



UNIVERSITÀ DEGLI STUDI DI PADOVA

---

DIPARTIMENTO D'INGEGNERIA DELL'INFORMAZIONE

*Corso di Laurea in Ingegneria dell'Automazione*

**STRATEGIE DI PARTIZIONAMENTO DISTRIBUITE  
PER IL PATTUGLIAMENTO DI UN PERIMETRO CON  
UNA RETE DI VIDEOCAMERE**

*Laureando*

**Giuseppe Belgioioso**

*Relatore*

**Prof. Angelo Cenedese**

---

ANNO ACCADEMICO 2014/2015



Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two facilities, which we may call intuition and ingenuity

*Alan Turing*



## INDICE

1. <i>Introduzione</i> . . . . .	1
1.1 Stato dell'arte . . . . .	3
1.2 Contributo originale . . . . .	4
1.3 Struttura della tesi . . . . .	5
2. <i>Pattugliamento di un perimetro</i> . . . . .	7
2.1 Impostazione del problema . . . . .	7
2.2 Partizionamento di un segmento . . . . .	10
2.3 Partizionamento di un perimetro . . . . .	13
2.3.1 Caso banale . . . . .	15
2.3.2 Caso non banale . . . . .	16
2.4 Angolo d'Incidenza Visuale e funzione penalità . . . . .	17
3. <i>Ottimizzazione distribuita e protocolli di comunicazione</i> . . . . .	23
3.1 Pattugliamento di un perimetro, scenario distribuito . . . . .	23
3.2 Protocolli di comunicazione e aggiornamento . . . . .	25
3.2.1 Gossip Sincrono . . . . .	26
3.2.2 Gossip Simmetrico . . . . .	26
3.2.3 Gossip Asimmetrico . . . . .	28
4. <i>Algoritmi di Partizionamento</i> . . . . .	31
4.1 Algoritmo Perimetrico (AP) . . . . .	31
4.1.1 Metodo grafico . . . . .	33
4.1.2 AP nella pratica . . . . .	37
4.2 Algoritmo Perimetrico - Angolo d'Incidenza (AP $\Psi$ ) . . . . .	40
4.2.1 Criterio Centramento . . . . .	41
4.2.2 Criterio Area . . . . .	43
4.2.3 Metodo Grafico . . . . .	46
5. <i>Simulazioni in Matlab</i> . . . . .	49
5.1 Simulazioni di AP . . . . .	49
5.2 Simulazioni di AP $\Psi$ . . . . .	57
5.2.1 Algoritmo di Centramento AC . . . . .	57

5.2.2	Algoritmo Area <b>AA</b> . . . . .	58
5.2.3	Simulazioni di <b>AP<math>\Psi</math></b> . . . . .	61
5.3	Scenario Reale . . . . .	69
6.	<i>Conclusione</i> . . . . .	79
6.1	Sviluppi futuri . . . . .	81
	<i>Bibliography</i> . . . . .	84

## ABSTRACT

In questa tesi si studia il problema di partizionamento distribuito per il pattugliamento di un perimetro con una rete di videocamere di sorveglianza. L'obiettivo è dividere il perimetro in segmenti non sovrapposti, il pattugliamento di ciascuno di essi viene affidato ad una videocamera PTZ, in grado di spostare il campo visivo orizzontalmente. Ogni videocamera della rete è in grado di vedere solo una porzione del perimetro, detto *Patrolling Range*, possiede una velocità limitata e deve pattugliare il segmento che le viene assegnato, detto *Patrolling Area*, muovendo il proprio campo visivo avanti ed indietro su di esso alla massima velocità. Si propongono due strategie di partizionamento, chiamate **AP** e **AP $\Psi$** : l'unico obiettivo della prima è minimizzare la probabilità che vi sia un'intrusione non rilevata, formalmente minimizza il tempo massimo che intercorre tra l'ispezione di un punto del perimetro e la successiva; la seconda inoltre garantisce alle videocamere una buona visuale sulla propria zona di pattugliamento. Entrambi gli algoritmi sono iterativi e distribuiti e il loro funzionamento richiede che le comunicazioni tra gli agenti della rete avvengano solo a livello locale, gli schemi di comunicazioni considerati sono Gossip Simmetrico e Gossip Asimmetrico. Attraverso delle simulazioni software di **AP** e **AP $\Psi$** , si mostra che le loro prestazioni di convergenza dipendono fortemente sia dal numero di agenti della rete sia dal tipo di protocollo di comunicazione adottato.



## 1. INTRODUZIONE

Negli ultimi anni la continua crescita della domanda di sicurezza ha portato alla proliferazione di sistemi di videosorveglianza in ambienti eterogenei: industriali, commerciali, domestici, etc. Per convincersene basta osservare quante videocamere per la sorveglianza siano piazzate nell'ambiente che ci circonda e in cui ci spostiamo ogni giorno.

Un'applicazione chiave in questo campo è il pattugliamento di un perimetro, che consiste nel sorvegliare un intero perimetro, o una parte di esso, mediante una rete di videocamere, al fine di rilevare eventuali intrusioni o semplicemente monitorarne lo stato.

Per tale compito possono venire impiegate videocamere fisse che, una volta piazzate, inquadrano sempre la stessa zona mentre attualmente è sempre più comune l'uso di videocamere mobili, *videocamere PTZ* (Pan Tilt Zoom) nel gergo tecnico. Tali videocamere sono in grado di spostare il loro campo visivo lungo i tre assi, se le necessità lo richiedono: come ad esempio per seguire gli spostamenti di un intruso in un edificio sorvegliato (problema di *tracking*), oppure per monitorare il traffico lungo un attraversamento pedonale. Il motivo per cui le videocamere PTZ hanno rapidamente sostituito quelle fisse è molto semplice: la possibilità di spostare il campo visivo permette di monitorare un territorio anche molto esteso con un numero modesto di videocamere, sempre inferiore all'equivalente numero di videocamere fisse. Con l'avvento di grandi reti di videocamere è sorto anche il problema di controllarle. La soluzione attualmente più diffusa consiste in una "sala di controllo" dove vengono piazzati diversi monitor, generalmente uno per ogni agente della rete, sui quali si mostra quello che ogni videocamera vede in quel momento. Il rilevamento di un'intrusione, la chiamata d'allarme e il controllo del movimento della videocamera PTZ sono tutti incarichi affidati a una squadra di operatori umani. L'inefficienza di un simile scenario è evidente: risulta arduo per un operatore mantenere la concentrazione mentre controlla più monitor allo stesso momento e la possibilità di non rilevare un'intrusione a causa di una distrazione non va trascurata. Inoltre mentre un operatore è concentrato su un preciso incarico, come ad esempio comandare il movimento di una videocamera PTZ, gli è impossibile controllare tutti gli altri monitor che gli sono assegnati. Per risolvere questi problemi recentemente

sono stati introdotti sistemi di sorveglianza automatizzati, in grado di pattugliare il perimetro di una struttura, rilevare eventuali intrusioni e seguire gli spostamenti dell'intruso senza alcuna partecipazione umana. Alcuni di tali sistemi sono già impiegati per il controllo del traffico, ed effettuano la lettura automatica delle targhe dei veicoli, oppure nei sistemi di sicurezza più all'avanguardia.

La prima scelta che bisogna affrontare quando si sta progettando un sistema di sorveglianza con una rete di videocamere autonome è se utilizzare una soluzione centralizzata o distribuita. Entrambe le soluzioni hanno vantaggi e svantaggi: la prima è solitamente più facile da implementare, il controllo viene eseguito da un'unità di calcolo centrale che può comunicare con tutti gli agenti oltre che comandarli. L'onere computazionale di questa unità aumenta col numero di videocamere della rete, che però possono evitare elettronica di controllo incorporata al loro interno, risultando quindi più economiche. Il punto debole di tale approccio è la robustezza: se si verifica un guasto all'unità centrale di controllo il funzionamento dell'intero sistema è compromesso. L'approccio distribuito necessita di videocamere intelligenti, in altri termini, devono essere dotate di una certa potenza di calcolo, seppur minima nella pratica. Occorre perciò incorporare in ciascuna di esse dell'elettronica di controllo. L'aumento di prezzo che ne consegue è compensato dall'assenza di un'unità centrale di calcolo e da un sostanziale passo in avanti per quanto concerne la robustezza del sistema. I principali vantaggi dell'approccio distribuito sono i seguenti:

- le comunicazioni avvengono solo a livello locale, più precisamente solo le videocamere adiacenti comunicano, questo si traduce in una semplificazione dei collegamenti fisici della rete;
- il sistema che adotta l'approccio distribuito è robusto a eventuali guasti delle videocamere e si adatta a eventuali cambiamenti dello scenario;
- un algoritmo distribuito ben concepito non richiede modifiche se qualche nodo viene aggiunto o sottratto alla rete.

L'obiettivo di questa tesi è di progettare e analizzare alcuni algoritmi che permettono ad una rete di videocamere PTZ di gestire autonomamente il problema di pattugliamento ottimo. In pratica devono dividersi il perimetro da pattugliare autonomamente, in modo da minimizzare l'intervallo di tempo in cui ogni punto del perimetro rimane incontrollato, il tutto senza la presenza di un'unità centrale di calcolo e interagendo solo a livello locale.

---

## 1.1 *Stato dell'arte*

Il problema del pattugliamento è ampiamente discusso nella letteratura collegata al controllo distribuito. Il caso più comunemente trattato riguarda agenti autonomi che muovendosi su un'area predefinita cercano di determinare la regione ottima da pattugliare. Nel nostro caso invece gli agenti sono in posizioni fisse e il problema viene ridotto alla determinazione della sezione ottima del perimetro da pattugliare attraverso l'elaborazione di informazioni ricevute solo a livello locale. La sfida principale del problema di pattugliamento consiste nel trovare il modo ottimo per partizionare il perimetro. Negli articoli [1], [2], [3] e [4] vengono esaminati nel dettaglio alcuni algoritmi per il partizionamento ottimo con una rete di videocamere PTZ e particolare attenzione viene posta nei protocolli di comunicazione utilizzati.

