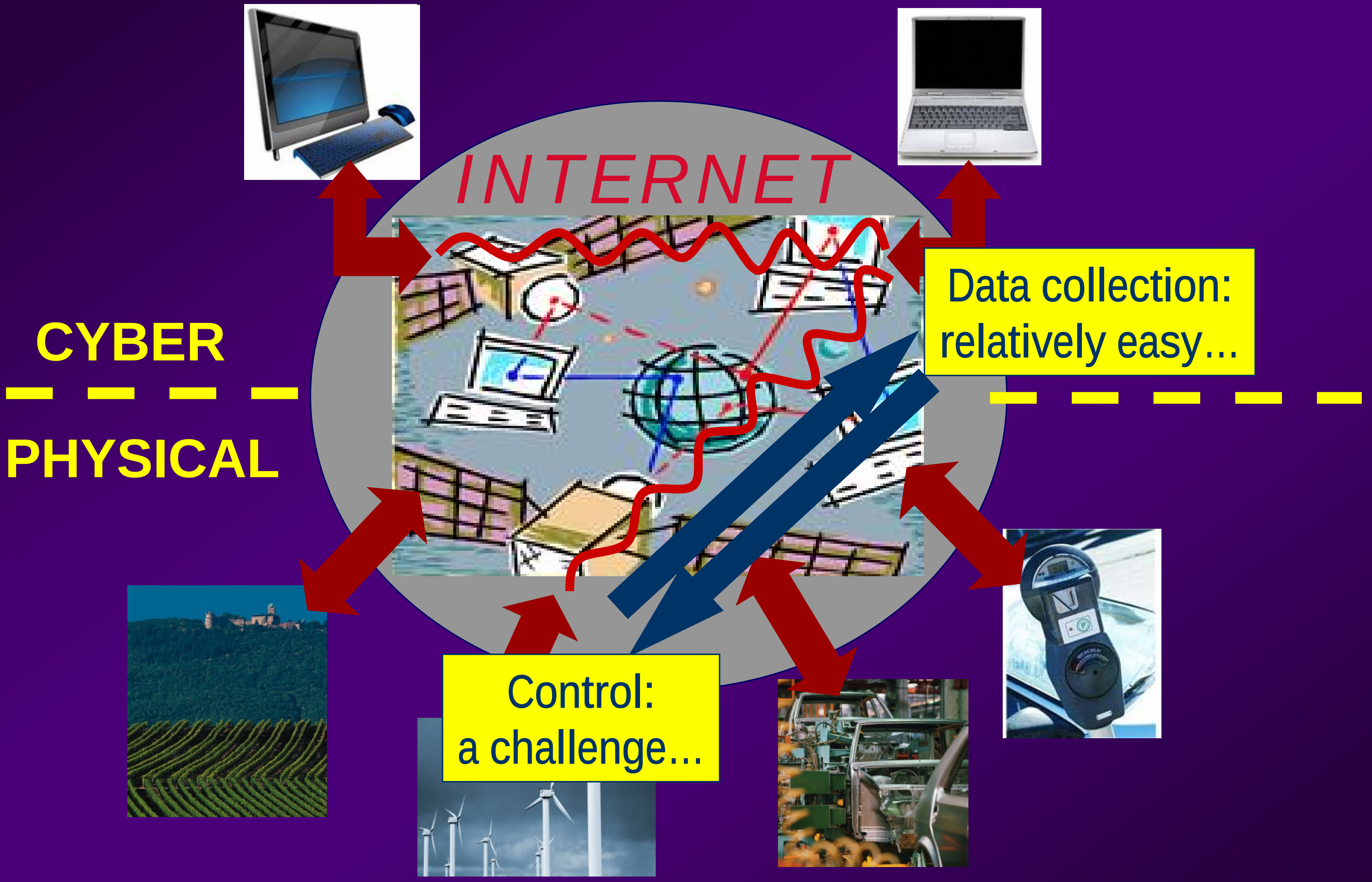
CONTROL AND OPTIMIZATION IN CYBERPHYSICAL SYSTEMS: FROM SENSOR NETWORKS TO "SMART PARKING" APPS

C. G. Cassandras

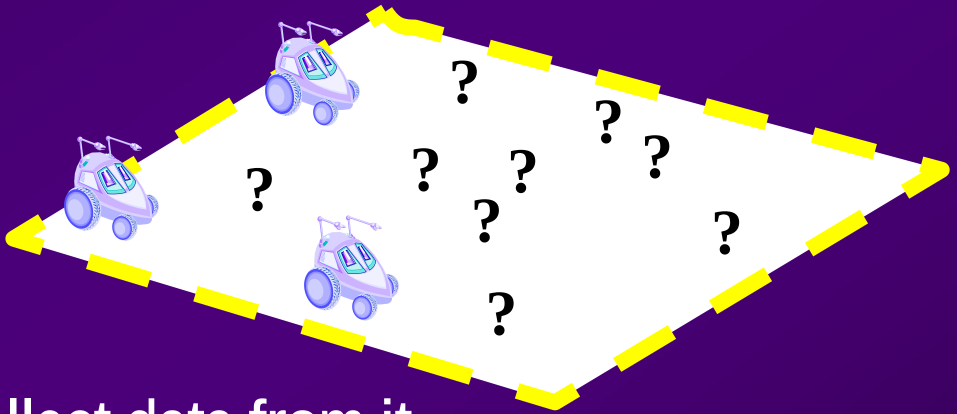
Division of Systems Engineering
and Dept. of Electrical and Computer Engineering
and Center for Information and Systems Engineering
Boston University

CYBER-PHYSICAL SYSTEMS



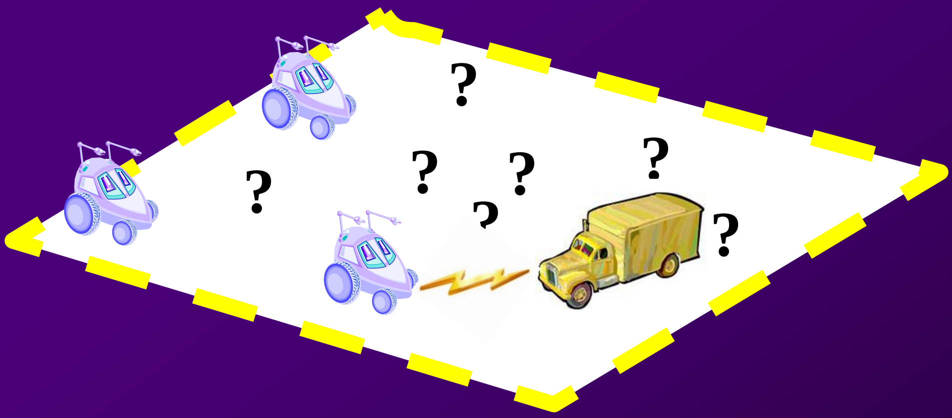
SENSOR NETWORK AS A CONTROL SYSTEM

What is the function of a SENSOR NETWORK?



1. Seek and detect "Data Sources" (or "Targets")

2. Once a Data Source is detected, collect data from it, track it if mobile



3. Continue to seek data sources while collecting data from detected sources

OUTLINE

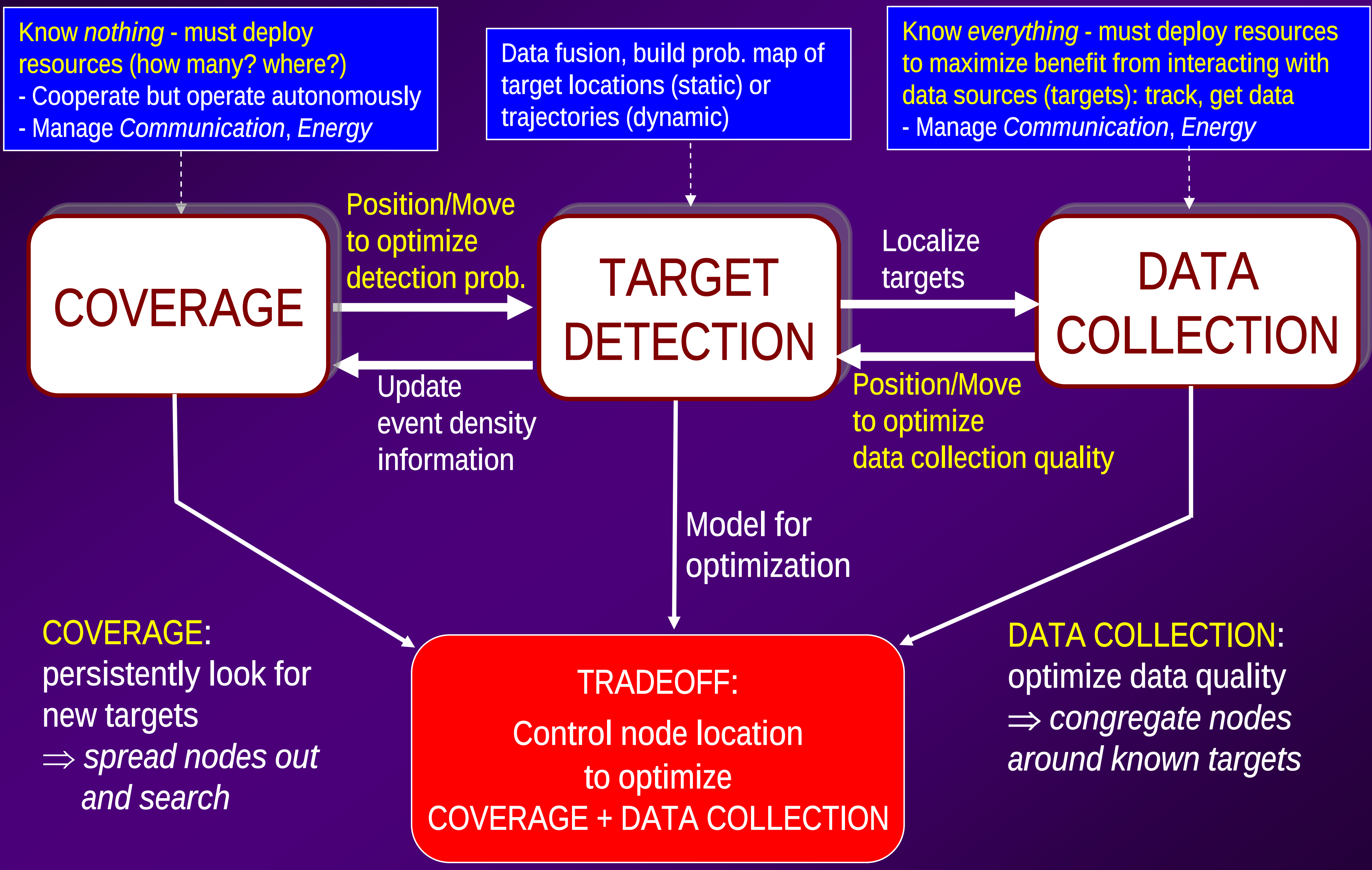
- Sensor Networks as Control Systems
- No knowledge of mission space:

  Coverage control, Persistent Monitoring
- Full knowledge of mission space:

  Data Collection, Data Harvesting, Reward Maximization
- Distributed Optimization Framework
- Information exchange among nodes:

  Event-driven communication
- Sensor + Actuation Networks: “Smart Parking” system

SENSOR NETWORK AS A CONTROL SYSTEM

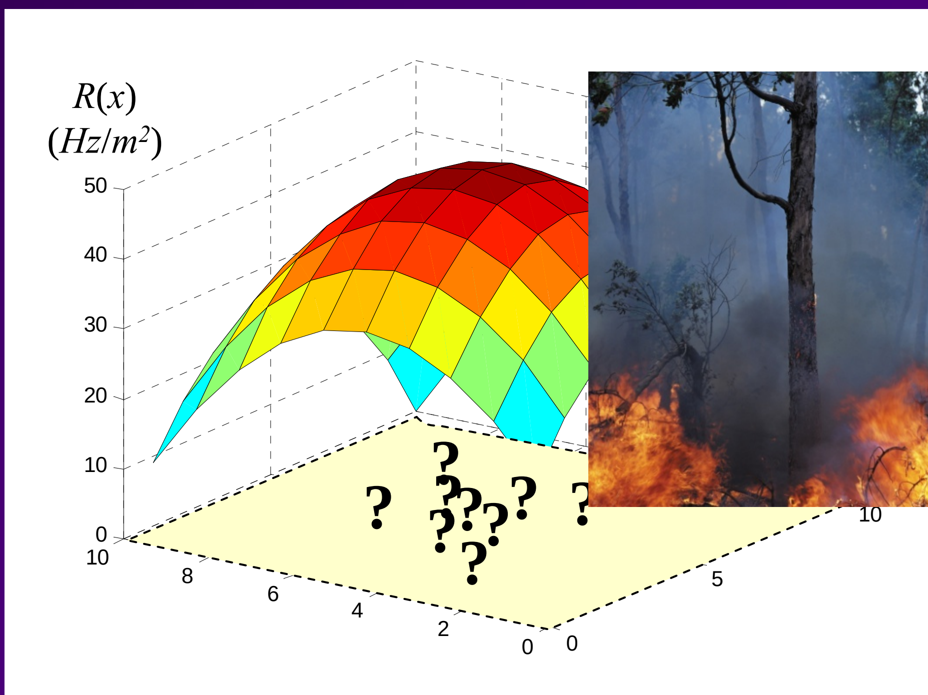


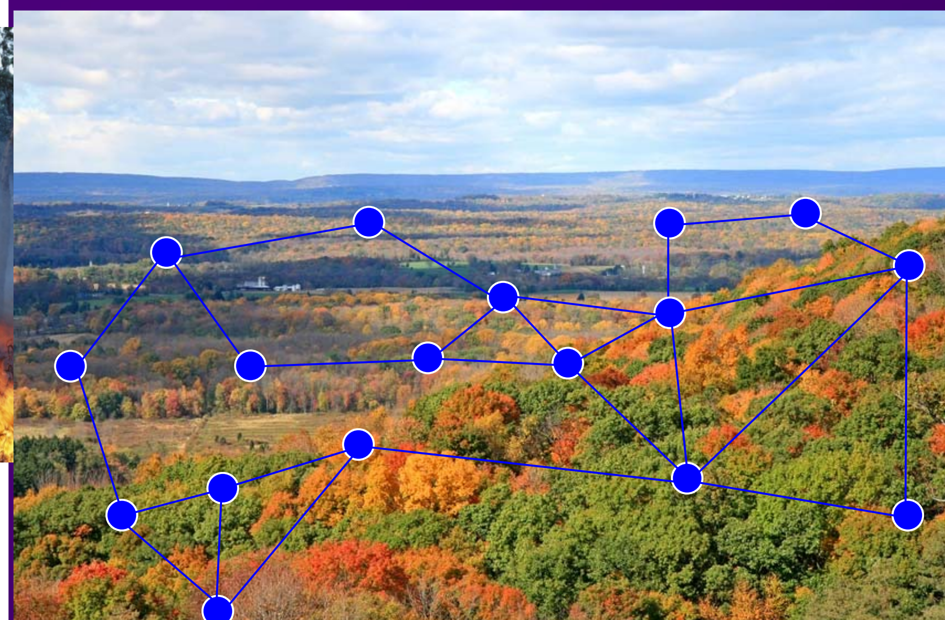
COVERAGE

MOTIVATIONAL PROBLEM: *COVERAGE CONTROL*

Deploy sensors to maximize "event" detection probability

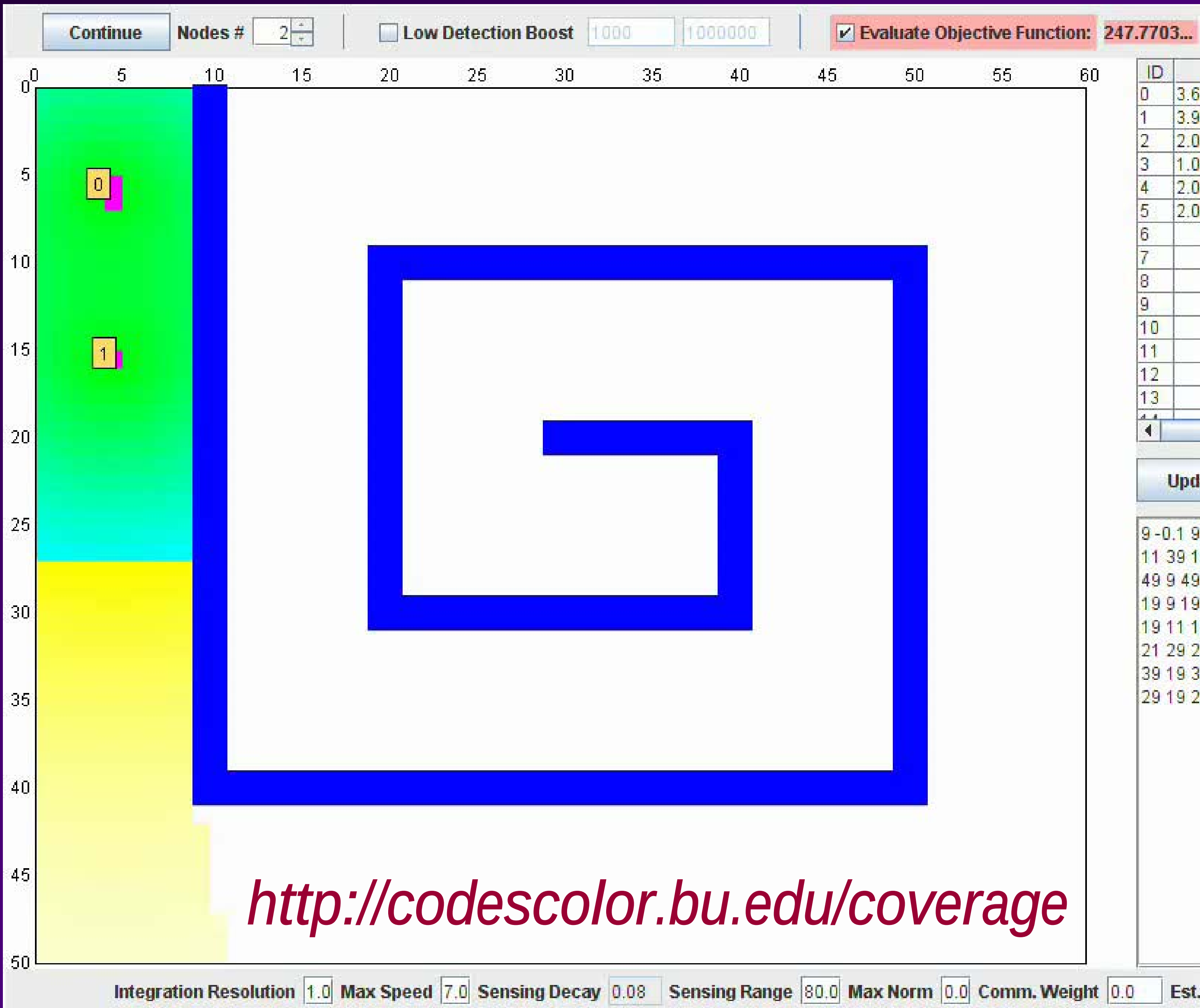
– unknown event locations
– event sources may be mobile
– sensors may be mobile

