Kestrel Institute: e-Merge-ANT

Stephen Fitzpatrick & Cordell Green (PIs)
Lambert Meertens

http://ants.kestrel.edu/

Kestrel Institute,
3260 Hillview Avenue,
Palo Alto,
California

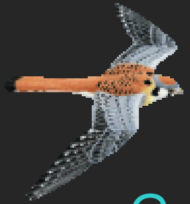
29-31 May 2002
Captiva Island, Florida

Administration

- Project Title: e-Merge-ANT: A Toolkit to Create Run-time Ant Generators, Aggregators and Synthesizers — and a Demonstration Application
- PM: Dr Vijay Raghavan
  PI, and co-Pis: Drs Stephen Fitzpatrick & Cordell Green
- PI(s) Phone Number and e-mail:
  - 650-493-6871
  - fitzpatrick@kestrel.edu & green@kestrel.edu
- Company/Institution: Kestrel Institute
- Contract Number: F30602-00-C-0014
- AO Number: J123
- Award Start Date: 26 January 2000
- Award End Date: January 2003
- Agent Name and Organization: Robert Paragi, AFRL/IFTD

Collaboration

- Univ. of Washington at St. Louis have been performing experiments on Kestrel's distributed constraint optimization algorithm (CFP):
  - phase transitions in activation probability
  - performance vs. Distributed Breakout Algorithm (Yokoo et al.)
    - DBA seems to be the main contender with CFP for distributed graph coloring
    - other variants of CFP are under investigation
      - e.g., choosing a better value each step, not necessarily an optimal value
  - so far CFP out-performs DBA in most cases
- Altarum have been looking at the stability/convergence aspects of CFP with respect to entropy
  - instability as a result of too many choices
  - Kestrel may be able to exploit such notions in techniques for adapting the activation probability



Problem Description & Approach

- Goal: real-time resource management in large sensor networks
  - support flexible, robust sensor networks containing thousands of sensors
- Distributed resource management is realized using resource-centric scheduling “agents”
  - a resource’s agent schedules the resource’s actions over medium-term
  - scheduling helps to overcome latency: predict rather than react
- Scalability is achieved by making use of only local interaction
  - objective is to achieve good, local sensor coordination everywhere
    - thesis is that good global behavior “emerges”
  - per-resource costs are determined by the number of a resource’s neighbors
- Real-time adaptivity is achieved by continual, anytime rescheduling
  - each agent continually adapts to changes in the local environment
  - adaptation optimizes local, soft constraints
    - less vulnerable to phase-transitions
- Robustness is achieved through pervasive, local interaction
  - local failures have local consequences/repairs
  - small changes to the environment are absorbed by continual rescheduling

Success Criteria & Metrics

- High-quality solutions: scanning must be better than greedy/random
  - measure quality obtained using variety of control schedulers
    - solution quality reflects success in scanning targets
- Responsiveness: solution adapts quickly to environmental change
  - measure how time-averaged solution quality varies with rate of change of environment
- Stability: adaptation should be commensurate with size of change
  - measure how size of change in solution varies with size/rate of environmental change
    - thrashing (continual, widespread change without improvement) should be avoided

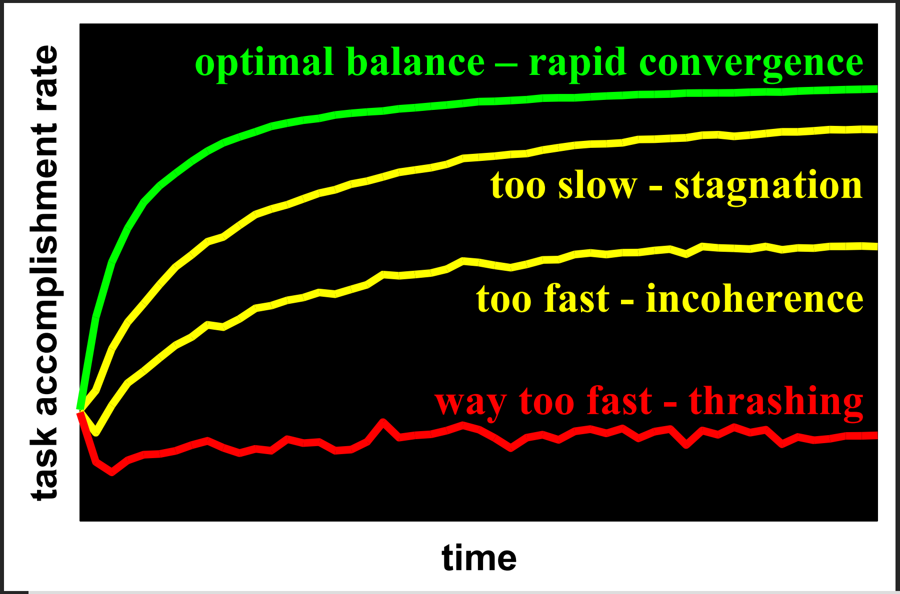
(*cont*.)

Success Criteria & Metrics (cont.)

- Scalability in costs: bounded per-node costs as number of nodes grows
  - measure how time-averaged per-node costs vary with network size
    - computational, storage & communication costs
- Scalability in quality of solution: maintained as number of nodes grows
  - measure how time-averaged solution quality varies with network size
- Robustness: costs and solution quality approximately maintained as small fraction of nodes fail
  - measure how costs & solution quality vary with extent/rate of failure
    - graceful degradation as failure rate increases moderately
- Robustness: failure should be localized
  - measure size of region affected by localized failures

e-Merge-ANT

Trade-off between adaptation rate and coherence in distributed resource management



Impact

Enables cost-effective deployment of large networks of simple, low-cost sensors

- **Real-time** responsiveness guaranteed
- **Scalability** to arbitrarily large networks
- **Mission success** rate improved (more tasks accomplished by resources)
- **Low overheads** for resource management even under dynamically varying loads

New Ideas

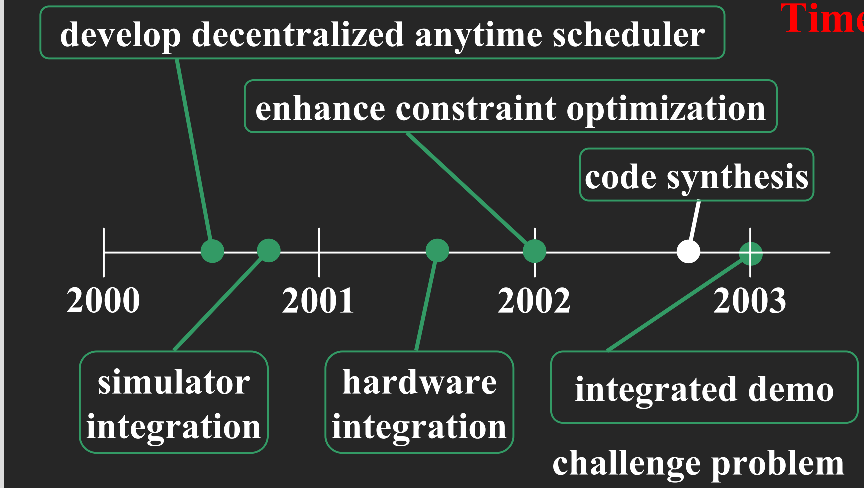
Decentralized, localized, anytime scheduling for resource management in large, distributed networks

