DISTRIBUTED OPTIMIZATION FOR COOPERATIVE MISSIONS IN UNCERTAIN ENVIRONMENTS

C. G. Cassandras

Center for Information and Systems Engineering
Boston University

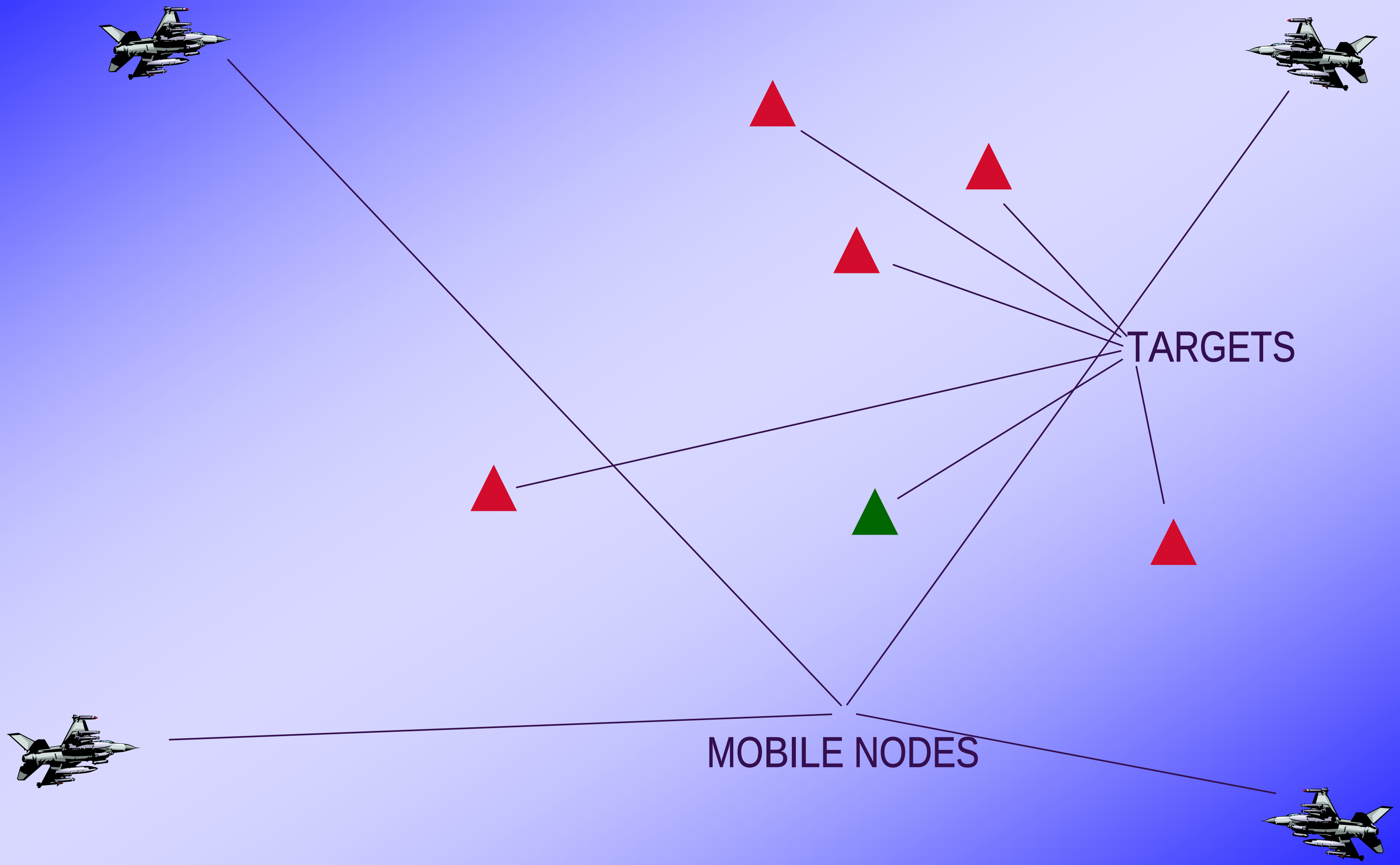
OUTLINE

- COOPERATIVE "MISSION" SETTING
- REWARD MAXIMIZATION MISSIONS
- COOPERATIVE RECEDING HORIZON (CRH) CONTROL
- COVERAGE CONTROL MISSIONS
- DEMOS: Applets and Movies

ACKNOWLEDGEMENTS:

PhD Students: Wei Li (PhD 2006), Ning Xu, Minyi Zhong
Sponsors: NSF, AFOSR, ARO, DOE, Honeywell
Members of Boston U. Sensor Networks Consortium