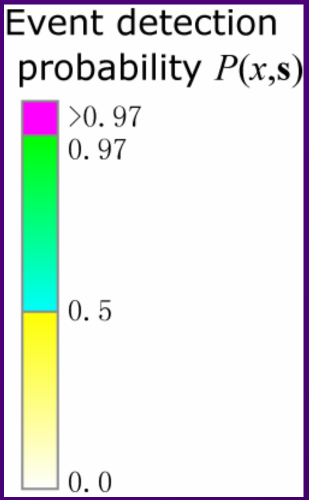




Perceived event density (data sources) over given region (mission space)

OPTIMAL COVERAGE IN A MAZE

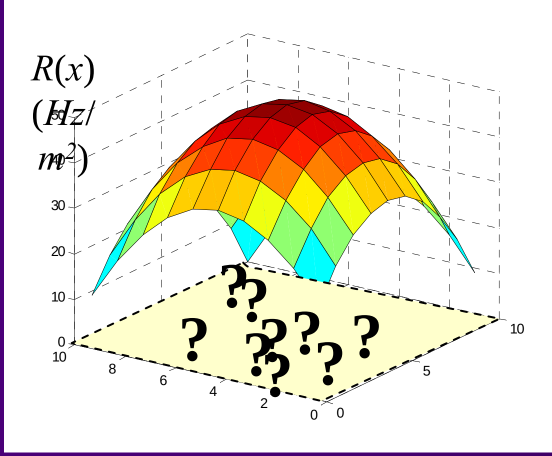




http://codescolor.bu.edu/coverage

Zhong and Cassandras, 2008

COVERAGE: PROBLEM FORMULATION



- N mobile sensors, each located at $s_i \in \mathbf{R}^2$
- Data source at x emits signal with energy E
- Signal observed by sensor node i (at s_i)
- SENSING MODEL:

$$p_i(x, s_i) \equiv P[\text{Detected by } i \mid A(x), s_i]$$

($A(x)$ = data source emits at x)

- Sensing attenuation:
  $p_i(x, s_i)$ monotonically decreasing in $d_i(x) \equiv \|x - s_i\|$

COVERAGE: PROBLEM FORMULATION

- Joint detection prob. assuming sensor independence ($\mathbf{s} = [s_1, \ldots, s_N]$: node locations)

Event sensing probability

$$P(x, \mathbf{s}) = 1 - \prod_{i=1}^{N} [1 - p_i(x, s_i)]$$

- OBJECTIVE: Determine locations $\mathbf{s} = [s_1, \ldots, s_N]$ to maximize total *Detection Probability*:

$$\max_{\mathbf{s}} \int_{\Omega} R(x) P(x, \mathbf{s}) dx$$

Perceived event density

DISTRIBUTED COOPERATIVE SCHEME

- Set

$$H(s_1,\ldots,s_N)=\int_\Omega R(x)\left\{1-\prod_{i=1}^{N}\left[1-p_i(x)\right]\right\}dx$$

- Maximize $H(s_1,\ldots,s_N)$ by forcing nodes to move using gradient information:

$$\frac{\partial H}{\partial s_k}=\int_\Omega R(x)\prod_{i=1,i\neq k}^{N}\left[1-p_i(x)\right]\frac{\partial p_k(x)}{\partial d_k(x)}\frac{s_k-x}{d_k(x)}dx$$

$$s_i^{k+1}=s_i^k+\beta_k\frac{\partial H}{\partial s_i^k}$$

Desired displacement $= V\cdot\Delta t$

Cassandras and Li, 2005
Zhong and Cassandras, 2011

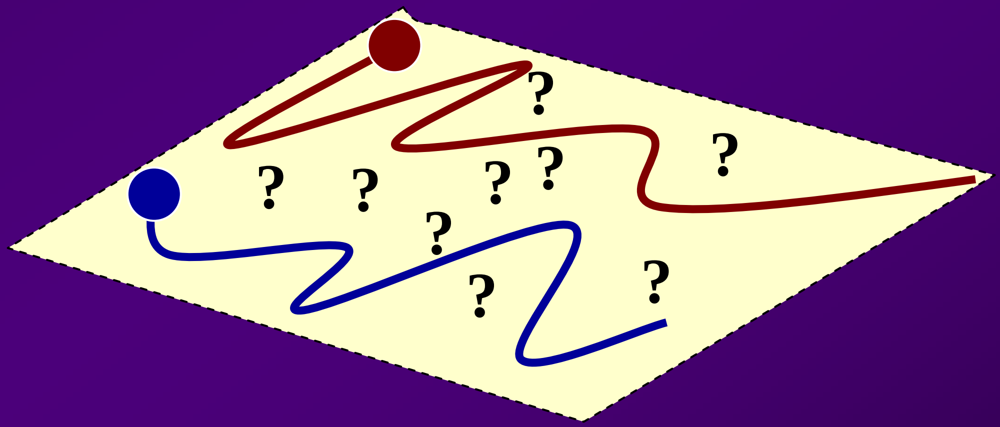
PERSISTENT MONITORING

(PERSISTENT SEARCH, SURVEILLANCE)

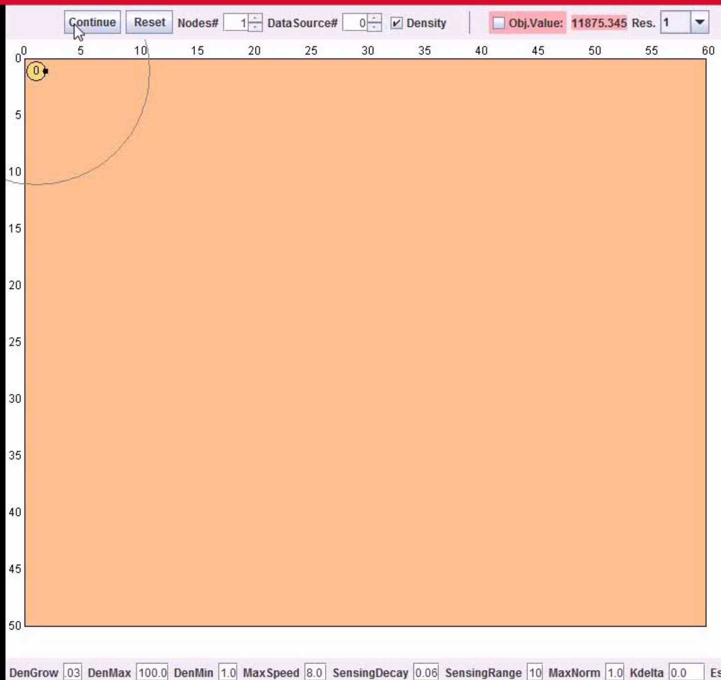
COVERAGE CONTROL v PERSISTENT MONITORING

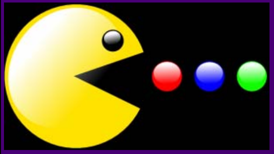
PERSISTENT MONITORING:

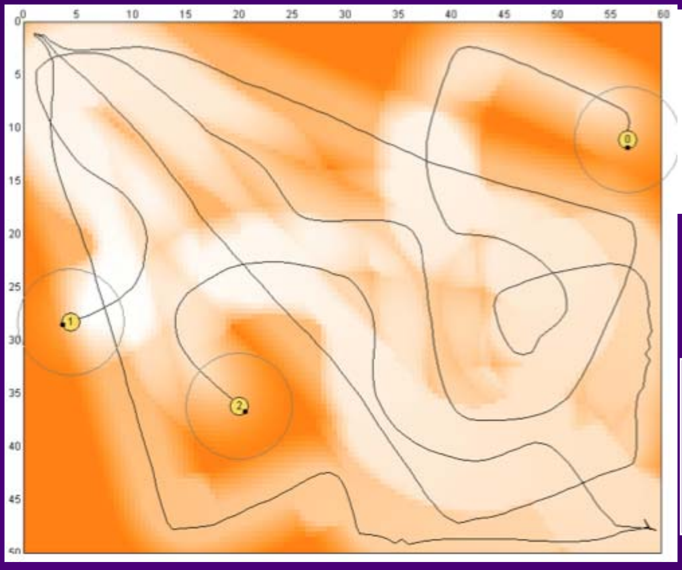
- environment cannot be fully covered by stationary team of nodes
- all areas of mission space must be visited infinitely often
- minimize some measure of overall uncertainty



PERSISTENT SEARCH IN 2D MISSION SPACE







Dark brown:
HIGH uncertainty

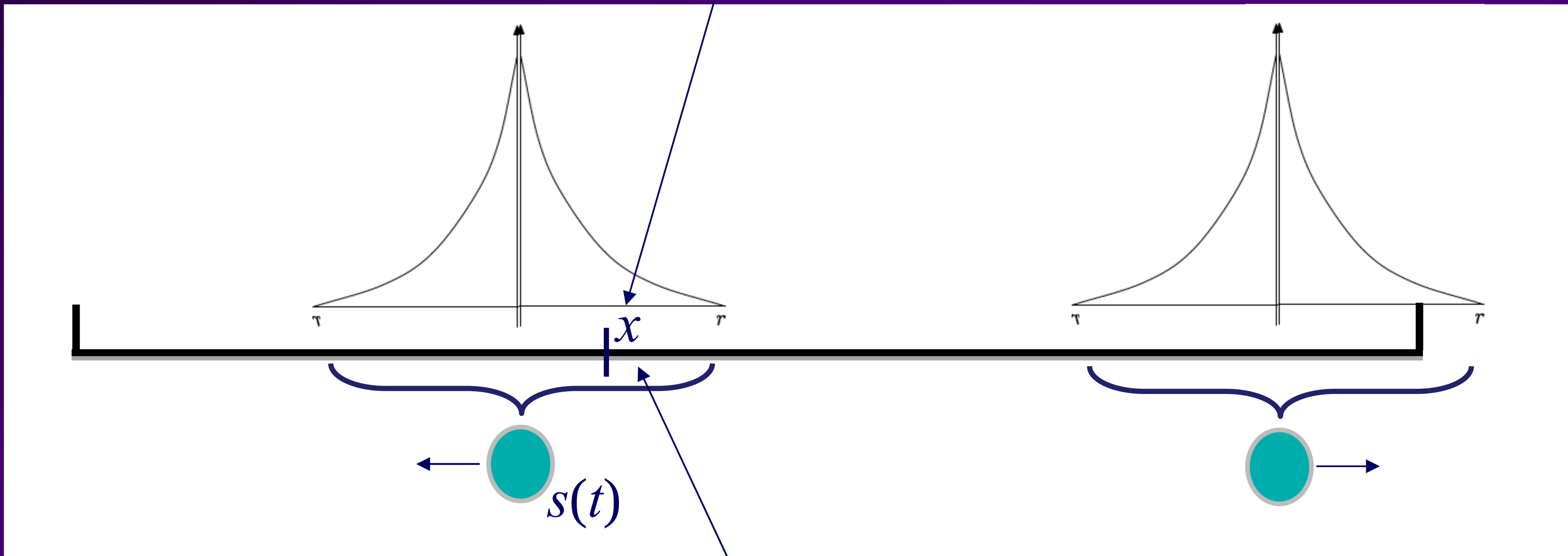
White:
NO uncertainty

Agents play a cooperative PACMAN game against "uncertainty" which continuously regenerates…

JAVA multi-agent simulator designed to interactively test various controllers. Polygonal obstacles may be added to the environment.
http://codescolor.bu.edu/simulators/density/density.html

PERSISTENT MONITORING PROBLEM

SENSING MODEL: $p(x,s)$ Probability agent at s senses point x

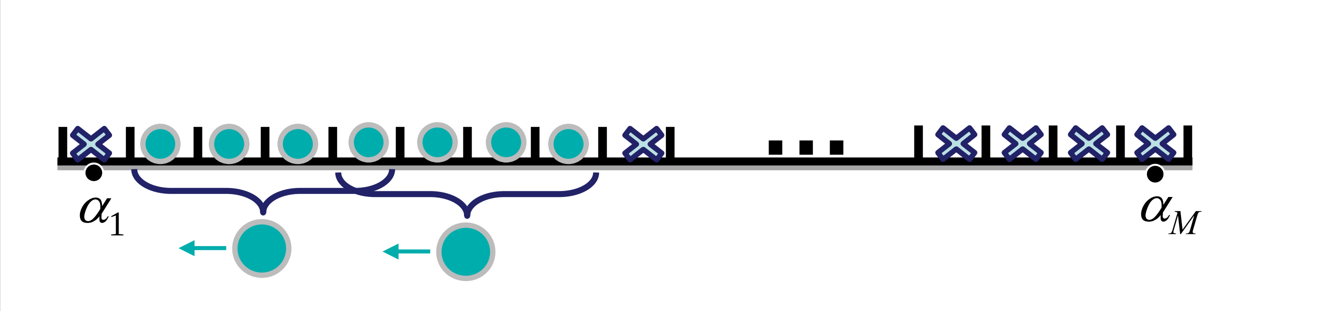


UNCERTAINTY MODEL: Associate to x *Uncertainty Function* $R(x,t)$ such that

$$\dot{R}(x,t) = \begin{cases} 0 & \text{if } R(x,t)=0,\ A(x) < Bp(x,s(t)) \\ A(x) - Bp(x,s(t)) & \text{otherwise} \end{cases}$$

PERSISTENT MONITORING PROBLEM

Partition mission space $\Omega = [0,L]$ into M intervals:



For each interval $i = 1,\ldots,M$ define *Uncertainty Function* $R_i(t)$:

$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i < BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}$$

$$P_i(\mathbf{s}) = 1 - \prod_{j=1}^{N} \left[1 - p_i(s_j)\right]$$

$$p_i(s_j) \equiv p_j(\alpha_i, s_j)$$

where $P_i(\mathbf{s})$ = joint prob. $\boldsymbol{i}$ is sensed by agents located at $\mathbf{s} = [s_1,\ldots,s_N]$

OPTIMAL CONTROL PROBLEM

Determine $u_1(t),\ldots,u_N(t)$ such that

$$\min_{u_1,\ldots,u_N} J = \frac{1}{T}\int_0^T \sum_{i=1}^{M} R_i(t)\,dt$$

Uncertainty measure

s.t.

$$\dot{s}_n = u_n, \quad |u_n(t)| \le 1, \quad 0 \le s_n(t) \le L$$

Agent dynamics

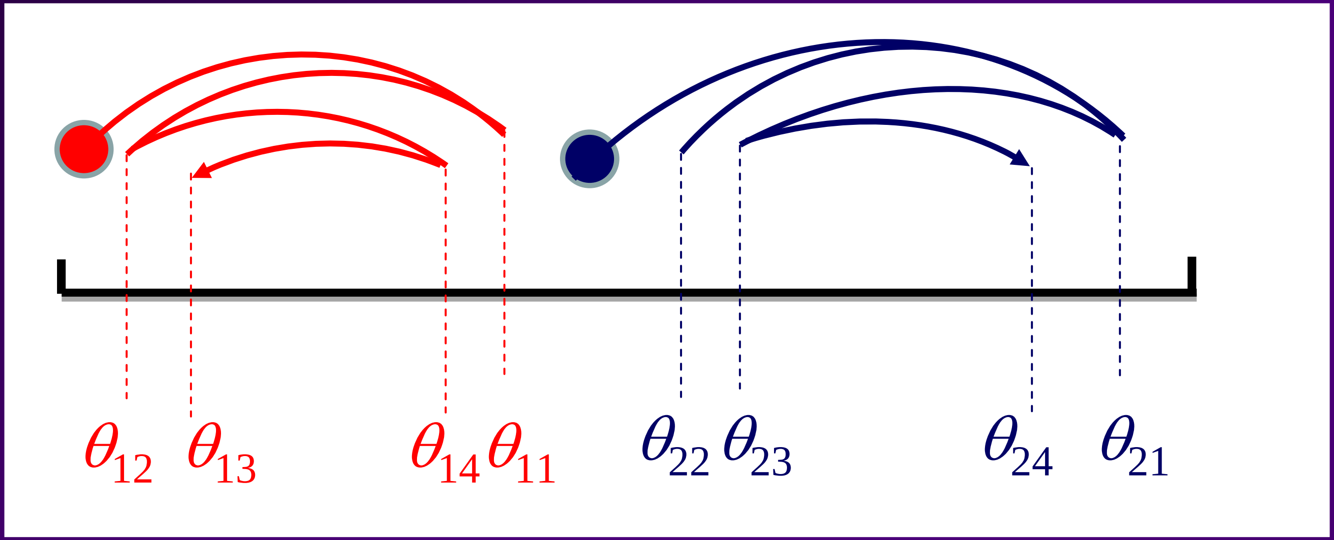
$$\dot{R}_i(t) = \begin{cases} 0 & \text{if } R_i(t) = 0,\ A_i < BP_i(\mathbf{s}(t)) \\ A_i - BP_i(\mathbf{s}(t)) & \text{otherwise} \end{cases}$$