Reflective scheduling explicitly accounts for time and resources appropriated by scheduler

Stochastic soft-constraint optimization balances response time against coherence to ensure convergence and maintains tolerable computational costs under dynamic task loads

Formal modeling & synthesis techniques produce analyzable, reusable & transferable components

Timeline



develop decentralized anytime scheduler

enhance constraint optimization

code synthesis

simulator integration

hardware integration

integrated demo

challenge problem

Kestrel Institute: Stephen Fitzpatrick & Cordell Green

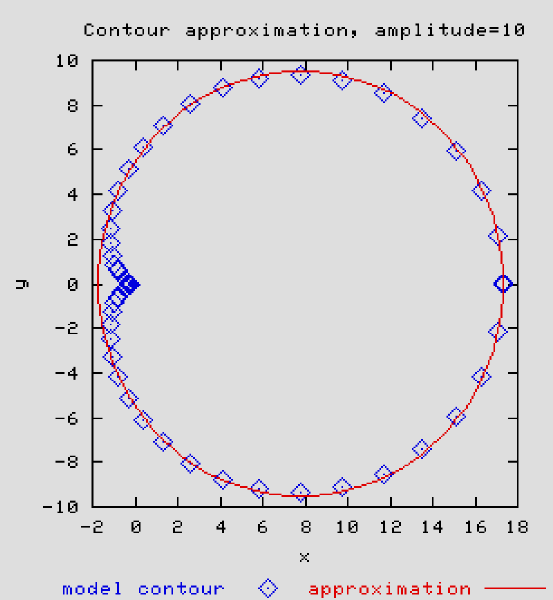
Project Status: Update

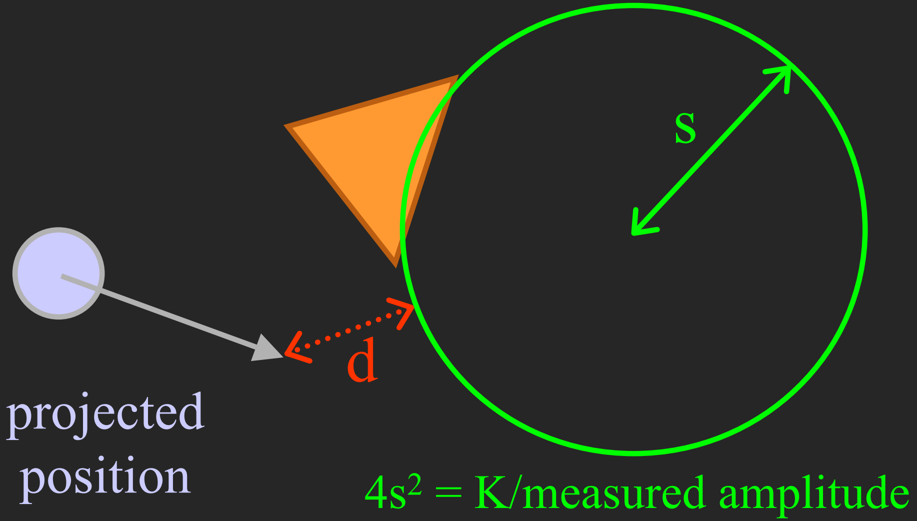
- Improved approach to challenge problem:
  - designed & implemented new metric to support track generation & update
    - should avoid excessive track generation problem previously encountered
  - designed & implemented new metric on scan schedules
    - used by scheduling agents to coordinate sensors
    - based on scan schedules of nearby nodes, local target estimates, local measurements and measurements from nearby nodes
  - new communication strategy: stochastic squirts
    - short messages at essentially random times
    - local locks to prevent transmitter-scanner interference
  - details later
- New dynamics experiments on abstract constraint problem (graph coloring)
  - investigated effect of phase transitions on anytime solution quality
  - investigated effect of asynchronous execution/communication latency
  - investigated effect of network interconnection density
  - details later

Challenge Problem

New Measurement-Target Association Method

- Need to associate measurements with targets (known or new)
  - based on a measurement-target “distance” metric
- Improved metric
  - project target’s position to time at which measurement was taken
  - measurement-target distance is the shortest distance between amplitude contour and projected position
    - amplitude contour = {(R, ϕ): amp(R, ϕ) = measurement’s amplitude}
  - for efficiency, approximate the contour with a circle