In [1] viene introdotta l'idea che il partizionamento ottimo sia quello che minimizza l'intervallo di tempo tra due ispezioni successive di uno stesso punto del perimetro. Quest'idea viene poi formalizzata matematicamente definendo un funzionale e un problema di ottimizzazione che rappresenta il punto di partenza per tutti gli algoritmi in [1], [2], [3], [4] e anche in questa tesi. Inoltre, sempre in [1], viene proposto un algoritmo di partizionamento con protocollo di comunicazione sincrono e viene sviluppata un interessante soluzione al problema di tracking.

Negli articoli [2] e [3] vengono proposti degli algoritmi di partizionamento distribuiti che utilizzano protocolli di comunicazione di Gossip sincrono, simmetrico e asimmetrico. Per i primi due viene fornita una prova formale di convergenza, per l'ultimo invece la convergenza viene solo supportata attraverso diverse simulazioni.

In [4] il problema di partizionamento viene affrontato in maniera innovativa: a differenza di [1], [2] e [3] il perimetro viene suddiviso in zone non secondo la loro lunghezza bensì guardando l'ampiezza angolare dello spostamento che ogni videocamera attua per coprire la propria. L'idea vincente di quest'approccio, ripresa anche in questa tesi, è calcolare gli aggiornamenti delle aree di pattugliamento attraverso un metodo grafico: si elaborano, attraverso traslazioni e inversioni, le funzioni che mappano i punti del perimetro in spostamenti angolari delle videocamere e poi si cerca la loro intersezione. Quest'approccio risulta molto utile ed efficace poiché non è necessario conoscere l'espressione analitica delle funzioni citate.

## 1.2 Contributo originale

Una caratteristica che accomuna tutti gli algoritmi di partizionamento proposti in letteratura è assumere per alcuni agenti della rete dei vincoli speciali sulle zone di pattugliamento: per due videocamere adiacenti, identificate come prima e ultima, le zone di pattugliamento sono semifisse, in altri termini solo un estremo può variare liberamente, mentre l'altro è vincolato per entrambe a restare in un determinato punto del perimetro. Sotto quest'ipotesi è possibile fornire una descrizione matematica molto solida e vantaggiosa del problema di pattugliamento grazie alla quale si riesce a provare la convergenza degli algoritmi di partizionamento proposti.

Il tallone d'Achille di quest'approccio sta nel fatto che all'atto pratico si va a risolvere un problema di partizionamento su un segmento e non su un perimetro, con ripercussioni sia sulle performance del pattugliamento (secondo il criterio definito in [1]), sia sull'efficacia e la robustezza dell'algoritmo distribuito.

Un'altra caratteristica che accomuna gli algoritmi di partizionamento in letteratura è che il perimetro viene suddiviso in zone con l'unico fine minimizzare l'intervallo di tempo tra due visite successive di uno stesso punto e senza tener conto della qualità della visuale delle videocamere sulla zona a loro assegnata. Per ovvie ragioni, legate sia alla risoluzione che all'effetto di possibili ostruzioni, è preferibile che una videocamera pattugli una zona del perimetro vicina alla sua posizione fisica, sulla quale possiede una vista nitida e perpendicolare piuttosto che una zona lontana sulla quale ha una visuale obliqua.

In questa tesi viene definito formalmente, attraverso un problema di ottimizzazione, un nuovo approccio per il pattugliamento di un perimetro con una rete di videocamere PTZ, in cui:

- tutte le videocamere possono variare entrambi gli estremi della zona di pattugliamento (in altri termini senza assumere due videocamere con zone semifisse),
- si tiene conto della qualità della visuale delle videocamere nell'effettuare il partizionamento, pur mantenendo sempre come obiettivo principale l'ottimizzazione del pattugliamento in termini di minimizzazione della probabilità di non rilevare un'intrusione.

Basandosi su queste specifiche viene progettato un algoritmo distribuito che:

- funziona iterativamente,

- utilizza solo comunicazione locali, in particolare protocolli di comunicazione di tipo Gossip, il cui vantaggio più grande è non richiedere sincronizzazione tra le videocamere,
- converge sempre alla partizione ottima associata al nuovo problema di ottimizzazione definito.

### 1.3 *Struttura della tesi*

Nel primo capitolo della tesi si formalizza matematicamente il problema di partizionamento di un perimetro da parte di una rete di videocamere, successivamente si definiscono e analizzano due nuovi problemi di ottimizzazione: per uno scenario in cui tutte le videocamere possono variare entrambi gli estremi della zona di pattugliamento, e per uno che tiene conto della visuale delle videocamere, ottenuto aggiungendo alla funzione costo classica una funzione penalità appositamente progettata.

Nel secondo capitolo viene delineata l'impostazione del problema di partizionamento secondo uno scenario iterativo e distribuito, in particolare viene descritta la configurazione assunta al passo iniziale dell'algoritmo, ossia come le zone di pattugliamento vengono inizializzate. Sempre nel capitolo 2 viene descritto nel dettaglio il funzionamento dei protocolli di comunicazione e aggiornamento considerati: Gossip Sincrono, Simmetrico e Asimmetrico.

Nel capitolo 3 si presentano gli algoritmi di partizionamento progettati, descrivendo in profondità le regole per l'aggiornamento delle zone di pattugliamento. Grazie all'ausilio di numerose figure vengono illustrati i metodi grafici utilizzati dai vari algoritmi per individuare i punti del perimetro su cui aggiornare le zone di pattugliamento a seguito di una comunicazione tra due videocamere.

Infine nel capitolo 4 vengono presentati i risultati ottenuti dalla simulazione in Matlab dell'applicazione degli algoritmi su vari scenari. Questi risultati vengono poi interpretati ricavando utili informazioni riguardo il funzionamento, le performance e la convergenza degli algoritmi proposti.



## 2. PATTUGLIAMENTO DI UN PERIMETRO

In questo capitolo si descrive in termini più concreti il problema di partizionamento di un perimetro da parte di una rete di videocamere. Nella prima sezione viene presentata l'impostazione formale al problema, che è simile a quelle già proposte in letteratura e segue in particolar modo le notazioni adottate in [3]. Nella seconda sezione viene mostrato come i vincoli sulle zone di pattugliamento della prima e dell'ultima videocamera, assunti comunemente in letteratura, trasformino il problema di pattugliamento di un perimetro in un problema di pattugliamento di un segmento. I vantaggi nel considerare questo problema sono molteplici: è dimostrata l'esistenza di un'unica soluzione ottima e sono stati proposti diversi algoritmi che convergono ad essa. Sempre in questa sezione vengono richiamate alcune proposizioni da [1] e [3] che caratterizzano la soluzione al problema di partizionamento ottimo sul segmento e risulteranno utili nel seguito della tesi.

Nella sezione successiva si analizza il problema di partizionamento su un perimetro reale: ottenuto assicurando ad ogni videocamera della rete di poter variare entrambi gli estremi della propria area di pattugliamento. Si procede innanzitutto alla sua formalizzazione matematica, attraverso l'introduzione di un nuovo problema di ottimizzazione. Successivamente si analizza il problema introdotto, provando l'esistenza di infinite soluzioni secondo il criterio di ottimalità considerato finora. Nell'ultima sezione si traduce matematicamente il concetto di "qualità della visuale" di una videocamera, al fine di progettare una funzione penalità da introdurre nella funzione costo. Da quest'operazione nasce un nuovo problema di ottimizzazione, che, rispetto al precedente, ha il vantaggio di possedere un'unica soluzione ottima, corrispondente al partizionamento del perimetro che oltre a minimizzare l'intervallo di tempo tra due ispezioni successive di uno stesso punto, garantisce alle videocamere la visuale più perpendicolare possibile sulla loro zona.

### 2.1 *Impostazione del problema*

Si vuole pattugliare un perimetro 1-dimensionale di lunghezza finita con un numero finito di videocamere, ottimizzando il pattugliamento secondo un criterio che verrà descritto nel seguito della sezione. Il perimetro è una linea

spezzata chiusa di lunghezza  $L > 0$ , costituita da  $r$  segmenti consecutivi, in cui il segmento iniziale è consecutivo all' $r$ -esimo. Una volta scelto l'inizio del perimetro, ciascun punto viene identificato univocamente con la distanza, lungo il perimetro in senso antiorario, che lo separa dal punto iniziale, ossia il punto 0. Il perimetro è quindi rappresentabile come un anello  $\mathcal{L}$  di lunghezza  $L$ , dove il punto successivo a  $L$  è 0. Sia  $N$  la cardinalità dell'insieme delle videocamere, etichettate da 1 ad  $N$  ( $V_1, \dots, V_N$ ), si assume che le videocamere abbiano:

- (a) profondità di campo fissa (1-d.o.f.), perciò il loro campo visivo (f.o.v.) può cambiare solo se si cambia la panoramica;
- (b) raggio di copertura fisso, ciò significa che durante i movimenti panoramici il raggio di copertura non è alterato dalla prospettiva;
- (c) campo visivo puntiforme.

Inoltre si introducono le seguenti notazioni:

- $D_i = [\underline{d}_i \ \bar{d}_i] \subset \mathcal{L}$  è il **Patrolling Range** (PR), ossia l'area massima (sarebbe più corretto scrivere segmento) potenzialmente pattugliabile dalla videocamera  $i$ -esima, data la sua configurazione e la topologia dello scenario.  $\underline{d}_i$  e  $\bar{d}_i$  sono gli estremi sinistro e destro di quest'area e verranno anche chiamati vincoli fisici di  $V_i$ ;
- $\bar{v}_i \in \mathbb{R}^+$  è la velocità massima della videocamera  $i$ -esima durante le rotazioni sul piano orizzontale, quindi  $|v_i(t)| \leq \bar{v}_i$ ;
- $A_i = [\ell_i \ r_i] \subseteq D_i$  viene chiamata **Patrolling Area** (PA): è l'area attivamente pattugliata dalla videocamera  $i$ -esima, ovviamente deve essere un sottoinsieme dell'area potenzialmente pattugliabile  $D_i$ ;  $\ell_i$  e  $r_i$  sono rispettivamente l'estremo sinistro e destro di  $A_i$ ;
- $z_i(t) : \mathbb{R}^+ \rightarrow D_i$  è la funzione continua che mappa la posizione del f.o.v della videocamera  $i$ -esima in funzione del tempo.