Uncertainty dynamics

$$p_j(x, s_j) = \begin{cases} 1 - \dfrac{|x - s_j|}{r_j} & \text{if } |x - s_j| \le r_j \\ 0 & \text{if } |x - s_j| > r_j \end{cases}$$

Sensing model

OPTIMAL CONTROL SOLUTION



Optimal trajectory is fully characterized by parameter vectors:

$$\theta_j = [\theta_{j1} \cdots \theta_{jS}], \quad j = 1, \ldots, N$$

such that agent j switches

from $u_j^*(t) = 1$ to $u_j^*(t) = -1$ at $s_j = \theta_{jk}$, if k is odd

from $u_j^*(t) = -1$ to $u_j^*(t) = 1$ at $s_j = \theta_{jk}$, if k is even

Cassandras, Lin, Ding, 20012

DATA COLLECTION

COVERAGE + DATA COLLECTION

Recall tradeoff:

COVERAGE:
persistently look for new targets
$\Rightarrow$ *spread nodes out*

TRADEOFF:
Control node location to optimize COVERAGE + DATA COLLECTION

DATA COLLECTION:
optimize data quality
$\Rightarrow$ *congregate nodes around known targets*

MODIFIED DISTRIBUTED OPTIMIZATION OBJECTIVE:

collect info from detected data sources (targets) while maintaining a good coverage to detect future events

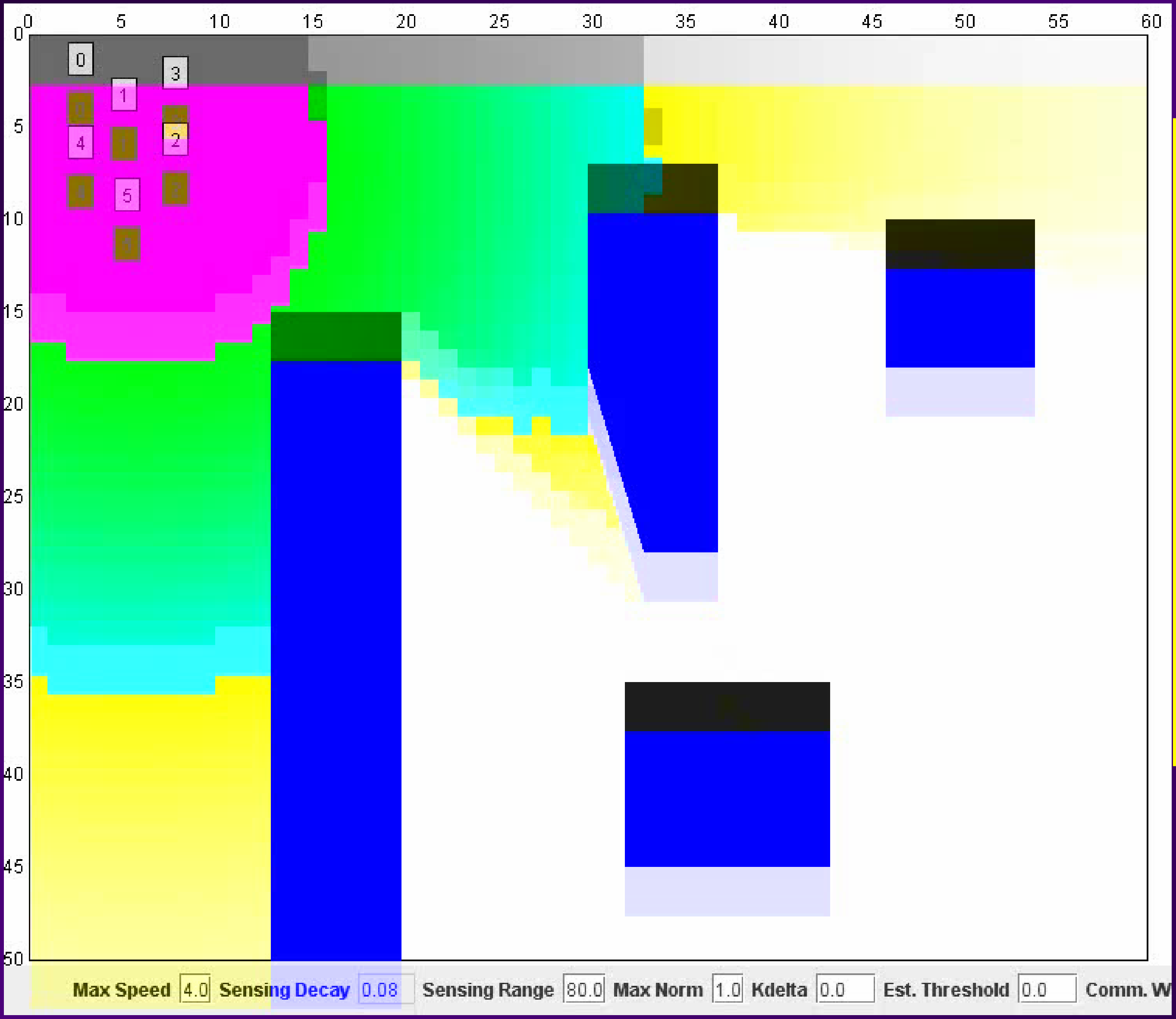
$S(u)$: data source value

$$H(\mathbf{s},t) = \int_{\Omega} R(x)P(x,\mathbf{s})dx + \boxed{\beta \sum_{u \in \mathcal{D}_t} S(u)F(u,\mathbf{s})}$$

$\mathcal{D}_t$: set of data sources, estimated based on sensor observations

$F(u,\mathbf{s})$: joint data collection quality at u (e.g., covariance)

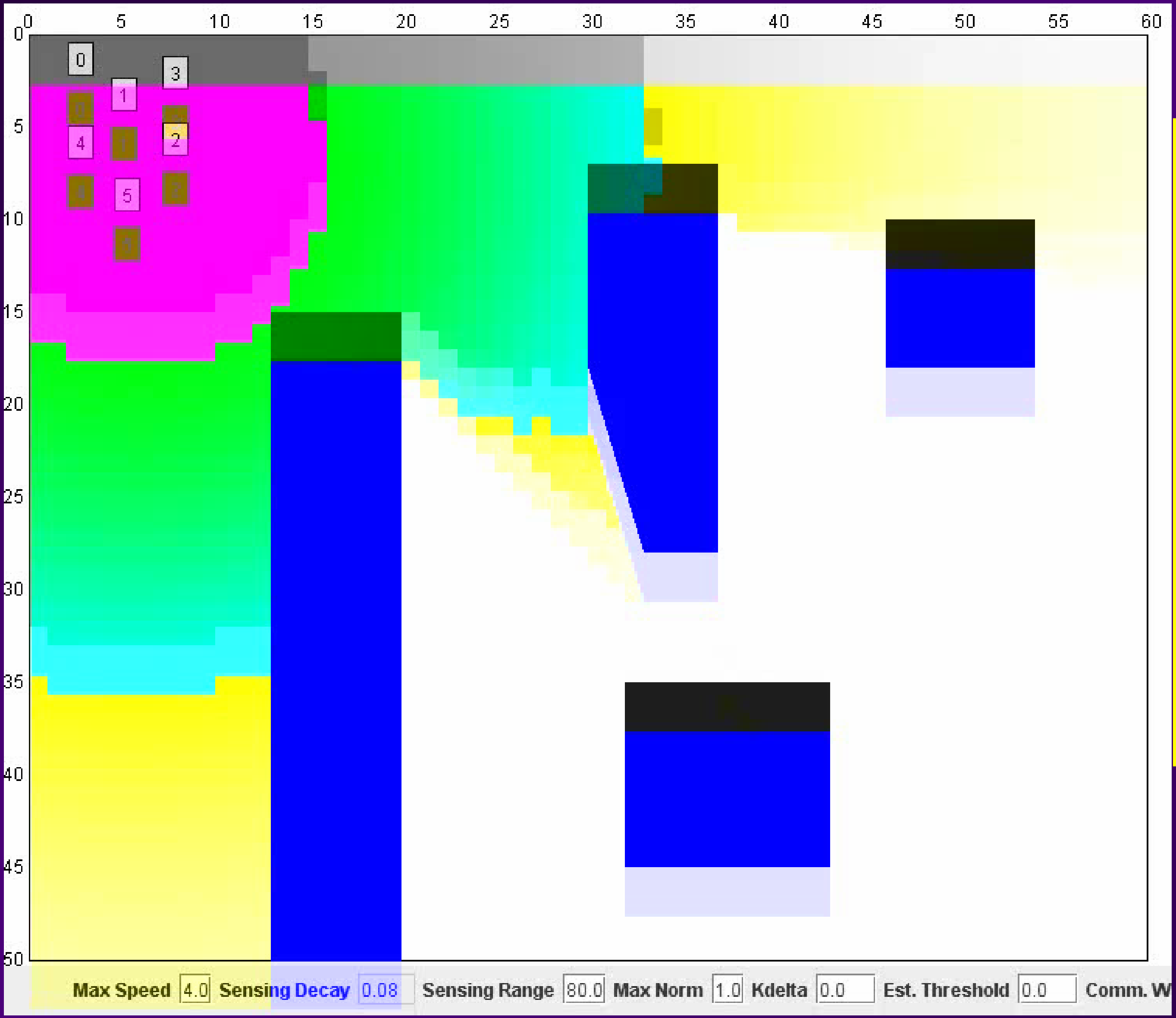
DEMO: REACTING TO EVENT DETECTION



Important to note:

There is no external control causing this behavior. Algorithm includes tracking functionality automatically

DEMO: REACTING TO EVENT DETECTION

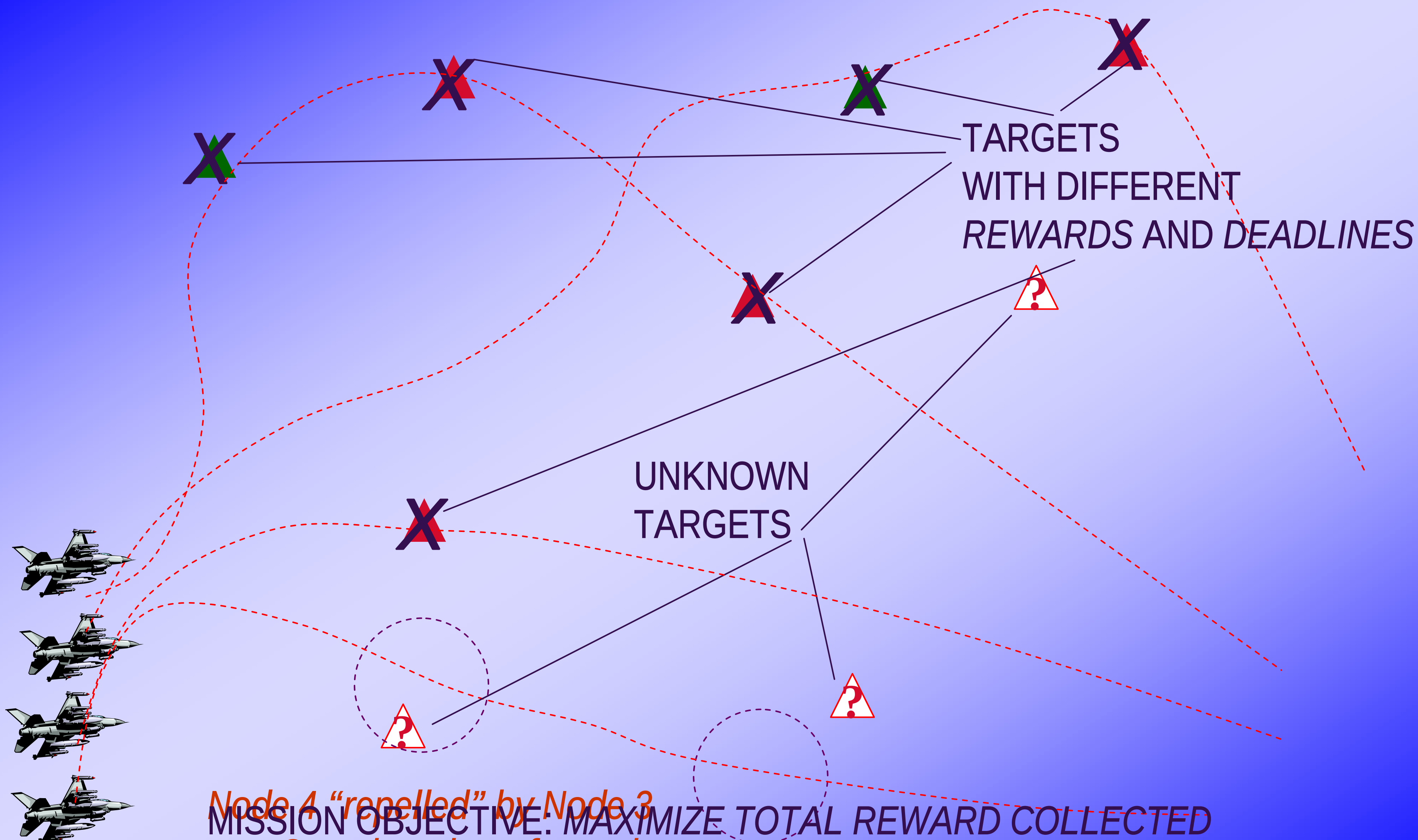


Important to note:

There is no external control causing this behavior. Algorithm includes tracking functionality automatically

DATA COLLECTION: REWARD MAXIMIZATION, DATA HARVESTING

REWARD MAXIMIZATION MISSION



Node 4 “repelled” by Node 3
→ Search task performed

MISSION OBJECTIVE: *MAXIMIZE TOTAL REWARD COLLECTED*
BY VISITING TARGETS BEFORE THEIR “DEADLINES” EXPIRE

REWARD MAXIMIZATION MISSION

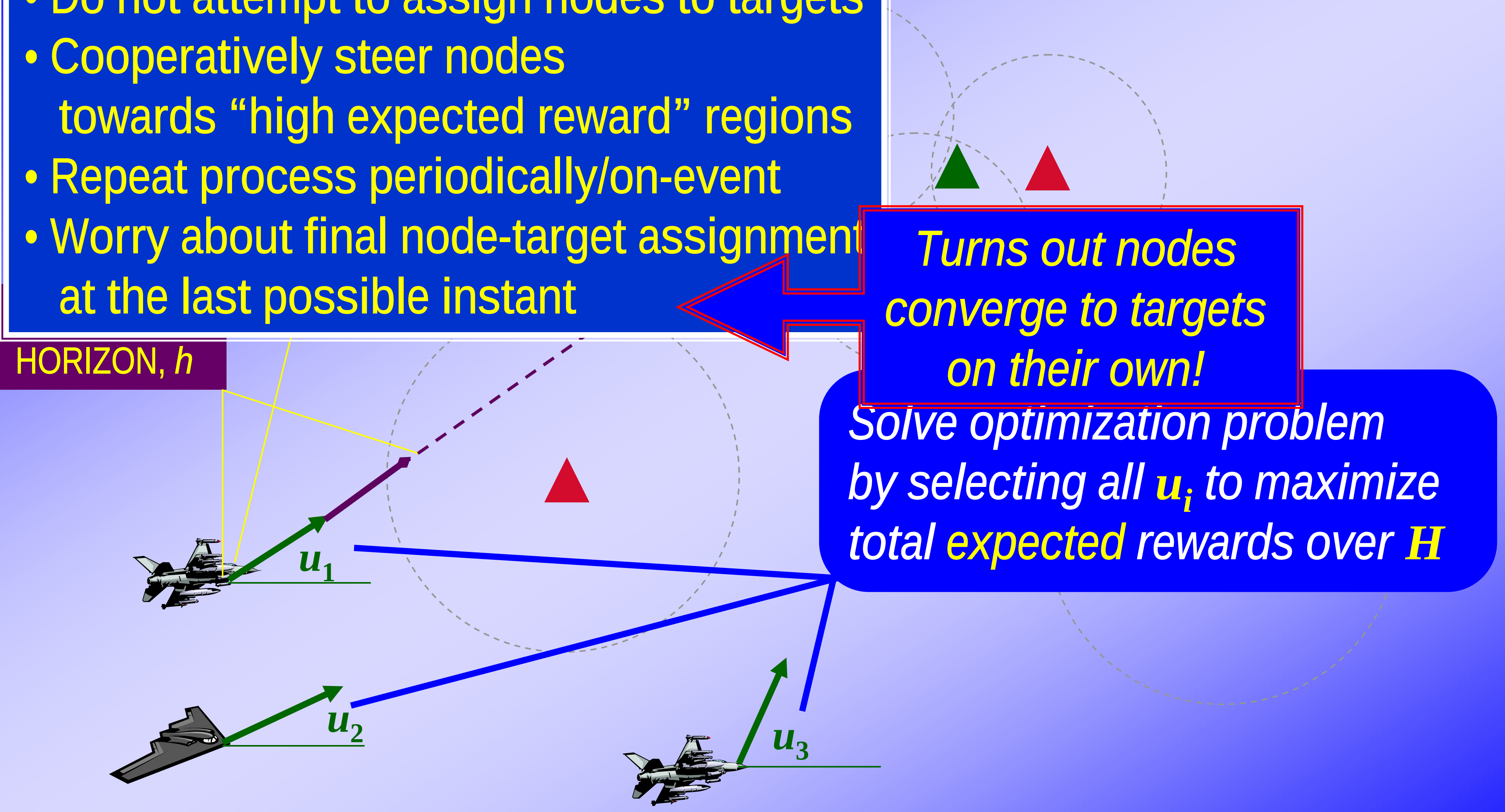
CONTINUED

This is like the notorious TRAVELING SALESMAN problem, except that…

- … there are multiple (cooperating) salesmen
- … there are deadlines + time-varying rewards
- … environment is stochastic
  (nodes may fail, threats damage nodes, etc.)