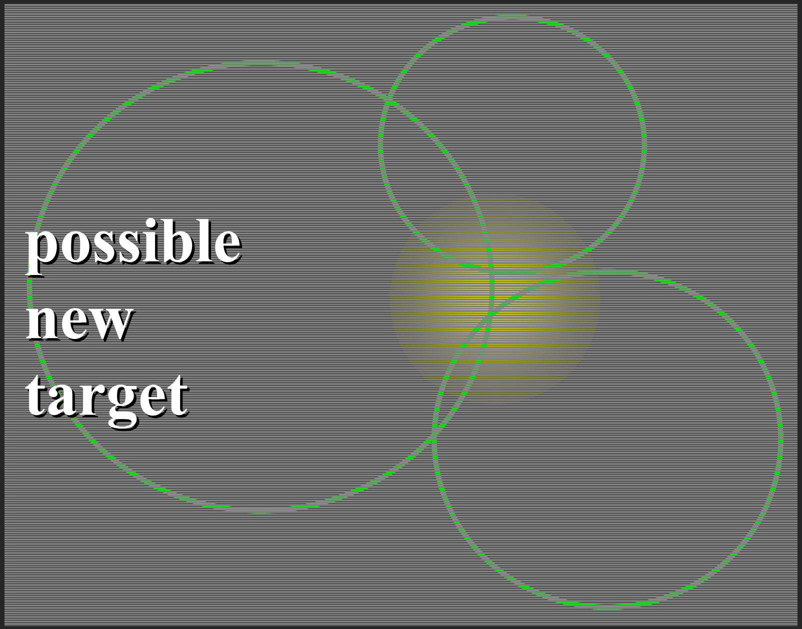


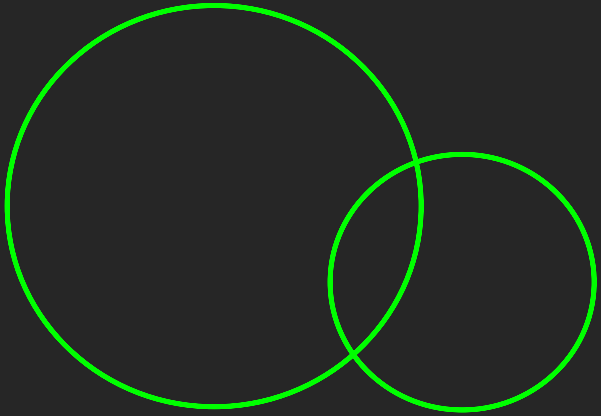
projected position

$4s^2 = K/\text{measured amplitude}$

New Target Detection Method using Contour Clustering

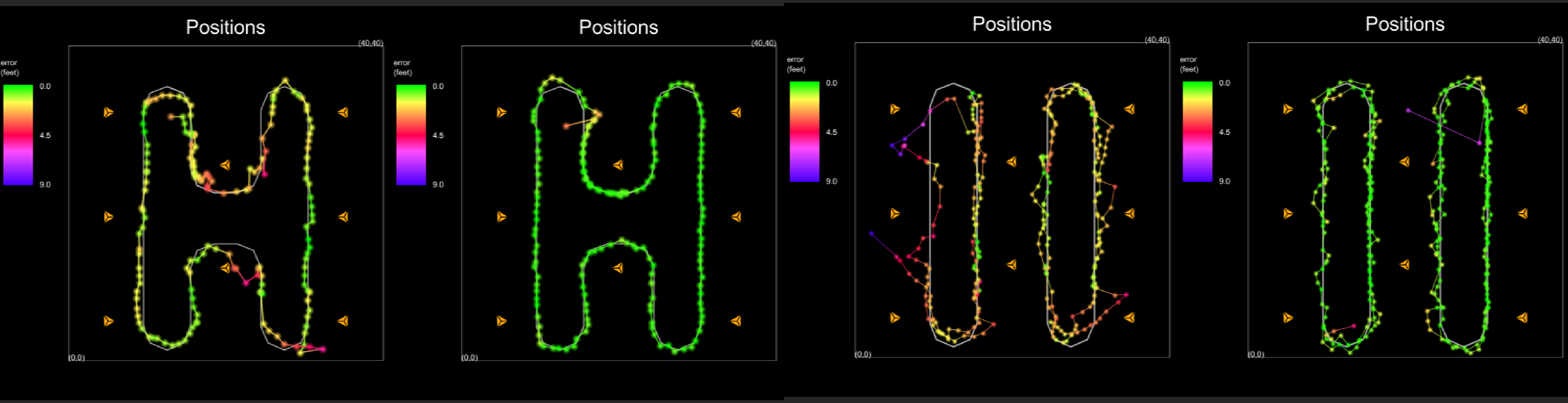
- Identify cliques of (approximately) simultaneous, unassociated measurements that have (approximately) intersecting contours
- For each clique of size > 2, generate a new potential target
  - position initialized using grid-search technique of BAE tracker
  - velocity initialized to zero
  - target becomes “real” if and when it is subsequently updated with new measurements





Results of New Association Method

- High-quality tracking achieved with Radsim and **old** noise model



BAE: 1.6 **SC: 0.4** **BAE: 3.0** **SC: 1.3**

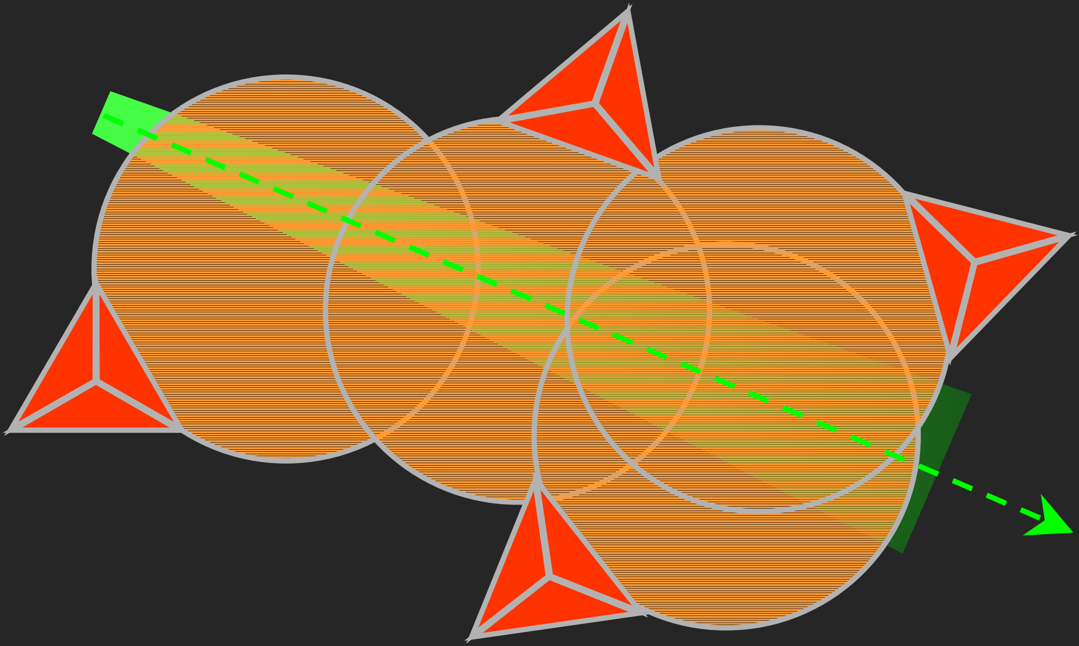
rms error/ft

Context for New Metric on Scan Schedules

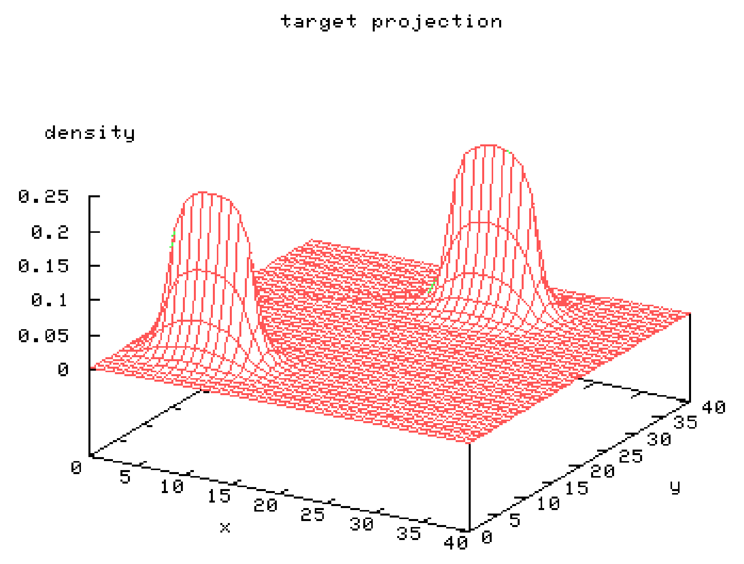
- Main idea:
  - determine where & when the sensors should scan based on measurements & target projections
  - given scan schedules, determine how well the sensors collaborate in scanning each point in (x, y, t)
  - optimize trade-off between:
    i) average quality of scanning, weighted by where targets are expected to be
    ii) operational costs and penalties for constraint violations
- Abstract approach: optimize expected values over evolution of multiple-target probability distribution
  - probability distribution may reflect uncertainty arising from noisy measurements
  - or it may reflect targets that are only quasi-predictable (e.g., that may turn)
  - computationally expensive
    - lots of multi-dimensional integrals