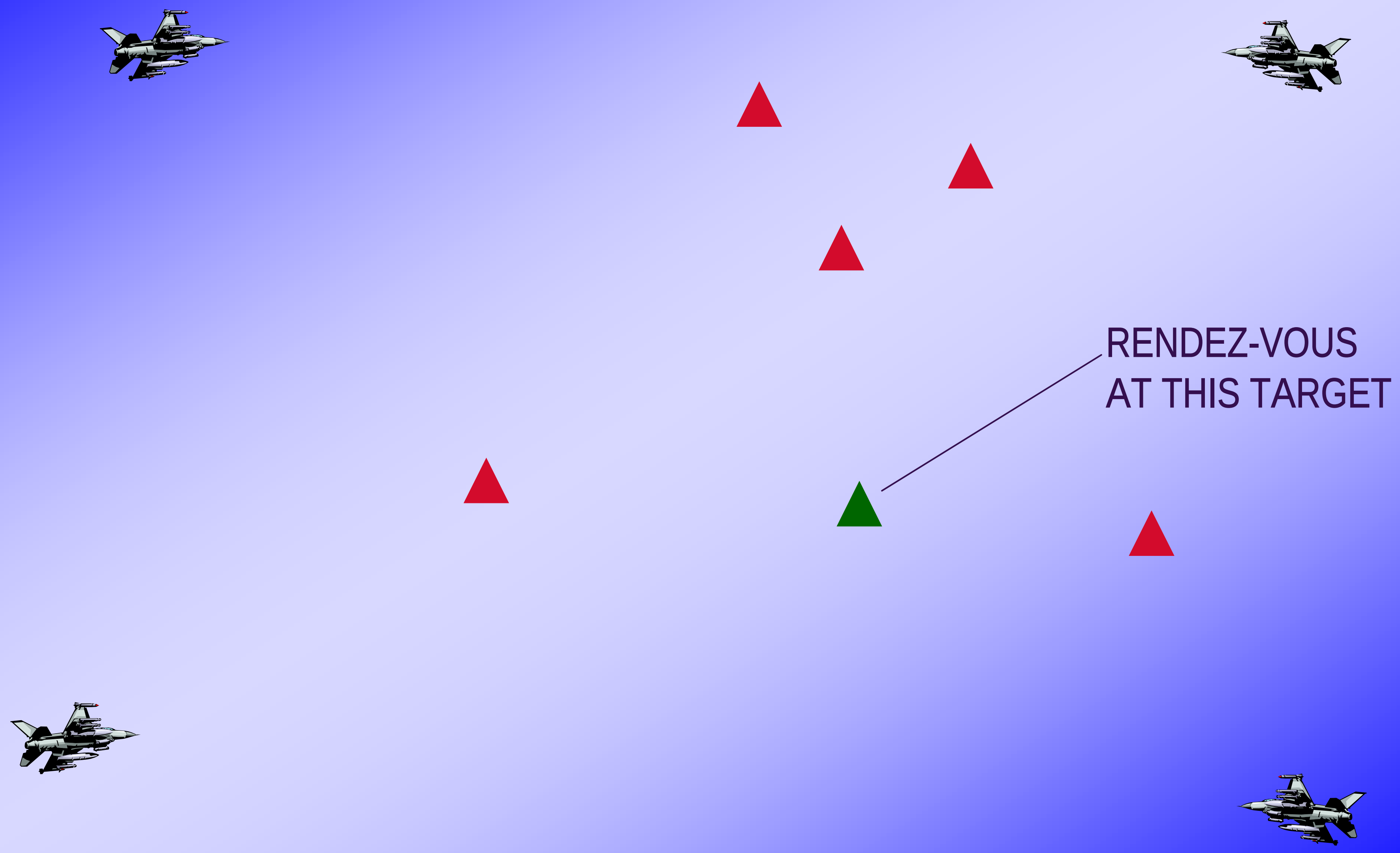
COOPERATIVE MISSION SETTING



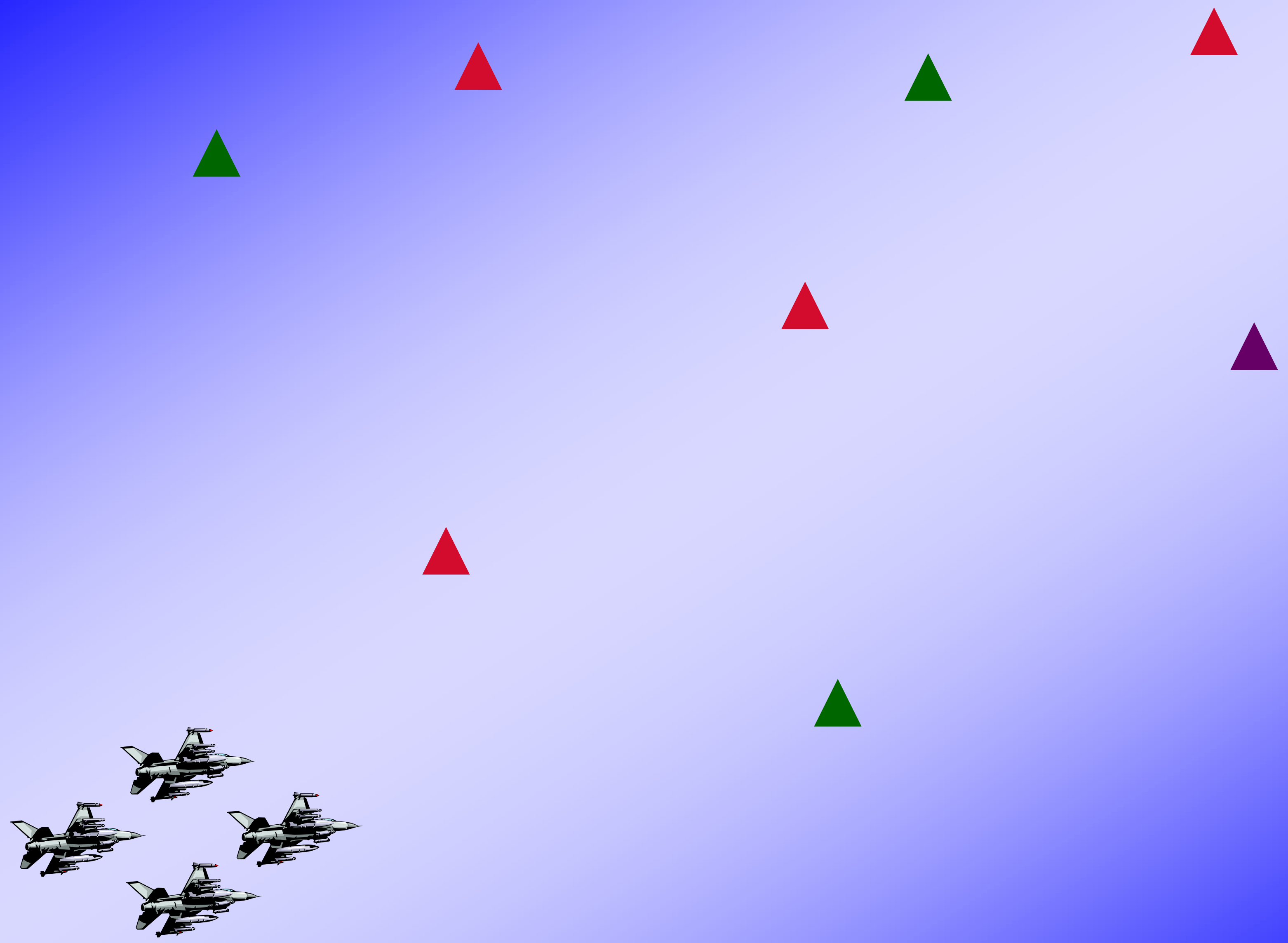
DIFFERENT COOPERATIVE MISSION TYPES

- RENDEZ-VOUS AT SOME TARGET POINT
- FORMATION MAINTENANCE
- REWARD MAXIMIZATION
- COVERAGE CONTROL

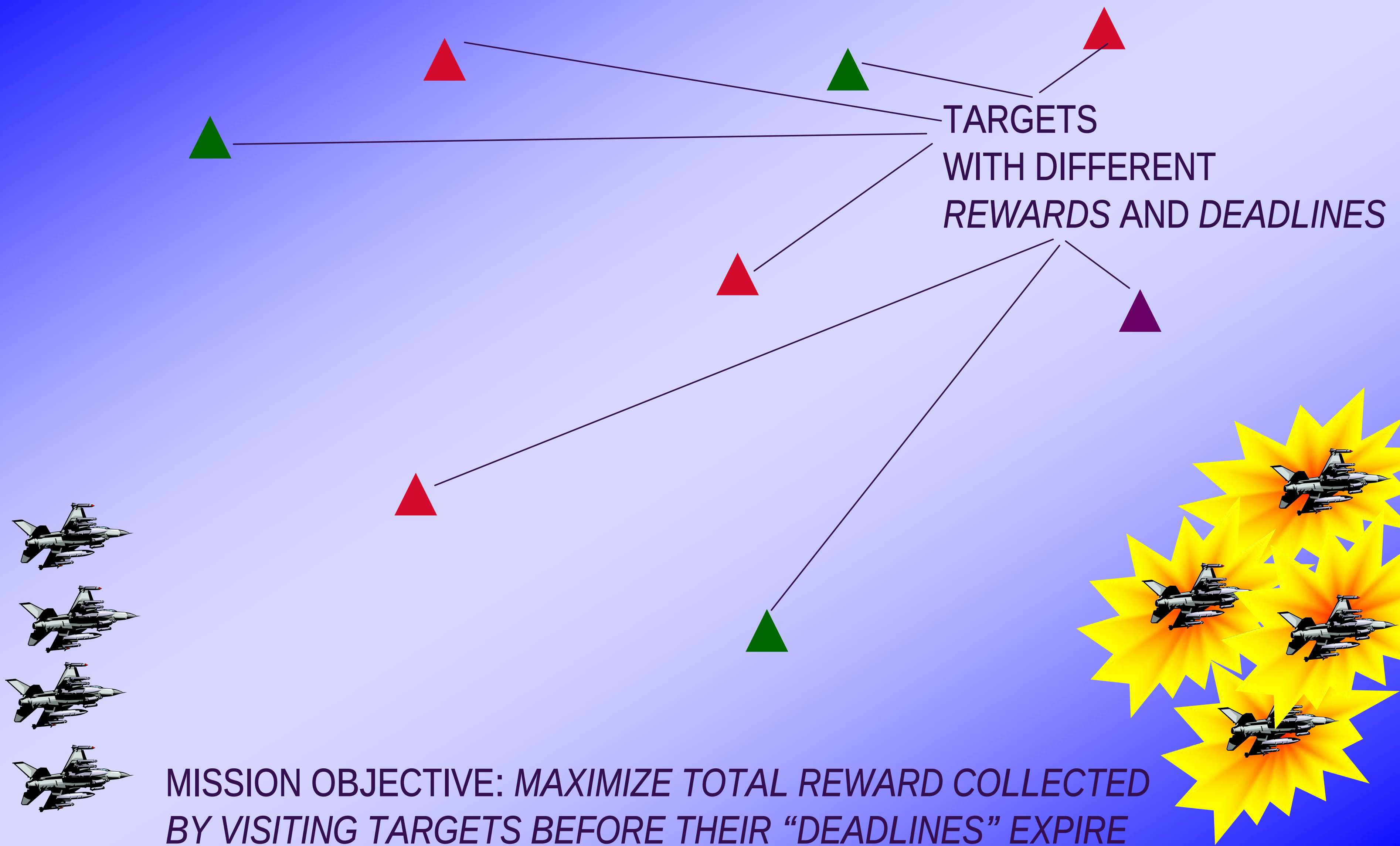
RENDEZ-VOUS MISSION



FORMATION MAINTAINANCE MISSION



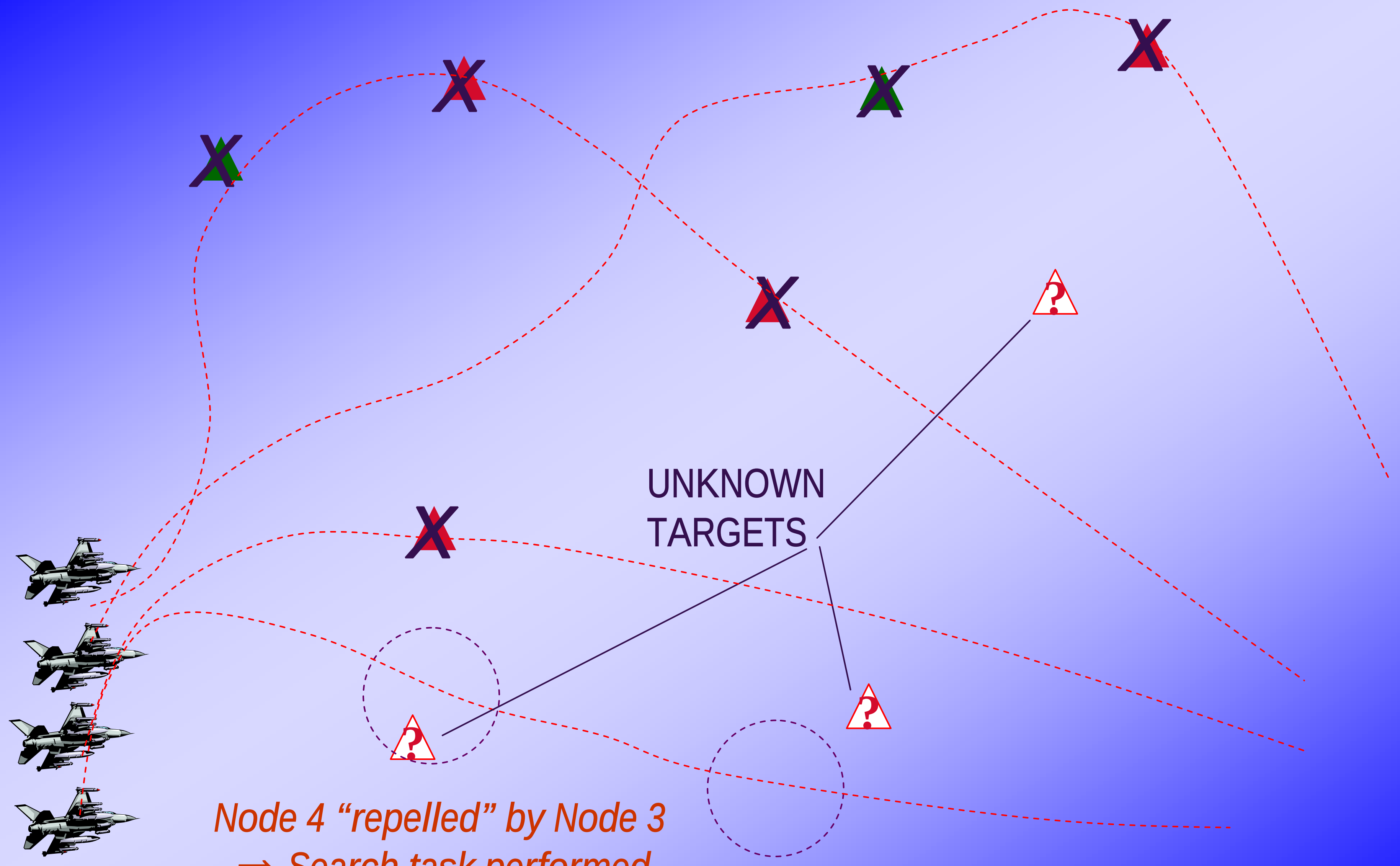
REWARD MAXIMIZATION MISSION



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

REWARD MAXIMIZATION MISSION

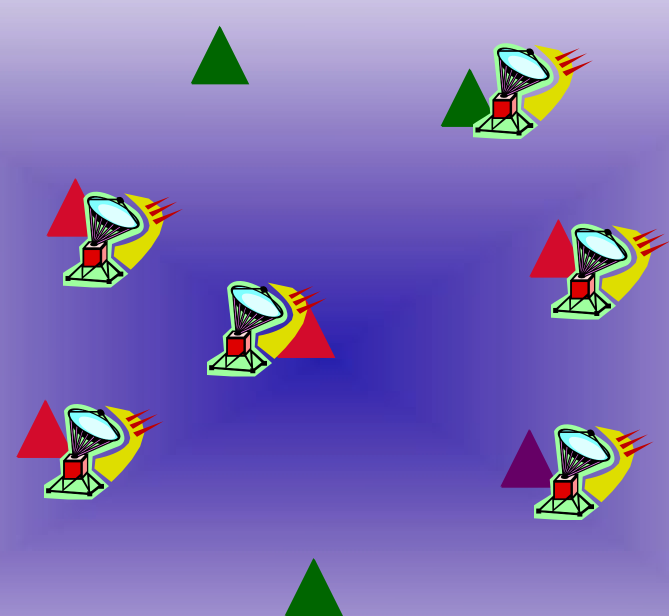
CONTINUED

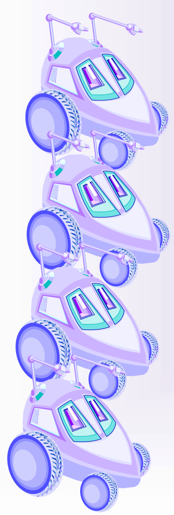


Node 4 "repelled" by Node 3
⇒ Search task performed

COVERAGE CONTROL MISSION

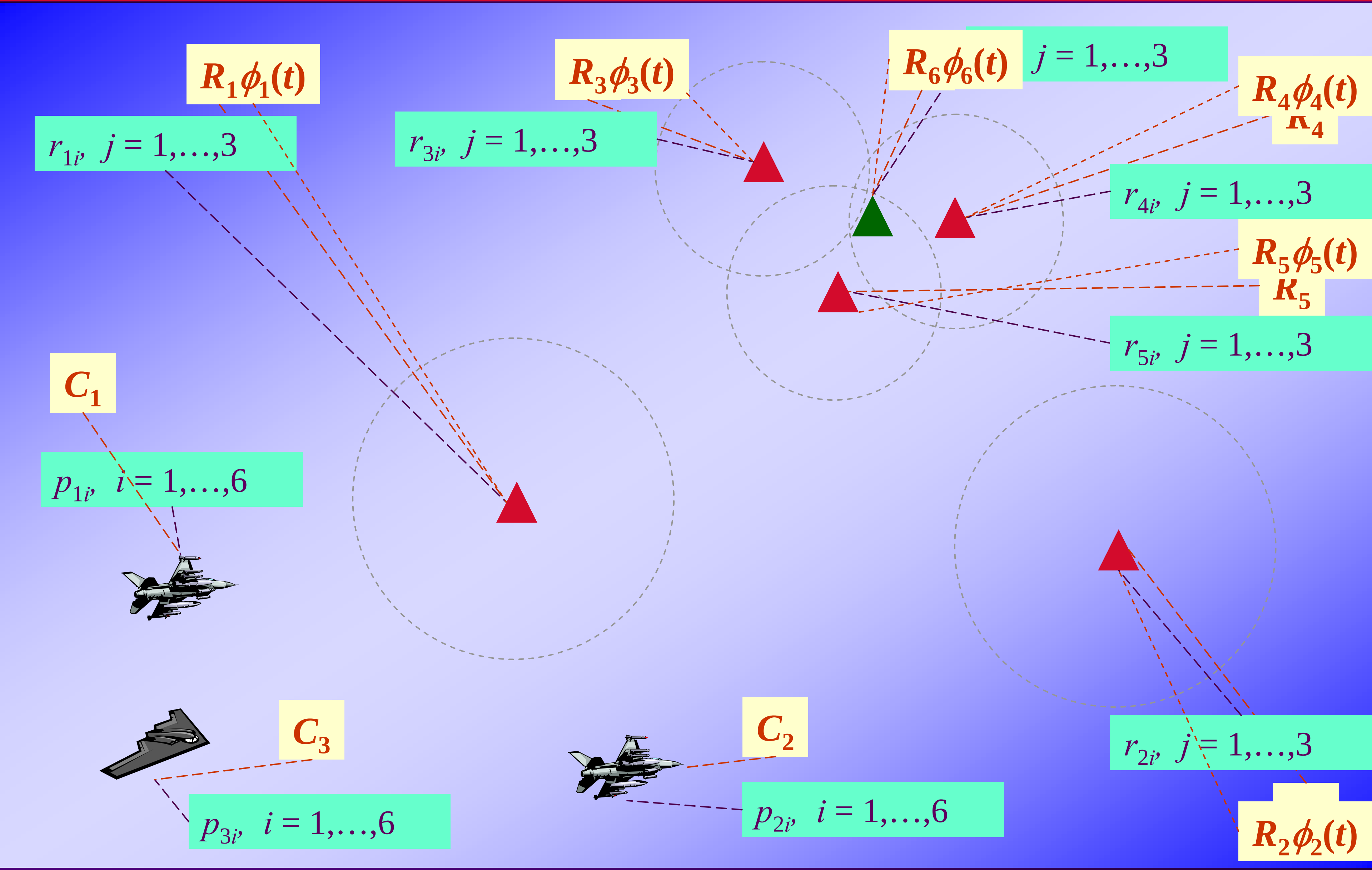
SENSOR FIELD WITH UNKNOWN DATA SOURCES
- ONLY DENSITY FUNCTION ASSUMED





- *Meguerdichian et al, INFOCOM, 2001,*
- *Cortes et al, IEEE Trans. on Robotics and Auto., 2004*
- *Cassandras and Li, Euro. J. of Control, 2005*