Per definire in modo formale il problema di pattugliamento occorre introdurre una funzione obiettivo  $J$  e stabilire un criterio di ottimalità. La funzione costo che si prende in esame è quella proposta in [1], e ripresa anche in [2] e [3]. Si tratta di una funzione monotona di  $T_{lag}$  (time lag), definito come il massimo intervallo di tempo tra due visite consecutive dello stesso punto del perimetro. Entrando nel dettaglio, per ciascun istante  $t$  e per ciascuna posizione  $x \in \mathcal{L}$ ,  $J$  è 0 se  $x$  è attualmente vista da qualche videocamera ( $\exists i \ t.c. \ z_i(t) = x$ ), altrimenti ha un valore positivo che incrementa al

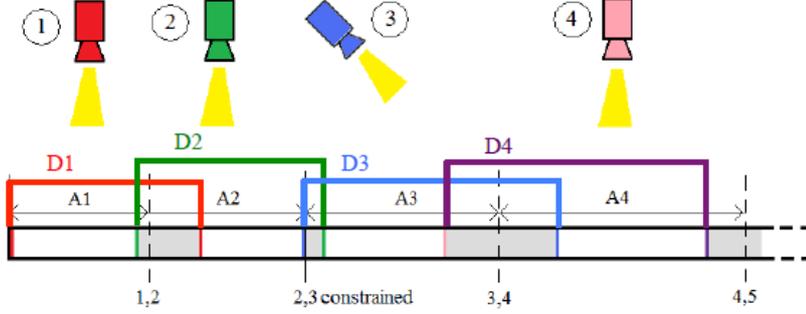


Fig. 2.1: Porzione di perimetro pattugliata da 4 videocamere PTZ, in evidenza le Patrolling Areas e i Patrolling Ranges.

passare del tempo dall'ultima visita di una videocamera in  $x$ . Il problema di minimizzazione per  $J$  corrisponde al calcolo del minimo  $T_{lag}$  vincolato alla dinamica del sistema.

$$\dot{z}_i(t) = v_i(t), \quad \begin{cases} |v_i(t)| \leq \bar{v}_i \\ z_i(t) \in D_i \end{cases} \quad (2.1)$$

La **Proposizione II.1** in [3] mostra che se  $A_i = [\ell_i \ r_i]$  è l'area di pattugliamento attiva della videocamera  $i$ -esima, il minimo  $T_{lag}$  per quest'area si ottiene assumendo che essa si muova alla sua massima velocità  $\bar{v}_i$ , avanti e indietro per  $A_i$ , con un movimento periodico di periodo  $T_i^*$ ,

$$T_i^* = T_{lag}^*(A_i) = \frac{2|A_i|}{\bar{v}_i} = \frac{2(r_i - \ell_i)}{\bar{v}_i}. \quad (2.2)$$

Ciò significa che una volta assegnata una PA  $A_i$  ad una videocamera, l'ottimo viene raggiunto muovendo la videocamera avanti e indietro per  $A_i$  alla massima velocità. La conseguenza più importante di questa proposizione è che il problema di minimizzare globalmente il  $T_{lag}$  si riduce al problema di progettare le PA  $\{A_i\}_{i=1}^N$ , ossia a un **problema di partizionamento**. Formalmente, il problema di ottimizzazione che si intende risolvere è lo stesso introdotto in [3], che viene riportato nel seguito:

$$\min_{A_1, \dots, A_N} \max_i \{T_{lag}^*(A_i)\} \quad (2.3)$$

$$(\mathcal{P}_1) : \quad t.c. \quad A_i \subseteq D_i \quad i = 1, \dots, N \quad (2.4)$$

$$\cup_1^N A_i = \mathcal{L} \quad (2.5)$$

Da (1.3) si vede che l'obiettivo è minimizzare il più grande  $T_{lag}$  tra quelli determinati dalle aree  $\{A_i\}_{i=1}^N$ , rispettando i vincoli fisici (1.4) e garantendo la copertura totale del perimetro (1.5). Nel resto della tesi la ricerca dell'ottimo, secondo il criterio introdotto, sarà sempre effettuata attraverso strategie di partizionamento delle PA  $\{A_i\}_{i=1}^N$ , inoltre si assumerà che le videocamere si muovano sempre alla loro velocità massima  $\bar{v}_i$ .

## 2.2 Partizionamento di un segmento

Il problema di pattugliamento del perimetro  $\mathcal{L}$  può essere rilassato, assumendo che i Patrolling Ranges  $D_i$  soddisfino le seguenti disequazioni, chiamate **vincoli di allacciamento**:

$$\underline{d}_i \leq \underline{d}_{i+1} \leq \bar{d}_i \leq \bar{d}_{i+1}, \quad i = 1, 2, \dots, N - 1 \quad (2.6)$$

e imponendo che

$$\underline{d}_1 = 0 \quad e \quad \bar{d}_N = L, \quad (2.7)$$

ossia che il limite inferiore del PR  $D_1$  di  $V_1$  sia il punto iniziale di  $\mathcal{L}$ , mentre il limite superiore del PR  $D_N$  di  $V_N$  corrisponda con il punto finale di  $\mathcal{L}$ . Vengono ora riportati alcuni risultati ottenuti per il problema di pattugliamento di un perimetro sotto le ipotesi (1.6) ed (1.7), che saranno utilizzati nel seguito della tesi.

- La **Proposizione II.3** in [3] dimostra che è possibile trovare una soluzione ottima del problema  $\mathcal{P}_1$  risolvendo il seguente problema di ottimizzazione:

$$\min_{\{r_i\}_{i=1}^{N-1}, \{\ell_i\}_{i=2}^N} \sum_{i=1}^N \frac{(r_i - \ell_i)^2}{\bar{v}_i} \quad (2.8)$$

$$(\mathcal{P}_3) : \quad t.c. \quad \underline{d}_i \leq \ell_i, r_i \leq \bar{d}_i \quad i = 1, \dots, N \quad (2.9)$$

$$r_i \geq \ell_{i+1} \quad i = 1, \dots, N - 1 \quad (2.10)$$

Il vantaggio di risolvere  $\mathcal{P}_3$  anziché  $\mathcal{P}_1$  è duplice: trattandosi di un problema di ottimizzazione convessa ha un'unica soluzione ottima, ossia un unico insieme di minimizzatori, inoltre, data la particolare forma della funzione costo il calcolo risulta parallelizzabile, aprendo la strada all'utilizzo di algoritmi distribuiti.

- La **Proposizione 1** in [1] caratterizza la soluzione ottima del problema di partizionamento in assenza di vincoli fisici per le videocamere ( $D_i =$

$\mathcal{L}$ ,  $\forall i$ ), fornendo un'espressione per le PA e per i relativi periodi di pattugliamento:

$$A_i = \bar{v}_i T_0 \quad (2.11)$$

$$\bar{T} = 2T_0 = \frac{2L}{\sum_{i=1}^N \bar{v}_i} \quad (2.12)$$

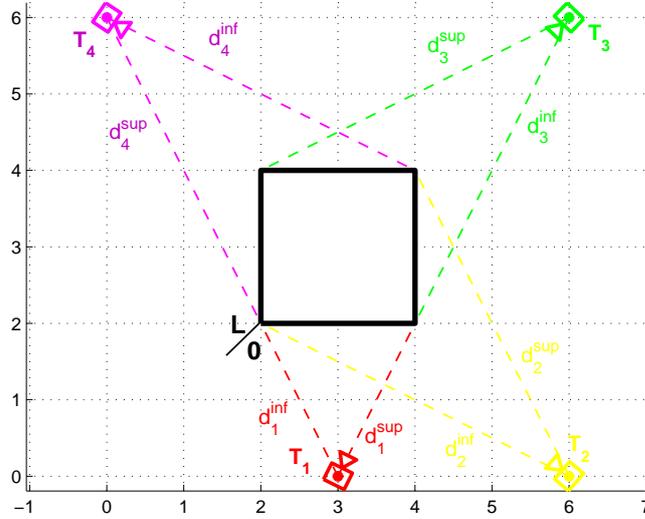


Fig. 2.2: Un esempio del problema di pattugliamento rilassato: il perimetro ha forma quadrata, le linee tratteggiate incidenti su di esso rappresentano i vincoli fisici  $\underline{d}_i$  ( $d_i^{inf}$  in figura) e  $\bar{d}_i$  ( $d_i^{sup}$  in figura) per ciascuna videocamera. Lo scenario soddisfa i vincoli di allacciamento, inoltre risulta  $\underline{d}_1 = 0$  e  $\bar{d}_N = L$ .

- La **Proposizione 2**, sempre in [1], dimostra che in generale la soluzione ottima del problema di partizionamento in presenza di vincoli fisici differisce da quella senza vincoli fisici. L'uguaglianza vale solo se la soluzione senza vincoli verifica:  $A_i \subseteq D_i$ ,  $\forall i$ . Nel caso in cui una videocamera violi i suoi vincoli fisici, ad esempio la videocamera 1 (senza perdita di generalità), ossia  $A_1 \not\subseteq D_1$ , la soluzione ottima si ottiene assegnando alla videocamera 1 la massima zona da essa pattugliabile ( $D_1$ ), e ricalcolando la soluzione ottima per le rimanenti N-1 videocamere su  $\mathcal{L} \setminus D_1$ :

- 1)  $A_1 = D_1$ ,

$$2) |A_i| = \bar{v}_i \cdot T_{0,c}, \quad i \neq 1$$

con

$$T_{0,c} = \frac{L - |D_1|}{\sum_{i=2}^N \bar{v}_i}, \quad T_{0,c} \geq T_0. \quad (2.13)$$

Il problema di partizionamento con (2.6) e (2.7) sebbene sia vantaggioso in fase risolutiva (unicità della soluzione e possibilità di calcolo distribuito e convergente), introduce anche una grande limitazione. Dal punto di vista pratico, imporre che  $V_1$  abbia come limite fisico inferiore 0 e che  $V_N$  abbia come limite fisico superiore  $L$ , significa spezzare l'anello del perimetro: impedisce a qualsiasi comunicazione tra  $V_1$  e  $V_N$  di essere produttiva, cioè in grado di aggiornare le rispettive PA:  $A_1$  ed  $A_N$ . Di fatto il problema fin'ora analizzato è un problema di partizionamento su un segmento: è sempre possibile "srotolare" il perimetro  $\mathcal{L}$  nel segmento  $[0 L]$  di lunghezza  $L$ , senza che nulla cambi all'atto pratico. Il fatto che in realtà il segmento rappresenti un perimetro è solo una nostra convenzione. Tale situazione è indesiderabile per due principali motivi:

