

Problemi multi-agente

Esercitazione 1

Esempi di domini multi-agente

- ◇ Virtual USAR Robots
- ◇ DVMT
- ◇ Rescue
- ◇ Seacoast Surveillance

Virtual USAR Robots



Laser+exploration sw
Camera+search victim sw



Virtual USAR Robots II

- ◇ Arene simulano scenario post-disastro con crescente livello di difficoltà (scale, rampe, buche...).
- ◇ Robot con differenti funzionalità (definite dai differenti sensori, software a bordo) hanno differenti abilità: movimento, localizzazione, mapping, rilevamento vittime...
- ◇ Obiettivo: esplorare autonomamente tutta l'arena e indicare la posizione delle vittime.

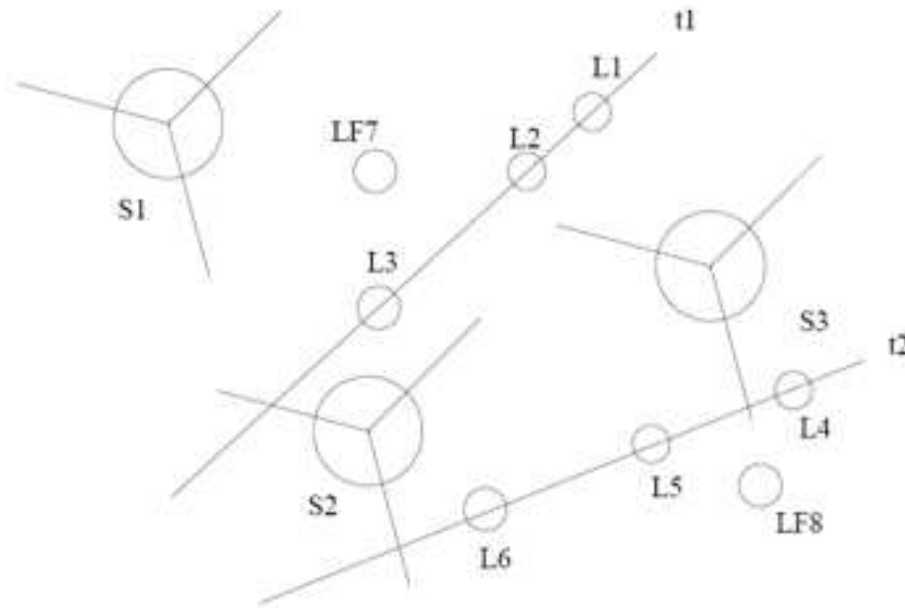
USAR: Problemi di coordinamento

- Ampliamento delle funzionalità dei robot grazie a collaborazione.
 - Esempio: Dopo un periodo di esplorazione, due robot entrano in comunicazione e fondono le mappe locali
 - Formare la squadra più adatta per esplorare e trovare le vittime, date le funzionalità dei singoli robot e le caratteristiche dell'arena.
- Esplorazione coordinata.
 - Esempio: Dato un obiettivo, coordinamento autonomo di una squadra di robot per l'assegnazione dell'obiettivo al robot più adatto
 - Dato un insieme di obiettivi, coordinamento autonomo di una squadra di robot per l'assegnazione degli obiettivi ai robot secondo le loro posizioni

USAR

- ◇ Possibile protocollo: Usare il contract net protocol per istanziare le richieste di tipo `goToPoint()` che arrivano dall'operatore.
- ◇ → Filmato coordinamento USAR

DVMT



DVMT - Descrizione agenti e dominio

- ◇ Un agente può controllare un sensore
- ◇ S_i può avere un solo settore acceso per timeslot
- ◇ $\langle S_{ij}, t_k \rangle$ il settore j del sensore i è attivo al tempo t_k
- ◇ per rilevare una traccia almeno due sensori devono leggere entro X time slot dalla prima lettura nella zona della traccia supposta
- ◇ sono possibili letture fantasma: il secondo sensore non rileva una lettura nella zona della traccia supposta

DVMT: Problema di coordinamento

- ◇ Obiettivo: **verificare** quante più tracce possibili
- ◇ Un agente non può realizzare da solo l'obiettivo. Gli agenti devono comunicare in modo che gli spostamenti dei settori accesi verifichino le tracce.
- ◇ Possibili strategie:
 - Acquaintance Network
 - Contract Net Protocol (CNP)