COOPERATIVE *REWARD MAXIMIZATION* MISSION



COOP. *REWARD MAXIMIZATION* MISSION

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

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

SOLUTION APPROACHES

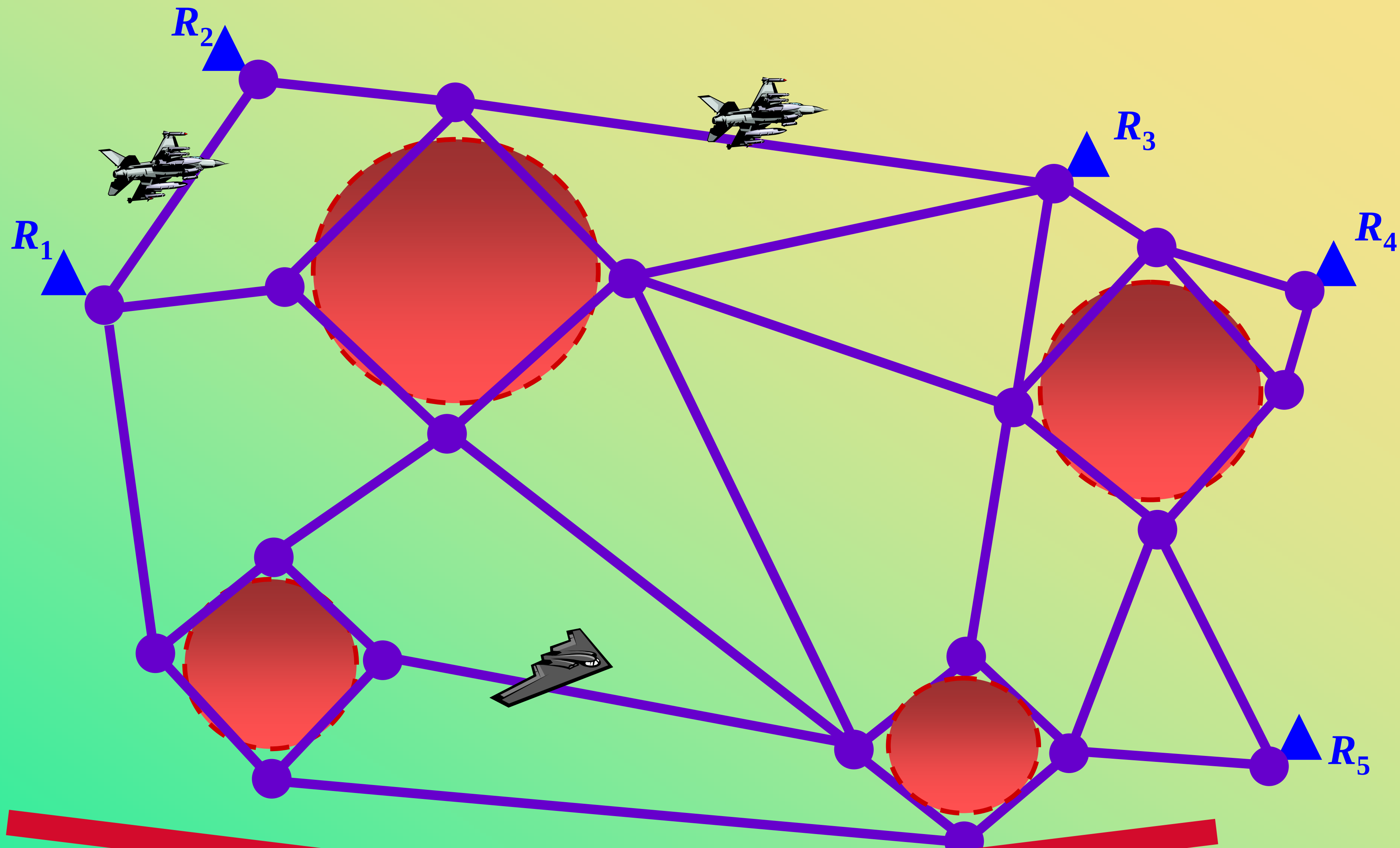
- Stochastic Dynamic Programming – *Wohletz et al, 2001*

  Extremely complex…

- Functional Decomposition:
  - Dynamic Resource Allocation – *Castanon and Wohletz, 2002*
  - Assignment Problems through Mixed Integer Linear Programming – *Bellingham et al, 2002*

    Combinatorially complex…
  - Path Planning – *Hu and Sastry, 2001, Lian and Murray 2002, Gazi and Passino, 2002, Bachmayer and Leonard, 2002*

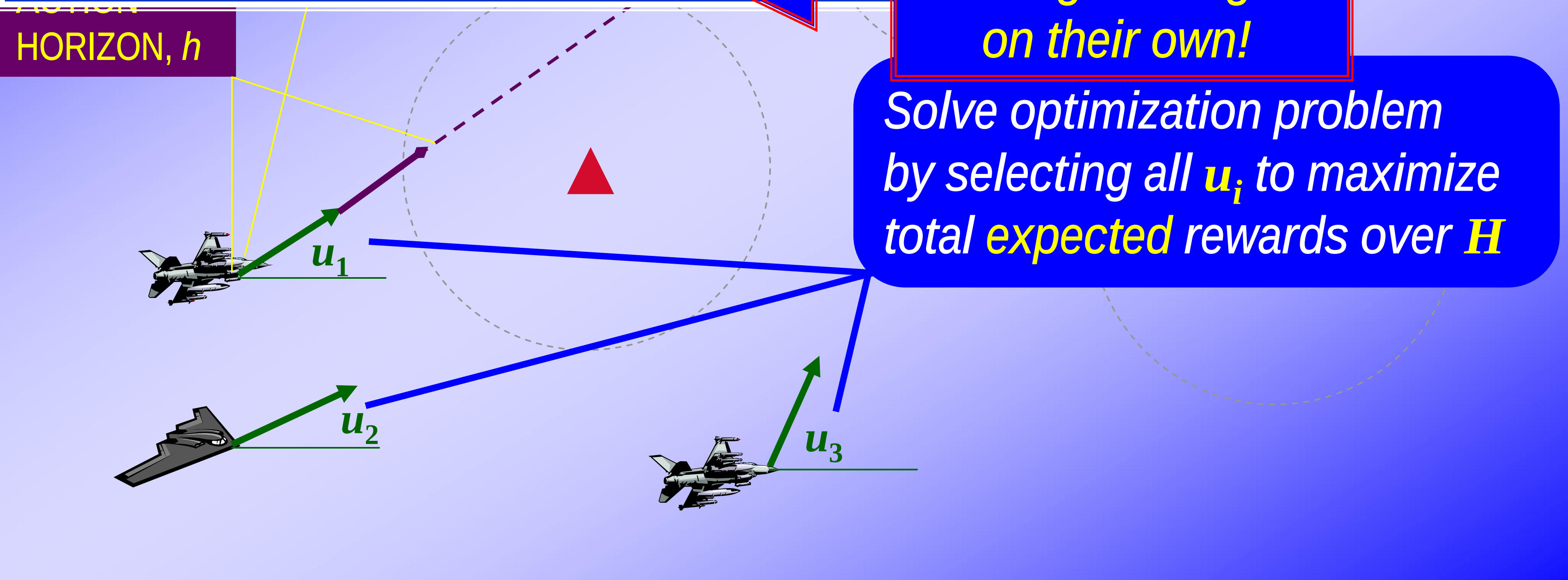
COMBINATORIAL + STOCHASTIC COMPLEXITY



~~1. Target Assignment → 2. Routing → 3. Path Control~~

RECEDING HORIZON (RH) 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



CONTRAST APPROACHES

HEDGE-AND-REACT	VS	ESTIMATE-AND-PLAN

HEDGE-AND-REACT

- Delay decisions until last possible instant
- No stochastic model
- Simpler opt. problems

⇓

Compare to *Model Predictive Control* (MPC)

VS

ESTIMATE-AND-PLAN

- Need accurate stochastic models
- Curse of dimensionality

CRH CONTROL PROBLEM FORMULATION

- Target positions ($i = 1,\ldots,N$): $y_i \in \mathbb{R}^2$

- Node dynamics ($j = 1,\ldots,M$):
  - State: $x_j(t) \in \mathbb{R}^2$ position of jth node at time t
  - Control: $u_j(t)$ Node heading at time t

$$\dot{x}_j(t) = V_j \begin{bmatrix} \cos u_j(t) \\ \sin u_j(t) \end{bmatrix}, \quad x_j(0) = x_j^0$$

- At *kth* iteration, time t_k ($k=1,2,\ldots$):
  - Planning Horizon: H_k
  - Node position at time t_k+H_k: $x_j(t_k + H_k) = x_j(t_k) + \dot{x}_j(t_k)H_k$

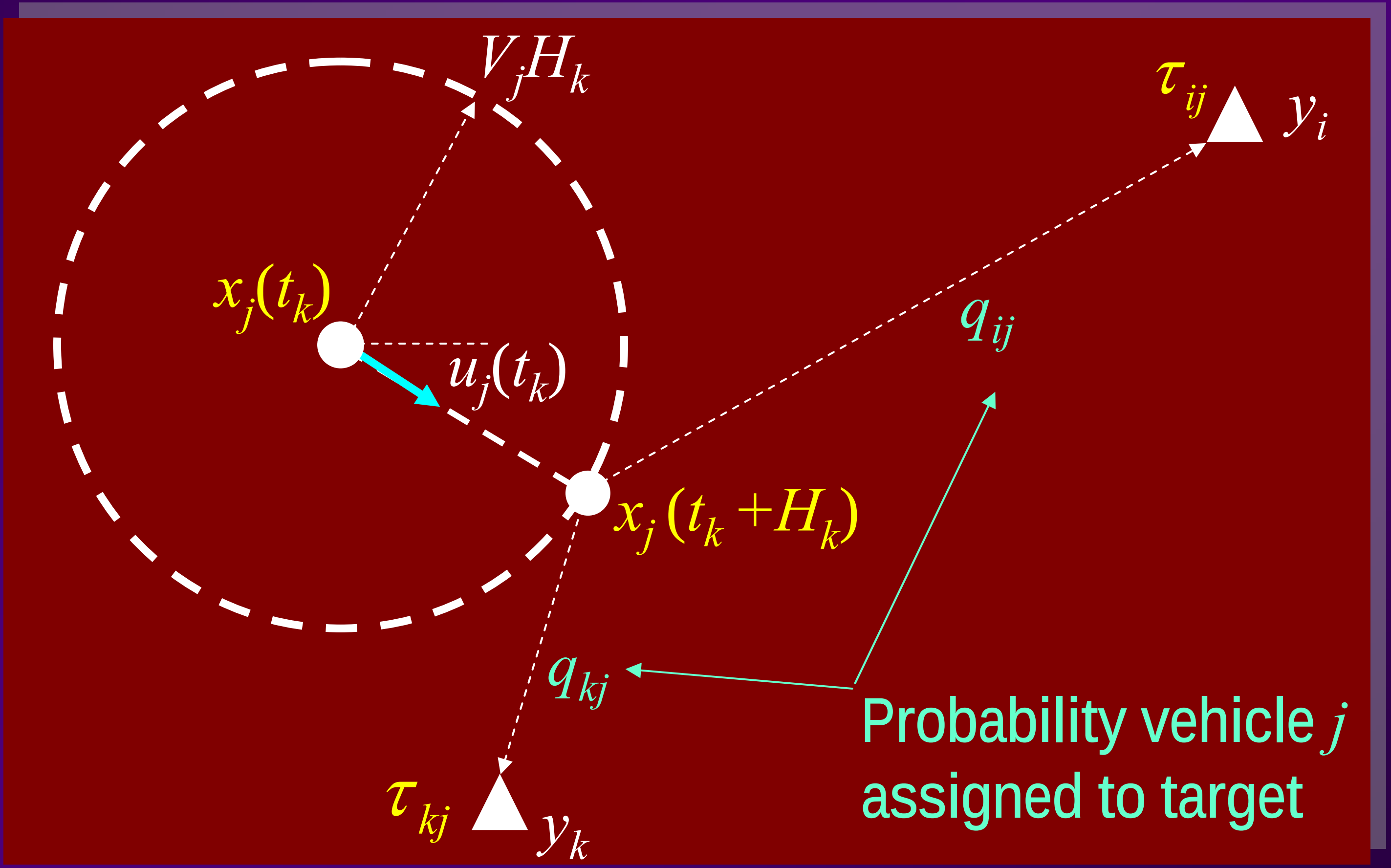
RH PROBLEM FORMULATION

CONTINUED

- At *kth* iteration (k=1,2,…):

Earliest time node j can reach target i under control $u_j(t_k)$:

$$\tau_{ij}(u_j(t_k),t_k) = (t_k + H_k) + \| x_j(t_k + H_k) - y_i \| / V_j$$



CRH PROBLEM FORMULATION

CONTINUED

- Objective at kth iteration:

Maximize *EXPECTED REWARD* over horizon H_k

Target i value attainable by node j [depends on $u_j(t)$]

node j value

$$\max_{\mathbf{u}} \sum_{i=1}^{M}\sum_{j=1}^{N} R_i\phi_i(\tau_{ij})\, p_{ij}(\tau_{ij})\, q_{ij}(t_k+H_k) - \sum_{i=1}^{M}\sum_{j=1}^{N} C_j r_{ij}(t_k+H_k)$$

Control node headings

Prob. node j collects target i reward [depends on $u_j(t)$]