- **Limita i vantaggi derivanti dall'utilizzo di un algoritmo distribuito.** Uno dei punti di forza dell'utilizzo di un algoritmo distribuito per la gestione del partizionamento è quello di poter sopperire al guasto di una delle videocamere, affidando la zona da essa pattugliata alle due contigue. Nel caso la videocamera in questione sia  $V_1$  o  $V_N$ , la sua zona dovrebbe essere affidata ad una sola videocamera (rispettivamente a  $V_2$  o a  $V_{N-1}$ ), anziché a due, in maniera molto inefficiente, poiché porterebbe un aumento drastico del  $T_{lag}^*$  (e perciò della funzione costo  $J$ ) che invece sarebbe evitabile se la zona scoperta venisse divisa equamente dalle due videocamere contigue.
- **La soluzione ottima sul segmento è in generale diversa da quella sul perimetro.** La scelta del punto iniziale del perimetro determina univocamente il problema di partizionamento di tipo  $\mathcal{P}_3$  e quindi la soluzione ottima. Si tratta però della soluzione ottima sul segmento  $[0 L]$  e non sul perimetro  $\mathcal{L}$ . Nulla vieta che scegliendo un punto iniziale diverso si arrivi ad una soluzione migliore: con  $T_{lag}$  minore e quindi  $J$  minore. Sarebbe quindi più conveniente che anche il punto iniziale, inteso come limite fisico sinistro e destro rispettivamente di  $V_1$  e  $V_N$ , fosse una variabile del problema di ottimizzazione.

### 2.3 Partizionamento di un perimetro

Per risolvere il problema di partizionamento su un perimetro reale bisogna evitare di fissare a priori un limite per  $V_1$  e  $V_N$ , rinunciando alle condizioni (2.6), e permettere una zona comune tra i PR  $D_1$  e  $D_N$ . Per questo motivo si assumerà d'ora in avanti che il PR  $D_1$  sia parte a destra e parte a sinistra del punto 0 di  $\mathcal{L}$ :

$$D_1 = [0 x] \cup [y L], \quad x, y \in \mathcal{L}, \quad y > x \quad (2.14)$$

Sotto quest'ipotesi si ha che  $\underline{d}_1 > \bar{d}_1$ , ossia per  $V_1$  il valore del suo limite fisico sinistro è maggiore di quello del suo limite fisico destro. Per semplificare l'analisi del problema si ipotizzerà che solo  $V_1$  abbia questa caratteristica, cioè sia l'unica con PR a cavallo dello zero.

In conseguenza a (2.14) i vincoli di allacciamento (2.6) non sono mai verificati per  $i = 1$ , in quanto ad esempio  $\underline{d}_1 > \bar{d}_1$ . Di seguito si scrive l'equivalente dei vincoli di allacciamento (2.6) nel caso  $D_1$  sia del tipo (2.14):

$$\underline{d}_i \leq \underline{d}_{i+1} \leq \bar{d}_i \leq \bar{d}_{i+1}, \quad i = 2, \dots, N - 1 \quad (2.15)$$

$$0 \leq \underline{d}_2 \leq \bar{d}_1 \leq \bar{d}_2 \quad (2.16)$$

$$\underline{d}_N \leq \underline{d}_1 \leq \bar{d}_N \leq L \quad (2.17)$$

Poiché  $\mathcal{P}_3$  non è più valido sotto queste ipotesi, emerge la necessità di definire un nuovo problema di ottimizzazione  $\mathcal{P}'_3$  valido per lo scenario descritto.

Si consideri il segmento  $S = [\underline{d}_1 \bar{d}_N] \subset \mathcal{L}$ , ossia il pezzo di perimetro che va dal limite fisico sinistro  $\underline{d}_1$  di  $V_1$  fino al limite fisico destro  $\bar{d}_N$  di  $V_N$  (si veda ad esempio Fig. 2.3). In linea teorica per giungere alla partizione ottima si potrebbe assumere sequenzialmente come punto iniziale di  $\mathcal{L}$  ogni punto di  $S$  e risolvere per ciascuno di essi un problema di tipo  $\mathcal{P}_3$ . La soluzione con  $T_{lag}$  più piccolo tra quelle calcolate è la soluzione ottima del problema originale. Si intuisce che il procedimento è applicabile empiricamente solo nel caso si introduca una quantizzazione di  $S$ , altrimenti i problemi  $\mathcal{P}_3$  da risolvere sarebbero infiniti.

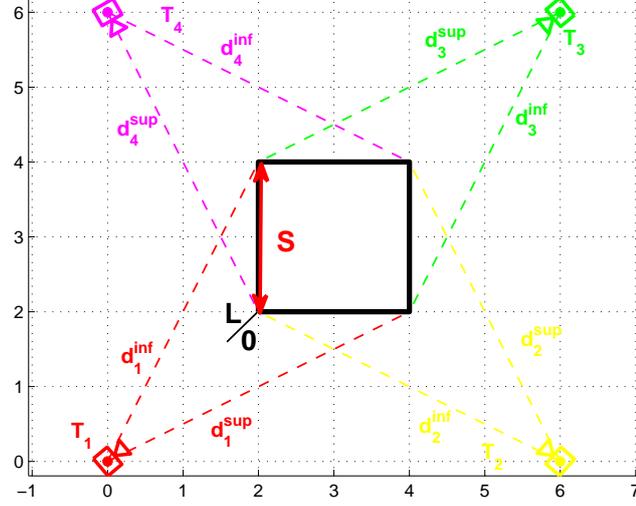


Fig. 2.3: Un esempio del problema di pattugliamento sul nuovo scenario: il perimetro ha forma quadrata come in Fig. 2.2, ma la configurazione delle videocamere è diversa. In particolare  $V_1$  è posta in corrispondenza del vertice in basso a sinistra, in modo che il suo PR  $D_1$  sia del tipo (2.14); il segmento  $S = [\underline{d}_1 \bar{d}_N]$  è rappresentato in rosso e corrisponde all'ultimo lato del quadrato .

Seguendo questa logica si è scritto il nuovo problema di ottimizzazione  $\mathcal{P}'_3$ .

$$\begin{aligned}
 (\mathcal{P}'_3) : \quad & \min_{\{\ell_i\}_{i=2}^N, \{r_i\}_{i=1}^N, \varepsilon} \sum_{i=1}^N \frac{(r'_i - \ell'_i)^2}{\bar{v}_i} \\
 & t.c. \quad 0 \leq \varepsilon \leq \bar{d}_N - \underline{d}_1 \\
 & \underline{d}'_i = \underline{d}_i + \varepsilon \quad i = 2, \dots, N \\
 & \bar{d}'_i = \bar{d}_i + \varepsilon \quad i = 1, \dots, N - 1 \\
 & \ell'_i = \ell_i + \varepsilon \quad i = 2, \dots, N \\
 & r'_i = r_i + \varepsilon \quad i = 1, \dots, N - 1 \\
 & \underline{d}'_i \leq \ell'_i, r'_i \leq \bar{d}'_i \quad i = 1, \dots, N \\
 & r'_i \geq \ell'_{i+1} \quad i = 1, \dots, N - 1
 \end{aligned}$$

con

$$\underline{d}'_1 = \ell'_1 = 0, \quad \bar{d}'_N = r'_N = L \quad e \quad \ell_1 = r_N$$

Una volta fissato  $\varepsilon$ ,  $\mathcal{P}'_3$  diventa a tutti gli effetti un problema  $\mathcal{P}_3$ , di cui sappiamo esistere un'unica soluzione ottima (Proposizione II.3 in [3]) e anche

le sue caratteristiche (Proposizioni 1 e 2 in [1]). Purtroppo  $\mathcal{P}'_3$  non è un problema di ottimizzazione convessa e, come vedremo nel seguito, non è per nulla detto che esista un'unica soluzione ottima. Vale ora la pena di indagare sulle proprietà del problema  $\mathcal{P}'_3$ , innanzitutto si distinguono due casi: uno banale e un altro non banale.

### 2.3.1 Caso banale

All'inizio della sezione si è ipotizzato che  $D_1$  e  $D_N$  avessero un segmento di  $\mathcal{L}$  in comune, nulla invece si è detto per gli altri PR eccetto che debbano verificare le condizioni (2.15). Ipotizziamo ci sia una coppia di videocamere contigue,  $V_i$  e  $V_{i+1}$  in cui il limite fisico superiore della prima corrisponda al limite fisico inferiore della seconda:  $\exists i \text{ t.c. } \bar{d}_i = \underline{d}_{i+1}$ . In questo caso la soluzione al problema  $\mathcal{P}'_3$  è unica. Per provarlo si ridefinisce il problema ponendo  $\bar{d}_i$  come inizio (punto 0) di  $\mathcal{L}$  e si rinominano le videocamere in modo che  $V_{i+1} = V_1, V_{i+2} = V_2, \dots, V_i = V_N$ . A questo punto il nuovo segmento  $S = [\underline{d}_1 \bar{d}_N]$  conterrà un solo elemento, il punto precedentemente identificato con  $\bar{d}_i$ , quindi per trovare la soluzione ottima di  $\mathcal{P}'_3$  si dovrà risolvere un solo problema  $\mathcal{P}_3$ , che sappiamo avere un'unica soluzione ottima. Più intuitivamente si poteva notare che qualsiasi soluzione ammissibile (ossia che garantisca la copertura totale del perimetro) di  $\mathcal{P}'_3$  deve porgere:  $r_i = \bar{d}_i$  e  $\ell_{i+1} = \underline{d}_{i+1}$ , che nelle nuove notazioni diventano:  $\ell_1 = \underline{d}_1 = 0$  e  $r_N = \bar{d}_N = L$ , analoghe a (2.7). Ciò fornisce la giustificazione per ricondurre  $\mathcal{P}'_3$  a un problema di tipo  $\mathcal{P}_3$  con punto iniziale del perimetro in  $\underline{d}_{i+1}$ .