DVMT: Esempio protocollo di coordinamento

- ◇ Adattiamo il Contract Net Protocol (CNP) con molti Manager al nostro dominio
- ◇ Nella versione del DVMT dell'esercizio, le tracce sono verificate solo se vengono inquadrate contemporaneamente. L'agente che vede la traccia nella posizione $\langle x, y \rangle$ fa da Manager e chiede conferma facendo broadcast di una richiesta per $\langle x, y \rangle$, solo se essa si trova nel settore su cui può verificare più tracce. Ad ogni t_k accetta offerta a costo minore.
- ◇ Solo gli agenti interessati rispondono. Ogni agente che può rispondere all'offerta, offre come costo il primo slot di tempo libero sicuramente libero.
- ◇ Un agente non può rifiutare un compito dopo che il manager per quel compito lo ha decretato vincitore. (Questo evita un nuovo scambio di messaggi manager-vincitore.)

DVMT: Progetto protocollo di coordinamento

◇ execute()

for all endBidTime(o_k) **do**

$(a_i, t_j) = \text{selectWinner}(); \text{schedule}(s(o_k), t_j); \text{send_accept}(o_k, a_i);$

for all $a_l \neq a_i$ **do**

$\text{send_refusal}(o_k, a_l)$

end for

end for

for all msgReceived m **do**

if $m = \text{req}(o_k, a_i, t_{bid}, [t_{min}, t_{max}])$ **then**

$\text{send_propose}(o_k, a_i, \text{firstFree}([t_{min}, t_{max}]));$

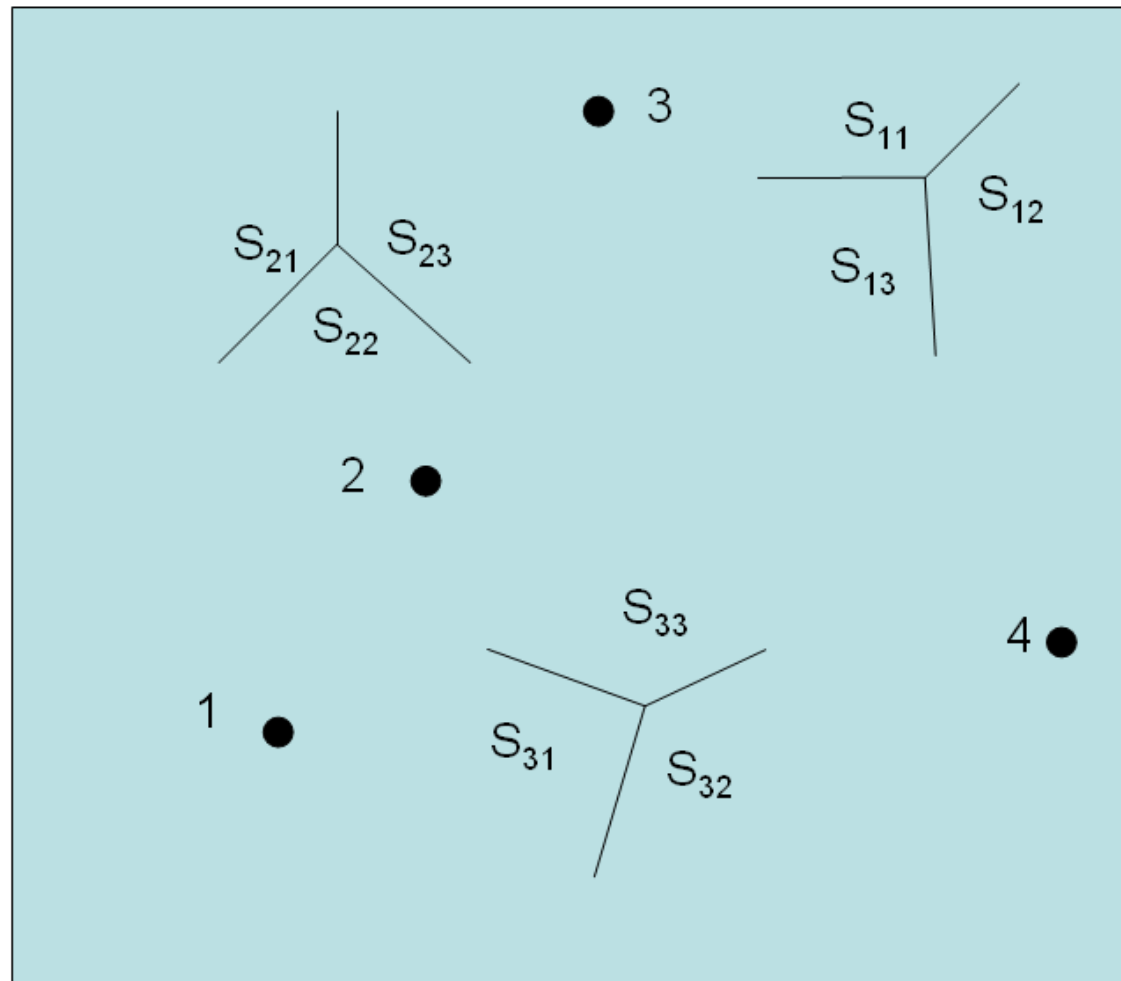
else if $m = \text{accept}(o_k, a_i, t_{task})$