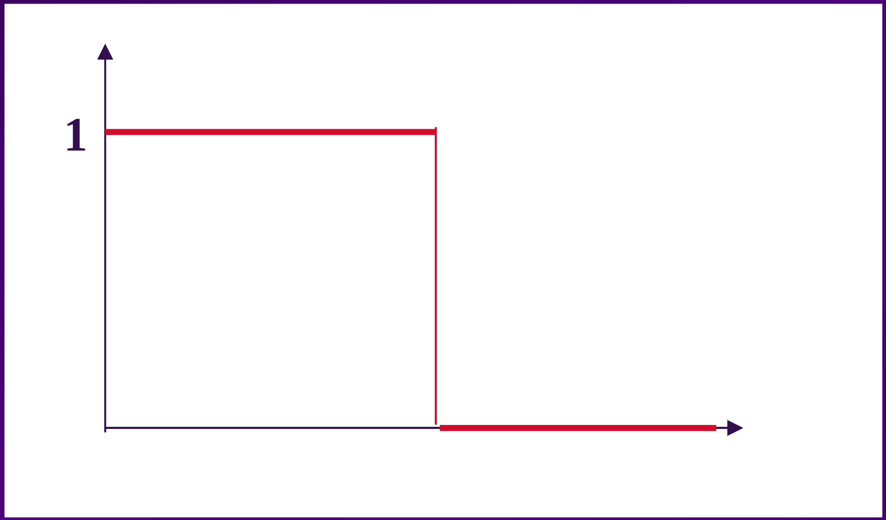
Earliest time when node j can collect reward from target i [depends on $u_j(t)$]

Prob. node j assigned to target i [depends on $u_j(t)$]

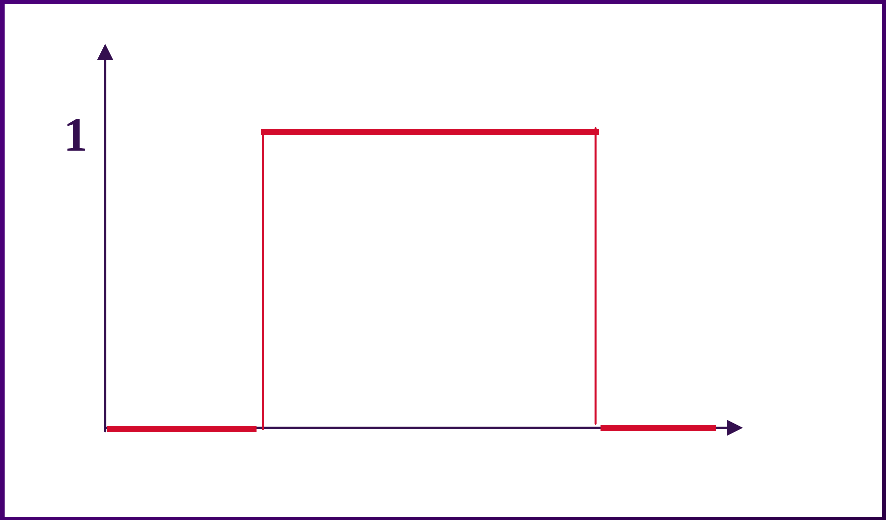
Prob. node j destroyed by target i [depends on $u_j(t)$]

THE FUNCTION $\phi_i(t)$ [REWARD DISCOUNTING FUNCTION]

- Targets with deadlines:



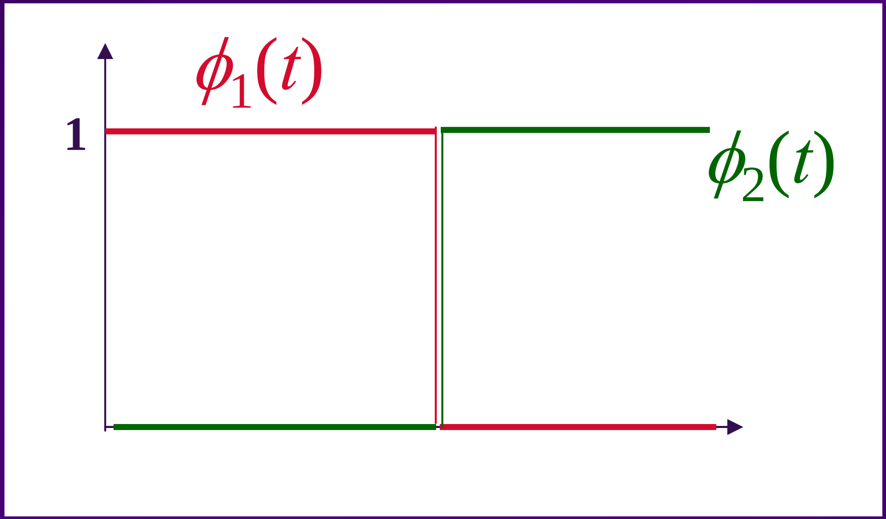
- Targets with time windows:



THE FUNCTION $\phi_i(t)$

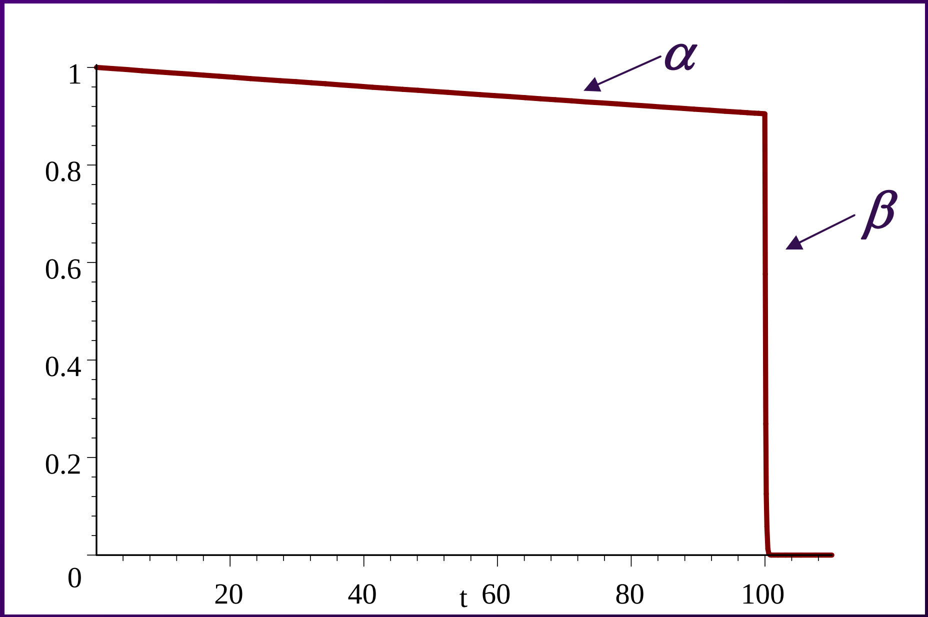
CONTINUED

- Sequencing targets:



- A general purpose ϕ–function:

$$\phi_i(t) = \begin{cases} e^{\frac{\ln(1-\alpha)}{D_i}t} & \text{if } t \le D_i \\ e^{\frac{\ln(1-\alpha)}{D_i}t} e^{-\beta(t-D_i)} & \text{if } t > D_i \end{cases}$$



THE FUNCTION q_{ij} [TARGET ASSIGNMENT FUNCTION]

- Node-to-target distance: $d_{ij} = \left\| x_j - y_i \right\|$

- Relative distance:

$$\delta_{ij} = \frac{d_{ij}}{\sum_{m=1}^{M} d_{im}}$$

or: *b closest nodes to j only*

- Target assignment function $q_{ij}(\delta_{ij})$:

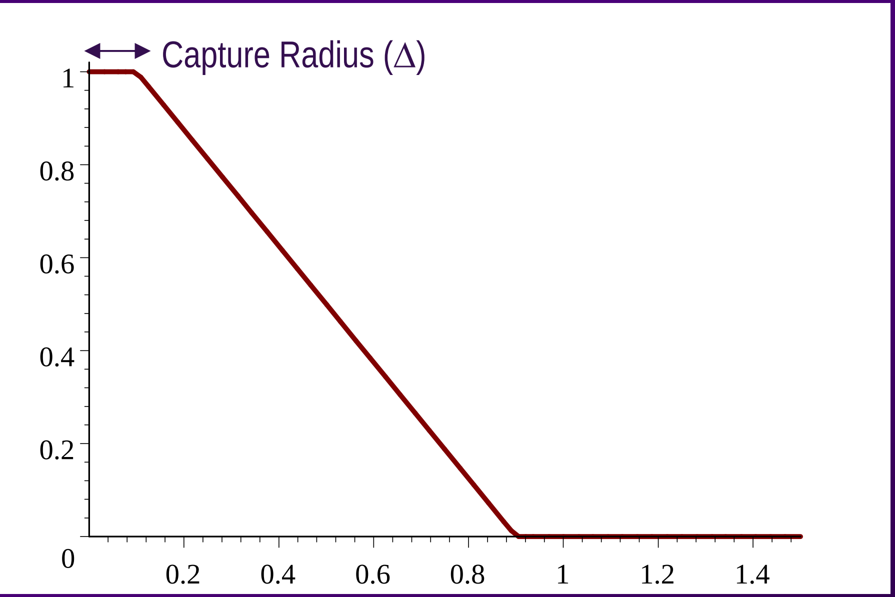
Monotonically non-increasing and s.t.

$$q_{ij}(0) = 1, \quad q_{ij}(1) = 0$$

THE FUNCTION q_{ij}

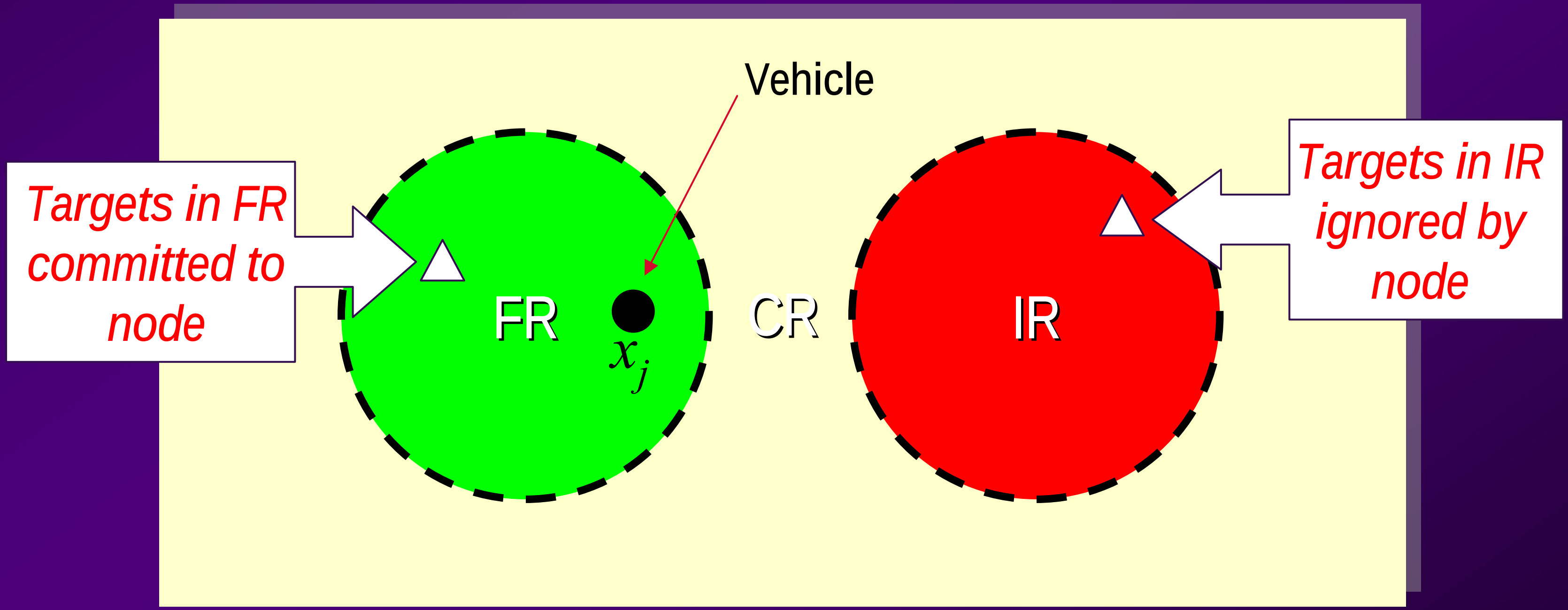
- A example of q_{ij} function **(M=2)**:

$$q_{ij}(\delta_{ij}) = \begin{cases} 1 & \text{if } \delta_{ij} \le \Delta \\ \dfrac{1}{1-2\Delta}\left[(1-\Delta)-\delta_{ij}\right] & \text{if } \Delta < \delta_{ij} \le 1-\Delta \\ 0 & \text{otherwise} \end{cases}$$