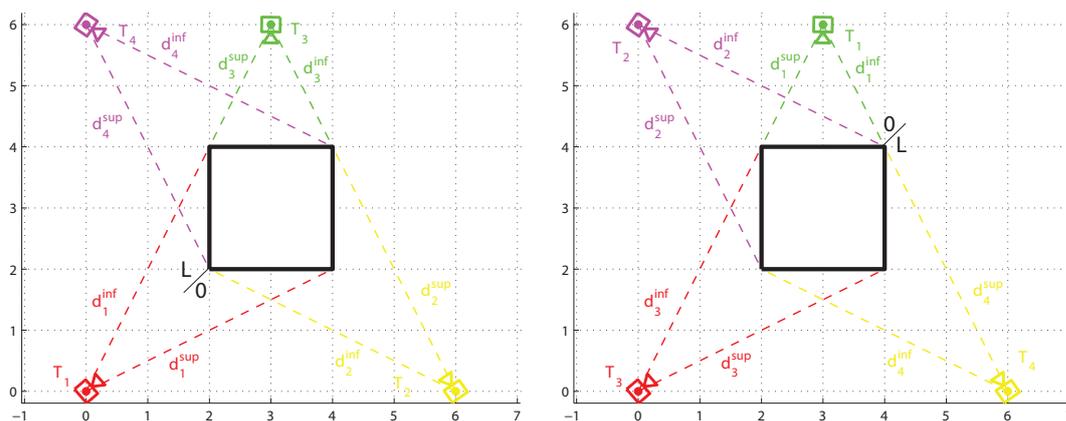


Fig. 2.4: Un esempio del caso banale: la figura a sinistra rappresenta il problema  $\mathcal{P}'_3$  in cui  $\bar{d}_2 = \underline{d}_3$ ; la figura a destra mostra lo stesso scenario dopo che è stato spostato il punto iniziale e sono state rinominate le videocamere, si tratta di un problema  $\mathcal{P}_3$ .

## 2.3.2 Caso non banale

Nel caso in cui tutti i Patrolling Ranges adiacenti abbiano segmenti di perimetro in comune, non è possibile ricondurre  $\mathcal{P}'_3$  a  $\mathcal{P}_3$  e quindi nulla si può dire sul numero di soluzioni ottime del problema. Nel seguito della tesi vedremo che si può trovare una soluzione ottima  $\xi^*$  di  $\mathcal{P}'_3$  attraverso un algoritmo distribuito, descritto nel capitolo 4. Una volta identificata  $\xi^*$  si utilizza la seguente proposizione che fornisce un'indicazione sul numero di soluzioni ottime di  $\mathcal{P}'_3$ .

**Proposizione I** Sia  $\xi^* = (\ell_1^*, r_1^*, \ell_2^*, r_2^*, \dots, \ell_N^*, r_N^*, \varepsilon^*)$  una soluzione ottima del problema di ottimizzazione  $\mathcal{P}'_3$ .  $\xi^*$  è unica se e solo se  $\exists i, j$  t.c.  $\ell_i = \underline{d}_i$  e  $r_j = \bar{d}_j$ ,  $i, j \in \{1, 2, \dots, N\}$ . Se  $\xi^*$  non è unica allora esistono infinite soluzioni di  $\mathcal{P}'_3$ .

La proposizione ci dice che la soluzione ottima  $\xi^*$  è l'unica partizione ottimale di  $\mathcal{L}$  se e solo se in essa ci sono almeno: una videocamera  $V_i$  in cui il limite inferiore della sua PA  $\ell_i^*$  corrisponde con il limite inferiore del suo PR  $\underline{d}_i$ , e una videocamera  $V_j$  in cui il limite superiore della sua PA  $r_j^*$  corrisponde con il limite superiore del suo PR  $\bar{d}_j$  (potrebbe anche essere  $i = j$ ).

*Prova:* Si proverà innanzitutto l'implicazione ( $\Rightarrow$ ), mostrando che se in  $\xi^*$  nessun  $\ell_i^*$  coincide con il limite fisico inferiore  $\underline{d}_i$ , allora  $\xi^*$  non è l'unica soluzione di  $\mathcal{P}'_3$ , anzi ne esistono infinite. Ipotizziamo che in  $\xi^*$  nessun  $\ell_i^*$  coincida con  $\underline{d}_i$ , si definisce  $\delta = \min_{i=1, \dots, N} (\ell_i^* - \underline{d}_i) > 0$  e si prende  $k \in \mathbb{R} : 0 < k \leq \delta$ . Si consideri ora:

$$\begin{aligned} \xi^{*'} &= (\ell_1^{*'}, r_1^{*'}, \ell_2^{*'}, r_2^{*'}, \dots, \ell_N^{*'}, r_N^{*'}, \varepsilon^{*'}) = \\ &= (\ell_1^* - k, r_1^* - k, \ell_2^* - k, r_2^* - k, \dots, \ell_N^* - k, r_N^* - k, \varepsilon^* - k), \end{aligned}$$

si può verificare che  $\xi^{*'}$  appartiene ancora all'insieme dei vincoli di  $\mathcal{P}'_3$  e inoltre il valore della funzione costo rimane invariato:

$$\begin{aligned} f_3(\xi^{*'}) &= \frac{(r_1^{*'} + \varepsilon^{*'})^2}{\bar{v}_1} + \sum_{i=2}^{N-1} \frac{(r_i^{*'} - \ell_i^{*'})^2}{\bar{v}_i} + \frac{((L - \varepsilon^{*'}) - \ell_N^{*'})^2}{\bar{v}_N} = \\ &= \frac{((r_1^* - k) + (\varepsilon^* + k))^2}{\bar{v}_1} + \sum_{i=2}^{N-1} \frac{((r_i^* - k) - (\ell_i^* - k))^2}{\bar{v}_i} + \frac{((L - (\varepsilon^* - k)) - (\ell_N^* - k))^2}{\bar{v}_N} = \\ &= \frac{(r_1^* + \varepsilon^*)^2}{\bar{v}_1} + \sum_{i=2}^{N-1} \frac{(r_i^* - \ell_i^*)^2}{\bar{v}_i} + \frac{((L - \varepsilon^*) - \ell_N^*)^2}{\bar{v}_N} = f_3(\xi^*), \end{aligned}$$

quindi anche  $\xi^{*'}$  è una soluzione ottima del problema  $\mathcal{P}'_3$ . Quello che si è fatto in pratica è stato traslare di  $k$  in senso orario tutte le PA  $\{A_i\}_{i=1}^N$

assicurandosi che continuassero a restare all'interno dei rispettivi PR  $\{D_i\}_{i=1}^N$ , la funzione costo del nuovo partizionamento ha lo stesso valore di prima poiché sono rimaste invariate le lunghezze di ciascuna  $A_i$ . In generale qualsiasi  $k \in (0 \delta]$  porta a una partizione ottima, perciò esistono infinite soluzioni ottime con  $\{A_i\}_{i=1}^N$  della stessa lunghezza. Un analogo ragionamento si può fare con gli  $r_i$  e i limiti fisici superiori  $\bar{d}_i$ , traslando le partizioni in senso antiorario di  $q$ , con  $0 \leq q \leq \eta$  e  $\eta = \min_{i=1, \dots, N} (\bar{d}_i - r_i^*) > 0$ .

L'implicazione inversa ( $\Leftarrow$ ) si dimostra facendo vedere che se  $\xi^*$  ha almeno un  $\ell_i = \underline{d}_i$  e un  $r_j = \bar{d}_j$  allora qualsiasi altra soluzione ammissibile di  $\mathcal{P}'_3$  ha un costo maggiore di  $\xi^*$ . Si nota intuitivamente che il procedimento utilizzato per dimostrare l'esistenza di più soluzioni non è applicabile: non si può infatti traslare le partizioni  $A_i$  né verso destra né verso sinistra preservandone le lunghezze. Inoltre la modifica delle  $A_i$  non può essere unilaterale: se  $\exists i \text{ t.c. } |A'_i| < |A_i| \Rightarrow \exists j \text{ t.c. } |A'_j| > |A_j|$  e siccome le videocamere si stanno già muovendosi alla loro massima velocità  $T'_0 = T'_j > T_0$ . Quindi a questa partizione è associata una funzione costo  $J$  maggiore.

## 2.4 Angolo d'Incidenza Visuale e funzione penalità

La Proposizione I assicura l'esistenza di un' unica soluzione ottima di  $\mathcal{P}'_3$  solo se in essa una videocamera o un gruppo di videocamere saturano completamente le loro zone di competenza. Non è per niente anomalo il caso in cui esistano infinite partizioni ottime di  $\mathcal{L}$ , soluzioni di  $\mathcal{P}'_3$ . Quest'ultima situazione è poco desiderabile poiché non si può assicurare la convergenza di nessun algoritmo per la ricerca dell'ottimo, che invece potrebbe oscillare tra le varie partizioni che minimizzano  $f_3$ , la funzione costo di  $\mathcal{P}'_3$ .

Occorre innanzitutto introdurre un criterio per scegliere una soluzione ottima "migliore"  $\xi^{**}$  tra tutte quelle di  $\mathcal{P}'_3$ . Successivamente si progetterà una funzione penalità seguendo questo criterio, da aggiungere alla funzione costo  $f_3$  per rendere  $\xi^{**}$  l'unica soluzione ottima del nuovo problema di ottimizzazione.

A tal scopo si definisce l'**angolo d'incidenza visuale**  $\psi$ . Data una videocamera  $V_i$  e un punto  $x \in \mathcal{L}$  tale che  $x \in D_i$ , sia  $r$  la retta passante per  $x$  e per la posizione fisica di  $V_i$  e sia  $s$  la retta passante per  $x$  e perpendicolare ad  $\mathcal{L}$ . Se ipotizziamo che  $x$  sia la posizione del f.o.v. di  $V_i$ , allora l'angolo d'incidenza visuale  $\psi$  di  $V_i$  su  $x$ , è l'angolo tra la retta  $r$  e la retta  $s$ .

