**CAT Scan**: Constraint-based Approaches to Time-bounded Synthesis, Customization and Adaptation in Networked Embedded Systems

Weixiong (Wayne) Zhang *(PI)*
Ron K. Cytron *(co-PI)*

Computer Science Department
Washington University in St. Louis

http://www.cs.wustl.edu/~zhang/projects/catscan
Composition of the Team

委宣传ong (Wayne) Zhang, PI

- **AI**
  - Heuristic search (algorithms and complexity)
  - Phase transitions and problem structures
  - (Multi-agent systems)
- **Combinatorial optimization and algorithms**
  - e.g. the Traveling Salesman Problem
  - Parallel algorithm
- **Computational biology**

**Ron K. Cytron, co-PI**

- **Software**
  - Compiler
  - Middleware – TAO, ACE
- **Real-time systems** (real-time Java)

**Ph.D. level researcher, postdoc, graduate students**
Specific Problems Addressed (1)

- Combinatorial explosion in NEST
  - Most important problems are intractable – NP-hard
  - Distributed environment makes them even more difficult
  - Pervasive throughout design, development and deployment

- Integration of system analysis, design and efficient real-time problem-solving strategies
  - Use analysis/design information in real-time environments

- System reconfiguration and stabilization
  - Dynamically reconfigure in response to real-time changes
  - Dynamically reallocate system resources over time
  - Maintain system stabilization during dynamic reconfiguration
Outline of Our Approach

Dynamic NEST Systems

Constraint Models

Constraint-Based Modeling, Analysis, characterization, and problem solving for Real-time Synthesis in NEST

Phase Transitions

Phase-aware Problem Solving

Phase Diagrams

Real-time Synthesis

Improved Real-Time Performance

Phase-aware Problem Solving

Real-time System Response
Constraint-based Modeling (1)

- NEST system as a system of constraints
  - **Variables (Nodes)**
    - MEMS components: actuators, sensors, etc.
    - Information processing units, data storage devices, etc.
    - Coordination processes: middleware and other supporting components
  - **Constraints (links)**
    - Connections
    - Logical dependencies
    - Association relationships
  - **Weights of constraints**
    - Preferences
    - Importance and/or rewards
    - Costs and/or penalties
Constraint-based Modeling (2)

Goal: Constraint satisfaction and minimization

- **NEST** systems usually operate under pressure
  - Hard constraints (must be satisfied)
  - Soft constraints (may be violated but with penalties)

Finding a solution (variable assignments) to minimize the total weight of the constraints violated

Constraint satisfaction is a special case (where the system is underconstrained)

The problem is NP-complete and/or NP-hard!

- Special case (prepositional satisfiability) is difficult
  - 3-SAT is NP-complete (decision problem)
  - 2-SAT minimization is NP-hard (optimization problem)
Outline of Our Approach

Dynamic NEST Systems

Constraint-Based Modeling, Analysis, characterization, and problem solving for Real-time Synthesis in NEST

Phase Transitions

Phase-aware Problem Solving

Constraint Models

Phase Diagrams

Improved Real-Time Performance

Real-time Synthesis

Real-time Problem Solving Dimension

Satisfiable solutions, polynomial complexity (finding solutions directly)
Unsatisfiable solutions, exponential complexity (using problem transformation methods)

Unstable system configurations

System design dimension

Real-time problem solving dimension

Flexible, real-time, response

NEST Kickoff Meeting, Napa Valley, CA, 6/6/01

Washington University / CAT Scan
Complexity Analysis/Characterization

- NP-completeness is a worst-case measure
  - It only tells what it could be at one extreme
  - Worst cases are rare
- We are interested in complexity under different conditions (parameters)
  - Complexity under different constraint conditions
  - Average-case (or typical-case) complexity
- System characterization (emerging behavior)
  - Phase transitions
  - Backbones
Phase Transitions (1)

- A system’s property changes qualitatively and dramatically when certain parameters pass through critical values (order parameters).