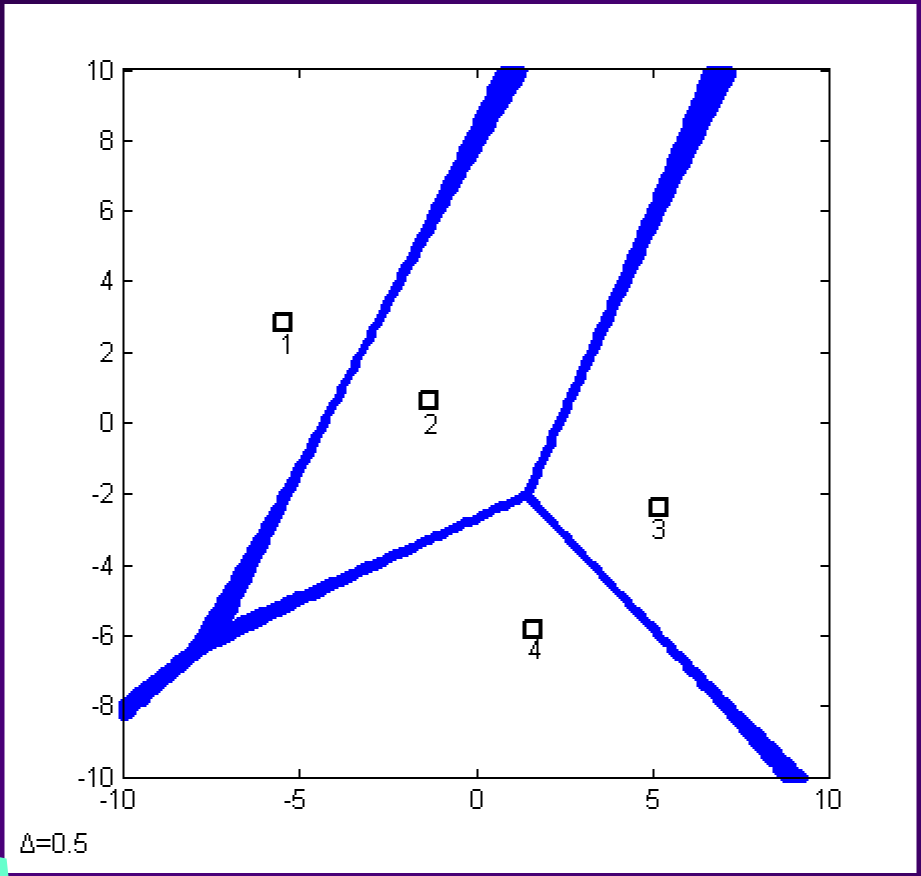
$q_{ij}(t)$ defines *DYNAMIC RESPONSIBILITY REGIONS* for vehicle j

- S_j – Full Responsibility Region (FR) $\delta_{ij} \leqslant \Delta$
- C_j – Cooperative Region (CR) $\Delta < \delta_{ij} \leqslant 1-\Delta$
- I_j – Invisibility Region (IR) $\delta_{ij} > 1-\Delta$



THE FUNCTION q_{ij}

CONTINUED



Voronoi partition

Partition of a plane with into n convex polygons such that each polygon contains exactly one point and every point in a given polygon is closer to its central point than to any other.

What happens as parameter Δ increases ?

2-VEHICLE CASE – DYNAMIC PARTITIONING

Possible Target Location

Vehicle Locations

II: Only vehicle 1 goes to target

III: Both vehicles go to target

IV: Only vehicle 2 goes to target (1 is repelled !)

PLANNING AND ACTION HORIZONS

PLANNING Horizon ***H(t)***:

$$H(t) = d_{\min}(t) \equiv \min_{i,j} d_{ij}(t)$$

ACTION Horizon ***h(t)***:

$$h(t) = \alpha_H + \beta_H H(t), \quad \alpha_H \geq 0, \quad 0 \leq \beta_H \leq 1$$

OR: Whenever next EVENT occurs

TARGET ASSIGNMENT

MAIN IDEA IN CRH APPROACH:

Replace complex *Discrete Stochastic Optimization* problem by a sequence of simpler *Continuous Optimization* problems

But how do we guarantee that vehicles ultimately head for the desired DISCRETE TARGET POINTS?

STABILITY ANALYSIS

- TARGETS: y_i
- UAVs: x_j

DEFINITION: Node trajectory $\mathbf{x}(t) = [x_1(t), \ldots, x_M(t)]$ generated by a controller is *stationary*, if there exists some $t_V < \infty$, such that $\|x_j(t_v) - y_i\| \leq s_i$ for some $i = 1, \ldots, N$, $j = 1, \ldots, M$.

Target Size

QUESTION:
Under what conditions is a CRH-generated trajectory stationary ?

STABILITY ANALYSIS

CONTINUED

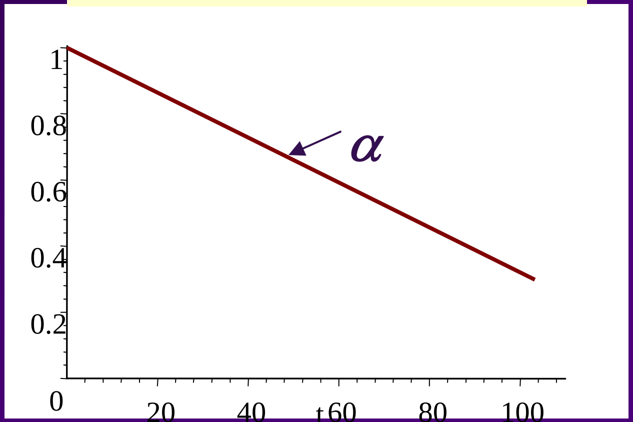
Recall objective function:

$$\max_{\mathbf{u}} \sum_{i=1}^{M}\sum_{j=1}^{N} R_i \phi_i(\tau_{ij}) p_{ij}(\tau_{ij}) q_{ij}(t_k + H_k) - \sum_{i=1}^{M}\sum_{j=1}^{N} C_j r_{ij}(t_k + H_k)$$

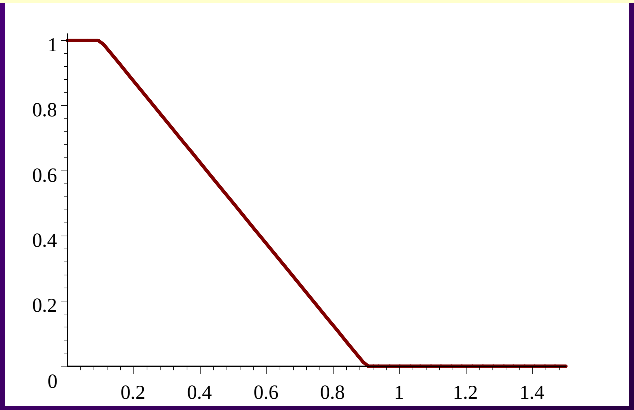
$$\phi_i(t) = 1 - \frac{\alpha}{T} t \quad t \in [0, T]$$

$$p_{ij} = 1$$

$$r_{ij} = 0$$



$$q_{ij}(\delta_{ij}) = \begin{cases} 1 & \text{if } j \in B_i, \delta_{ij} \le \Delta \\ \frac{1}{1-2\Delta}\left[(1-\Delta) - \delta_{ij}\right] & \text{if } j \in B_i, \Delta < \delta_{ij} \le 1-\Delta \\ 0 & \text{otherwise} \end{cases}$$



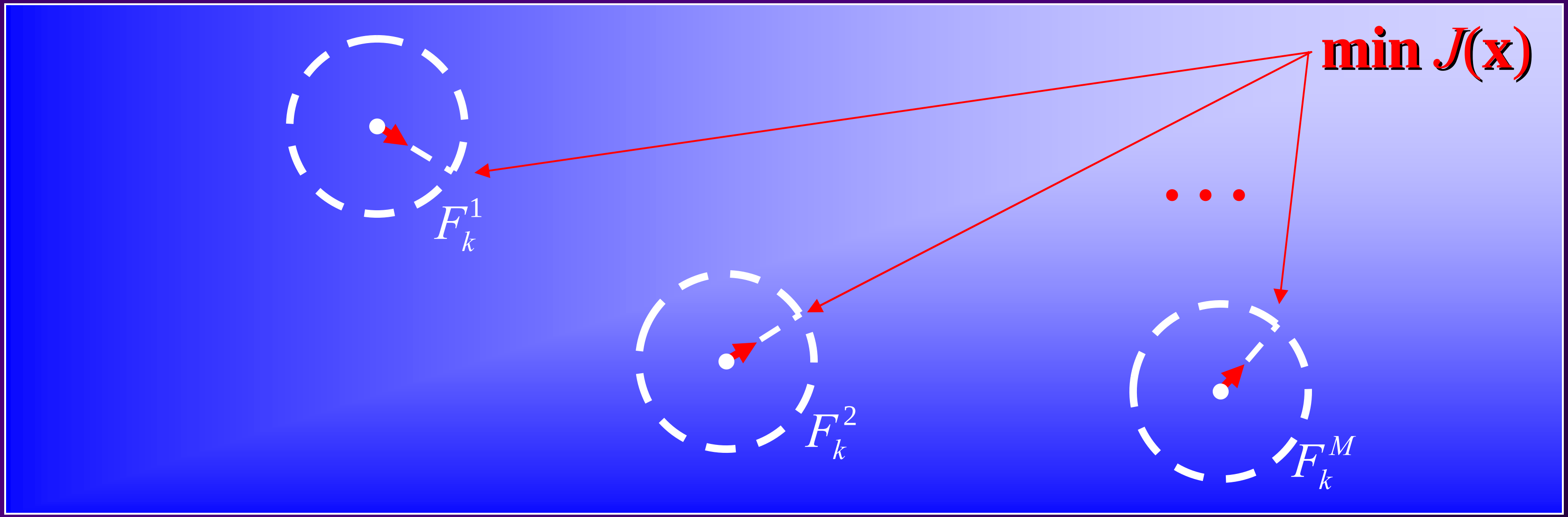
STABILITY ANALYSIS

CONTINUED

Objective function reduces to: $$J(\mathbf{x}) = \sum_{i=1}^{N}\sum_{j=1}^{M} R_i \left\| x_j - y_i \right\| q_{ij}$$

CRH controller solves optimization problem:
$$\begin{cases} \min\limits_{\mathbf{x}\in F_k} J(\mathbf{x}) \\ F_k = \left\{ \mathbf{w} : \left\| w_j - x_j(t_k) \right\| = VH_k \right\} \end{cases}$$

i.e., minimize the potential function $J(x)$ over a set of M circles:



MAIN STABILITY RESULT

Local minima of $J(x)$: $x^l = (x_1^l, \ldots, x_M^l) \in \mathbf{R}^{2M}, \quad l = 1, \ldots, L$

Vector of node positions
at *k*th iteration of CRH controller: $\mathbf{x}_k$

Theorem: Suppose $H_k = \min_{i,j} d_{ij}(t_k)$.
If, for all $l = 1,\ldots,L$, $x_j^l = y_i$ for some $i = 1,\ldots,N$, $j = 1,\ldots,M$,
then $J(\mathbf{x}_k) - J(\mathbf{x}_{k+1}) > b$ ($b > 0$ is a constant).

If all local minima coincide with targets, the CRH-generated trajectory is stationary

MAIN STABILITY RESULT

QUESTION:

When do all local minima coincide with target points?

1 Vehicle, *N* targets

If there exists a y_i s.t. $R_i - \left\| \sum_{j=1, j \neq i}^{N} R_j \frac{y_i - y_j}{\|y_i - y_j\|} \right\| > 0$

2 Vehicles, 1 target

2 Vehicles, 2 targets

TO RECAP…

- Limited look-ahead – control optimizes expectation over "planning horizon"
- Control updates – *event-driven* (*events are deterministic or random*) or *time-driven* (*for a given "action horizon"*)
- Target assignment – done implicitly, not explicitly: *No combinatorial problem involved*
- Assignment + Routing + Path Control – all done together

RH CONTROLLER FEATURES

CONTINUED

- Target values change – deadlines, target sequencing, return to base
- Node capabilities change - resource depletion, failures, damage
- Threat capabilities change - radar on/off, threat damage
- Target locations change - new targets, moving targets
- Obstacle avoidance - targets with negative values
- Randomness - new control actions in response to random events
- Constraints – heading change, heading-dependent costs, sensing tasks