In particolare, se  $\psi = 0$ , la videocamera ha una vista perpendicolare del perimetro, che considereremo la migliore possibile. Man mano che  $|\psi|$  aumenta

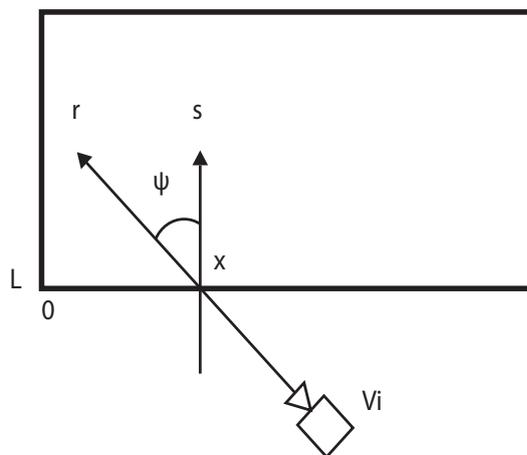


Fig. 2.5: L'angolo d'incidenza visuale  $\psi$  è compreso tra la retta  $s$  passante per  $x \in \mathcal{L}$  e perpendicolare a  $\mathcal{L}$  e la retta  $r$  che passa per  $x$  e la posizione fisica della videocamera.

la qualità della visuale della videocamera peggiora. Il caso limite  $|\psi| = 90$  (in pratica impossibile da raggiungere) è la situazione peggiore, in cui la visuale della videocamera è parallela al perimetro.

Tra tutte le partizioni ottime di  $\mathcal{P}'_3$ , che impongono a  $V_i$  una lunghezza fissa  $c$  di  $A_i$ , si vuole quella che ha la visuale più perpendicolare dei punti del segmento da pattugliare, ossia quella che presenta angoli d'incidenza  $\psi$  minori. Si fornisce di seguito un esempio grafico dell'applicazione di questo criterio.

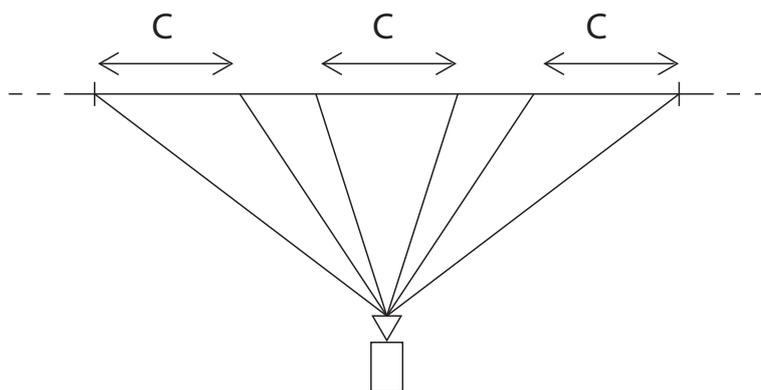


Fig. 2.6: Tra le tre PA di lunghezza  $c$  rappresentate la migliore è quella centrale, poiché in essa la videocamera ha una visuale più perpendicolare rispetto alle altre due.

Per formalizzare il ragionamento si definisce per ogni  $V_i$  una funzione:

$$\psi_i : \mathbb{R}^+ \longrightarrow \mathbb{R} : x \in D_i \longmapsto \psi \in \Psi = (-90 \ + 90), \quad (2.18)$$

che associa ad ogni punto  $x$  del PR di  $V_i$  un angolo d'incidenza visuale  $\psi$ . L'andamento delle funzioni  $\psi_i$  dipende dal posizionamento della videocamera rispetto al perimetro e dalla forma del perimetro stesso, in generale può essere molto diverso. Se però si evitano situazioni particolari, come rientranze all'interno del perimetro, le  $\psi_i$  sono funzioni strettamente decrescenti con valori sia positivi che negativi.

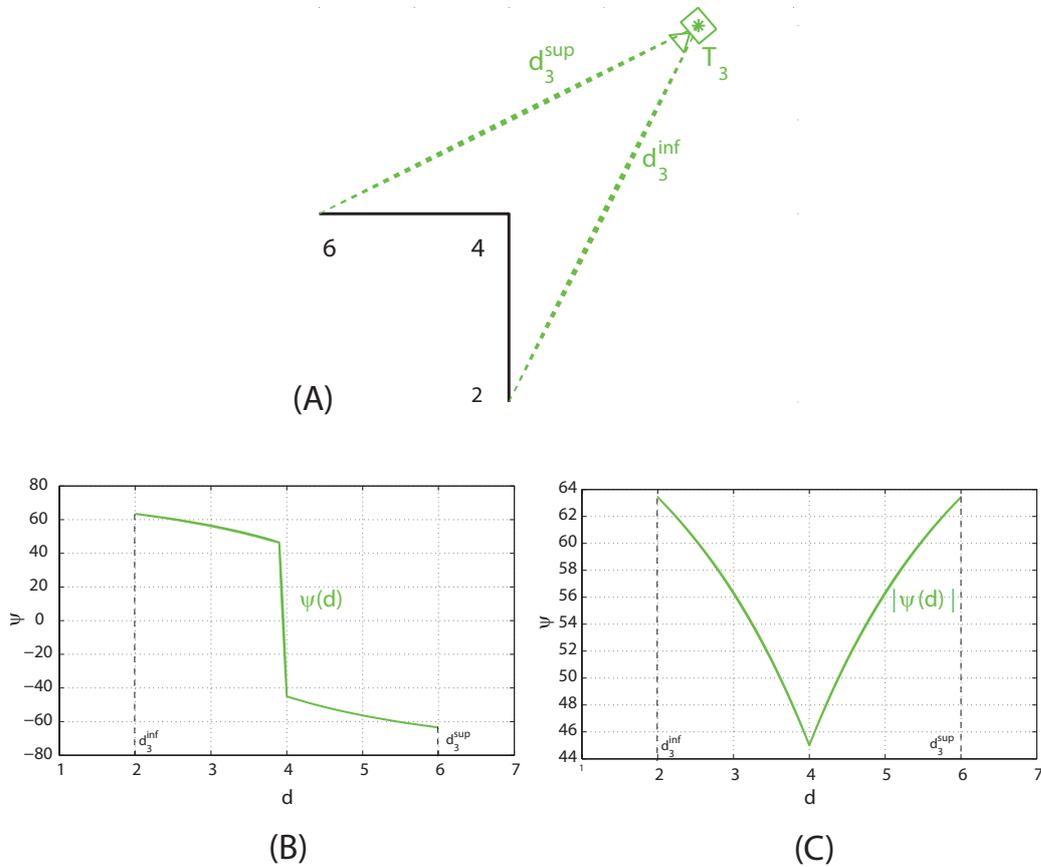


Fig. 2.7: (A): Videocamera 3 di Fig. 2.3; (B): Rappresentazione sul piano  $d$ - $\psi$  della funzione  $\psi(d)$  per la videocamera in (A); (C): Rappresentazione sul piano  $d$ - $\psi$  della funzione  $|\psi(d)|$  per la videocamera (A).  $d_3^{inf} = \underline{d}_3$  e  $d_3^{sup} = \bar{d}_3$

Siccome non si è interessati al segno dell'angolo di incidenza  $\psi$  ma solo al suo valore assoluto si deve prendere in considerazione la funzione  $|\psi_i|$ :

$$|\psi_i| : \mathbb{R}^+ \longrightarrow \mathbb{R}^+ : x \in D_i \longmapsto \psi \in \Psi = [0; +90),$$

ovviamente avrà sempre valori positivi o nulli, inoltre, data la monotonica decrescenza di  $\psi_i$ , avrà un minimo assoluto in corrispondenza del punto  $d_{min}$  dove la visuale di  $V_i$  è più perpendicolare ad  $\mathcal{L}$ .

Adesso si può fornire una descrizione matematica della Patrolling Area migliore  $A_i^*$ , tra tutte quelle di lunghezza  $c$ :

$$A_i^* = \arg \min_{A_i} \int_{A_i} |\psi_i(x)| dx \quad (2.19)$$

$$t.c. \quad |A_i| = c \quad (2.20)$$

Si cercano  $\ell_i$  e  $r_i$ , distanti  $c$ , in modo da minimizzare l'area sottesa alla curva  $|\psi_i(d)|$  e compresa tra essi. La soluzione ottima, ove possibile, si ottiene prendendo l'estremo destro e sinistro di  $A_i^*$ , rispettivamente prima e dopo  $d_{min}$ , in modo che  $|\psi_i(\ell_i^*)| = |\psi_i(r_i^*)|$ . Se la lunghezza del segmento  $c$  e la forma di  $|\psi_i(d)|$  non permettono questa soluzione, allora  $A_i^*$  è quella che minimizza il valore assoluto della differenza  $|\psi_i(\ell_i)| - |\psi_i(r_i)|$ .

Alla luce di queste osservazioni l'espressione (2.19) si può riscrivere:

$$A_i^* = [\ell_i^* \ r_i^*] = \arg \min_{\ell_i, r_i} ||\psi_i(\ell_i)| - |\psi_i(r_i)|| \quad (2.21)$$

$$con \quad \ell_i + r_i = c$$

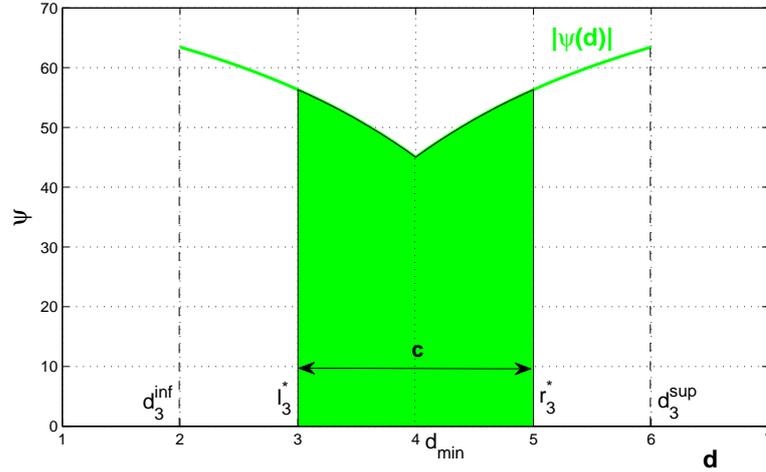


Fig. 2.8: Rappresentazione dell'area ottima  $Area_3^{ott}$  sottesa alla curva  $\psi_3(d)$  di  $A_3^*$  di lunghezza  $c = 2$  per la videocamera 3 di Fig. 2.3