- Examples
  - Melting of a solid (temperature)
  - Hamiltonian cycles in random graphs (node connectivity)

- Characteristics
  - Large number of components
  - Local, small actions/interactions collectively determine a global property/behavior
Phase Transitions (2)

Phase transitions in combinatorial optimization
(Karp & Pearl, 83; McDiarmid, 90; Zhang & Korf, 92, 93, 95; Zhang 99)
- State-space search for solving combinatorial problems
- Control parameter: expected number of children with the same cost as their parent – $bp_0$

\[ \begin{align*}
1 & \quad 2 \\
3 & \quad 4 & \quad 2 & \quad 4 \\
5 & \quad 5 & \quad 4 & \quad 6 \\
8 & \quad 6 & \quad 7 & \quad 9 & \quad 4 & \quad 0 & \quad 3 & \quad 0 & \quad 4 \\
\end{align*} \]

- $b$: mean branching factor
- $p_0$: prob. that an edge has cost zero
- $bp_0$: expected # of children having the same cost as their parent
Phase Transitions (3)

Phase diagram for combinatorial search problem

- $b$: mean branching factor
- $p_0$: prob. that an edge has cost zero
- $b p_0$: expected # of children having the same cost as their parent

Easy-hard transition in optimal tree search

- $b p_0 \geq 1$: polynomial region
- $b p_0 = 1$: transition boundary
- $b p_0 < 1$: exponential region
Phase Transitions (4)

Application - The asymmetric TSP (ATSP) (Zhang&Korf, 96)

- The average complexity of ATSP with distances from \{0,1,2,\ldots,r\} experiences phase transitions as r increases.
- The ATSP with distances 1 and 2 is NP-complete (Karp, 72)
- 1-2 distance ATSP is also NP-hard to approximate (Papadimitriou&Yannakakis, 93)

AP: assignment problem – cost to evaluate a node in search space
Phase Transitions (5)

3SAT: \( F = (x_1 \lor \neg x_2 \lor x_3) \land (x_2 \lor \neg x_4 \lor \neg x_6) \land \ldots \)

- Phase transition in 3SAT (decision problem) (Cheeseman, et. al. 91; Mitchell, et. al. 92)
- Phase transition in MAX 3SAT (optimization)

![Graph showing the easy-hard-easy transition in 3SAT](chart1.png)

![Graph showing the complexity of 3SAT and MAX-3SAT](chart2.png)
Phase Transitions (6)

Connection between Decision and Optimization
- Quality vs. Complexity Tradeoff (Zhang, 2001)
Phase Transitions (7)

**Backbone**: A set of variables having fixed values in all optimal solutions.
- Backbone is a special structure – critically constrained! – violating a backbone variable rules out all optimal solutions

- Backbone phase transition
- Correlation of two transitions

![Graph 1](image1)
![Graph 2](image2)
Phase Transitions (7)

- **Short term research topics**
  - Identifying backbone and use it to speed up search
  - Search in large problem space but with extremely small memory
  - Variable resolution search by exploiting phase transitions

- **Intermediate/long term topics**
  - Prediction of global property from local interactions
  - Distributed algorithms by exploiting phase transitions
Outline of Our Approach

Dynamic NEST Systems

Constraint Modeling

Constraint-Based Modeling, Analysis, characterization, and problem solving for Real-time Synthesis in NEST

Constraint Models

Phase Transitions

Phase-aware Problem Solving

Phase Diagrams

Dynamic, Real-time, fast responses

Real-time Synthesis

Improved Real-Time Performance
Phase Aware Problem Solving (1)

✦ Most problems in practice/NEST are difficult
✦ What can we do with a problem in the difficult region?
Phase Aware Problem Solving (2)

Main ideas:

- Exploiting quality/complexity tradeoff
- Exploiting problem structures and phase transitions

Approaches (Problem Transformations)

- Transformations in search space
  - Parametric transformations ** (Zhang&Pemberton, 94, 96)
  - Structural transformations (Zhang, 98, 2001)
- Optimization- to decision-problem transformation
  - Transforming an optimization problem into a series of decision problems
- Hierarchical constraint models (constraints of different roles to be considered in different iterations)
Phase Aware Problem Solving (3)

Parametric Transformation (Zhang&Pemberton, 94, 96)

- Treat a difficult problem in the exponential region as if it is an easy one in the polynomial region
Phase Aware Problem Solving (4)

- Iterative parametric transformation
- It improves real-time performance

\[ p = \text{probability of zero-cost edges} \]

\[ p_c, p_e \]

polynomial region

exponential region

\[ 0, 5, 10, 15, 20 \]

mean branching factor \( b \)

500-city ATSP \( c(i,j) \) uniformly chosen from \( \{0,1,2, \ldots ,2^{16}-1\} \).
Outline of Our Approach

Dynamic NEST Systems

Constraint Models

Constraint-Based Modeling, Analysis, characterization, and problem solving for Real-time Synthesis in NEST

Phase Transitions

Phase-aware Problem Solving

Phase Diagrams

Improved Real-Time Performance

NEST Kickoff Meeting, Napa Valley, CA, 6/6/01

Washington University / CAT Scan
Phase Diagrams

Phase diagram: A collection of information, in a compact format, of system behavior/performance and best methods under different operation conditions.