Construct *GRADIENT FIELD* instead of artificial potential field

$$F_{ij} = \frac{\partial J_i}{\partial x_j} = c_i(x_j) \cdot f_i^0(x_j)$$

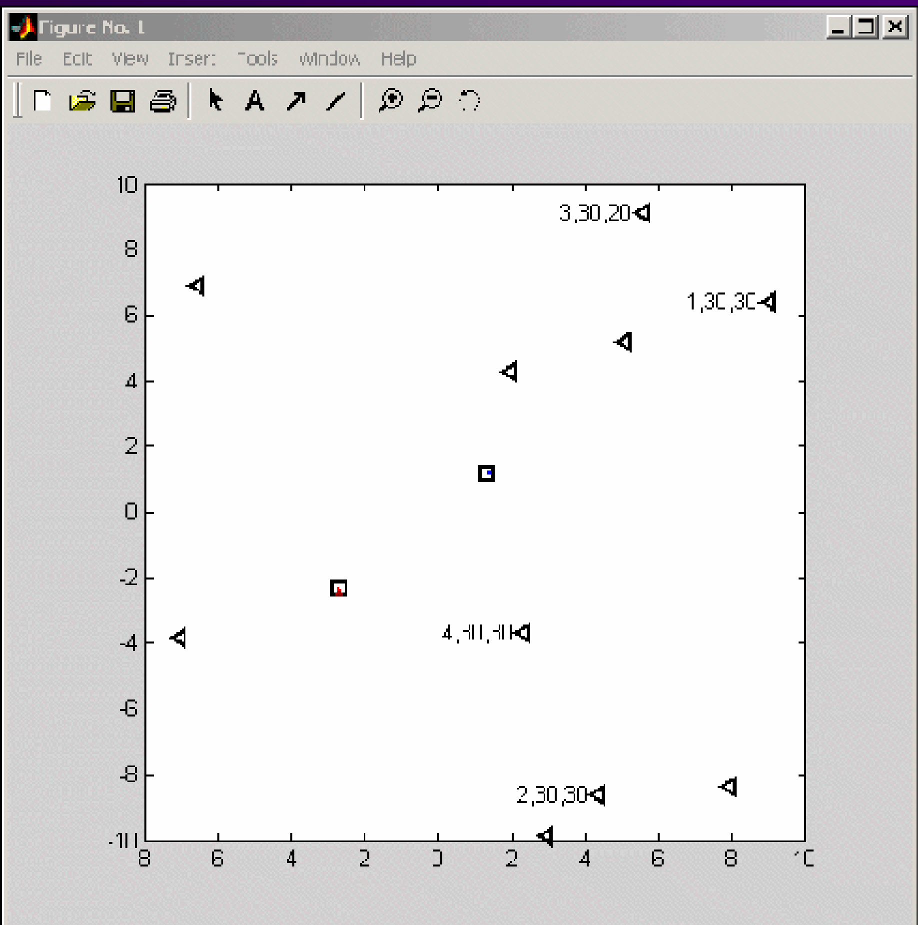
$$c_i(x_j) = \begin{cases} 1 & \text{if } y_i \in S_j \\ q_{ij} - \dfrac{(1-\delta_{ij})(2\delta_{ij}-1)}{(1-2\Delta)} & \text{if } y_i \in C_j \\ 0 & \text{if } y_i \in I_j \end{cases}$$

Cooperation coefficient

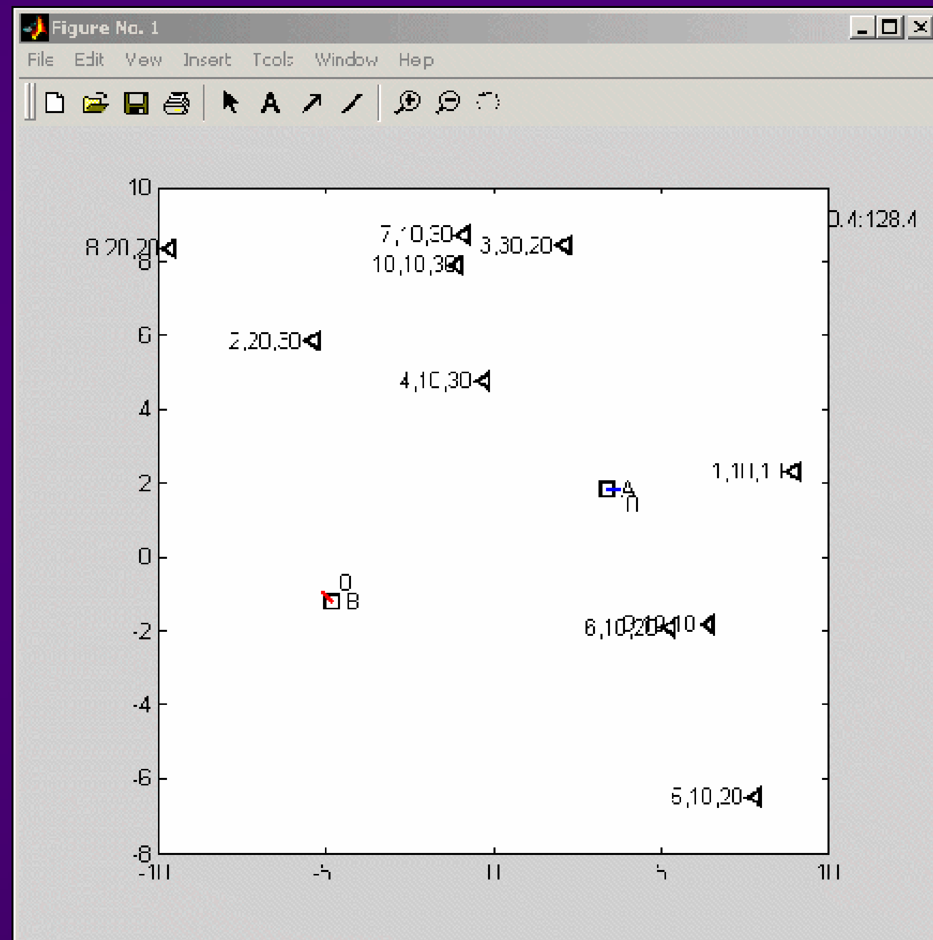
$$f_i^0(x_j) = \begin{cases} -\dfrac{R_i}{D_i} \dfrac{x_j - y_i}{\| x_j - y_i \|} & \text{if } x_j \neq y_i \\ 0 & \text{otherwise} \end{cases}$$

Force exerted by target i on node j given that it is the only node in the mission space

- 2 examples (*M*=2, *N*=10)



video01.avi



video02.avi

OTHER ISSUES

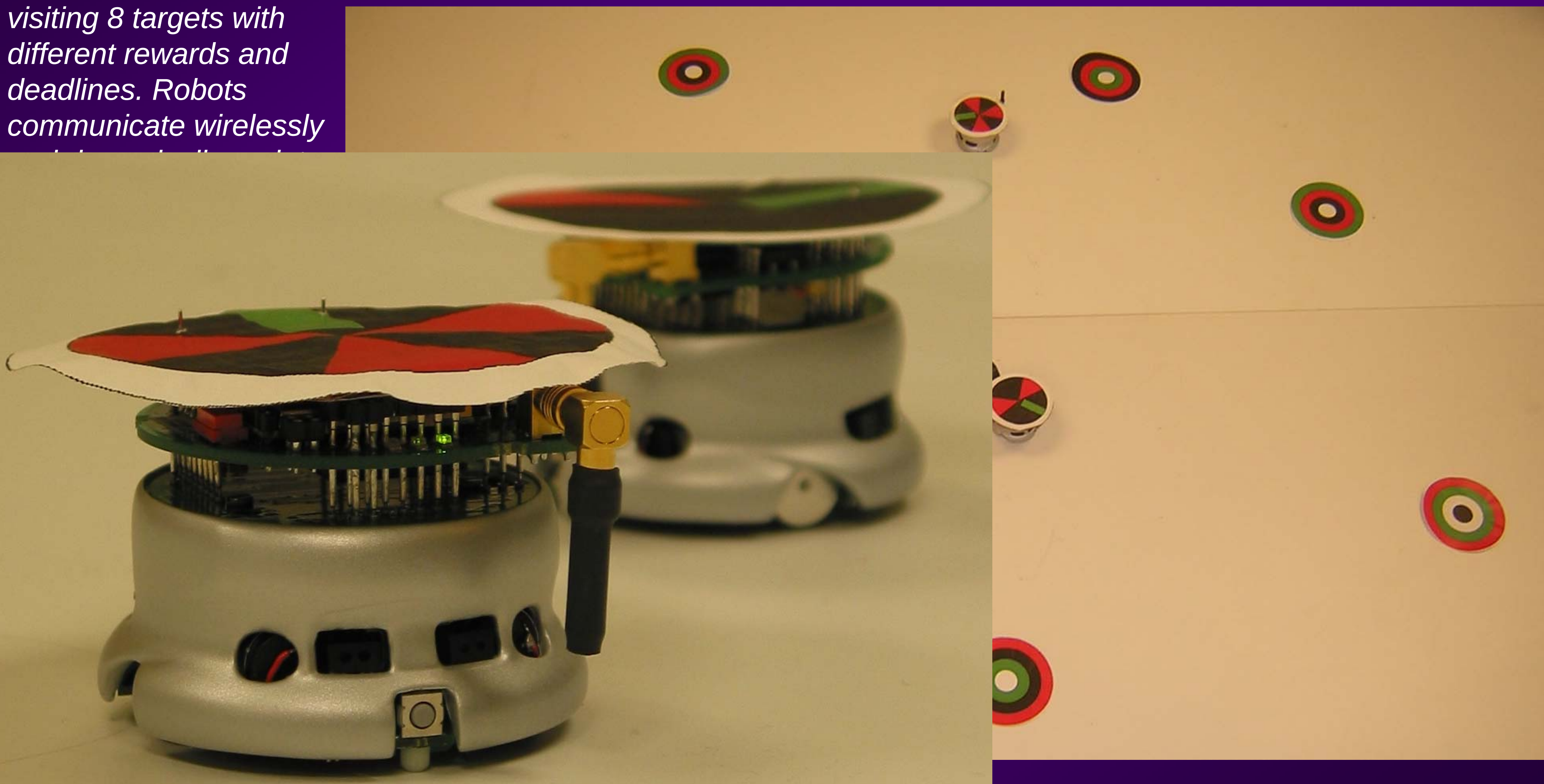
- Local optima in the CRH optimization problem
- Oscillatory vehicle behavior (*instabilities*)
- Additional path constraints,
  e.g., rendez-vous at targets
- Does CRH control generate optimal assignments?

REWARD MAXIMIZATION MISSION DEMO

MOVIES OF SUCH MISSIONS WITH SMALL ROBOTS:

3 Khepera robots executing mission: visiting 8 targets with different rewards and deadlines. Robots communicate wirelessly

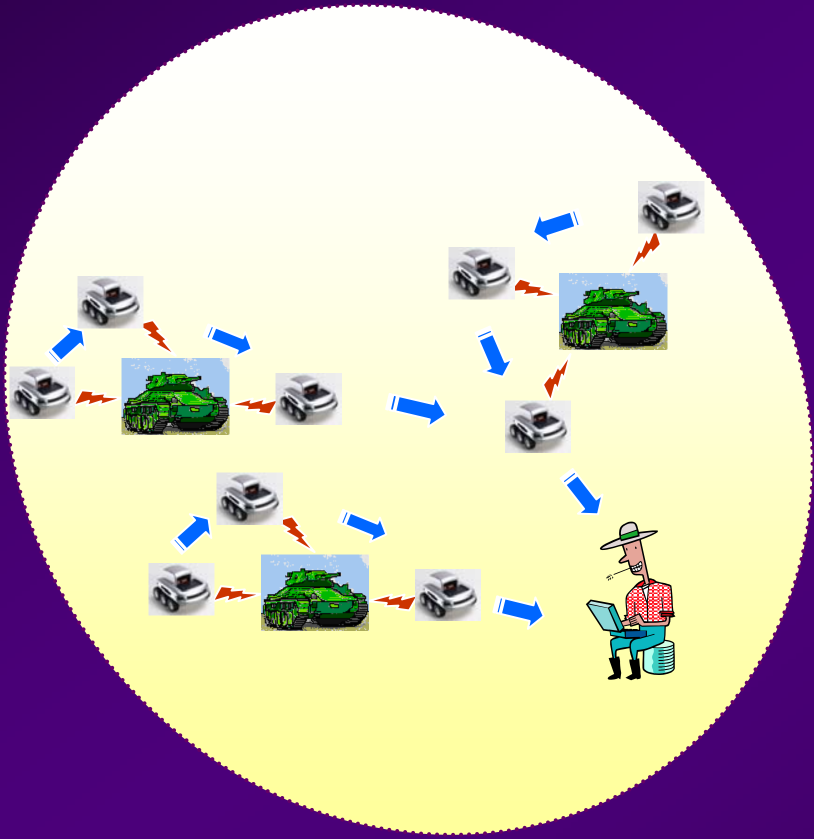
http://frontera.bu.edu/CoopCtrl.html

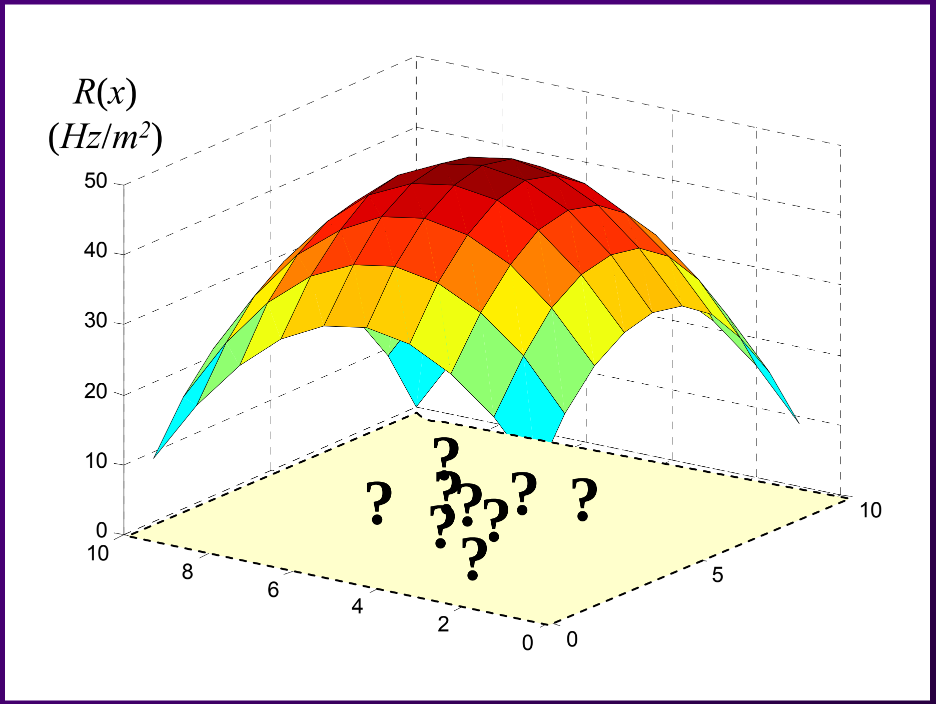


COVERAGE CONTROL MISSION

GOAL: Deploy mobile nodes to maximize data source detection probability

– unknown data sources
– data sources may be mobile





Perceived data source density over mission space

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:

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

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

- Sensing attenuation:

$p_i(x)$ is a decreasing function of $d_i(x) \equiv \|x - s_i\|$ (distance between x and s_i)

PROBLEM FORMULATION

- Joint detection prob. assuming sensor independence:

$$P(x) = 1 - \prod_{i=1}^{N} \left[1 - p_i(x)\right]$$

- OBJECTIVE:

  Determine locations $s_i \ (i = 1, \ldots, N)$
  to maximize total detection probability:

$$\max_{s_i \in \Omega} \int_{\Omega} R(x) \left\{ 1 - \prod_{i=1}^{N} \left[1 - p_i(x)\right] \right\} dx$$

Perceived data source density

DISTRIBUTED COOPERATIVE SCHEME

- Denote

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

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

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

DISTRIBUTED COOPERATIVE SCHEME *CONTINUED*

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

This has to be evaluated numerically.