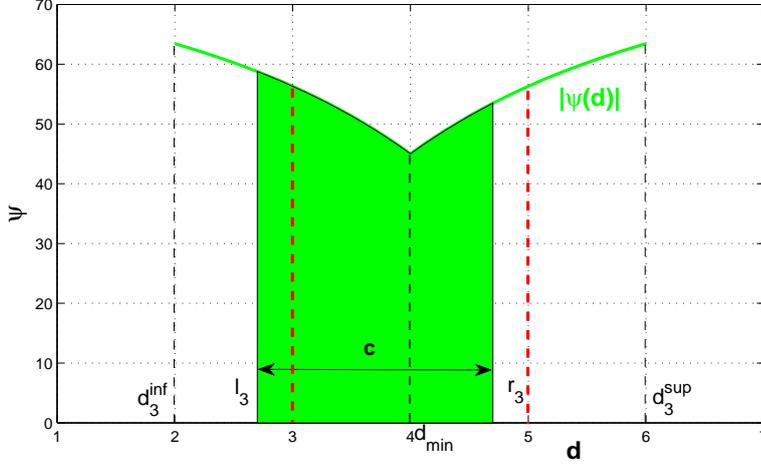


Fig. 2.9: Rappresentazione di un'altra area non ottima sottesa alla curva  $\psi_3(d)$  di  $A_3$  di lunghezza sempre  $c = 2$ : il valore di quest'area è maggiore di quello di Fig. 2.8, per convincersene basta confrontare due sezioni d'area: la prima  $S_1$  è compresa tra  $\ell_3$  e la linea tratteggiata rossa di sinistra, la seconda  $S_2$  è tra  $r_3$  e la linea tratteggiata rossa di destra; la nuova area totale  $Area_3$  è pari a  $Area_3^* + S_1 - S_2$ , siccome  $S_1 > S_2 \Rightarrow Area_3 > Area_3^*$ . Questo è vero per qualsiasi altra PA di lunghezza  $c$ .

La partizione ottima migliore  $\{A_i^*\}_{i=1}^N$  (associata alla soluzione  $\xi^{**}$  di cui si è parlato all'inizio della sezione), in cui viene minimizzato  $T_{lag}$  garantendo congiuntamente alle videocamere la visuale più perpendicolare possibile della loro PA, è quella, tra tutte le partizioni ottime del problema  $\mathcal{P}'_3$ , il cui valore del seguente indice è il minore:

$$P_\Psi = \sum_{i=1}^N \int_{\ell_i}^{r_i} |\psi_i(x)| dx \quad (2.22)$$

Se si considera  $P_\Psi$  come la funzione penalità cercata allora si può definire il nuovo problema  $\mathcal{P}_4$ , che ha lo stesso insieme di vincoli di  $\mathcal{P}'_3$  e funzione costo  $f_4$ , la cui espressione è data da:

$$f_4 = f_3 + \frac{1}{C} P_\Psi = \sum_{i=1}^N \frac{(r'_i - \ell'_i)^2}{\bar{v}_i} + \frac{1}{C} \sum_{i=1}^N \int_{\ell'_i}^{r'_i} |\psi'_i(x)| dx \quad (2.23)$$

dove  $C \in \mathbb{R}^+$  è una costante il cui significato verrà chiarito nel seguito.

In generale non è vero che la soluzione ottima  $\xi_4^*$  di  $\mathcal{P}_4$  sia anche una soluzione ottima di  $\mathcal{P}'_3$ . Le PA  $\{A_i\}_{i=1}^N$  corrispondenti a  $\xi_4^*$  avranno una lunghezza che

dipende sia dalla velocità delle rispettive videocamere (termine  $f_3$ ), come in  $\mathcal{P}'_3$ , sia dall'area sottesa alla loro funzione  $|\psi_i|$  (termine  $P_\Psi$ ). La costante  $C$  serve per pesare il contributo delle aree nel definire la partizione ottima:

- con  $C$  sufficientemente grande, il termine  $P_\Psi$  in  $f_4$  è molto ridotto e la soluzione ottima  $\xi_4^*$  risulta molto vicina a quella di  $\mathcal{P}'_3$  in termini di lunghezze delle PA, quindi  $\max_{A_i \in \xi_4^*} T_{lag}(A_i) \approx \max_{A_i \in \xi_3^*} T_{lag}(A_i)$ ;
- con  $C$  piccolo, le aree sottese alle funzioni  $\psi_i$  assumono un ruolo rilevante nell'identificazione della partizione ottima, in essa si potrebbe penalizzare  $T_{lag}$  per favorire la qualità della visuale delle videocamere,  $\max_{A_i \in \xi_4^*} T_{lag}(A_i) > \max_{A_i \in \xi_3^*} T_{lag}(A_i)$

Va osservato che per alcuni scenari, ad esempio configurazioni simmetriche con videocamere tutte con la stessa velocità massima  $\bar{v}_i$  e uguali funzioni angolo d'incidenza  $\psi_i$  (come in Fig. 2.3), la partizione associata a  $\xi_4^*$  risulta anche soluzione di  $\mathcal{P}'_3$  e quindi minimizza  $T_{lag}$ . In generale questo non è vero, ma se si sceglie un  $C$  sufficientemente grande,  $\xi_4^*$  è molto vicina a una soluzione ottima di  $\mathcal{P}'_3$ , garantendo  $T_{lag}$  prossimi al minimo, e inoltre fornisce il partizionamento migliore in termini di angoli d'incidenza  $\psi$ , ossia di qualità della visuale delle singole videocamere.

### 3. OTTIMIZZAZIONE DISTRIBUITA E PROTOCOLLI DI COMUNICAZIONE

Nella prima sezione di questo capitolo si descrivono le caratteristiche di un algoritmo distribuito per l'ottimizzazione in una rete multi-agente, dal punto di vista teorico.

Nella seconda sezione si fornisce l'impostazione al problema di partizionamento del perimetro secondo uno scenario iterativo tipico degli algoritmi di ottimizzazione distribuita. In particolare si spiega come le PA di ciascuna videocamera vengano settate all'inizializzazione del sistema, e quali siano i vincoli che devono rispettare ad ogni iterazione del partizionamento.

Nella sezione finale si illustrano le tre modalità considerate per la comunicazione tra le videocamere, basate su altrettanti protocolli di Gossip: sincrono, simmetrico, asimmetrico. Esse definiscono non solo quali videocamere comunicano ad ogni iterazione ma anche quali debbano aggiornare la propria PA, risultando cruciali per la velocità di evoluzione del sistema.

#### 3.1 *Pattugliamento di un perimetro, scenario distribuito*

L'obiettivo principale del pattugliamento è evitare che ci siano intrusioni non rilevate nel perimetro. Per tradurre matematicamente questa necessità è stato formulato  $\mathcal{P}_1$ , nel quale la quantità che si intende minimizzare è il massimo dei  $T_{lag}$  tra le videocamere della rete. Dalla forma della funzione costo (2.3) di  $\mathcal{P}_1$ , si vede che la sua soluzione richiede che ciascun agente della rete risolva un problema globale e quindi sia a conoscenza di tutte le caratteristiche delle altre videocamere (estremi delle PA, velocità massime  $\bar{v}_i$ , etc...). Per permettere l'elaborazione di un algoritmo distribuito per il partizionamento, viene introdotto  $\mathcal{P}_3$ , la cui funzione costo (2.8) si presta ad un calcolo parallelizzabile e quindi apre la strada per una soluzione distribuita.

In questa sezione si considerano i problemi di partizionamento di un perimetro reale all'interno di uno scenario iterativo e distribuito, prendendo spunto dalla formulazione proposta in [2] e [3] per  $\mathcal{P}_3$ . Si assume che ciascuna videocamera  $V_i$ , al tempo  $t = 0$  venga inizializzata con una PA  $A_i(0)$ , che in generale non coincide con la soluzione ottima. Ciascuna videocamera

può aggiornare la sua PA  $A_i(t)$  usando esclusivamente le informazioni locali provenienti dalle videocamere vicine:  $V_{i-1}$  e  $V_{i+1}$ . L'obiettivo è elaborare delle strategie che portino le videocamere a raggiungere asintoticamente la configurazione ottima e a stabilizzarsi in essa. La configurazione finale desiderata è il partizionamento ottimo del problema preso in considerazione ( $\mathcal{P}'_3$  o  $\mathcal{P}_4$ ). Nel seguito si descrivono nel dettaglio le impostazioni assunte per l'inizializzazione del sistema. Al tempo  $t = 0$  ciascuna videocamera  $V_i$  viene inizializzata con una PA  $A_i(0)$  che corrisponde al suo PR  $D_i$ :

$$A_i(0) = [\ell_i(0) \ r_i(0)] = [\underline{d}_i \ \bar{d}_i] = D_i, \quad i \in \{1, 2, \dots, N\}, \quad (3.1)$$

dove  $\ell_i(0)$  e  $r_i(0)$  sono rispettivamente l'estremo sinistro e destro di  $A_i(0)$ . Risulta facile verificare che l'insieme delle PA  $\{A_1(0), A_2(0), \dots, A_N(0)\}$  così inizializzate soddisfa i seguenti vincoli:

- vincoli fisici:

$$A_i(0) \subseteq D_i \quad i \in \{1, 2, \dots, N\}; \quad (3.2)$$

- vincoli di allacciamento:

$$\ell_i(0) \leq \ell_{i+1}(0) \leq r_i(0) \leq r_{i+1}(0), \quad i = 2, \dots, N \quad (3.3)$$

$$0 \leq \ell_2(0) \leq r_1(0) \leq r_2(0) \quad (3.4)$$

$$\ell_N(0) \leq \ell_1(0) \leq r_N(0) \leq L; \quad (3.5)$$

- vincolo di copertura del perimetro:

$$\bigcup_{i=1, \dots, N} A_i(0) = \mathcal{L}. \quad (3.6)$$

L'obiettivo è progettare un algoritmo di partizionamento iterativo che permetta alle videocamere di aggiornare gli estremi delle loro PA usando solo informazioni locali, provenienti dalle videocamere adiacenti, in modo che:

- (i) i vincoli fisici, di allacciamento e il vincolo di copertura del perimetro siano soddisfatti ad ogni iterazione;
- (ii) le PA convergano verso la partizione ottima definita dal problema di ottimizzazione considerato:  $\mathcal{P}'_3$  o  $\mathcal{P}_4$ .