New, Cheaper Approximation

- Compute a “target density field” $F(x, y, t)$ that represents the expected number of targets at each position-time coordinate
- Compute a “scanning field” $S(x, y, t)$ that represents the collaborative quality of scanning that results from individual sensor’s schedules
- Compute overall quality of schedules as the “inner product” $(F.S)$
- Operational costs and constraint penalties are straightforward
- Optimize schedules w.r.t. w_q.quality – (w_c.cost + w_p.penalties)



Computing the Target Field: Tracker



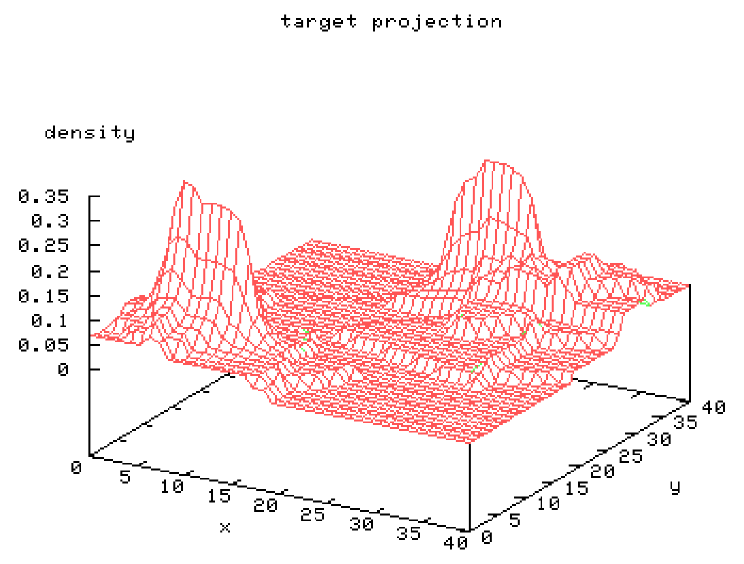
- given a target i at position (x_0, y_0) with velocity (u_0, v_0) at time t_0,
- project its position to time $t_0+\Delta t$ as $(x_0+u_0.\Delta t, y_0+v_0.\Delta t)$
- program's trackers do not yet publish confidence intervals: use a nominal width w $F_i(x, y, t_0+\Delta t) = e^{-d/w}$ where $d = |(x, y) - (x_0+u_0.\Delta t, y_0+v_0.\Delta t)|$

- Each target is linearly projected based on current estimates of its position & velocity
- In simplest case, the total field F is computed as the point-wise addition of F_i over targets i
- Can allow for multi-target interference:

$$F(x, y, t) = \max_i 2F_i(x, y, t) - \Sigma_i F_i(x, y, t)$$

Computing the Target Field: Raw Measurements

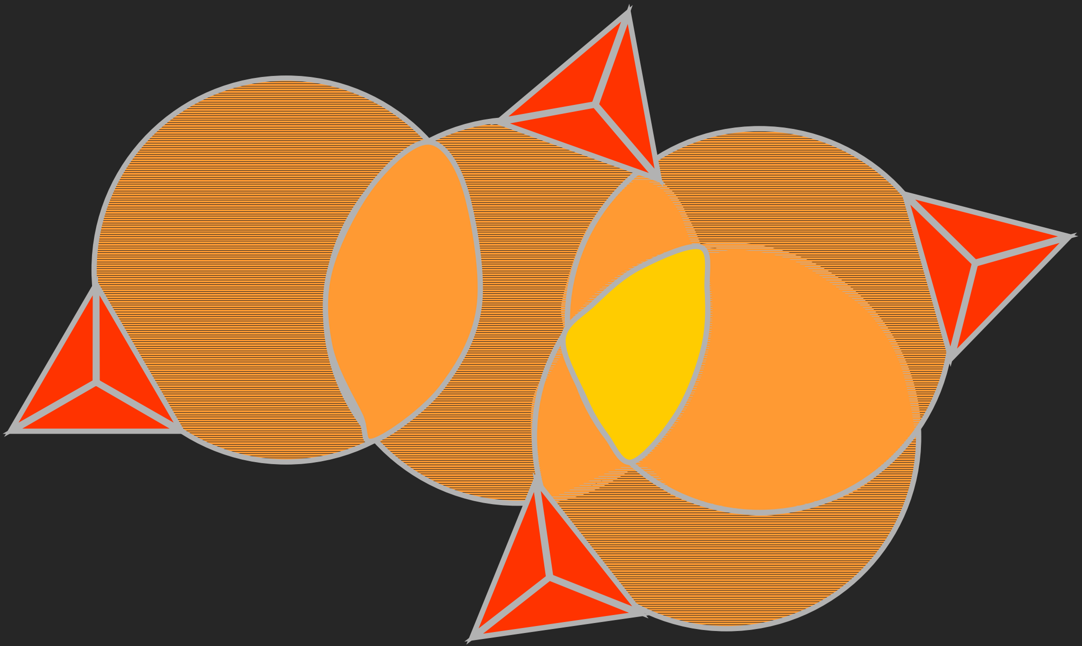
- given a single measurement m at time t_0:
  - $M(m, x, y) = \alpha(m.\text{amplitude})$
    if m is non-noise and (x,y) is in m.sector,
    where α is some monotonic function
  - $M(m, x, y) = -\alpha_0$
    if m is noise and if (x, y) is in m.sector,
    where α_0 is a (positive) constant
- measurement information devalues when projected
  - $M(m, x, y, t_0+\Delta t) = e^{-|\Delta t|/\beta} M(m, x, y)$,
    where β is a constant



- A single non-noise measurement tells us with high probability that there is a target within the scanned sector
  - so increase the target field in the area corresponding to sector
- A single noise measurement tells us the opposite
  - so decrease the target field in the area corresponding to sector
- Measurements represent cruder and less predictive, but more reliable information (en masse)
  - it is hard to project raw measurements over time