Not doable for a mobile sensor with limited computation capacity.

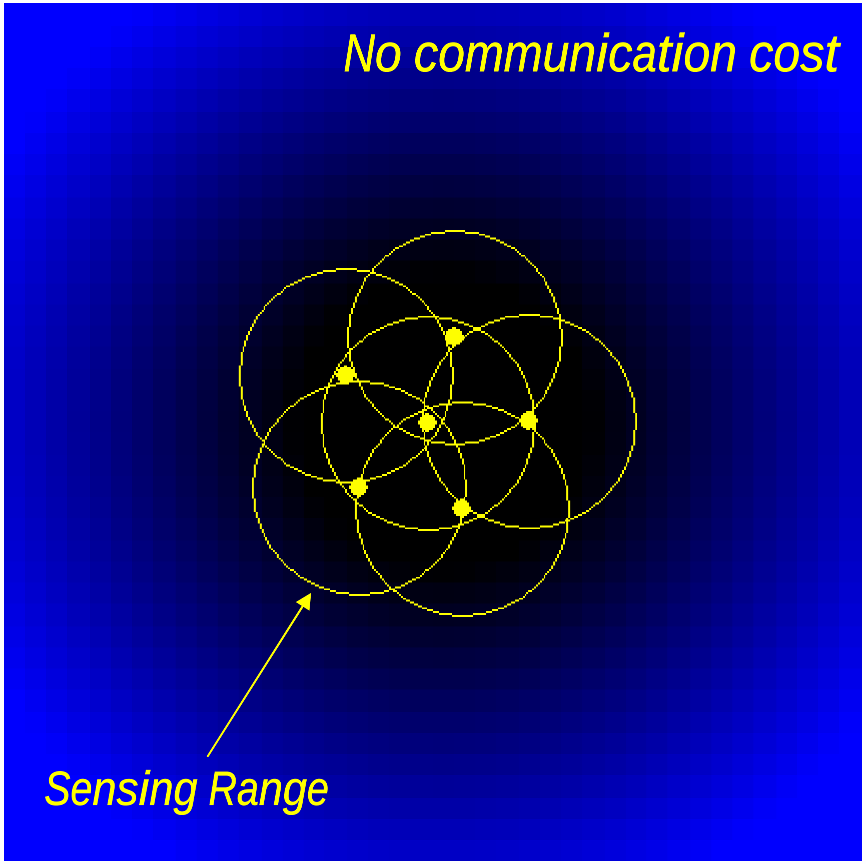
- Approximate $p_i(x)$ by truncating sensing attenuation
- Discretize $p_i(x)$ using a grid

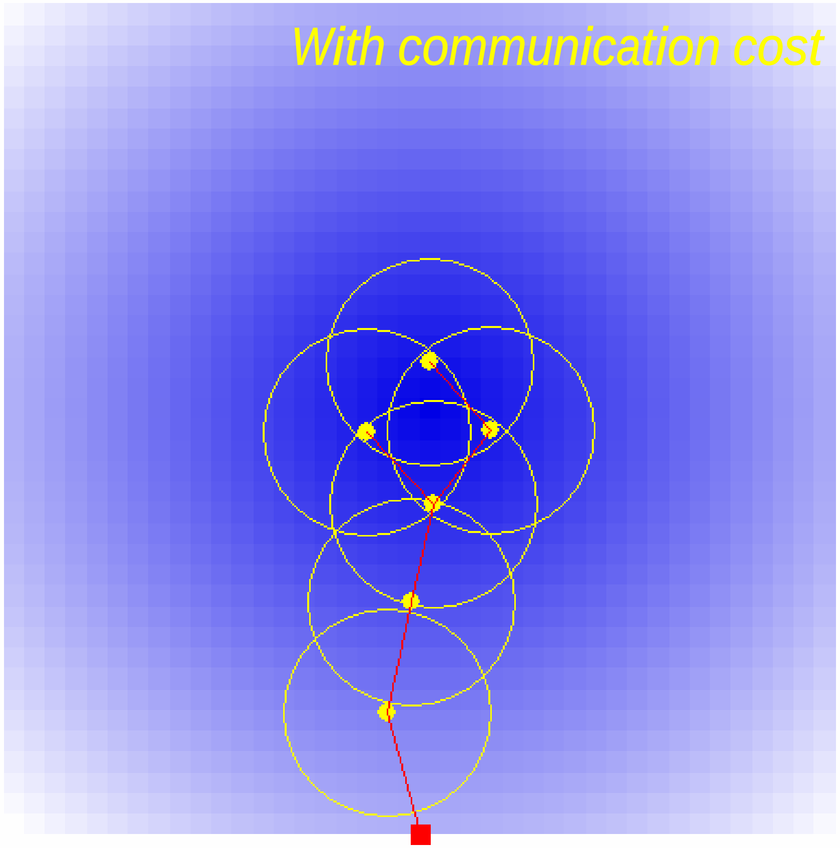
Details in *- Cassandras and Li, Euro. J. of Control, 2005*

COVERAGE CONTROL MISSION DEMO

SOFTWARE DEMO OF COVERAGE CONTROL ALGORITHM:

http://frontera.bu.edu/Applets/CoverageContr/index.html



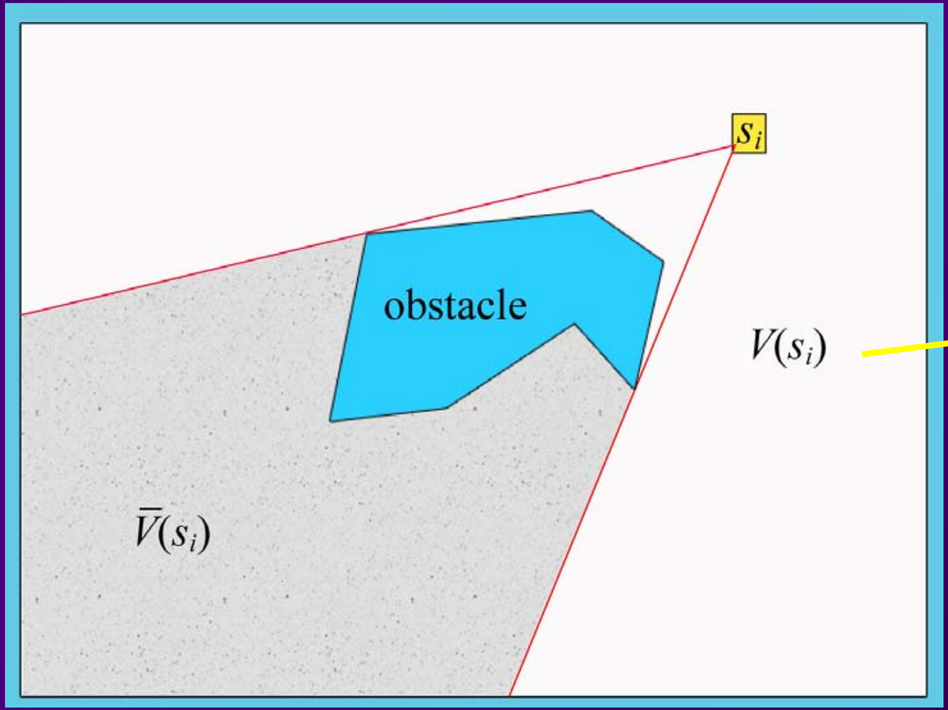


POLYGONAL OBSTACLES…

- Constrain the navigation of mobile nodes

- Interfere with the sensing

$$\hat{p}_i(x,s_i)=\begin{cases} p_i(x,s_i) & \text{if } x \text{ is visible from } s_i \\ 0 & \text{otherwise} \end{cases}$$



Visibility Region

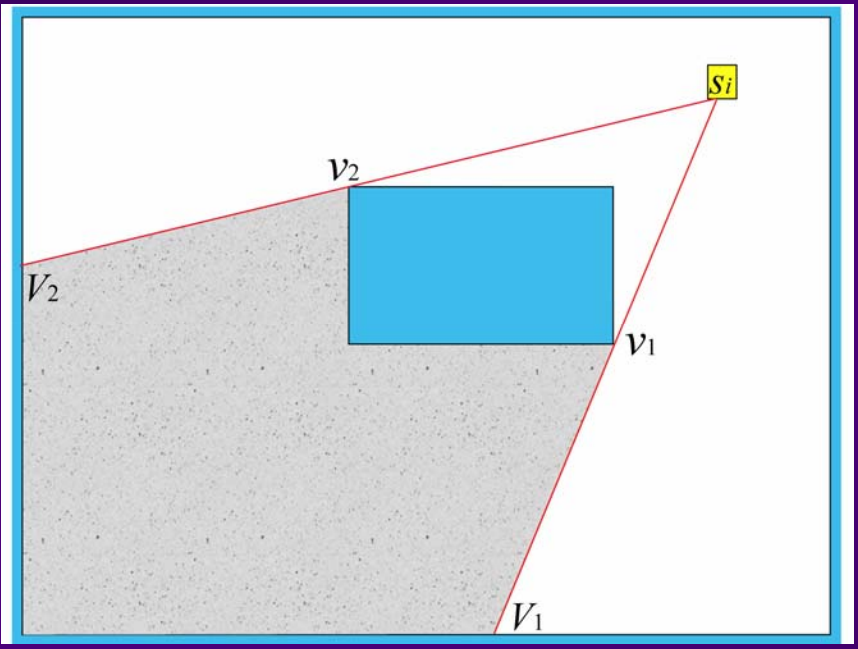
GRADIENT CALCULATION WITH OBSTACLES

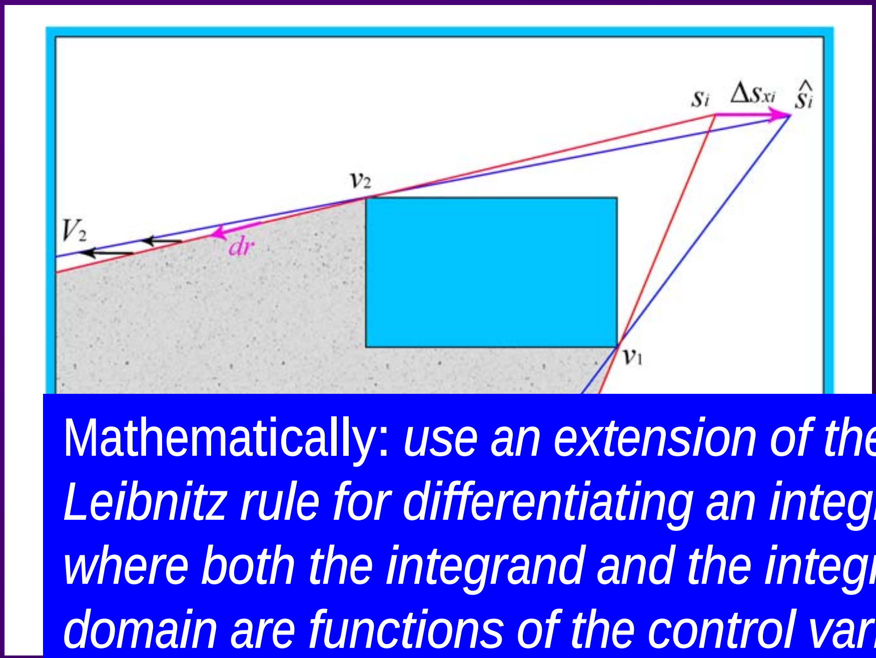
$$\frac{\partial H}{\partial s_i} = \int_{V(s_i)} R(x) \prod_{k=1, k\neq i}^{N} [1-\hat{p}_k(x,s_k)] \frac{\partial \hat{p}_i(x,s_i)}{\partial d_i(x)} \frac{s_i - x}{d_i(x)} dx + \sum_{j=1}^{Q(s_i)} A_j$$

$$\hat{p}_i = \begin{cases} p_i & \text{visible} \\ 0 & \text{invisible} \end{cases}$$

