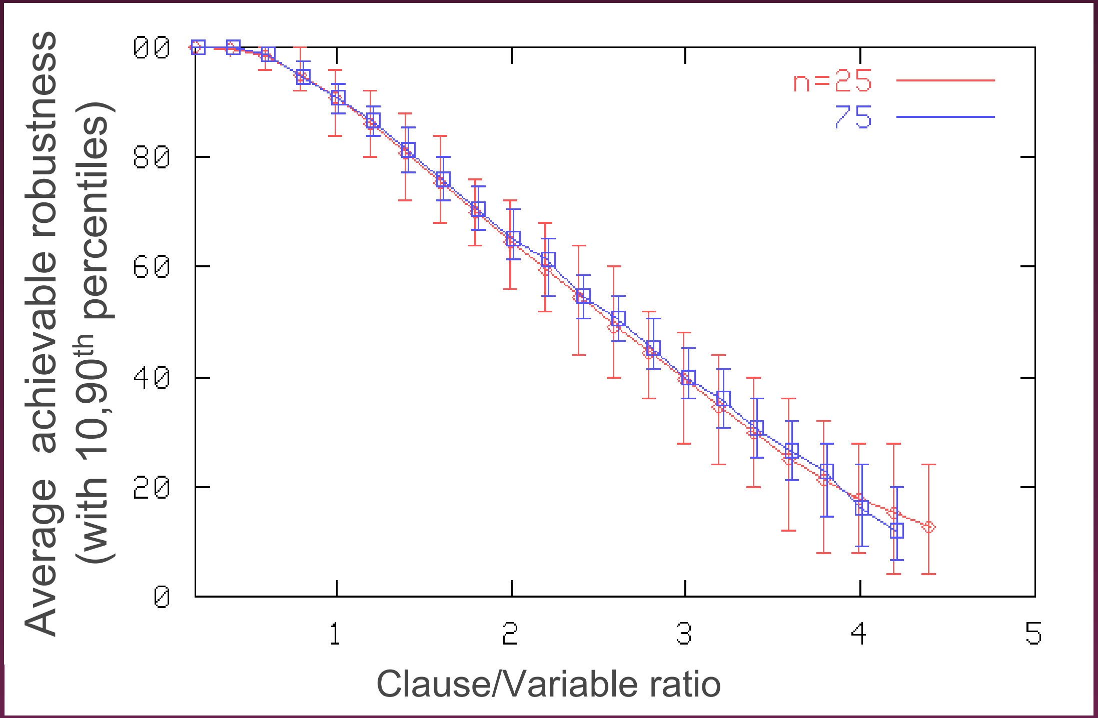
With constraint {x or y}, the solution x=y=true is most robust

- either of x or y can be reset
- achievable robustness is 100%

Robustness: Phase Transition

Random 3-SAT:

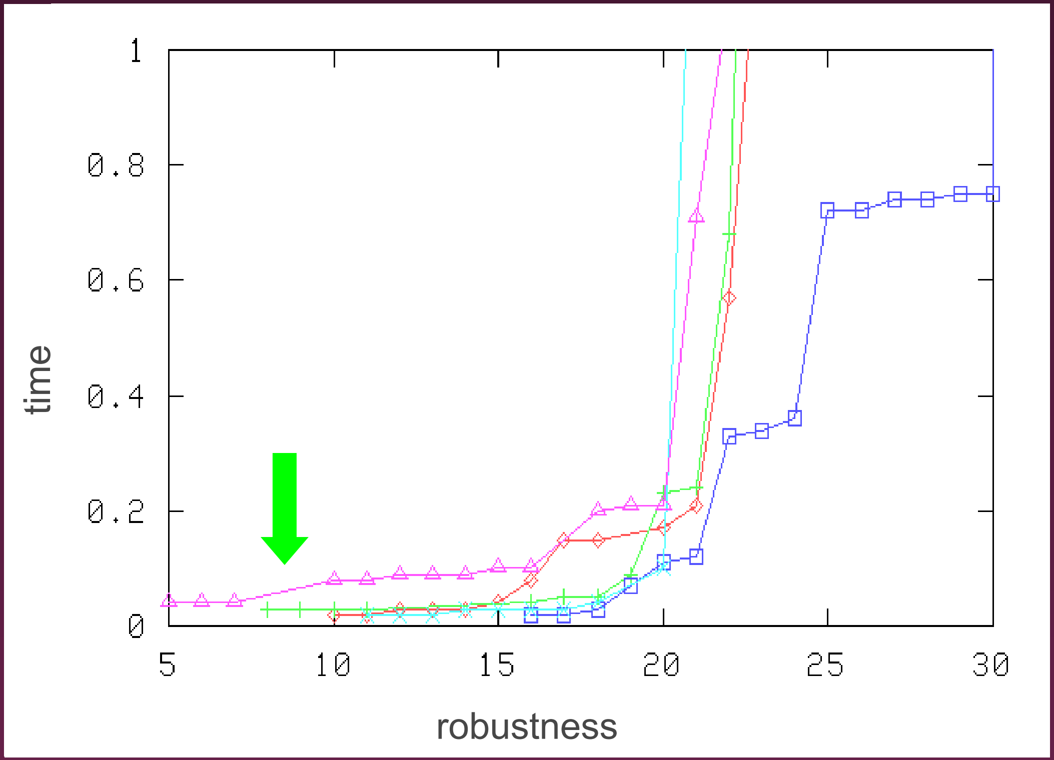
- achievable robustness depends on clause/ variable ratio
- sharp transition from almost always achievable to almost never achievable