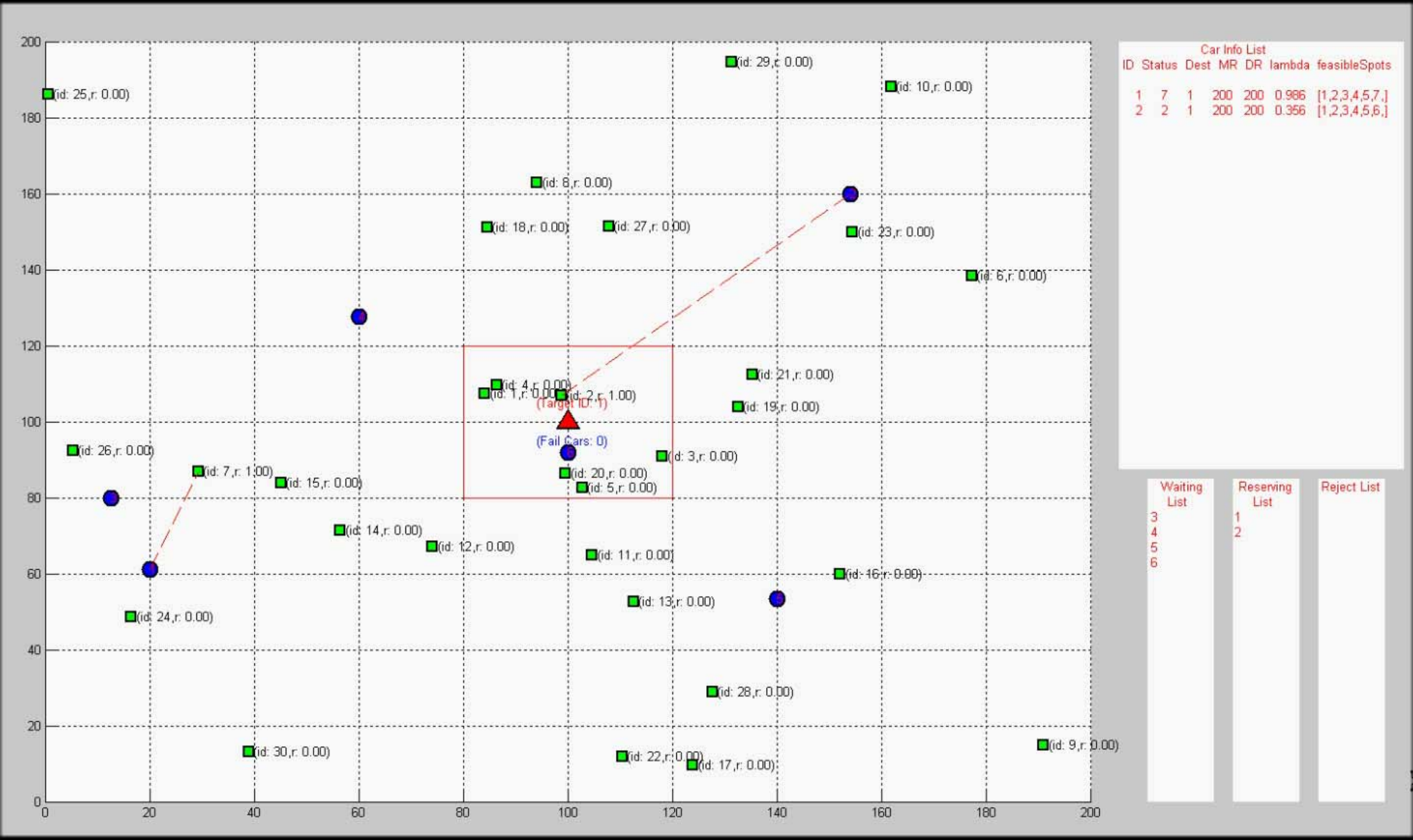
COOPERATIVE RECEDING HORIZON (CRH) CONTROL: *MAIN IDEA*

- Do not attempt to assign nodes to targets
- Cooperatively steer nodes towards "high expected reward" regions
- Repeat process periodically/on-event
- Worry about final node-target assignment at the last possible instant

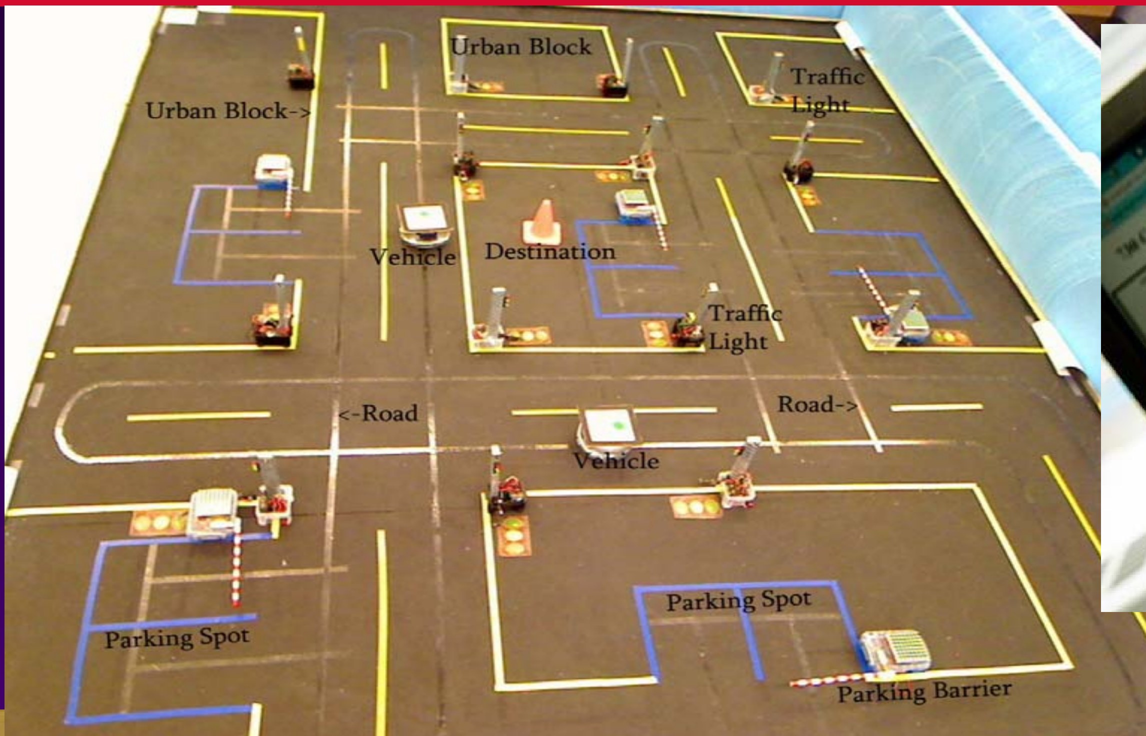


REWARD MAXIMIZATION DEMO

II. 2 Robots,
4 Targets Case


Car Info List
ID Status Dest MR DR lambda feasibleSpots
1 7 1 200 200 0.986 [1,2,3,4,5,7,]
2 2 1 200 200 0.356 [1,2,3,4,5,6,]
Waiting List
3
4
5
6
Reserving List
1
2
Reject List
(Target ID: 1)
(Fail Cars: 0)
(id: 29,r: 0.00)
(id: 10,r: 0.00)
(id: 25,r: 0.00)
(id: 8,r: 0.00)
(id: 18,r: 0.00)
(id: 27,r: 0.00)
(id: 23,r: 0.00)
(id: 6,r: 0.00)
(id: 21,r: 0.00)
(id: 4,r: 0.00)
(id: 1,r: 0.00)
(id: 2,r: 1.00)
(id: 19,r: 0.00)
(id: 26,r: 0.00)
(id: 3,r: 0.00)
(id: 7,r: 1.00)
(id: 15,r: 0.00)
(id: 20,r: 0.00)
(id: 5,r: 0.00)
(id: 14,r: 0.00)
(id: 12,r: 0.00)
(id: 11,r: 0.00)
(id: 16,r: 0.00)
(id: 13,r: 0.00)
(id: 24,r: 0.00)
(id: 28,r: 0.00)
(id: 9,r: 0.00)
(id: 30,r: 0.00)
(id: 22,r: 0.00)
(id: 17,r: 0.00)

BOSTON UNIVERSITY TEST BEDS



THE BIGGER PICTURE: DISTRIBUTED OPTIMIZATION

DISTRIBUTED COOPERATIVE OPTIMIZATION

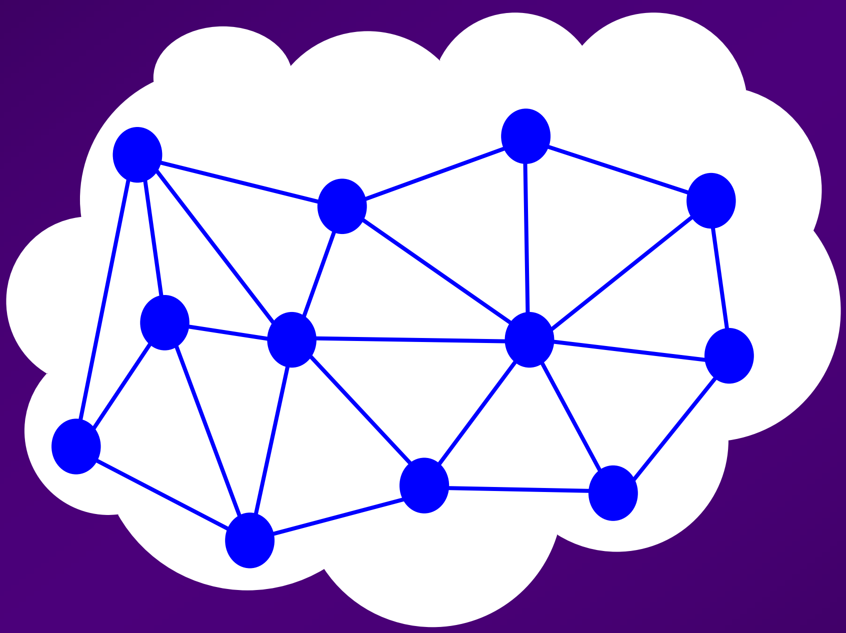
N system components (processors, agents, vehicles, nodes), one common objective:

$$\min_{s_1,\ldots,s_N} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on each } s_i$$

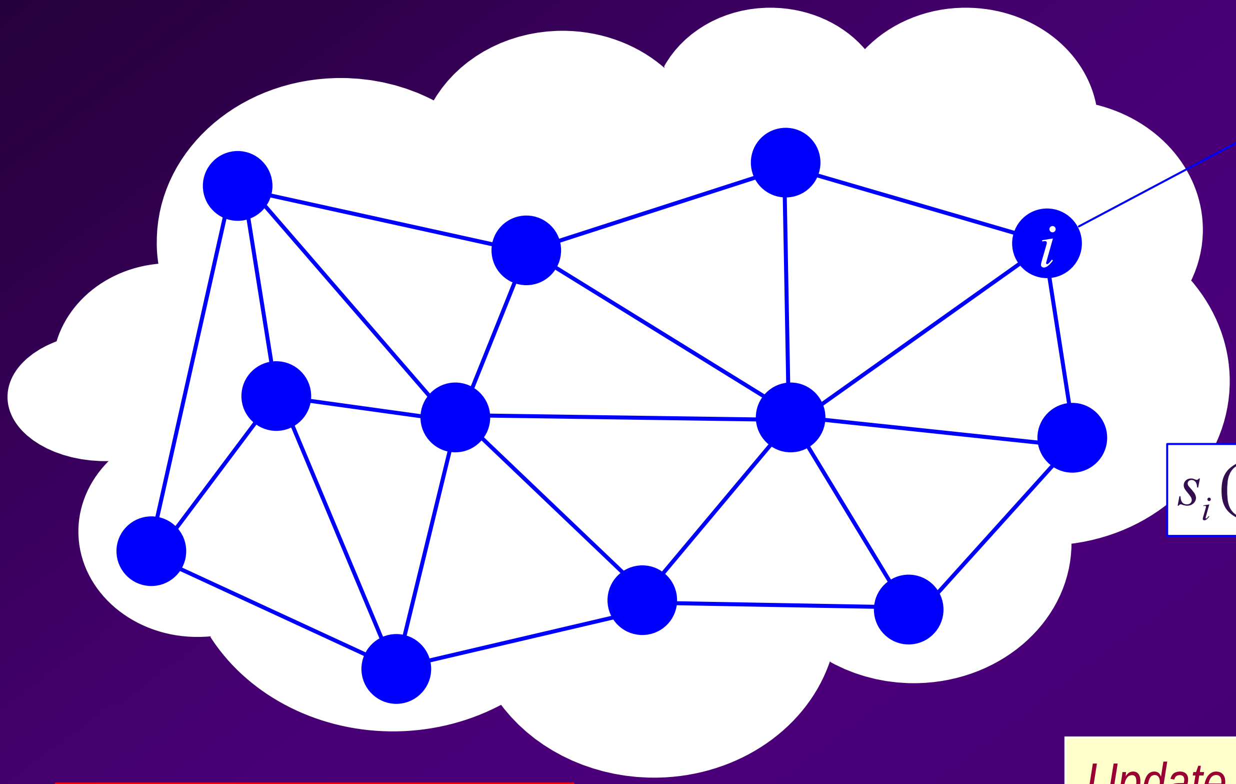
$$\min_{s_1} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on } s_1$$

⋮

$$\min_{s_N} H(s_1,\ldots,s_N)$$
$$s.t. \text{ constraints on } s_N$$



DISTRIBUTED COOPERATIVE OPTIMIZATION



Controllable *state*

$s_i, \; i = 1,\ldots,n_i$

$$s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k))$$

Step Size

Update Direction, usually

$$d_i(\mathbf{s}(k)) = -\nabla_i H(\mathbf{s}(k))$$

i requires knowledge of all $s_1,\ldots,s_N$

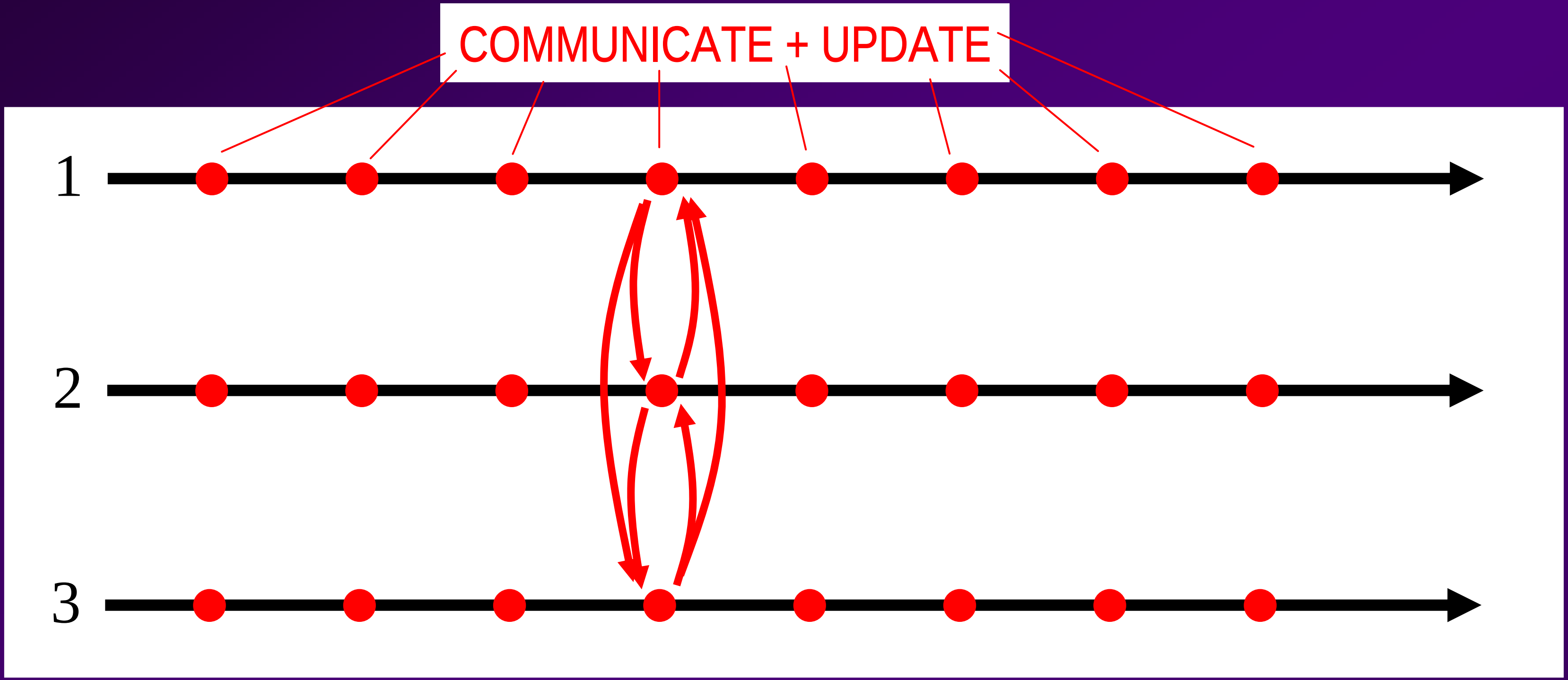
Inter-node communication

$$\min_{s_i} H(s_1,\ldots,s_N)$$

$$s.t. \text{ constraints on } s_i$$

HOW MUCH COMMUNICATION FOR OPTIMAL COOPERATION ?