$\text{schedule}(s(o_k), t_{task});$

end if

```
end for  
 $s_i = \text{chooseBestSector}();$   
for all  $o_k \in o(s_i)$  do  
     $t_m = \text{firstFree}(t_{cur} + 2, \infty);$   
     $\text{broadcast\_req}(o_k, t_{bid} = t_{cur} + 1, [t_m, t_m + 3))$   
end for  
if  $\text{scheduled}(t_{cur})$  then  
     $\text{turnOn}(\text{scheduled\_sector}(t_{cur}))$   
end if
```

DVMT: Situazione Iniziale



Esempio di esecuzione

$on(S_{12}), on(S_{21}), on(S_{31}), t = 1$

$A_1 broadcast_req(o_1, A_1, t_2, [t_3, t_6]); broadcast_req(o_2, A_1, t_2, [t_3, t_6]);$

$A_2 receives\ m = req(o_1, A_1, t_2, [t_3, t_6]) \rightarrow send_propose(o_1, A_1, t_3);$

$receives\ m = req(o_2, A_1, t_2, [t_3, t_6]) \rightarrow send_propose(o_2, A_1, t_3);$

$broadcast_req(o_3, A_2, t_2, [t_4, t_7]); broadcast_req(o_4, A_2, t_2, [t_4, t_7]);$

$A_3 receives\ m = req(o_1, A_1, t_2, [t_3, t_6]) \rightarrow send_propose(o_1, A_1, t_3);$

$receives\ m = req(o_2, A_1, t_2, [t_3, t_6]) \rightarrow send_propose(o_2, A_1, t_4);$

$receives\ m = req(o_3, A_2, t_2, [t_4, t_7]) \rightarrow send_propose(o_3, A_2, t_4);$

$receives\ m = req(o_4, A_2, t_2, [t_4, t_7]) \rightarrow send_propose(o_4, A_2, t_5);$

$t = 2$

$A_1 endBidTime(o_1) \rightarrow$

$send_accept(o_1, A_2, t_3); send_refusal(o_1, A_3); schedule(S_{13}, t_3)$

$endBidTime(o_2) \rightarrow$

$send_accept(o_2, A_2, t_3); send_refusal(o_2, A_3)$

receives $m = req(o_3, A_2, t_2, [t_4, t_7)) \rightarrow send_propose(o_3, A_2, t_4);$
receives $m = req(o_4, A_2, t_2, [t_4, t_7)) \rightarrow send_propose(o_4, A_2, t_5);$
 A_2 *endBidTime*(o_3) \rightarrow
 $send_accept(o_3, A_3, t_4); send_refusal(o_3, A_1); schedule(S_{23}, t_4)$
 $endBidTime(o_4) \rightarrow$
 $send_accept(o_4, A_3, t_5); send_refusal(o_4, A_1); schedule(S_{23}, t_5)$
receives $m = accept(o_1, A_2, t_3)$ and $m = accept(o_2, A_2, t_3) \rightarrow$
 $schedule(S_{22}, t_3)$
 A_3 *receives* $m = accept(o_3, A_3, t_4) \rightarrow schedule(S_{33}, t_4)$
receives $m = accept(o_4, A_3, t_5) \rightarrow schedule(S_{32}, t_5)$