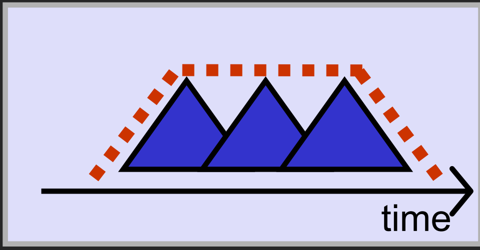
Scanning Field

- Scan schedules determine how well each point in space-time is scanned
  - compute combined scan quality at each point, resulting from individual scanning “fields”



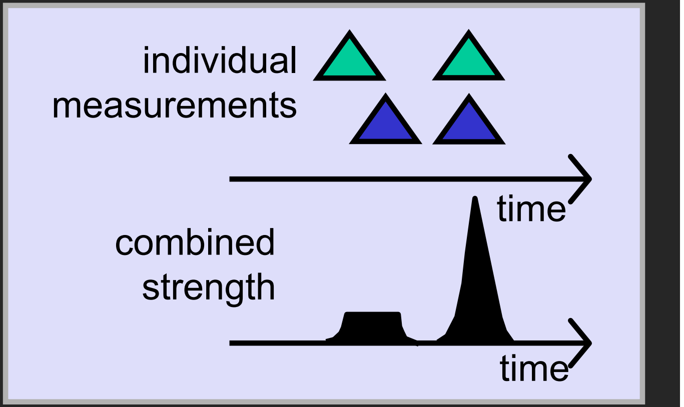
Computing the Scanning Field: Single Sensor

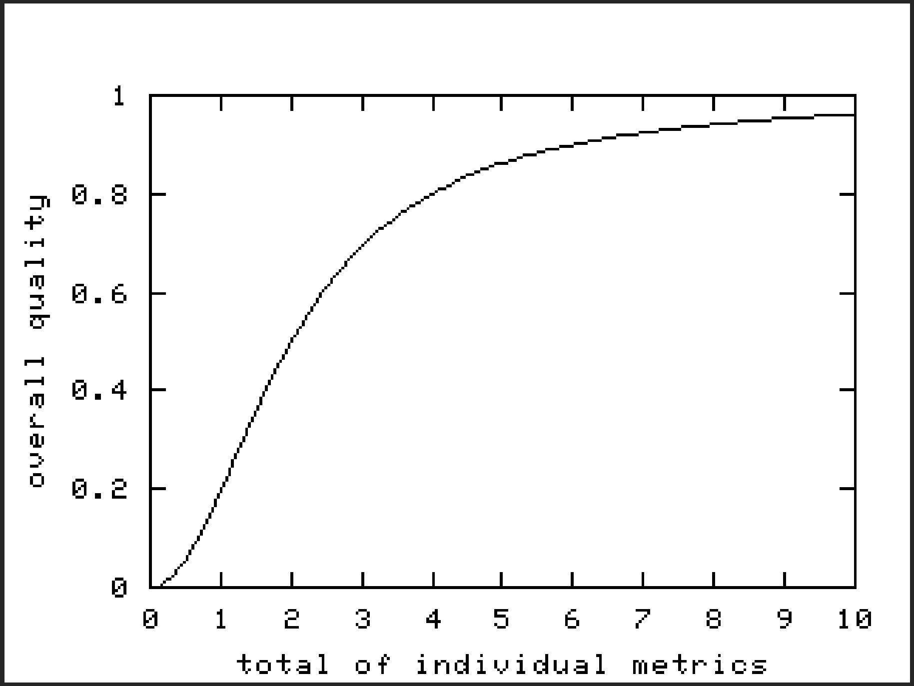
- For a single scan, the time-agnostic signal strength is
  - $amp(R,\phi) = K.exp(-A\phi^2)/R^\gamma$,
    where R is the distance from the sensor,
    f is the angle from the sensor's mid-beam,
    and K, A, γ are parameters of the sensor
- Take the quality to be some monotonic function of signal strength
  - e.g., $quality(R,\phi) = log(amp(R,\phi))/log(max\_amp)$
- Quality decays over time
  - $quality(R,\phi,t_0+\Delta t) = quality(R,\phi).max(0,\ 1-|\Delta t|/T)$
    where t_0 is the time the scan is scheduled to take place and T is some reasonable time period (e.g., 2 seconds)
- For any given time t, compute combined quality of a given sensor's scans by adding the qualities associated with individual scans
  - $totalQuality_n(R,\phi,t) = \Sigma_j\ quality_{nj}(R,\phi,t)$ for sensor node n



Scanning Field: Multiple Sensors

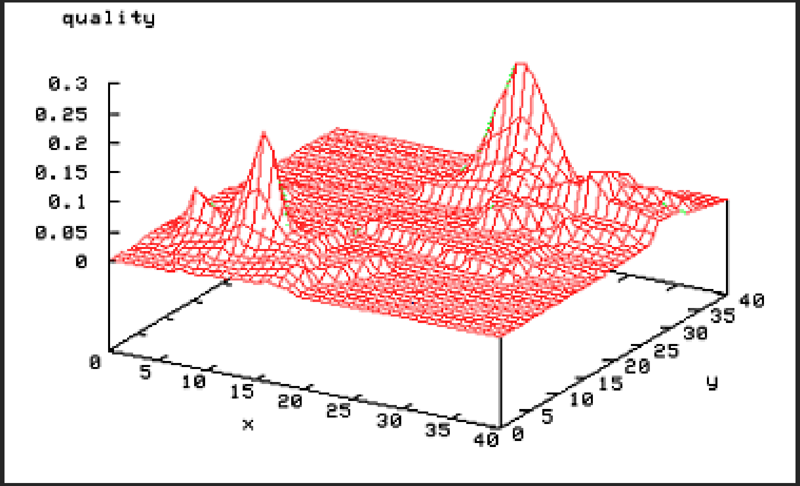
- Compute the overall quality of scanning at some point (x,y,t) by combining the individual quality metrics for each sensor
  - if there are two or three sensors with high quality, award a high score
  - if there are more sensors with high quality, award only slightly higher score
    - 10 sensors scanning a single target is only slightly better than 3 sensors
      - and costs a lot more
    - sensor energy can, perhaps, be better expended on other targets