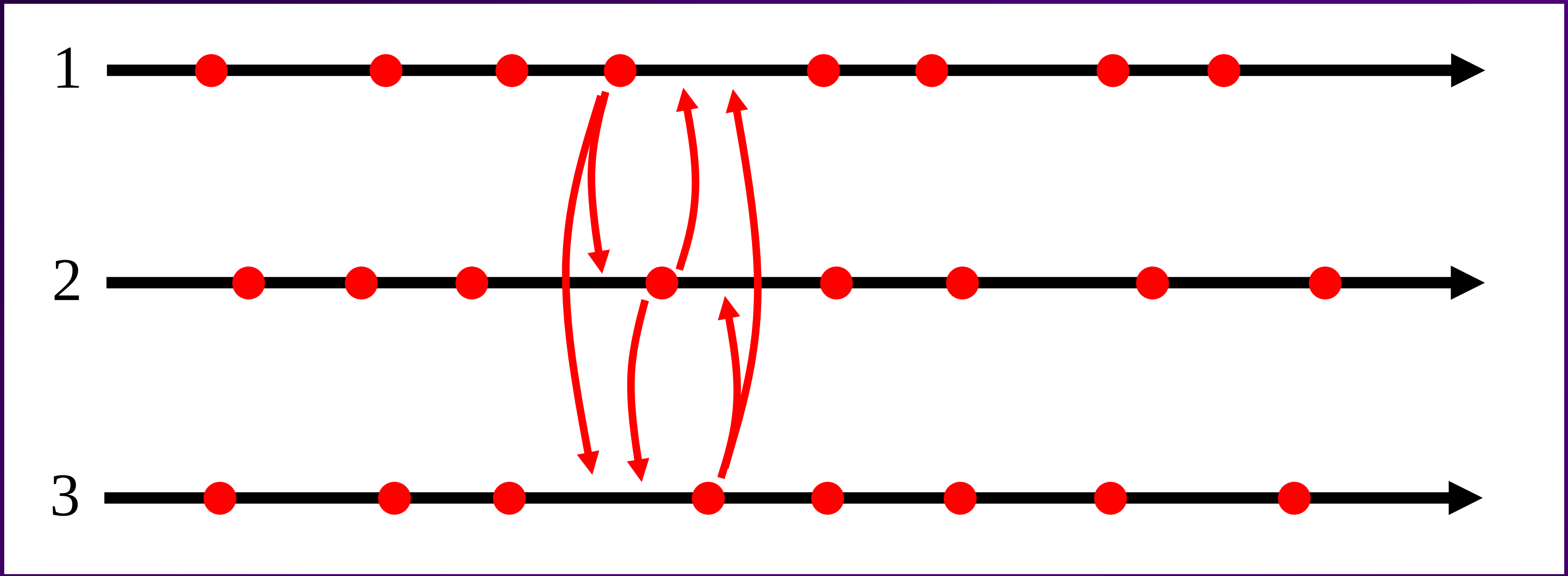
SYNCHRONIZED (TIME-DRIVEN) COOPERATION



Drawbacks:

- Excessive communication (critical in wireless settings!)
- Faster nodes have to wait for slower ones
- Clock synchronization infeasible
- Bandwidth limitations
- Security risks

ASYNCHRONOUS COOPERATION



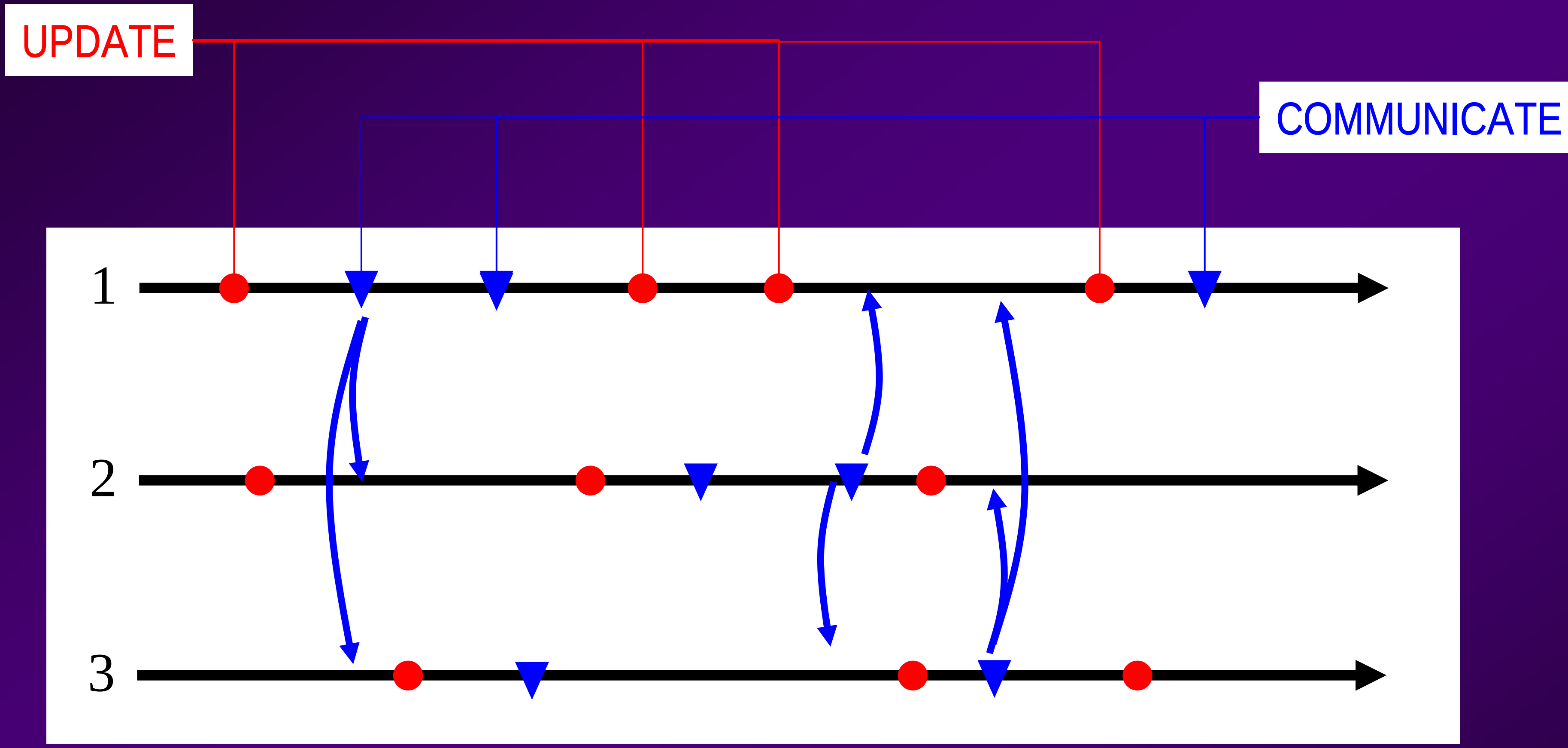
- Nodes not synchronized, delayed information used

Update frequency for each node is bounded
+
technical conditions

$\Rightarrow$ $s_i(k+1) = s_i(k) + \alpha_i d_i(\mathbf{s}(k))$ converges

Bertsekas and Tsitsiklis, 1997

ASYNCHRONOUS (EVENT-DRIVEN) COOPERATION



- UPDATE at i : locally determined, arbitrary (possibly periodic)
- COMMUNICATE from i : only when absolutely necessary

WHEN SHOULD A NODE COMMUNICATE?

Node state at any time t : $x_i(t)$

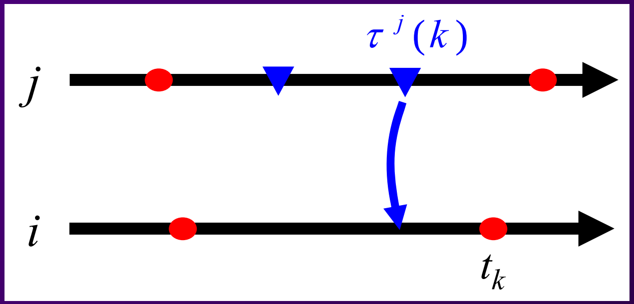
Node state at t_k : $s_i(k)$

$$\left.\begin{array}{l} x_i(t) \\ s_i(k) \end{array}\right\} \Rightarrow s_i(k) = x_i(t_k)$$

AT UPDATE TIME t_k : $s_j^i(k)$: node j state estimated by node i

Estimate examples:

➡ $s_j^i(k) = x_j(\tau^j(k))$ Most recent value



➡ $s_j^i(k) = x_j(\tau^j(k)) + \dfrac{t_k - \tau^j(k)}{\Delta_j} \cdot \alpha_i \cdot d_j\left(x_j(\tau^j(k))\right)$ Linear prediction

WHEN SHOULD A NODE COMMUNICATE?

AT ANY TIME t :

- $x_i^j(t)$: node i state estimated by node j
- If node i knows how j estimates its state, then it can evaluate $x_i^j(t)$
- Node i uses
  - its own true state, $x_i(t)$
  - the estimate that j uses, $x_i^j(t)$

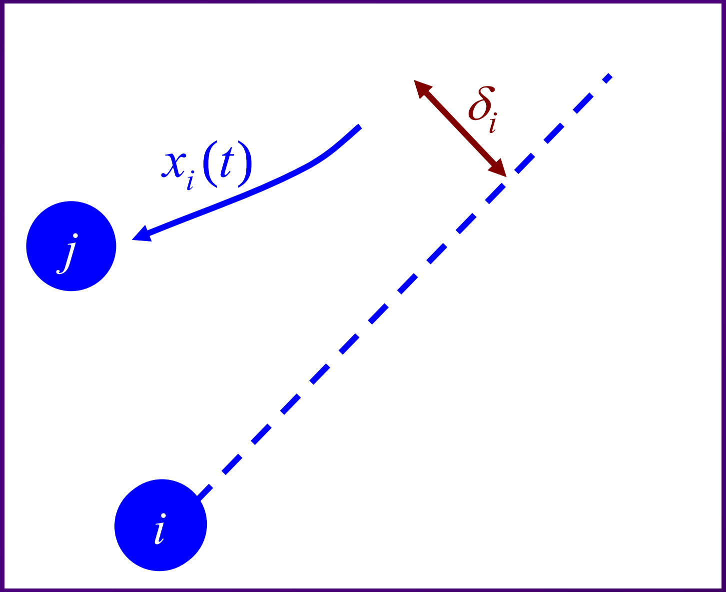
... and evaluates an ERROR FUNCTION $g\left(x_i(t), x_i^j(t)\right)$

Error Function examples: $\left\| x_i(t) - x_i^j(t) \right\|_1, \quad \left\| x_i(t) - x_i^j(t) \right\|_2$

WHEN SHOULD A NODE COMMUNICATE?

Compare ERROR FUNCTION $g\left(x_i(t), x_i^j(t)\right)$ to THRESHOLD δ_i

Node ***i*** communicates its state to node j only when it detects that its *true state* $x_i(t)$ deviates from ***j'*** *estimate of it* $x_i^j(t)$ so that $g\left(x_i(t), x_i^j(t)\right) \geq \delta_i$



$\Rightarrow$ *Event-Driven* Control

CONVERGENCE

Asynchronous distributed state update process at each i:

$$s_i(k+1) = s_i(k) + \alpha \cdot d_i(\mathbf{s}^i(k))$$

Estimates of other nodes, evaluated by node i

$$\delta_i(k) = \begin{cases} K_\delta \left\| d_i(\mathbf{s}^i(k)) \right\| & \text{if } k \text{ sends update} \\ \delta_i(k-1) & \text{otherwise} \end{cases}$$

THEOREM: Under certain conditions, there exist positive constants α and K_δ such that

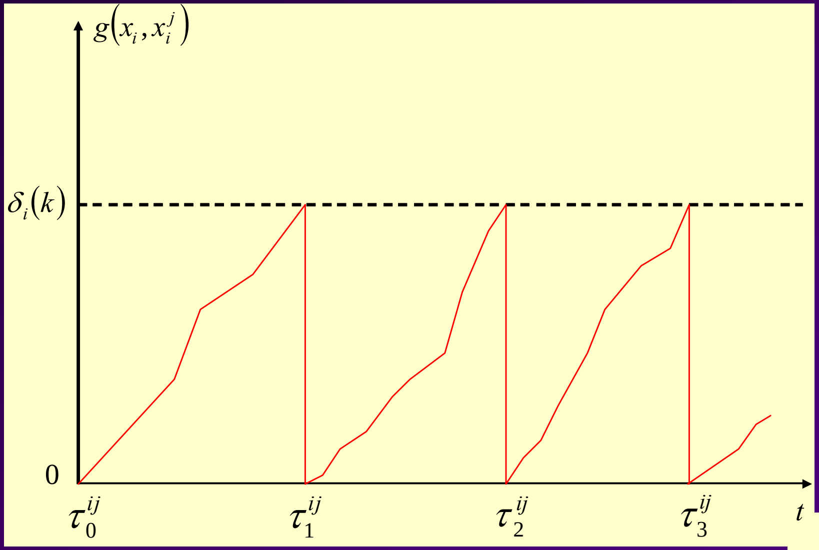
$$\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$$

Zhong and Cassandras, IEEE TAC, 2010

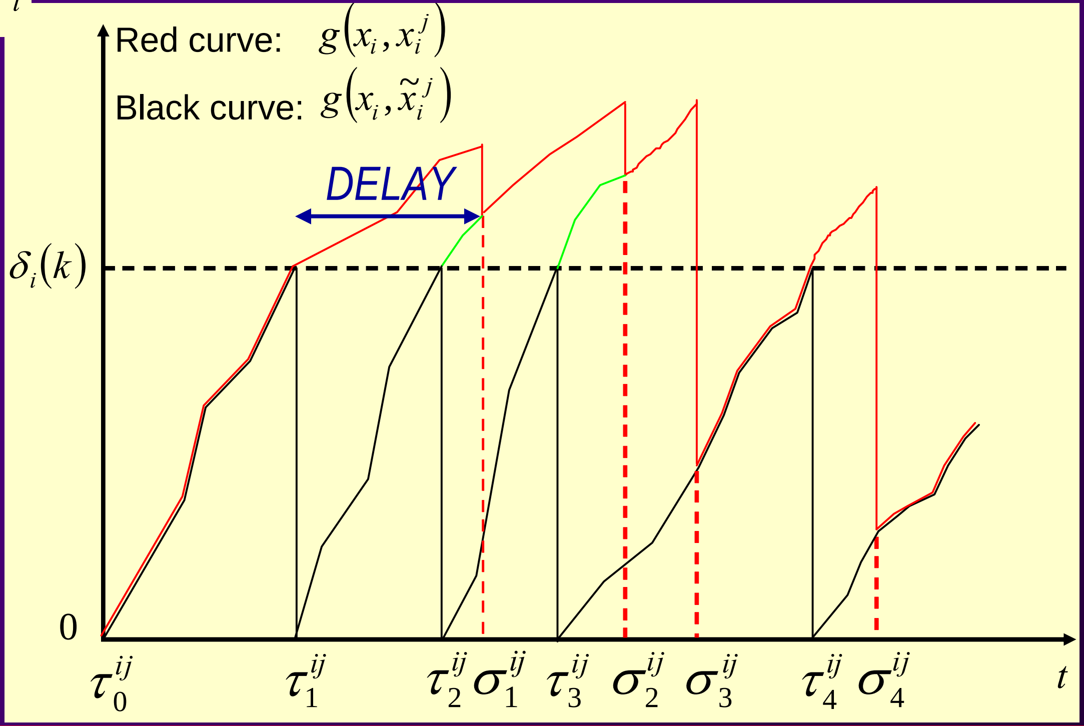
INTERPRETATION:

Event-driven cooperation achievable with minimal communication requirements $\Rightarrow$ energy savings

COONVERGENCE WHEN DELAYS ARE PRESENT



Error function trajectory with NO DELAY



COONVERGENCE WHEN DELAYS ARE PRESENT

Add a boundedness assumption:

ASSUMPTION: There exists a non-negative integer D such that if a message is sent before t_{k-D} from node i to node j, it will be received before t_k.

INTERPRETATION: at most $\boldsymbol{D}$ state update events can occur between a node sending a message and all destination nodes receiving this message.

THEOREM: Under certain conditions, there exist positive constants α and K_δ such that

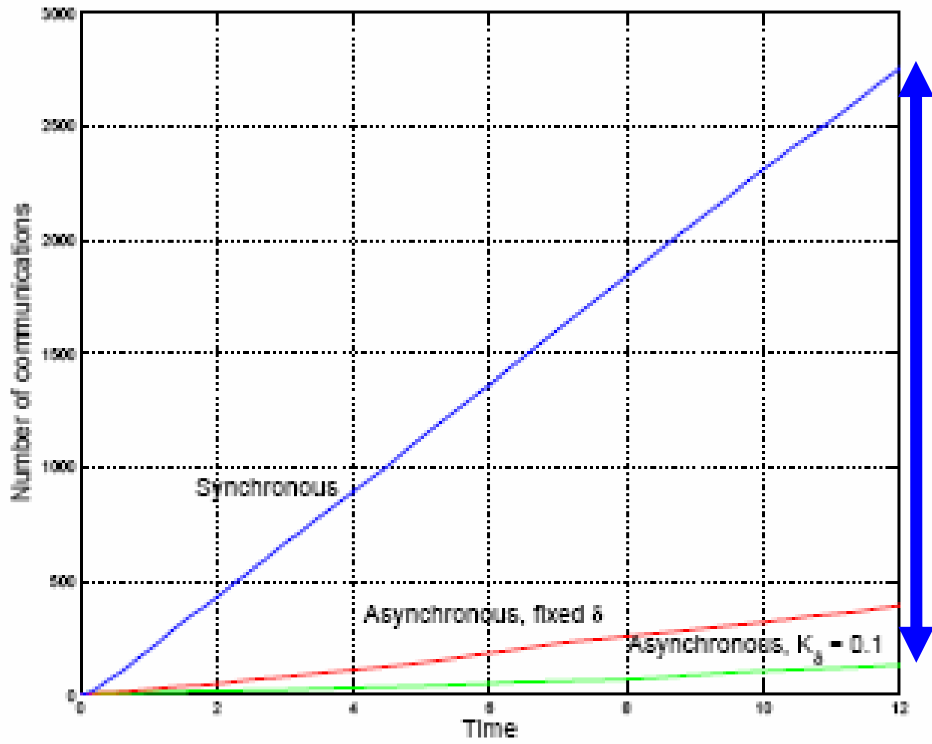
$$\lim_{k\to\infty} \nabla H(\mathbf{s}(k)) = 0$$

NOTE: The requirements on α and K_δ depend on $\boldsymbol{D}$ and they are tighter.