A_1	A_2	A_3
S_{13}	S_{22}	
	S_{23}	S_{33}
	S_{23}	S_{32}

Analisi del protocollo

- ◇ la soluzione non sempre ottima.
- ◇ $num_{msg} \leq num_{task} \times (3 \times num_{ag}) = o(num_{task} \times num_{ag})$
- ◇ $T_{tot} = (bidtime + 2) \times T_{msg} + T_{execTask}$

Rescue



Rescue II

- ◇ Ambulanze portano civili negli ospedali
- ◇ Vigili del fuoco spengono incendi
- ◇ Agenti polizia liberano strade
- ◇ Incendi si propagano dopo un certo periodo di tempo t ai palazzi vicini
- ◇ Civili e strade bloccate sono percepiti ad una distanza fissa
- ◇ Incendi hanno una distanza di visibilità variabile a seconda della loro intensità

Rescue: tecniche di coordinamento possibili

- ◇ Auctions
- ◇ CNP, acquaintance
- ◇ Pianificazione Distribuita (centr-distr, distr-centr)
- ◇ Plan Merging

Progetto di un protocollo di coordinamento

- ◇ Specifica dominio con alcune ipotesi semplificative:
 - Solo Vigili del fuoco ed incendi
 - Minimizzare comunicazione, massimizzare velocità di intervento
 - Evitare conflitti tra agenti (numero controllate di vigili del fuoco su ogni fuoco)
 - Ambiente rappresentato come griglia regolare
 - Visibilità solo da celle adiacenti (8-vicini)

Caratterizzazione degli agenti

$$Cap(f_i, t_j) = \begin{cases} \frac{DMAX}{d(i,j)} & \text{se non bloccato} \\ 0 & \text{se bloccato} \end{cases}$$

- ◇ $d(i, j)$ distanza di Manhattan
- ◇ f_i sa in ogni istante se è bloccato
- ◇ $Risorse(f_i) =$ numero di task che può gestire contemporaneamente = NMAX
- ◇ f_i conosce tutti gli altri agenti direttamente

Descrizione protocollo e Pseudo codice

◇ Ciascun task associato ad un token, ciascun agente quando percepisce nuovo un fuoco crea un token e decide se tenerlo o passarlo alla squadra, decisione presa in base a conoscenza globale della squadra (threshold sulla capacità)

◇ PercFuoco(T)

```
if  $T \notin KTS$  then  
    BroadCast( $T$ );  
     $KTS = KTS \cup T$   
     $TkS = TkS \cup Tk(T)$   
end if
```

◇ RicevoAnnuncio(T)

```
if  $T \notin KTS$  then  
     $KTS = KTS \cup T$ ;  
end if
```

Pseudo Codice II

◇ GestioneToken()

Ordino(TkS);

$ExTkS = \{ \}$

for all $Tk \in TkS$ **do**

if $\|ExTkS\| < NMAX \wedge Cap(myself, Tk.Task) > Threshold$
then

$ExTkS = ExTkS \cup Tk$

else

PassoTkA(Tk, ProssAg);

$ProssAg = (ProssAg + 1) \% NumAgTot;$

end if

end for

$TkS = ExTkS$

ScelgoTkMigliore(ExTkS);

EseguoTkMigliore();

Situazione Iniziale

4	A3		F1	A2
3				
2	F2	A1		
1	F4	F3		

Esempio di esecuzione

$\forall i \in 1, 2, 3 \ TkS_i = \{ \} \ KTS_i = \{ \} \ NMAX = 2 \ Threshold = 0$

A_1

percepisce F_4, F_3, F_2

annuncia $F_4 \rightarrow KTS_i = \{F_4\} \ TkS_1 = \{F_4\}$

annuncia $F_3 \rightarrow KTS_i = \{F_4, F_3\} \ TkS_1 = \{F_4, F_3\}$

annuncia $F_2 \rightarrow KTS_i = \{F_4, F_3, F_2\} \ TkS_1 = \{F_4, F_3, F_2\}$

$Ordino(TkS_1) = \{F_4, F_3, F_2\}$

$\|ExTk_1\| < 2 \wedge Cap(A_1, F_4) > 0 \rightarrow ExTk_1 = \{F_4\}$

$\|ExTk_1\| < 2 \wedge Cap(A_1, F_3) > 0 \rightarrow ExTk_1 = \{F_4, F_3\}$

$\|ExTk_1\| = 2 \wedge Cap(A_1, F_2) > 0 \rightarrow$

$PassoTkA_{A_1}(F_2, A_2) \rightarrow TkS_2 = \{F_2\} \ ProssAg_{A_1} = A_3$

Esempio di esecuzione

A_2

percepisce $F_1 \rightarrow KTS_i = \{F_4, F_3, F_2, F_1\}$ $TkS_2 = \{F_2, F_1\}$