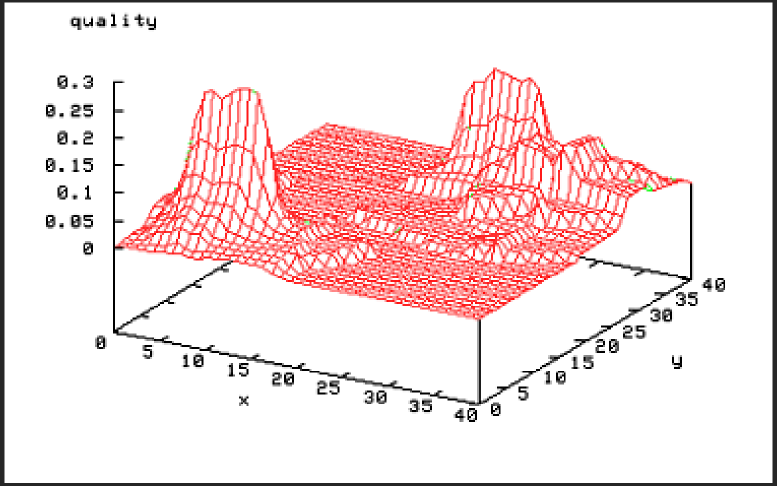


Overall Metric

- Overall scanning quality is the average over x,y,t of the point wise product of the target and scanning fields
- Overall operational costs is the average over t of the number of active sensors
- Overall constraint violation penalties easily computed
- Combined metric: w_q.quality – (w_c.cost + w_p.penalties)



random schedules



optimized schedules

Distributed Scan Schedule Optimization: Implementation for May Demonstration

- Each sensor periodically broadcasts its measurements and schedule
- Each sensor uses its neighbors' and its own measurements to compute local target trajectories
- Each sensor computes a local version of the global metric:
  - it computes fields over the region it can scan
  - it computes target density & scanning fields using the targets and measurements it knows about
- Each sensor optimizes its own schedule
  - with respect to the last schedules it has received from its neighbors
  - *details on next slide*
- Each sensor computes a locally-optimal schedule
  - initial experiments indicate that global quality is also good
  - though maybe not truly optimal
  - extensive experiments planned – details later

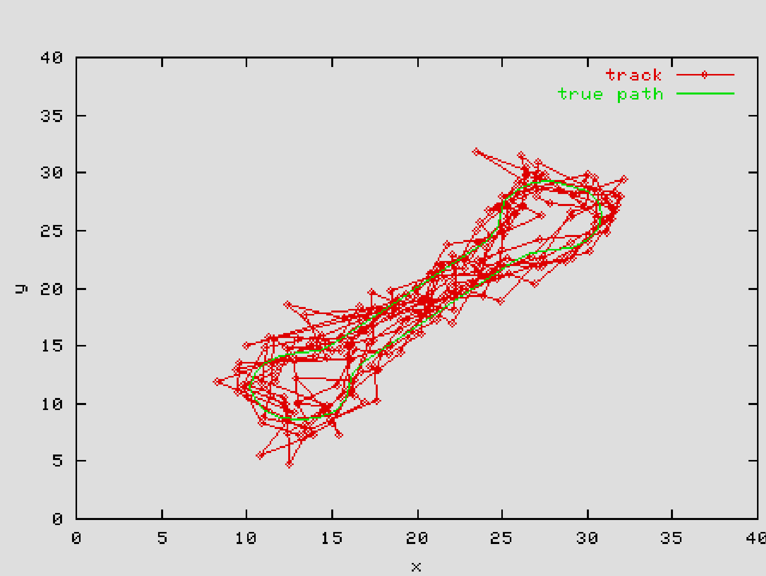
How Each Sensor Optimizes Its Scan Schedule

- Each node randomly initializes its scan schedule
- After initialization, each node periodically tries to improve its schedule using hill-climbing
  - a scan schedule is represented as a finite map from time slot to sector
  - in a single schedule, all scans have the same mode
    - amplitude-only or amplitude-and-frequency
  - a schedule may be transformed in a single hill-climbing step by changing the sector scanned in one time slot

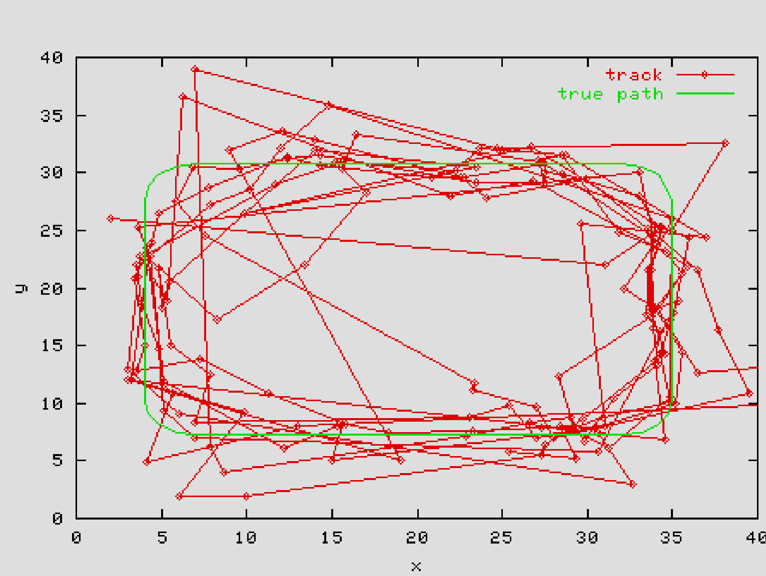
scan begin time/seconds	0.0	2.0	4.0	6.0	8.0
original schedule's sectors	0	0	1	2	0
transformed schedule's sectors	0	2	1	2	0

Simulator Results with New Noise Model

- Inner path is tracked well by itself
  - low error: self-evident, but don’t have the numbers yet
  - long tracks: 2 tracks are shown, having 80 track points & 184 track points
  - frequent updates: about every 2 seconds, on average



- Outer path is less well tracked, even by itself
  - higher error
  - shorter tracks: 15 tracks shown, total of 180 track points
  - updates less frequent: about every 4 seconds
  - periods when target is lost



- For both paths together, error in tracking outer path throws off tracking of inner path
- No attempt to reduce energy usage

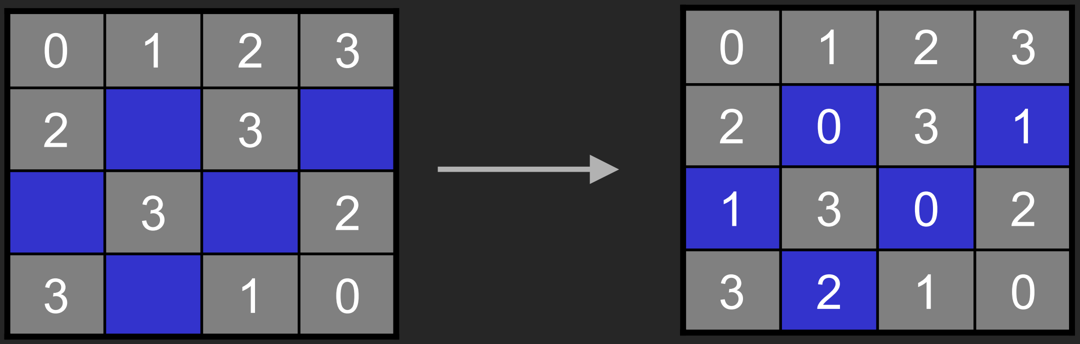
Abstract Dynamics

Simplified Test bed for Dynamics: Synchronous, Distributed Graph Coloring

- Abstract constraint problem: distributed, approximate graph coloring
  - objective is to minimize color conflicts
    - i.e., edges connecting nodes of the same color
  - degree of conflict is the fraction of edges that are conflicts
    - low is good
- Each node repeatedly executes the following computations synchronously:
  - the node activates if both of the following hold:
    - it currently has at least one color conflict with a neighbor
    - a randomly generated number is below some constant activation probability α
  - if it activates, the node chooses a color that minimizes its conflicts
    - based on what it currently believes are its neighbors’ colors
    - there may be several optimal colors – one is chosen at random
  - if the node changes color, it transmits its new color to its neighbors
    - there is always a communication latency of exactly 1