Zhong and Cassandras, IEEE TAC, 2010

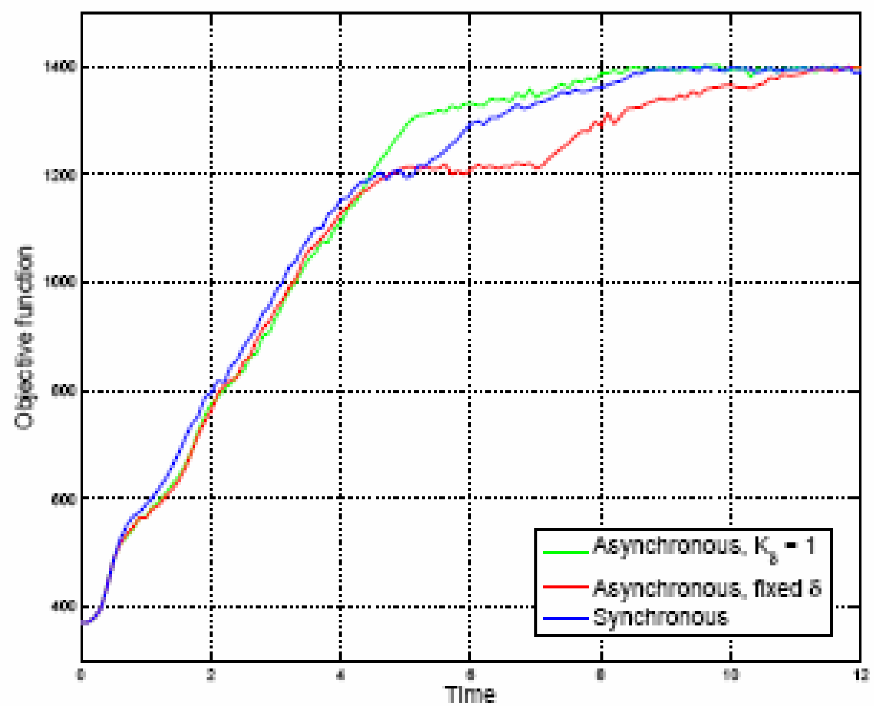
SYNCHRONOUS v ASYNCHRONOUS OPTIMAL COVERAGE PERFORMANCE

Energy savings + Extended lifetime



SYNCHRONOUS v ASYNCHRONOUS:

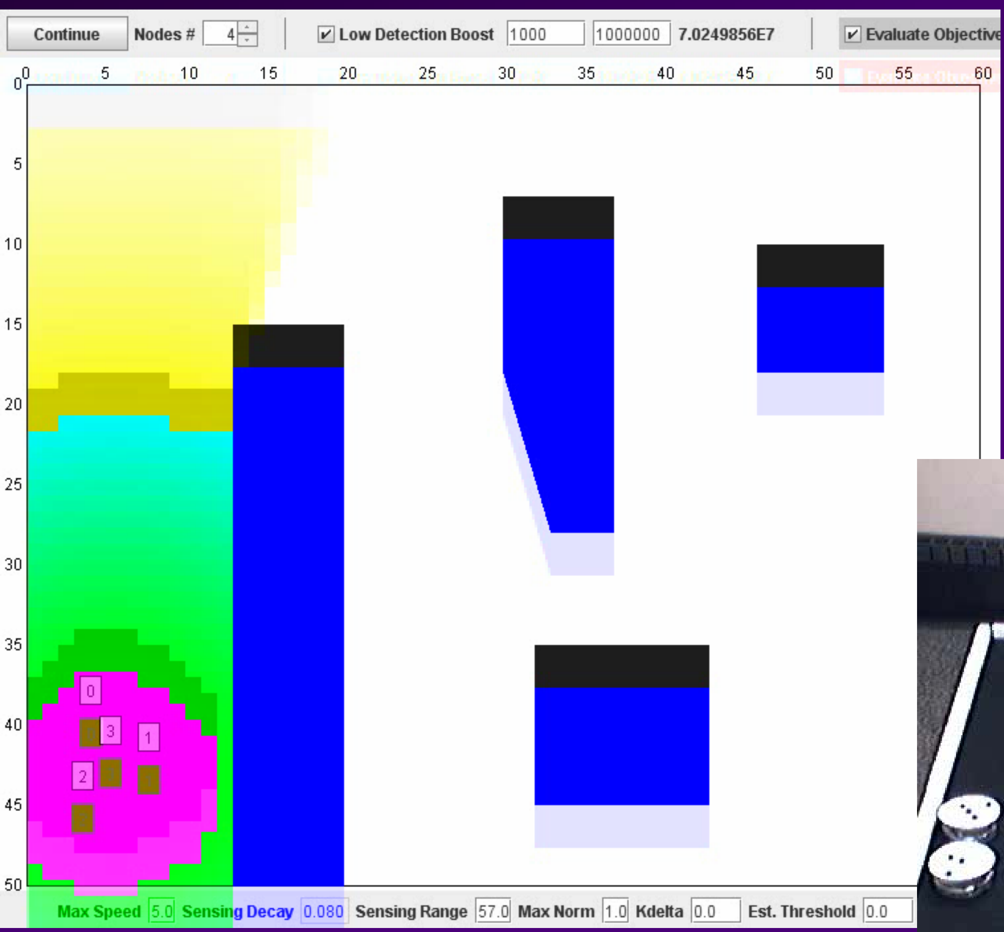
No. of communication events
for a deployment problem *with obstacles*

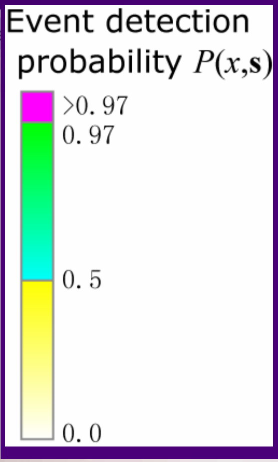


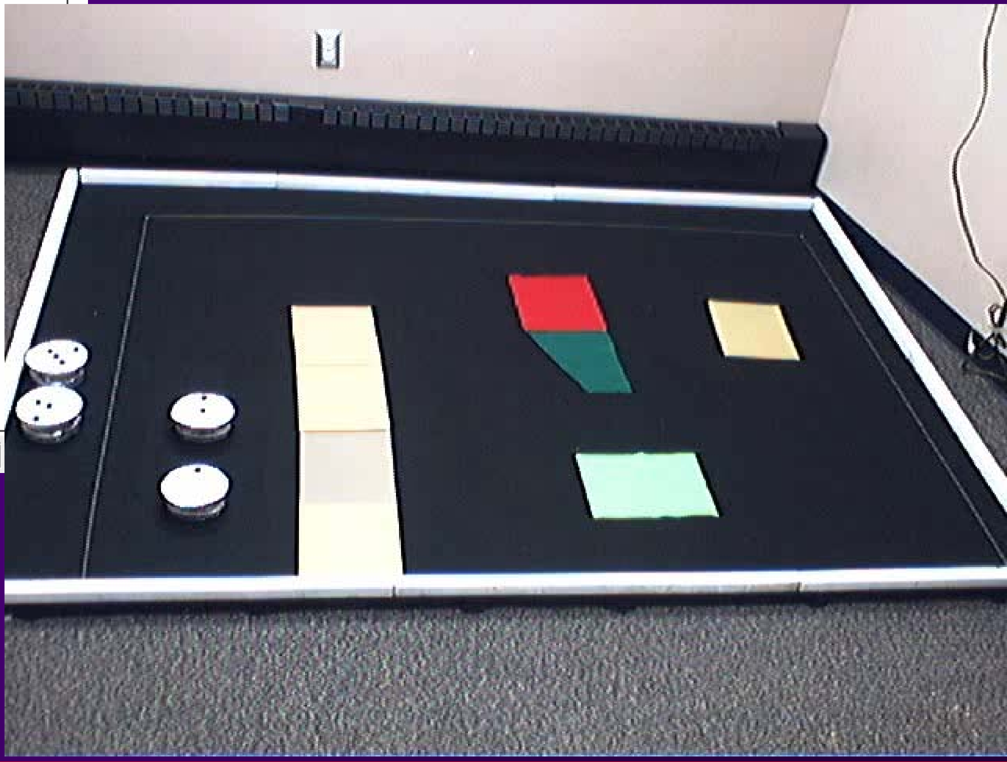
SYNCHRONOUS v ASYNCHRONOUS:

Achieving optimality
in a problem *with obstacles*

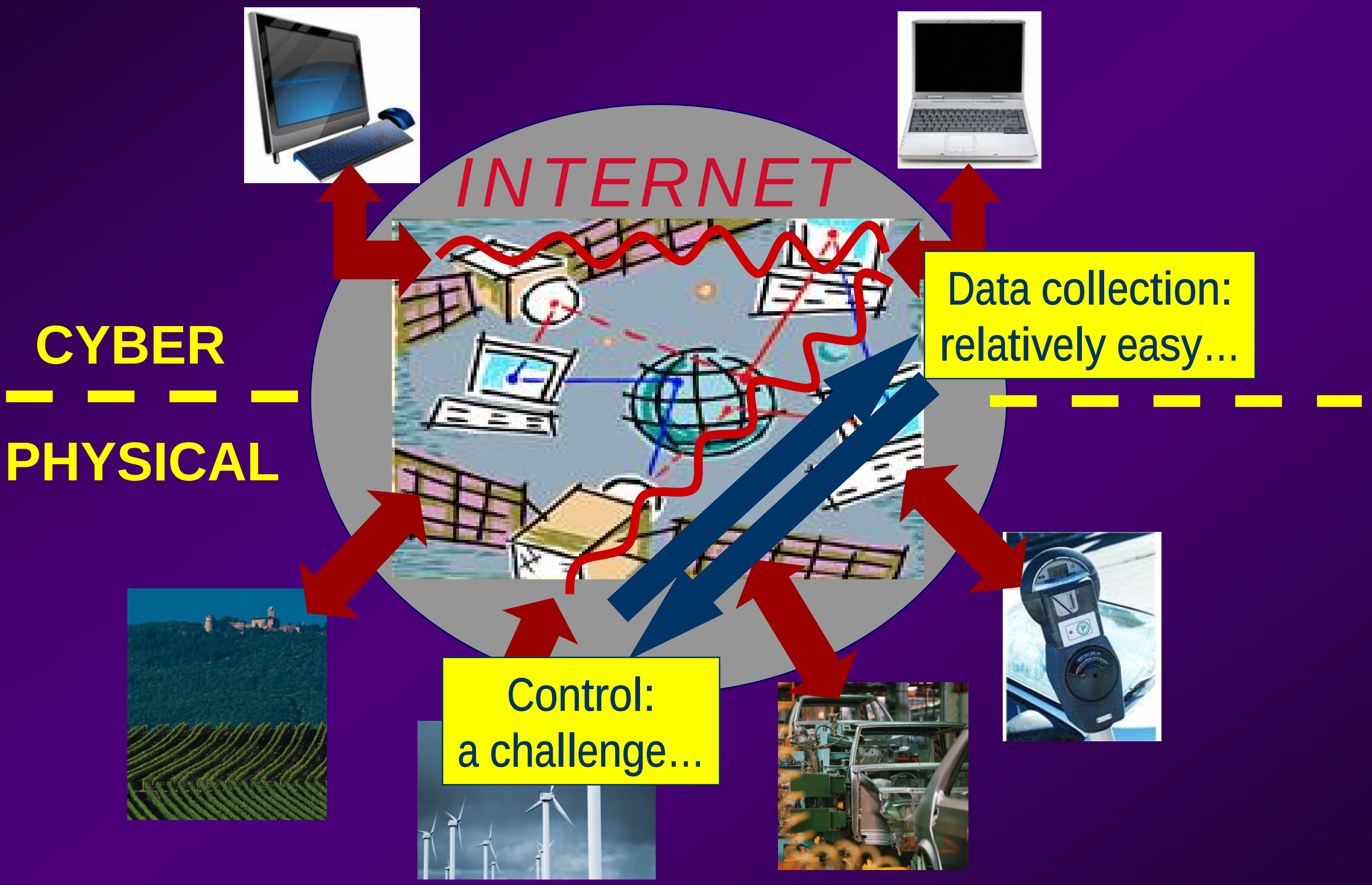
DEMO: OPTIMAL DISTRIBUTED DEPLOYMENT WITH OBSTACLES – *SIMULATED AND REAL*







SENSOR + *ACTUATION* NETWORK



SENSOR + ACTUATION: A "SMART PARKING" SYSTEM

"SMART PARKING" - MOTIVATION

30% of vehicles on the road in the downtowns of major cities are cruising for a parking spot. It takes the average driver 7.8 minutes to find a parking spot in the downtown core of a major city.

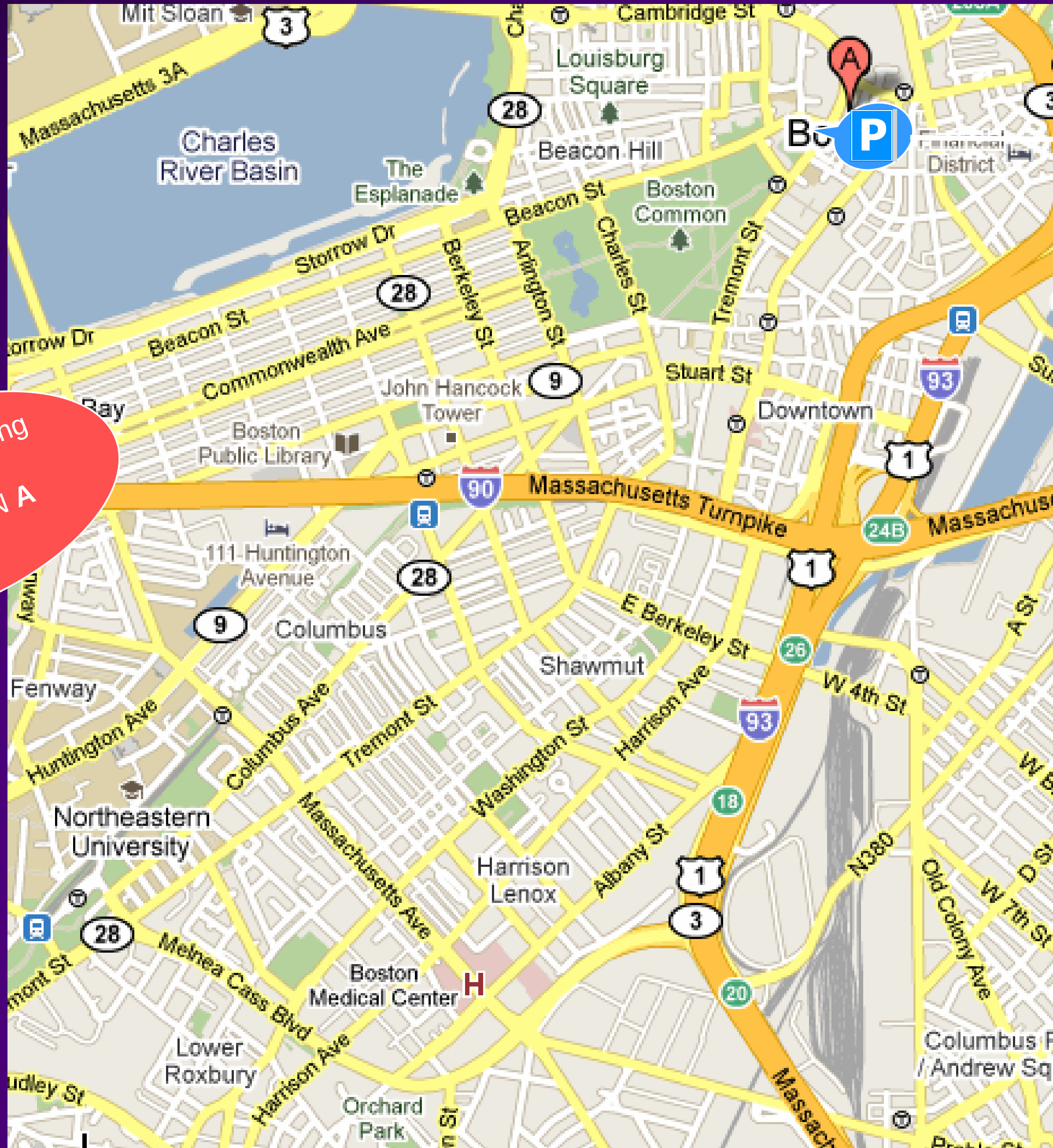
R. Arnott, T.Rave, R.Schob, *Alleviating Urban Traffic Congestion.* 2005

Over one year in a small Los Angeles business district, cars cruising for parking created the equivalent of 38 trips around the world, burning 47,000 gallons of gasoline and producing 730 tons of carbon dioxide.