Phase transition lets us predict achievable robustness.

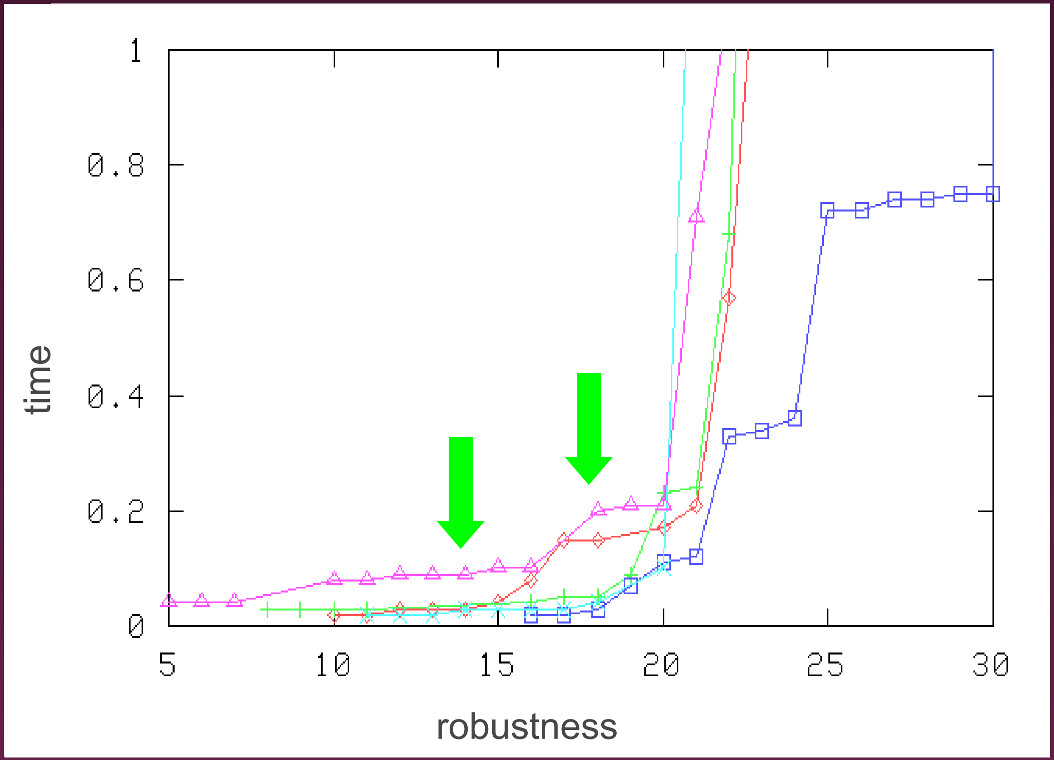
CIRL

Achievable Robustness



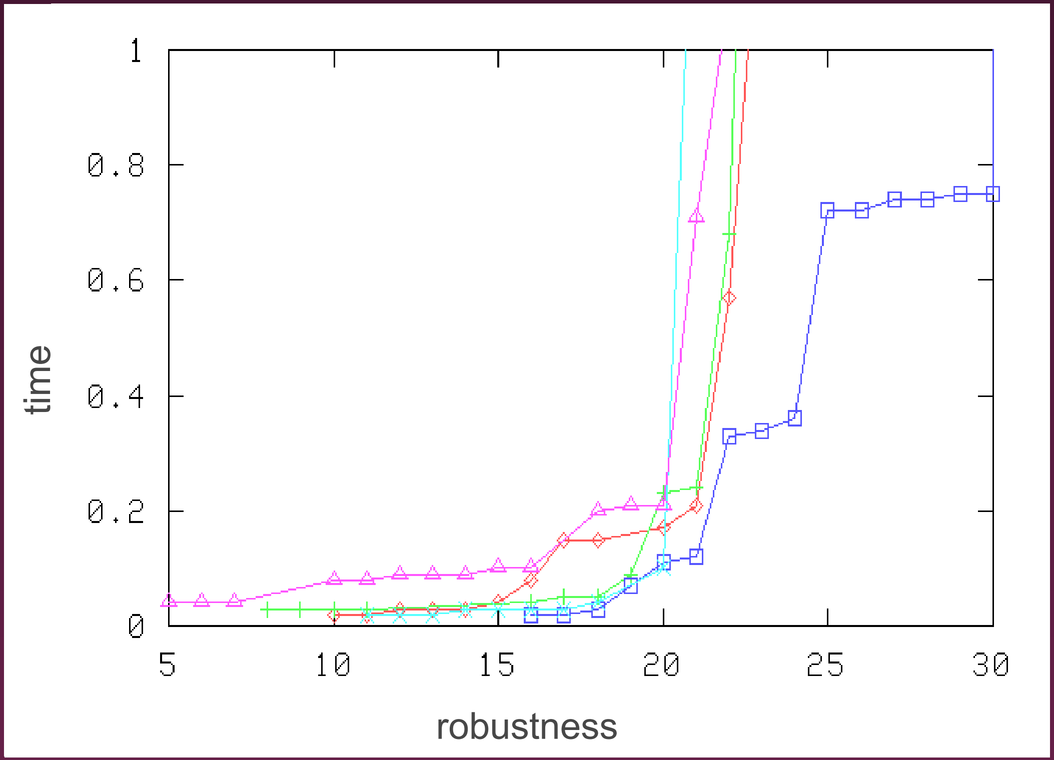
- initial solutions often brittle

Achievable Robustness



- initial solutions often brittle
- more robust solutions available cheaply