$Ordino(TkS_2) = \{F_1, F_2\}$

$\|ExTk_2\| < 2 \wedge Cap(A_2, F_1) > 0 \rightarrow ExTk_2 = \{F_1\}$

$\|ExTk_2\| < 2 \wedge Cap(A_2, F_2) > 0 \rightarrow ExTk_2 = \{F_1, F_2\}$

◇ Situazione finale $A_1 \rightarrow \{F_4, F_3\}$ $A_2 \rightarrow \{F_1, F_2\}$

Esempio di esecuzione II

$\forall i \in 1, 2, 3 \text{ } ExTk_i = \{ \} \text{ } TKS_i = \{ \} \text{ } NMAX = 2 \text{ } Threshold = 2$
 A_1

percepisce F_4, F_2, F_3

annuncia $F_4 \rightarrow KTS_i = \{F_4\} \text{ } TkS_1 = \{F_4\}$

annuncia $F_3 \rightarrow KTS_i = \{F_4, F_3\} \text{ } TkS_1 = \{F_4, F_3\}$

annuncia $F_2 \rightarrow KTS_i = \{F_4, F_3, F_2\} \text{ } TkS_1 = \{F_4, F_3, F_2\}$

$Ordino(TkS_1) = \{F_4, F_3, F_2\}$

$\|ExTk_1\| < 2 \wedge Cap(A_1, F_4) = 6 > 2 \rightarrow ExTk_1 = \{F_4\}$

$\|ExTk_1\| < 2 \wedge Cap(A_1, F_3) = 6 > 2 \rightarrow ExTk_1 = \{F_4, F_3\}$

$\|ExTk_1\| = 2 \wedge Cap(A_1, F_2) = 6 > 2 \rightarrow$

$PassoTkA_{A_1}(F_2, A_2) \rightarrow TkS_2 = \{F_2\} \text{ } ProssAg_{A_1} = A_3$

Esempio di esecuzione II

A_2

percepisce $F_1 \rightarrow KTS_i = \{F_4, F_3, F_2, F_1\}$ $TkS_2 = \{F_2, F_1\}$

$Ordino(TkS_2) = \{F_1, F_2\}$

$\|ExTk_2\| < 2 \wedge Cap(A_2, F_1) = 6 > 2 \rightarrow ExTk_2 = \{F_1\}$

$\|ExTk_2\| < 2 \wedge Cap(A_2, F_2) = 1.2 < 2 \rightarrow$

$PassoTkA_{A_2}(F_2, A_3) \rightarrow TkS_3 = \{F_2\}$ $ProssAg_{A_2} = A_1$

A_3

$Ordino(TkS_2) = \{F_2\}$

$\|ExTk_3\| < 2 \wedge Cap(A_3, F_2) = 3 > 2 \rightarrow ExTk_3 = \{F_2\}$

◇ Situazione finale $A_1 \rightarrow \{F_4, F_3\}$ $A_2 \rightarrow \{F_1\}$ $A_3 \rightarrow \{F_2\}$