Donald Shoup, *The High Cost of Free Parking.* 2005

“SMART PARKING” - CONCEPT

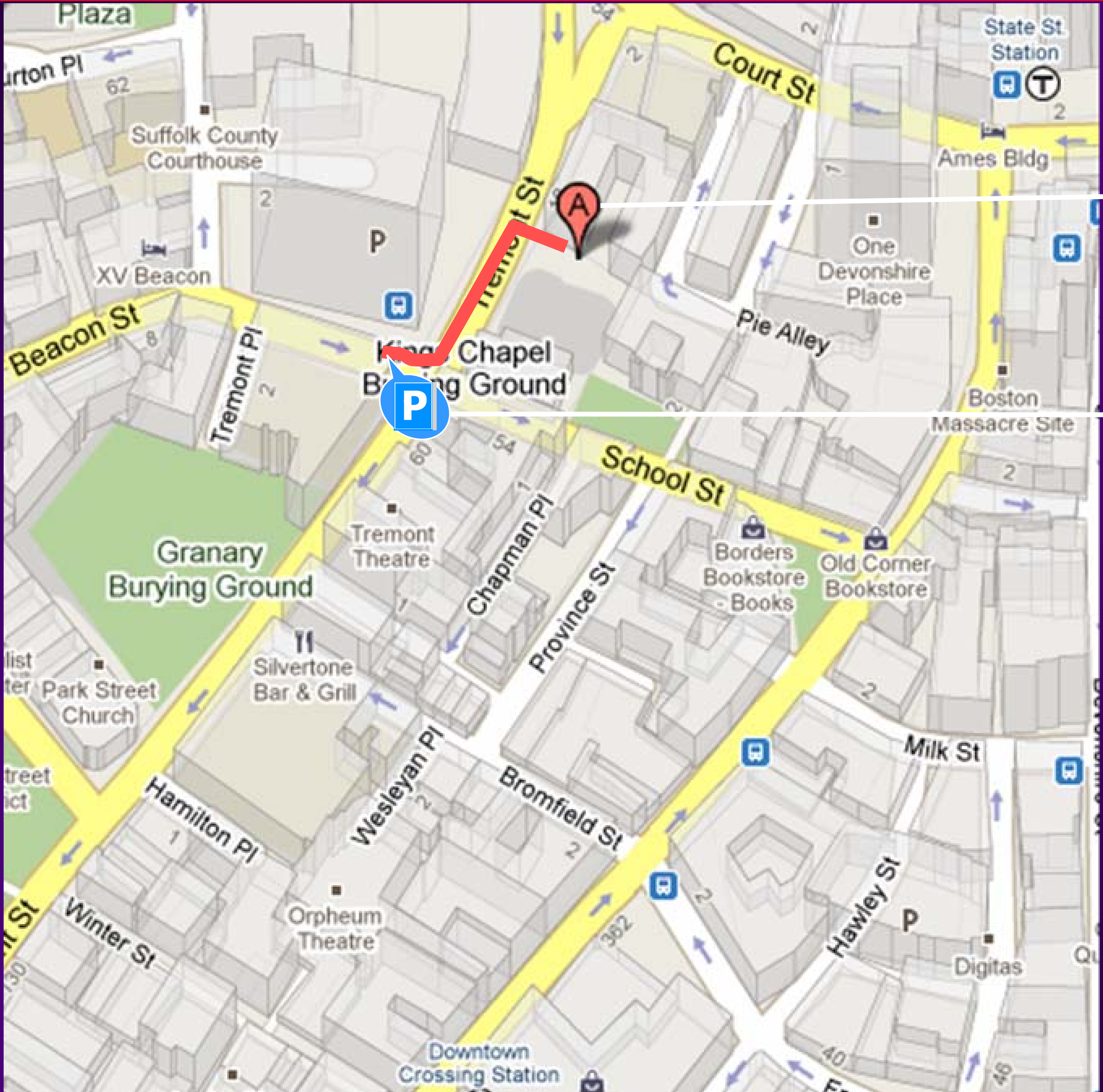
Find optimal parking spot for DESTINATION **A**



OPTIMAL PARKING SPOT

Minimize a function of COST and DISTANCE from A

"SMART PARKING" - CONCEPT



GUIDANCE-BASED PARKING – DRAWBACKS…

Drivers:

- May not find a vacant space
- May miss better space
- Processing info while driving

City:

- Imbalanced parking utilization
- May create ADDED CONGESTION (as multiple drivers converge to where a space exists)

Searching for parking ⇒ Competing for parking

SMART PARKING – NEW FEATURES

- System finds BEST parking space for driver
  (based on PROXIMITY to destination + parking COST)

- Space RESERVED $\Rightarrow$ guaranteed parking space

- System continuously IMPROVES assigned parking space

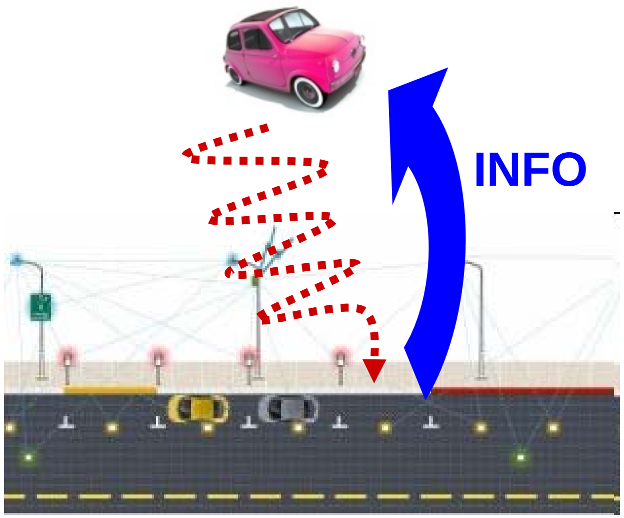
- System ensures FAIRNESS in parking space allocation

- Parking space UTILIZATION INCREASES

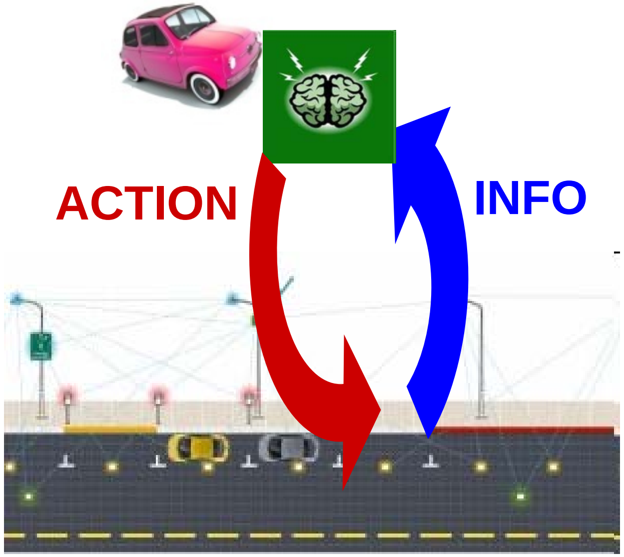
Driver makes decisions $\Rightarrow$ System makes *optimal* decisions for driver

GUIDANCE-BASED PARKING v "SMART PARKING"

COLLECTING DATA IS NOT "SMART", JUST A NECESSARY STEP TO BEING "SMART"



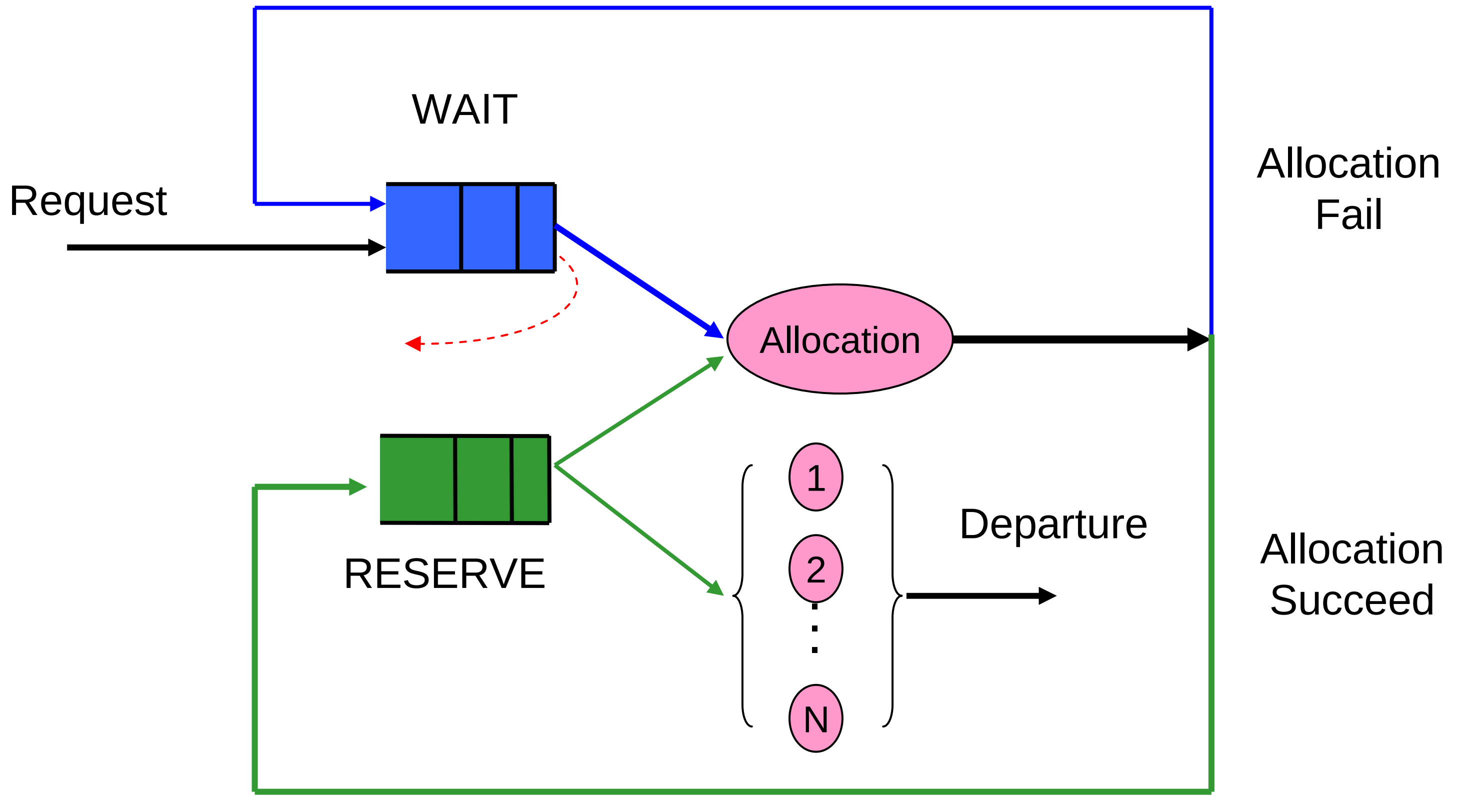
PROCESSING DATA TO MAKE GOOD DECISIONS IS "SMART"



SMART PARKING – IMPLEMENTATION

- Parking space availability detection
  - Standard sensors (e.g., magnetic, cameras)
  - Wireless sensor networking
- Vehicle localization
  - GPS
- System-Driver communication
  - Smartphone
  - Vehicle navigation system
- Parking reservation
  - Folding/Retreating barrier
  - Red/Green/Yellow light system

PROBLEM FORMULATION



OBJECTIVE FUNCTION

Objective function
at ***k*th** decision point:

$$J(k) = \min_{X} \sum_{i \in W(k) \cup R(k)} \sum_{j \in \Omega_i(k)} x_{ij} \cdot J_{ij}(k)$$

Decision variables:

$$x_{ij} = \begin{cases} 0 & \textit{if user i is NOT assigned to resource j} \\ 1 & \textit{if user i is assigned to resource j} \end{cases}$$

User cost function:

$$J_{ij} = \lambda_i \cdot \frac{M_{ij}(k)}{\boxed{M_i}} + (1 - \boxed{\lambda_i}) \cdot \frac{D_{ij}}{\boxed{D_i}}$$

cost upper bound

max proximity to dest.

weight

MIXED INTEGER LINEAR PROBLEM (MILP)

$$\min \underbrace{\sum_{i\in W(k)\cup R(k)} \sum_{j\in\Omega_i(k)} x_{ij}\cdot J_{ij}(k)}_{\text{Satisfied User Cost}} + \underbrace{\sum_{i\in W(k)} (1-\sum_{j\in\Omega_i(k)} x_{ij})}_{\text{Unsatisfied User Cost}}$$

s.t.

$$\sum_{i\in W(k)\cup R(k)} x_{ij} \le 1 \quad \forall j\in\Gamma(k)$$

$$\sum_{j\in\Omega_i(k)} x_{ij} \le 1 \quad \forall i\in W(k)$$

$$\boxed{\sum_{j\in\Omega_i(k)} x_{ij} = 1 \quad \forall i\in R(k)}$$

Reservation Guarantee

$$\boxed{\sum_{j\in\Omega_i(k)} x_{ij}\cdot J_{ij}(k) \le J_{iq_i(k-1)}(k) \quad \forall i\in R(k)}$$

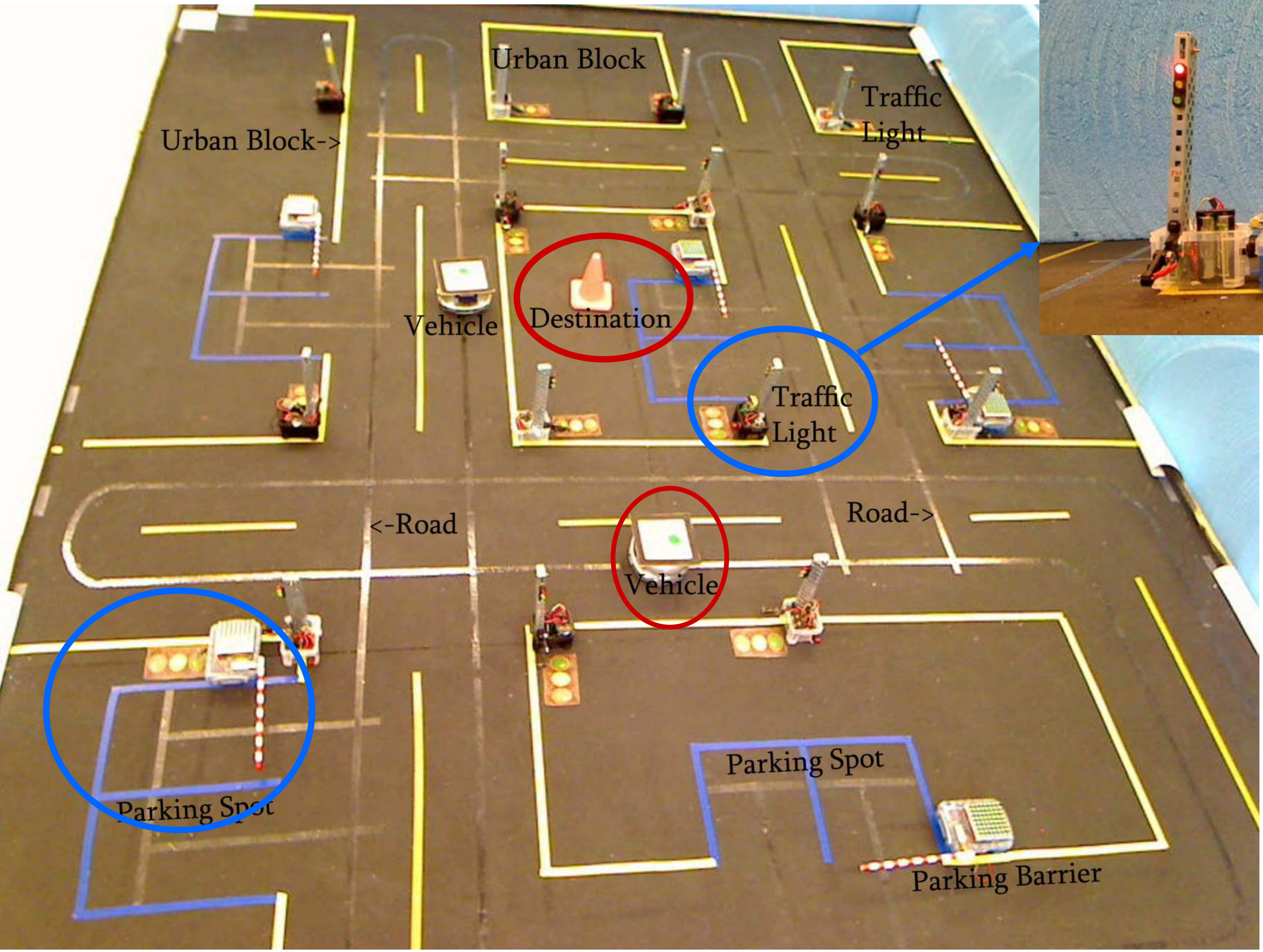
Reservation Upgrade

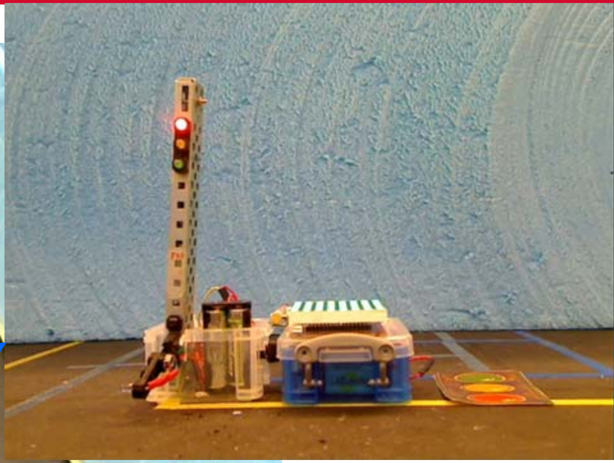
$$\boxed{\begin{aligned}(\sum_{n\in\Omega_i(k)} x_{in}) - x_{mj} \ge 0 \quad &\forall j\in\Gamma(k), i\in\{i \mid j\in\Omega_i(k)\},\\ &m\in\{m \mid j\in\Omega_m(k), t_{mj} > t_{ij}, m\in W(k)\}\end{aligned}}$$