New Result: Effect of Phase Transitions

- Experiments on coloring partial Latin squares (Gomes *et al.*)
  - a Latin square of order k is a k x k grid of nodes
  - each node is to be assigned an integer in the range 1 to k so that each integer occurs exactly once in each row and in each column
- A partial Latin square is a Latin square with some nodes' values erased
  - the problem is to complete the square by assigning values to the unassigned nodes
- Problem difficulty exhibits a phase transition in the fraction of holes
  - number of ways of completing the partial square increases as the number of holes increases
  - sudden change observed in the number of completions at critical fraction of holes



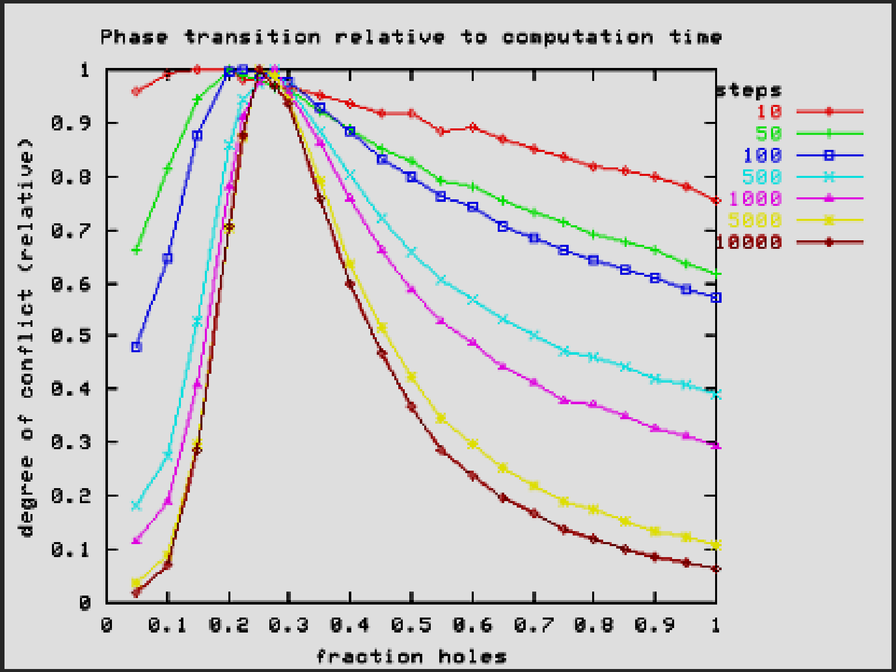
0	1	2	3
2		3	
	3		2
3		1	0

→

0	1	2	3
2	0	3	1
1	3	0	2
3	2	1	0

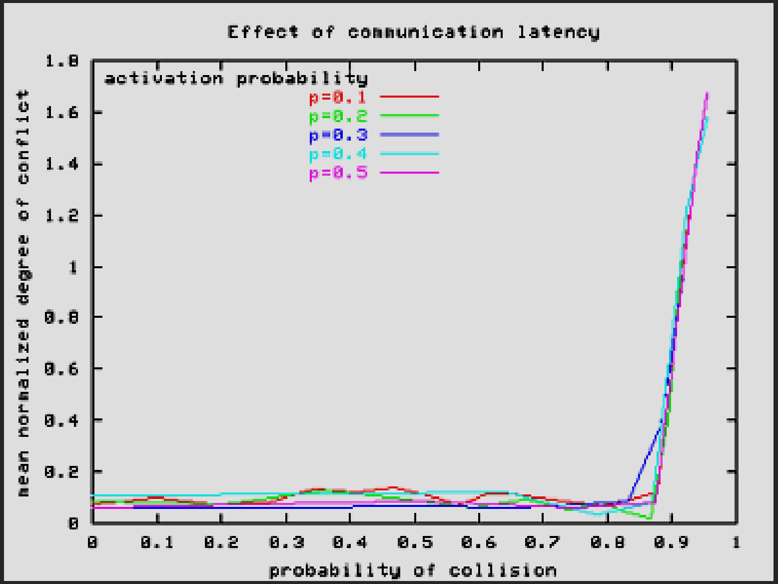
Reduced Severity of Phase Transition

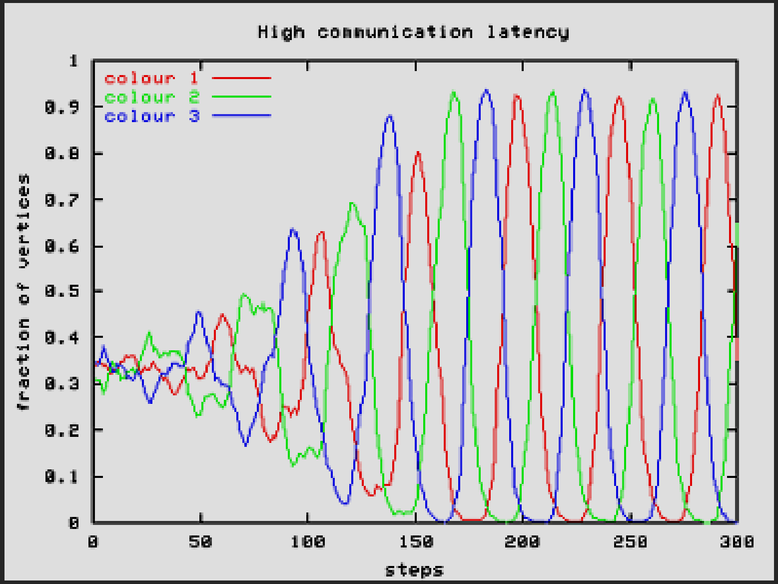
- Effects of phase transitions are less pronounced on short-term reductions in degree of conflict
  - more pronounced on long-term convergence values
- Alternative statement: severity of the effect of a phase transition increases with the required quality of solution
- NB these are relative effects
  - in absolute terms, phase transitions change the quality only slightly
  - may be more significant for other types of constraint
- Effect also dissipates in under-constrained problems
- Effect is secondary in over-constrained problems