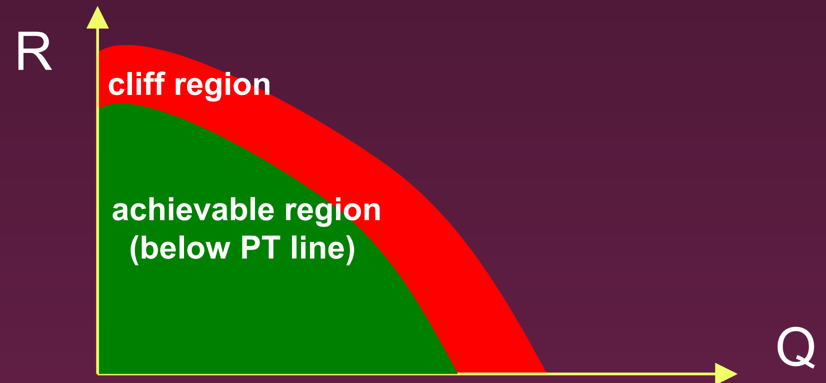
Achievable Robustness



Complexity arguments identify robustness as a reasonable computational goal in this environment.

Good and Robust

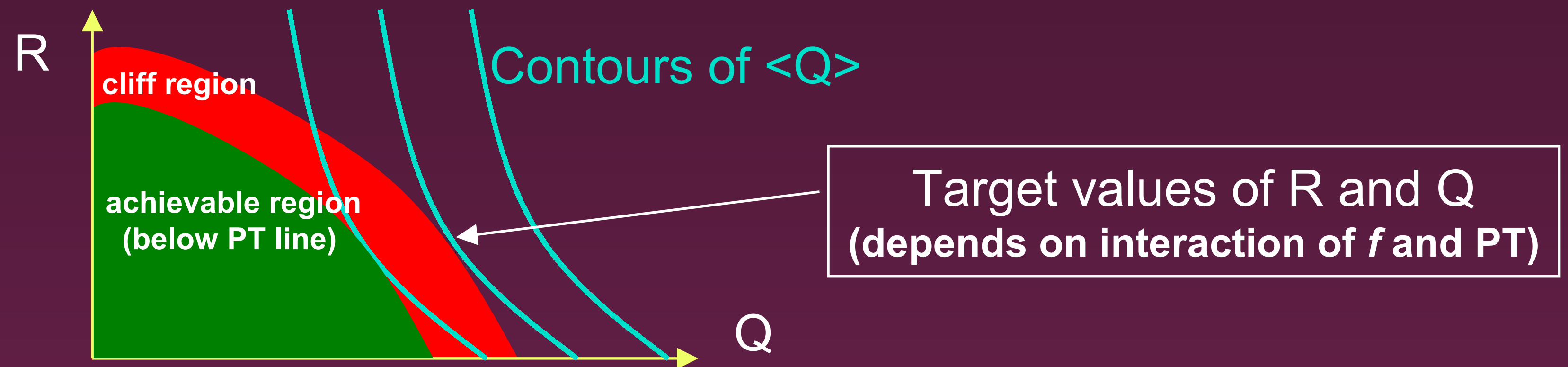
Q = nominal solution quality; R = solution robustness