$Q(s_i)$: # of occluding corner points

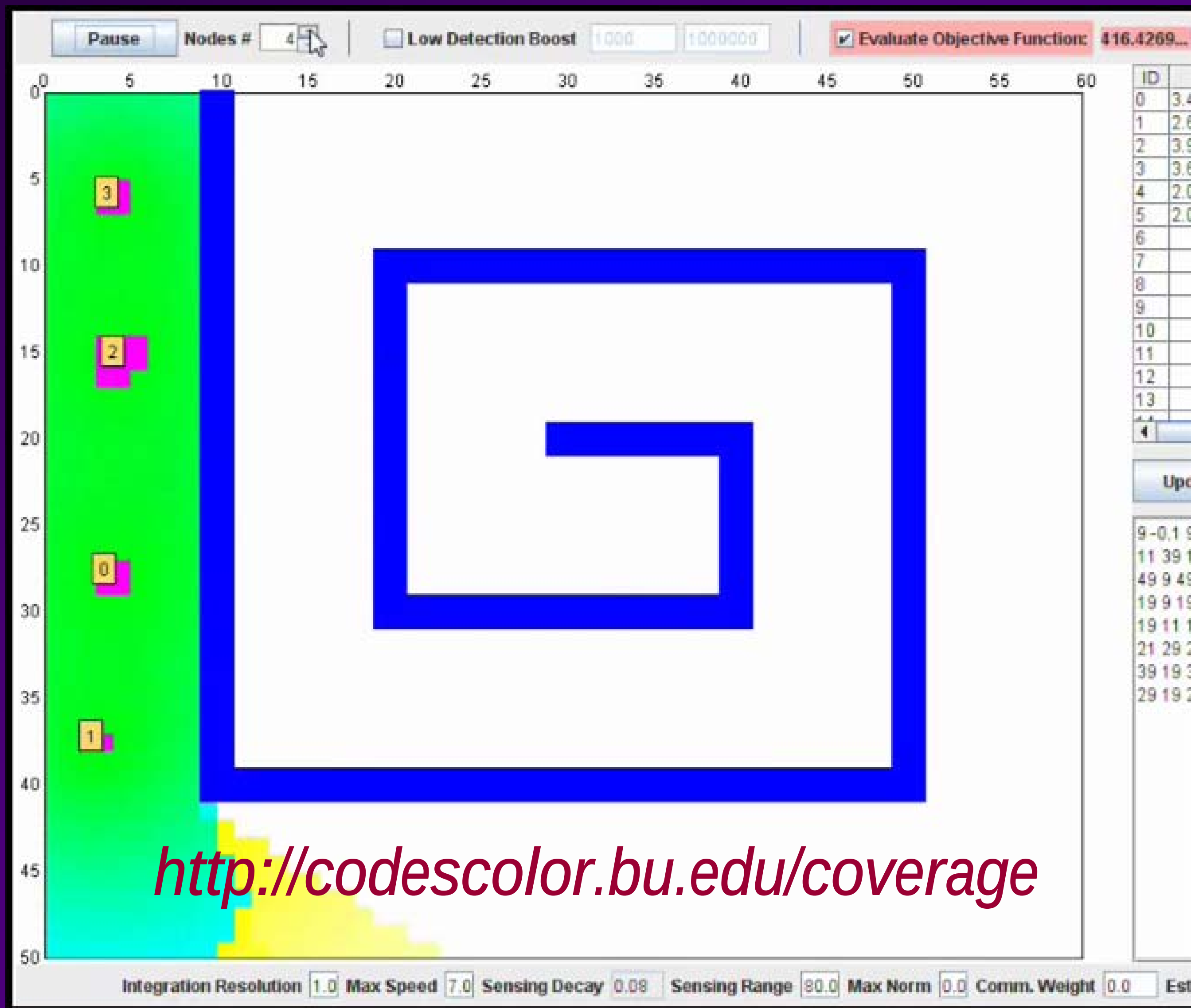
New term captures change in visibility region of s_i

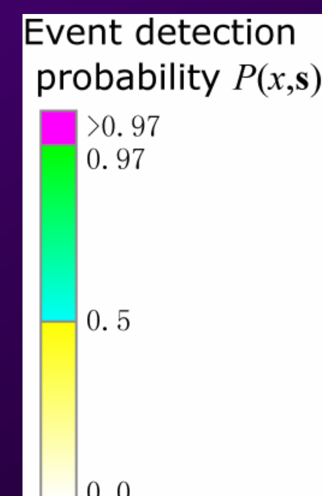




Mathematically: *use an extension of the Leibnitz rule for differentiating an integral where both the integrand and the integration domain are functions of the control variable*

DEPLOYMENT DEMO – WITH OBSTACLES





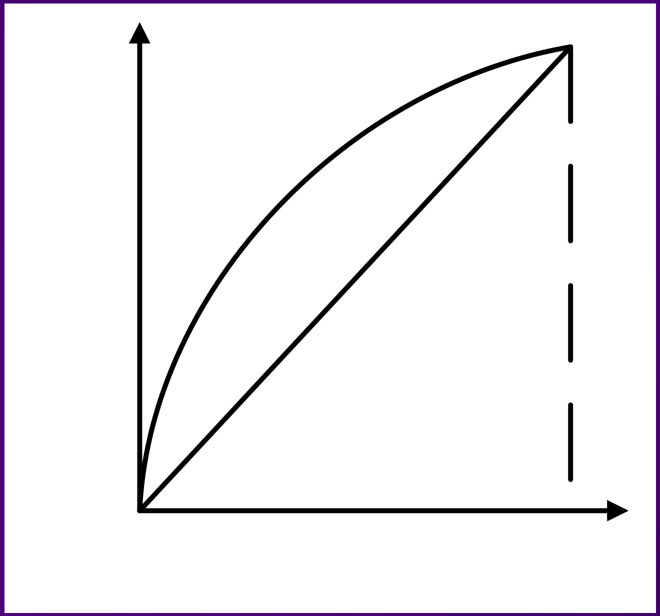
http://codescolor.bu.edu/coverage

A "FAIRNESS" ISSUE…

Some areas covered extremely well, while others not covered at all

SOLUTION: Assign higher reward to the same amount of marginal gain in $\boldsymbol{P(x,s)}$ in low coverage region

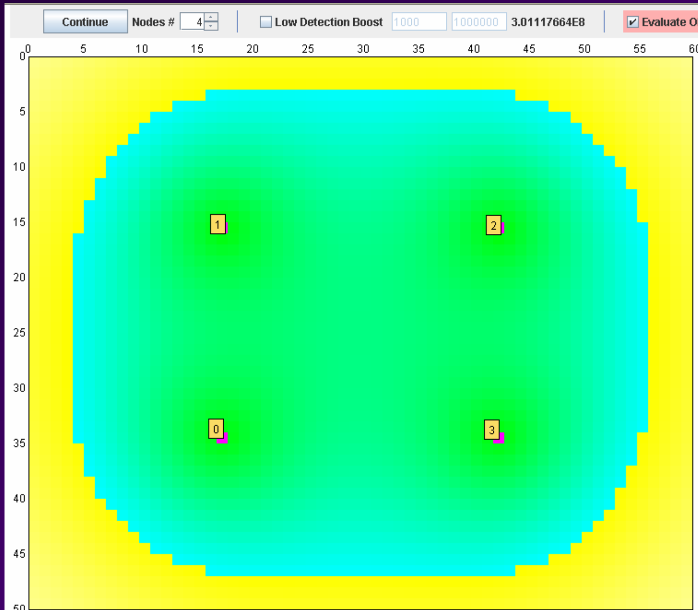
$$H(\mathbf{s}) = \int_F R(x)P(x,\mathbf{s})dx \quad \Longrightarrow \quad H_M(\mathbf{s}) = \int_F R(x)M\big(P(x,\mathbf{s})\big)dx$$

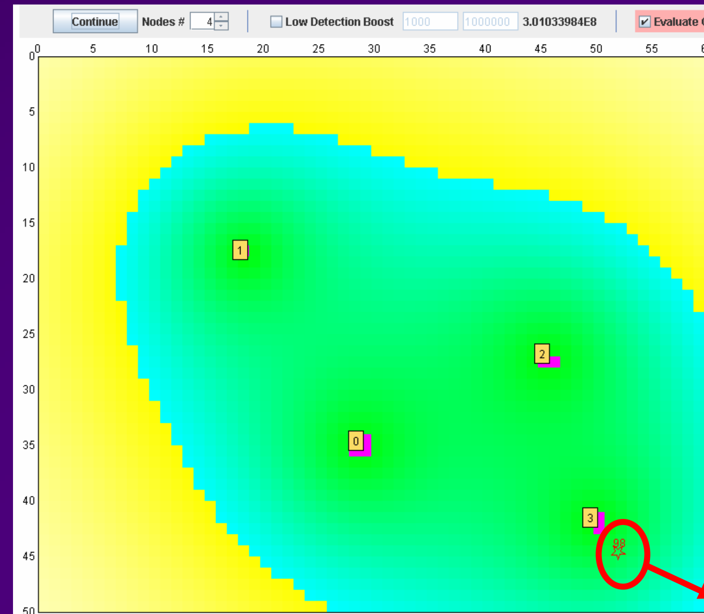


$M(\cdot) : [0,1] \to R$

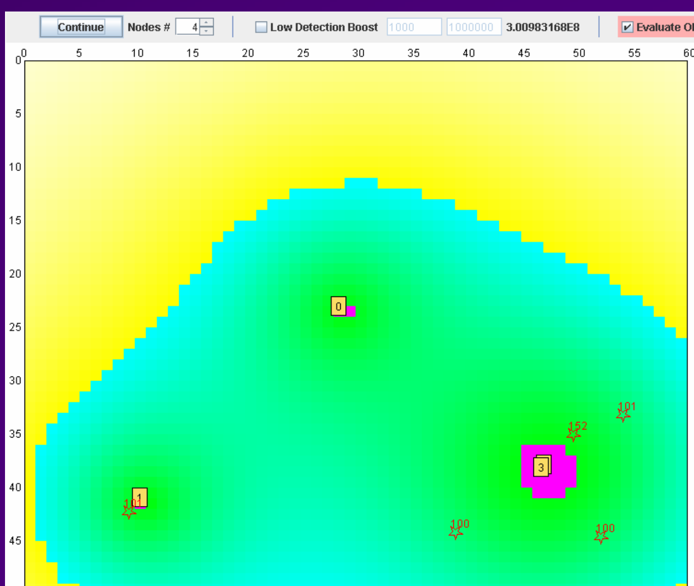
concave non-decreasing function

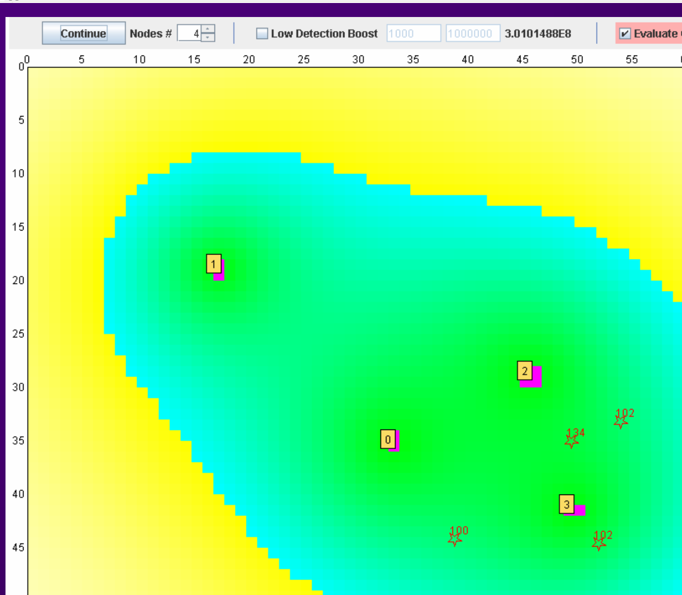
DEPLOYMENT DEMO – REACTION TO EVENTS





Event detected





ONGOING WORK:
SCALABLE, ASYNCHRONOUS, DISTRIBUTED OPTIMIZATION

- Small, cheap cooperating devices cannot handle complexity
  $\Rightarrow$ we need *DISTRIBUTED* control and optim. algorithms

- Cooperating agents operate asynchronously
  $\Rightarrow$ we need *ASYNCHRONOUS* control/optimization schemes

- Too much communication kills node energy sources
  $\Rightarrow$ communicate ONLY when necessary
  $\Rightarrow$ we need *ASYNCHRONOUS* control/optimization schemes

- Networks grow large, sensing tasks grow large
  $\Rightarrow$ we need *SCALABLE* control and optim. algorithms