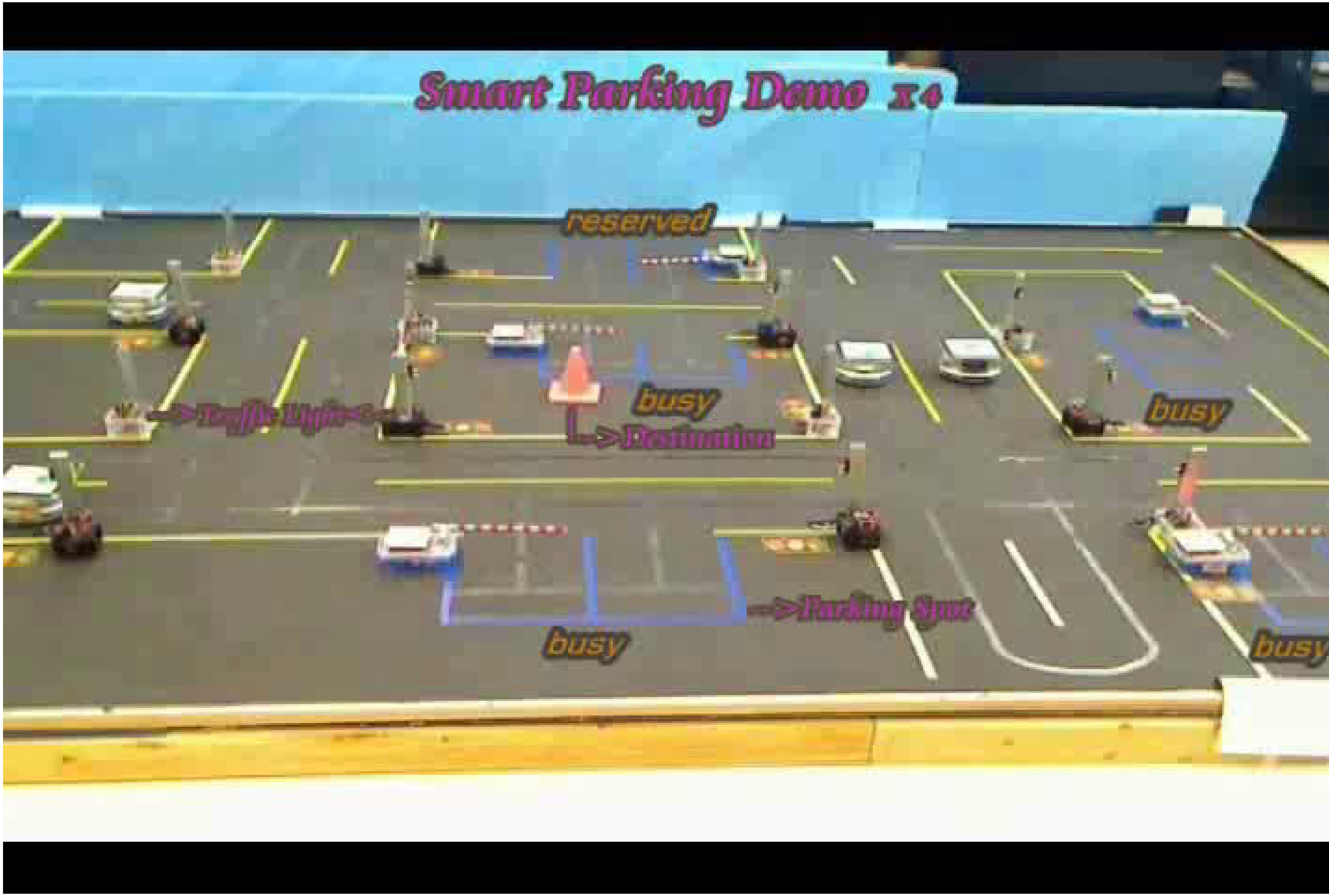
Fairness

$$x_{ij}\in\{0,1\} \quad \forall i\in W(k)\cup R(k), j\in\Omega_i(k)$$

"SMART PARKING" TEST BED






Smart Parking Demo x4
reserved
busy
busy
-->Traffic Light<--
-->Destination
-->Parking Spot
busy
busy

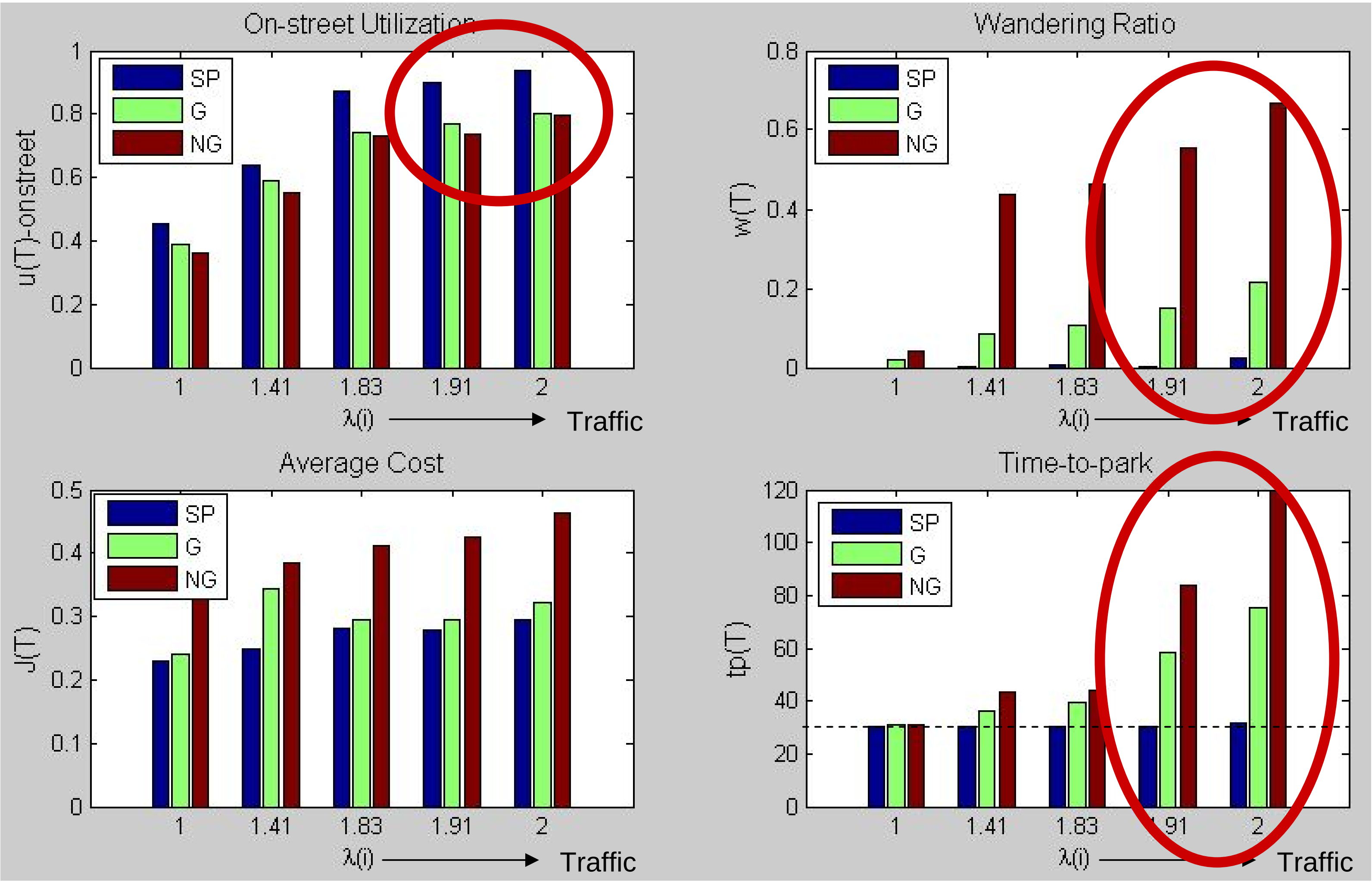
SIMULATION CASE STUDY

On-street parking spaces

Off-street parking spaces

Points of interest

CASE STUDY RESULTS



SP: BU Smart Parking system

G: Parking using guidance-based systems

NG: No guidance (status quo)

KEY CONCLUSIONS

1. 10-20% higher parking utilization
   $\Rightarrow$ HIGHER REVENUE,
   LOWER CONGESTION

2. % drivers searching for parking (wandering) < 2%
   $\Rightarrow$ HIGHER REVENUE,
   LOWER CONGESTION

3. 50% reduction in parking time under heavy traffic
   $\Rightarrow$ LOWER CONGESTION,
   LESS FUEL,
   DRIVER COMFORT

IMPLEMENTATION

“Smart Parking” proof-of-concept study implemented in a small (27 space) garage at Boston University during summer 2011:

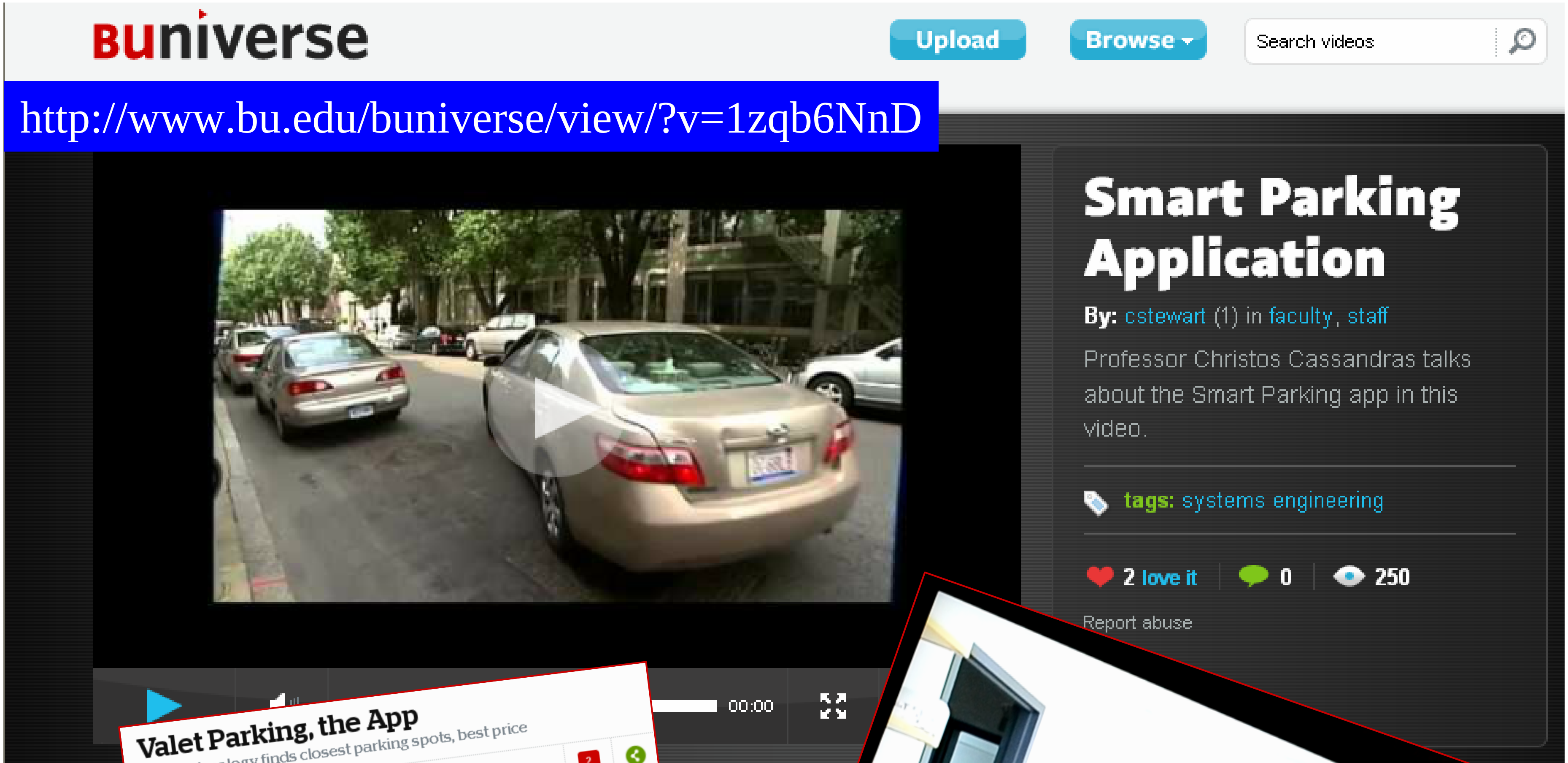
- ***Parking request through iPhone app.***

- ***Smart Parking Allocation Center (SPAC)*: Server located in CODES Lab**
SPAC determines optimal allocation for request (if one exists) and notifies driver through iPhone app showing the identity of reserved spot

- ***Garage gateway*: Laptop computer located in garage**

-***Sensor and light system device*: Custom-built device affixed on ceiling over each parking spot.**

http://www.bu.edu/buniverse/view/?v=1zqb6NnD




Valet Parking, the App

New technology finds closest parking spots, best price

08.30.2011 By Mark Dwortzan

http://www.necn.com/09/23/11/JoeBattParkingapp/landing_scitech.html?blockID=566574&feedID=4213

PROJECT TEAM, RECOGNITION

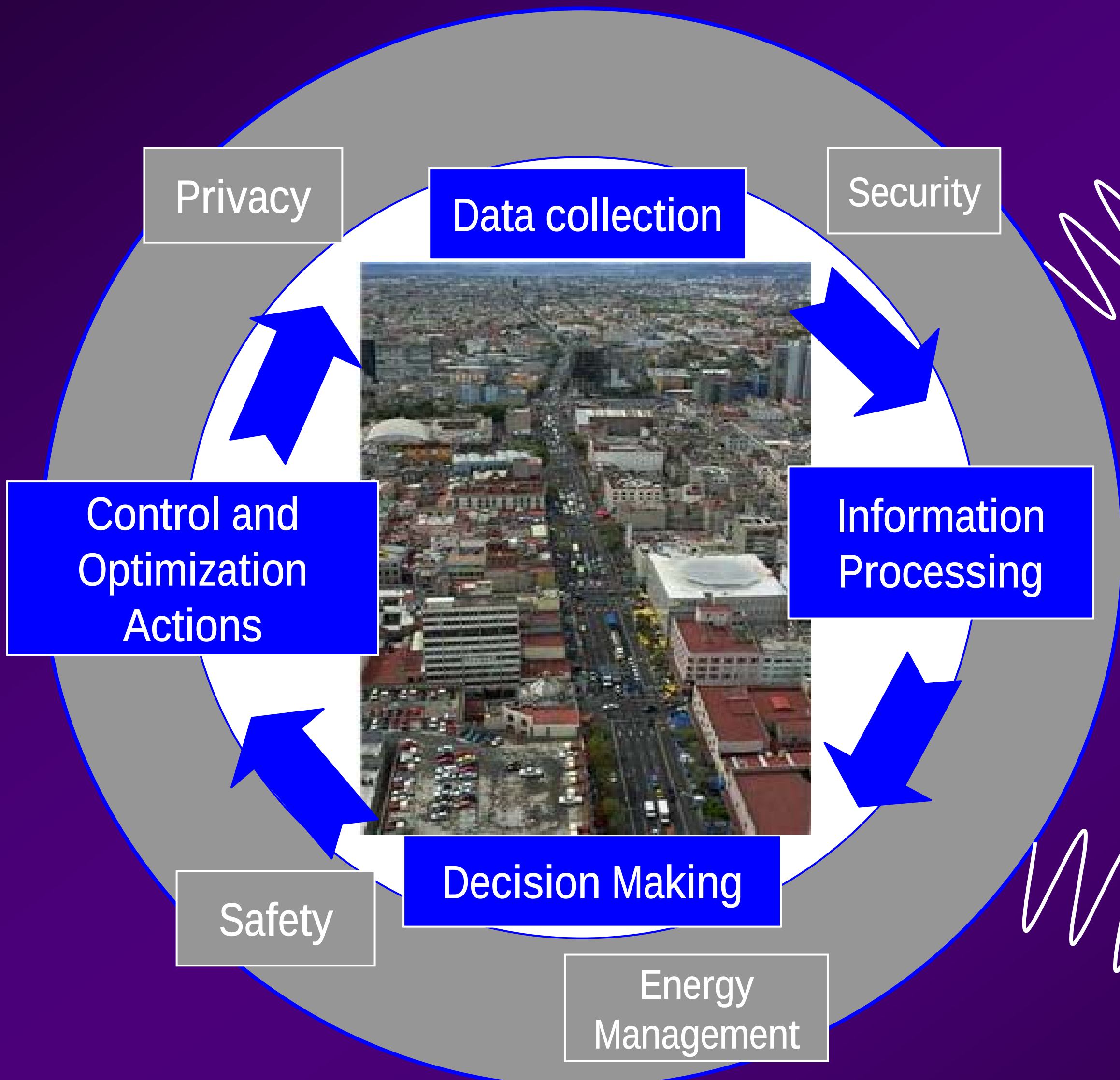
TEAM: Yanfeng Geng (PhD student), Ted Grunberg (Undergrad. Student), Andy Ochs, Mikhail Gurevich, Greg Berman (BU SOM students)

- *2011 IBM/IEEE Smarter Planet Challenge competition*, team won 2nd place prize
- *Best Student Paper Award*, Finalist, 2011 IEEE Multi-Conference on Systems and Control
- *Third prize poster* on "Smart Parking", INFORMS 2011 Northeastern Conference
- Ongoing implementation under BU OTD *"Ignition Award"*
- Working with City of Boston under *IBM Award* for "Combating Climate Change Through Smarter Urban Transportation Policies"

- Geng, Y., and Cassandras, C.G., "Dynamic Resource Allocation in Urban Settings: A "Smart Parking" Approach", Proc. of *2011 IEEE Multi-Conference on Systems and Control*, Oct. 2011.
- Geng, Y., and Cassandras, C.G., "A New "Smart Parking" System Based on Optimal Resource Allocation and Reservations", *Proc. of 14th IEEE Intelligent Transportation Systems Conf.*, pp. 979-984, Nov. 2011.

http://www.bu.edu/buniverse/view/?v=1zqb6NnD

“SMART CITY” AS A CYBER-PHYSICAL SYSTEM



TRAFFIC LIGHT CONTROL

Geng, Y., and Cassandras, C.G., “Traffic Light Control Using Infinitesimal Perturbation Analysis”, subm. to *51st IEEE Conf. Decision and Control*, 2012

SENSOR NETWORKS

“SMART PARKING”