Good and Robust

Expected solution quality (utility): <Q> = *f*(R,Q)

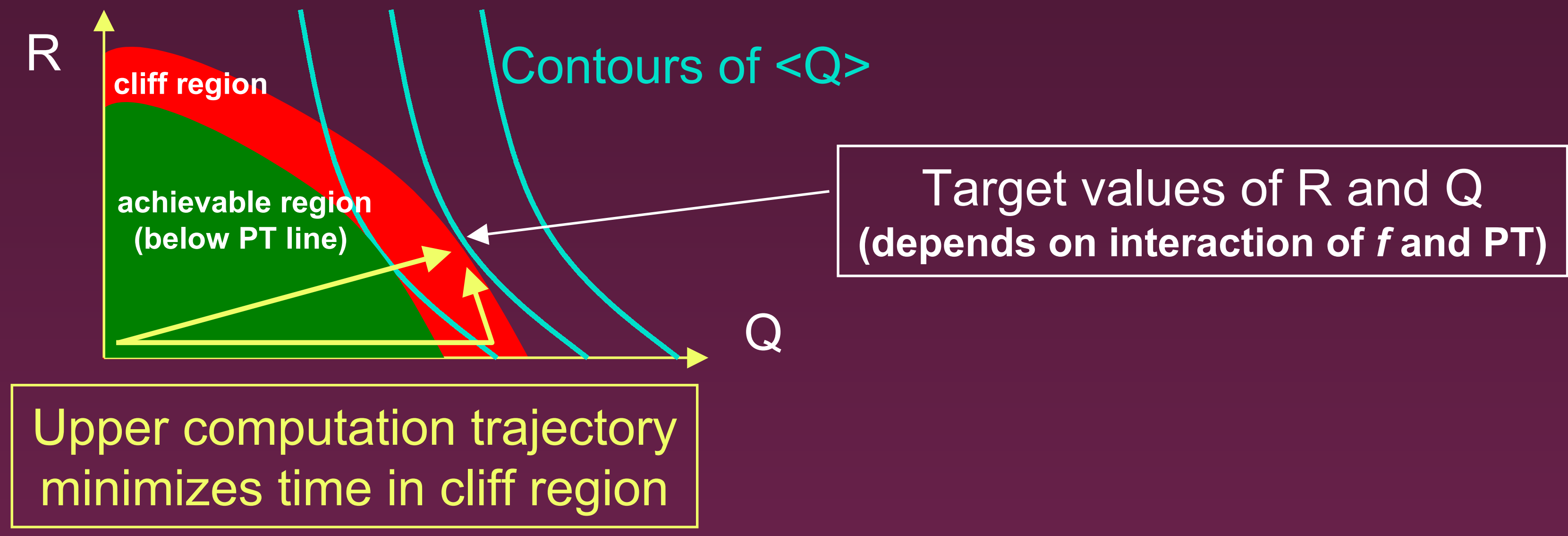
Q = nominal solution quality; R = solution robustness



Good and Robust

Expected solution quality (utility): <Q> = *f*(R,Q)

Q = nominal solution quality; R = solution robustness



- Robustness must be a goal from the outset.
- Complexity arguments underlie this insight.

Summary

Focusing on robust solutions:

- enhances self-reliance *and* cooperation
- aids in coping with lack of knowledge
- reduces communications demands

Phase transitions and computational cliffs:

- help predict achievable robustness
- guide anytime increases in robustness
- help manage “be robust” *vs* “act in time” conflict

What's Next?

Apply results to challenge problems

- incorporate into negotiation strategies

Focus robustness on most likely situations

  – e.g., unpredictable tracks in sensor challenge problem

- apply scenario-based methods (from OR)

Coalition structure and robustness strategies interact

- use breakdowns in robustness and decoupling to drive adjustments to coalition structure