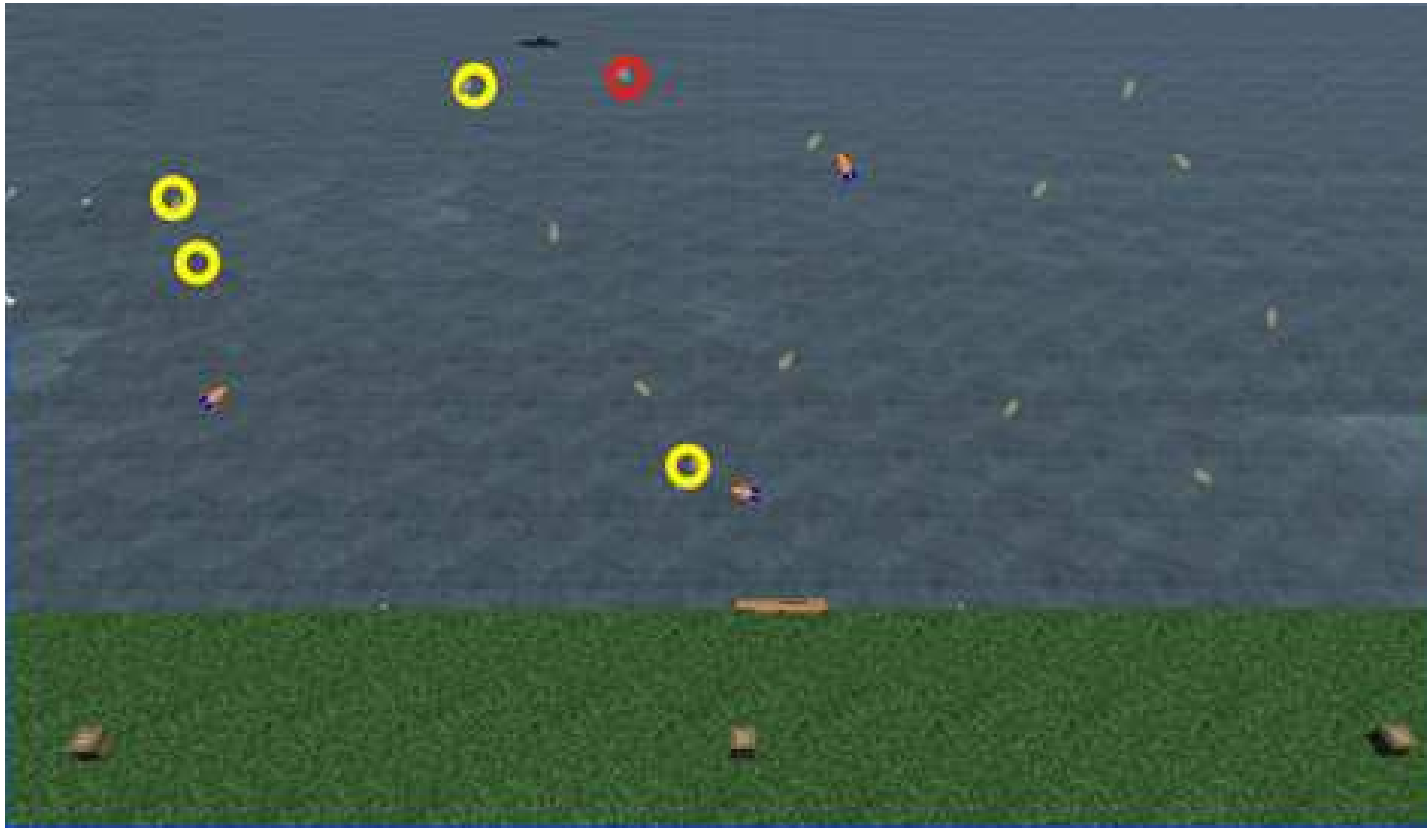
Analisi del protocollo

- ◇ fusione di CNP e acquaintance network
- ◇ a regime $T \times N + (T - (NMAX \times N)) \times nturni$
- ◇ performance dipendenti dal numero di risorse fissato per gli agenti
- ◇ Riallocazione di ciascun agente è locale → richiede solo un messaggio

Rescue: filmato

Filmato Token Passing

Seacoast Surveillance



Seacoast Surveillance II

◇ Gli agenti sono:

- Postazioni Radar: percepiscono tramite tracce radar, agiscono cambiando modalità di funzionamento del radar (punto di osserv., tipo di scansioni, filtri) e generando task da svolgere
- Pattugliatori: percepiscono tramite vista e sensori, agiscono cambiando posizione per svolgere task
- Operatori: percepiscono tramite monitor e collegamenti radio, possono decidere task da svolgere

◇ Obiettivo: Esecuzione attività di sorveglianza del tratto di costa attorno ad un porto.

Surveillance: Esempi di coordinamento possibili

- ◇ Plan deconfliction: rilevare/evitare situazioni in cui l'assegnazione dei task lascia scoperto un tratto di costa
- ◇ Task assignment: quale/i agente/i per quale/i attività
- ◇ Pianificazione Distribuita (centr-distr, distr-centr): formare un piano per un'attività che richiede una sequenza di azioni di uno o più agenti
- ◇ Formazione di squadre di intervento: eseguire attività che richiedono più di un agente contemporaneamente

Seacost surveillance: filmato

Simulazione coste