New Result: Strict Synchrony is Not Required

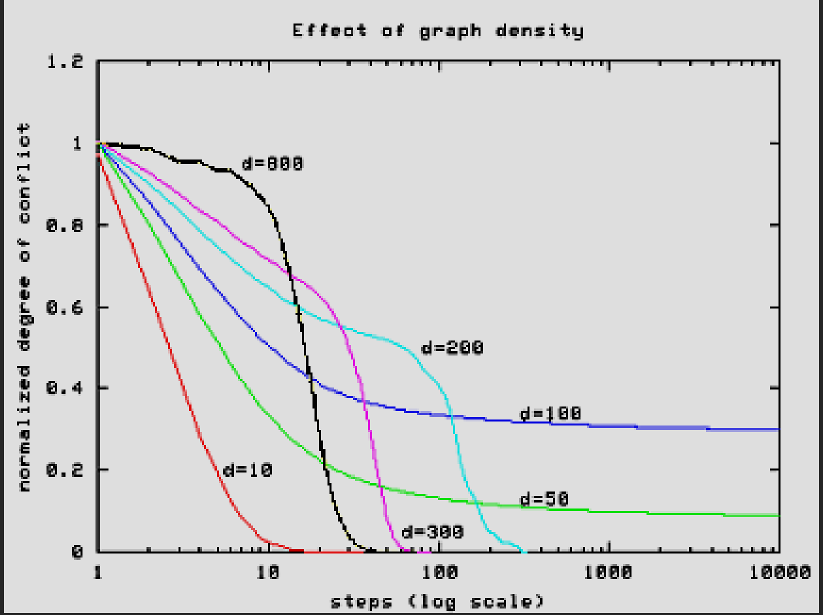
- Periodic but asynchronous coloring
  - simplifies implementation on distributed hardware
- Asynchronous execution is OK provided that the activation probability α is low with respect to communication latency L
  - “collision probability” along an edge = $1-(1-\alpha)^L <$ threshold
- Academic interest: extremely high communication latencies cause a “resonance” effect
  - each color is adopted in turn by almost every node simultaneously





New Result: Possible Phase Transition w.r.t. Interconnection Density

- For high-density graphs, the degree of conflict increases with the density for a while
- For very-high-density graphs, all conflicts are rapidly eliminated
  - presumably due to large number of backbone variables that implicitly guide the search
    - even a distributed search
- Practical consequences: CFP algorithm may actually be effective for dense graphs
  - was originally designed for sparse graphs



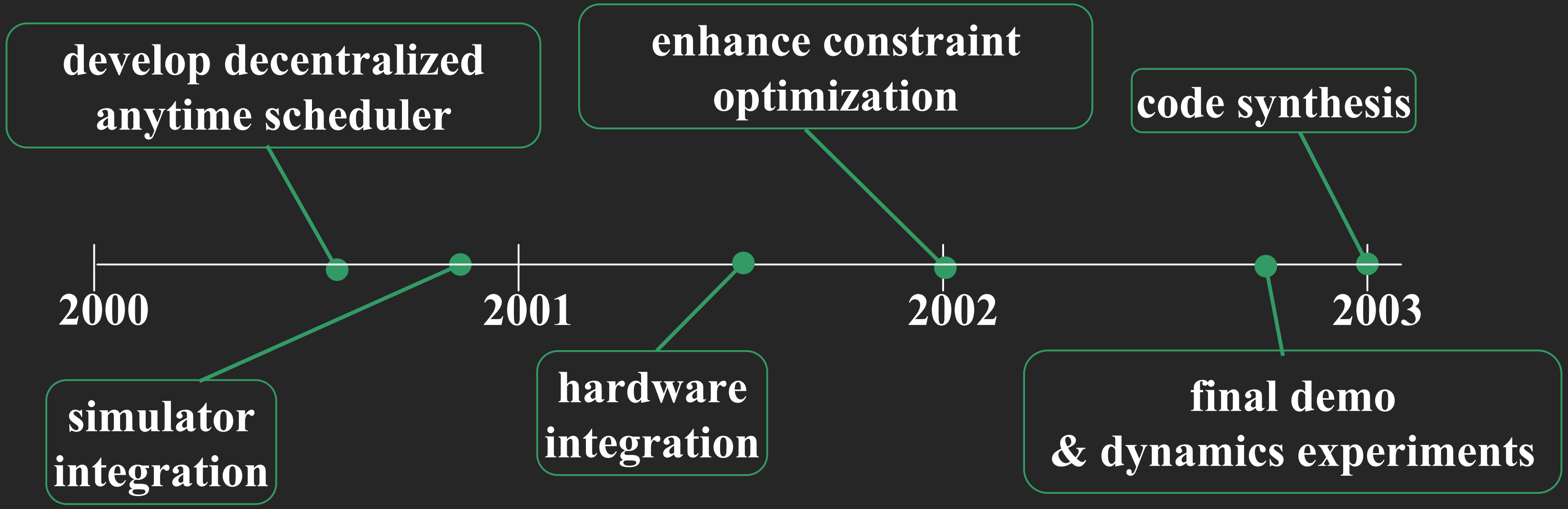
Random 20-colorable graphs
size ~ 2000 nodes
d is the mean degree

Wrap-Up

Project Plans

- Design & implement dynamics experiments on modified challenge problem
  - bypass the tracker(s) to directly assess resource manager
  - feed ground truth target positions to resource manager
    - with controllable noise added to simulate tracker uncertainty
  - resource manager produces scan schedules suited to the (noisy) target estimates
  - scanning quality is measured w.r.t. precise ground truth
  - quality achieved by Kestrel’s scheduler can be compared with that achieved by centralized/greedy/local schedulers
    - see metrics on slides 4 & 5
  - particular emphasis on scalability: does local coordination lead to good global results
- Improve challenge problem performance
  - in preparation for final demo

Project Schedule & Milestones



develop decentralized anytime scheduler

enhance constraint optimization

code synthesis

2000

2001

2002

2003

simulator integration

hardware integration

final demo & dynamics experiments

Technology Transition/Transfer

- Academia:
  - distributed constraint optimization growing as a field of study
  - Kestrel (and others) emphasizing pragmatic & real-time aspects
    - trying to counter usual tendency to focus on perfect solutions
  - presentations in relevant workshops:
    - Sixth Biennial World Conference on Integrated Design & Process Technology
    - Distributed Constraint Reasoning workshop (part of AAMAS 2002)
    - Probabilistic Search Techniques workshop (part of AAAI 2002)
    - Constraint Programming 2002 (submission)

Program Issues

- Large-scale experiments
  - thousands of sensors, dozens of targets
  - enough data to draw valid conclusions of scientific interest
- CP architecture currently requires each sensor to be represented as a unique JVM/process
  - even in the simulator
  - would require enormous computational power to perform large-scale experiments