Making information from static analysis and that used in design and development available to real-time operations – Integration of system design, development and deployment.

Performance analysis dimension

Real-time decision dimension

System design dimension

\[ F(\text{network speed}, \text{cpu speed}) \]

- Satisfiable solutions, polynomial complexity
- Unsatisfiable solutions, exponential complexity
- Unstable configuration

Nonsystematic search

Systematic search

algorithms
Adaptation

Ỡ Self reconfiguration
  🍀 Dynamic CSP/CMP models
    ✽ Variables and constraints in constraint model can be added and removed dynamically when conditions change and/or system reconfigures itself
    ✽ Complexity analysis and new algorithms

Ỡ Self stabilization
  🍀 View the problem as a graph search problem
    ✽ Nodes are viable system configurations and neighboring configurations are connected with edges.
    ✽ Search for a shortest path such that the system is stable at each reconfiguring step
Tasks and Deliverables

- Constraint analysis and constraint-based models (1.1)
  - Analytical results and models of selected NEST problems
- Complexity and phase transition analyses (1.2)
  - Complexity results on CSP and CMP
- Phase-aware, real-time problem-solving methods (1.3)
  - Implemented algorithms in C/C++/Java
- Phase diagrams (1.4)
  - Tables for selected NEST problems
- Software system for real-time synthesis (1.5)
  - Combining tasks 1.1-1.4 and methods for model building and selecting specific algorithms/methods and using phase diagrams
- Self reconfiguration and stabilization methods (2.1)
  - Implemented dynamic CSP/CMP models and algorithms
- An application system (2.2)
  - A working system for selected NEST applications
Schedule and Milestones

Task 2.2: application
Task 2.1: reconfig
Task 1.5: integration
Task 1.4: phase diagram
Task 1.3: aware method
Task 1.2: complexity
Task 1.1: models

Milestone 2: Phase transitions
Phase-aware methods
Most phase diagrams
Initial integration with others

Milestone 3: working system
With all algorithms/methods;
Demo on NEST challenges

NEST Kickoff Meeting, Napa Valley, CA, 6/6/01
Washington University / CAT Scan
Integrating our Results with Others

**Application-level adaptation**
- Help to decide what is solvable or reachable within given resources (computational power and time)
- Help to select the best problem-solving strategies and decision-making methods
- Using both Boeing and Berkeley OEPs

**Middleware adaptation** (with Boeing)
- Middleware reusability
  - Formalize adaptation solutions using patterns and frameworks
  - Implement using CORBA service model
- Middleware customization (footprint reduction)
Adaptation as a CORBA Service

Applications use CORBA (TAO) objects as representatives
Adaptation as a CORBA Service

- Applications use CORBA (TAO) objects as representatives
- Changes in environment are transmitted to the object using an Event Channel
Adaptation as a CORBA Service

- Applications use CORBA (TAO) objects as representatives
- Changes in environment are transmitted to the object using an Event Channel
- Adaptation algorithms are made available as a CORBA service
Adaptation as a CORBA Service

- Applications use CORBA (TAO) objects as representatives
- Changes in environment are transmitted to the object using an Event Channel
- Adaptation algorithms are made available as a CORBA service
- This obtains language independence, portability, and acceptance via a standard interface
Footprint Reduction

Background

• CORBA-compliant implementations handle all cases

• Footprint size of “all cases” code can be large

Diagram:

- Dispatch
- Types
- Event Channel
- Scheduling
- Filtering
Footprint Reduction

Background

- CORBA-compliant implementations handle all cases
- Subsetting is difficult

Specialization is possible

- Dispatch
- Event Channel
- Scheduling
- Filtering

Types

- collocated
- no “anys”

- reg expr filtering
- fixed priorities

Washington University / CAT Scan
Footprint Reduction

Background
- CORBA-compliant implementations handle all cases
- Subsetting is difficult

- Specialization is possible
- Feature accrual through aspects can yield smaller footprint

- collocated
- no “anys”

- reg expr filtering
- fixed priorities