Si è precedentemente parlato di videocamere “vicine” e comunicazioni locali, nel seguito se ne fornisce una definizione formale. Data una videocamera  $V_i$ , se  $i \in \{2, 3, \dots, N - 1\}$ , le videocamere vicine sono  $V_{i-1}$  e  $V_{i+1}$ , rispettivamente precedente e successiva, e sono le uniche due videocamere della rete

con le quali  $V_i$  può scambiare informazioni. Le vicine di  $V_1$  sono  $V_N$  e  $V_2$ , precedente e successiva, mentre quelle di  $V_N$  sono  $V_{N-1}$  e  $V_1$ .

La disposizione dei collegamenti forma un anello logico di comunicazione tra gli agenti della rete, dove  $V_1$  segue  $V_N$ , come in Fig. 3.1b. A differenza di quanto presupposto in [1] e [3] per il problema di partizionamento sul segmento (dove  $V_1$  ha come unica vicina  $V_2$  mentre  $V_N$  ha come unica vicina  $V_{N-1}$ , si veda Fig. 3.1a) si permettono comunicazioni anche tra le videocamere iniziale e finale. Grazie all'ipotesi (2.14) sulla forma del PR di  $V_1$  e all'introduzione del link che chiude l'anello di comunicazione, le videocamere  $V_1$  e  $V_N$  si comportano come qualsiasi altro agente della rete.

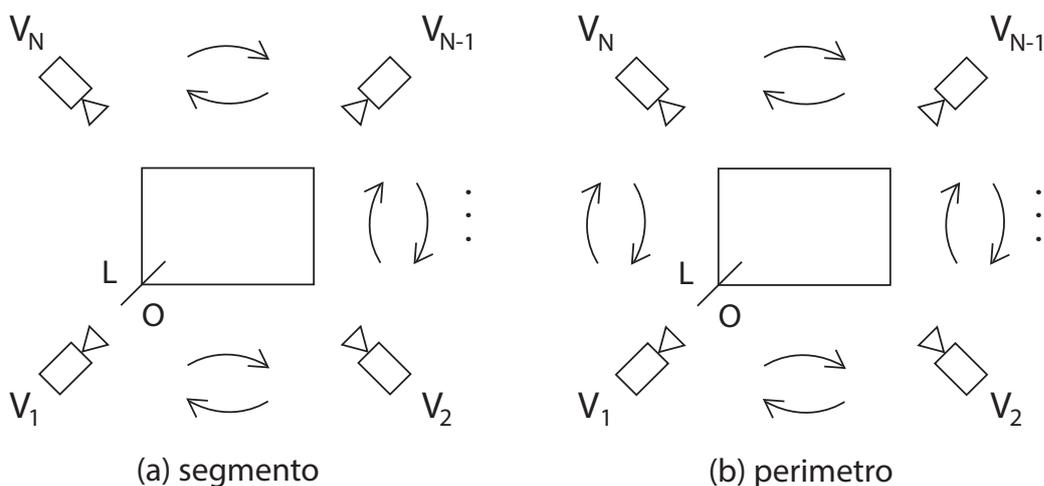


Fig. 3.1: Configurazione dei link di comunicazione tra le videocamere per il partizionamento del segmento (a) e del perimetro (b).

Le modalità con le quali le videocamere si scambiano i dati necessari per l'aggiornamento delle PA, vengono descritte dettagliatamente nella prossima sezione.

### 3.2 Protocolli di comunicazione e aggiornamento

I metodi di comunicazione e aggiornamento utilizzati si basano su protocolli di comunicazioni di tipo Gossip, particolarmente adatti a sistemi multi-agente, facili da implementare e molto robusti. Vengono considerate tre diverse modalità per lo scambio dati e l'aggiornamento delle PA delle videocamere: Gossip Sincrono, Gossip Simmetrico, Gossip Asimmetrico. Ciascuna prende il nome del tipo di protocollo di comunicazione Gossip su cui si basa.

Nel seguito viene descritto nel dettaglio il funzionamento di ognuna di queste tre tecniche.

### 3.2.1 Gossip Sincrono

Ad ogni istante  $t \in \mathbb{N}$ , corrispondente ad una iterazione dell'algoritmo di partizionamento, ciascuna videocamera  $V_i$ ,  $i = 1, 2, \dots, N$ , trasmette alle sue vicine i valori degli estremi  $\ell_i(t)$  e  $r_i(t)$  della sua attuale PA  $A_i(t)$ . Inoltre ad ogni iterazione  $t \in \mathbb{N}$  tutte le videocamere sono abilitate ad aggiornare gli estremi della loro PA, seguendo la regola di aggiornamento dettata dalla strategia di partizionamento utilizzata. In Fig. 3.2 si rappresentano i link di comunicazione attivi e le PA aggiornate durante una generica iterazione  $t$  di un algoritmo di partizionamento che utilizza Gossip Sincrono.

La rapidità con cui evolve il sistema è il principale vantaggio di questa tecnica, poiché lo porta a raggiungere la configurazione finale in poche iterazioni, quindi in tempi brevi. D'altro canto richiede che una grande mole di dati (minimo  $4 \cdot N$  valori di estremi di PA) sia trasmesso sulla rete ad ogni iterazione.

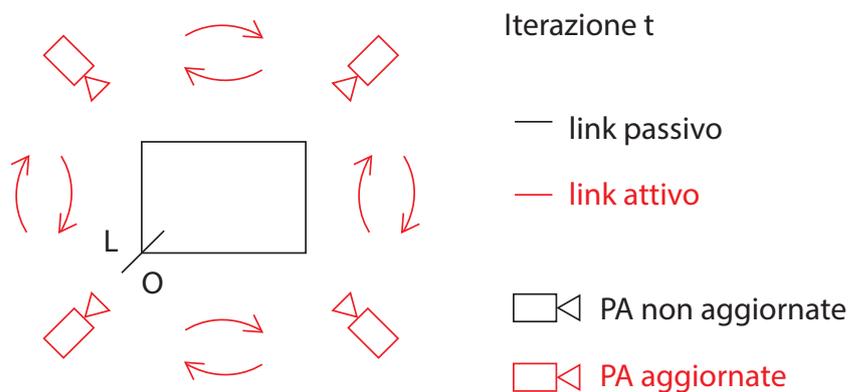


Fig. 3.2: Trasmissioni e aggiornamenti delle PA durante un'iterazione del Gossip Sincrono.

### 3.2.2 Gossip Simmetrico

Con questo metodo si rilassa la domanda di dati da trasmettere, assumendo che ad ogni iterazione dell'algoritmo di partizionamento solo una coppia di videocamere adiacenti,  $V_i$  e  $V_{i+1}$ , comunichi tra loro, scambiandosi vicendevolmente i valori degli estremi delle attuali PA,  $A_i(t)$  e  $A_{i+1}(t)$ . La scelta della coppia di videocamere da impegnare nella comunicazione all'iterazione  $t$  è completamente randomica. Durante questa stessa iterazione il resto delle

videocamere  $V_j$  della rete, con  $j \notin \{i, i+1\}$ , non trasmette né riceve alcuna informazione. Similmente, ad ogni iterazione le uniche videocamere abilitate ad aggiornare le loro PA sono le stesse che hanno comunicato. In particolare  $V_i$  aggiornerà solo il valore dell'estremo destro  $r_i(t)$ , mentre  $V_{i+1}$  aggiornerà il valore dell'estremo sinistro  $\ell_{i+1}(t)$ . In Fig. 3.3 si rappresentano i link di comunicazione attivi e le PA aggiornate durante una generica iterazione  $t$  di un algoritmo di partizionamento che utilizza Gossip Simmetrico.

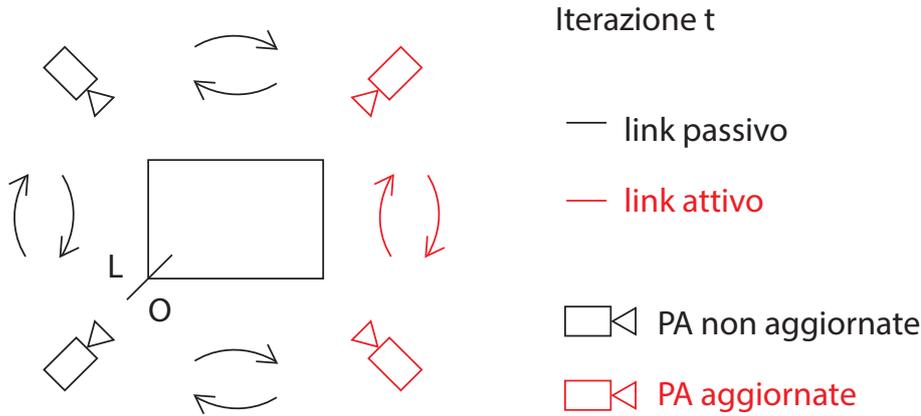


Fig. 3.3: Trasmissioni e aggiornamenti delle PA durante un'iterazione del Gossip Simmetrico.

La quantità di dati trasmessa sulla rete per ogni iterazione è  $\frac{1}{N}$  volte quella richiesta dal Gossip Sincrono, d'altro canto solo due PA alla volta vengono aggiornate, quindi la convergenza alla configurazione finale è sicuramente più lenta. In particolare il numero di iterazioni richieste per la convergenza è maggiore o uguale a  $N$  volte quelle richieste dal Gossip Sincrono, a parità di strategia di partizionamento adottata. Il margine d'incertezza sulla relazione è causato dall'aleatorietà con la quale vengono scelte le coppie di videocamere comunicanti ad ogni iterazione.

Si consideri, ad esempio, il caso in cui due iterazioni successive,  $t$  e  $(t+1)$ , coinvolgano la stessa coppia di videocamere  $V_i$  e  $V_{i+1}$ : l'iterazione  $(t+1)$  non avrebbe alcun effetto sulla configurazione del sistema, cioè  $A_j(t) = A_j(t+1)$ ,  $\forall j \in \{1, 2, \dots, N\}$ , in quanto  $A_i(t)$  e  $A_{i+1}(t)$ , le uniche PA a poter cambiare durante questa iterazione, hanno già raggiunto il consensus all'iterazione precedente,  $t$ . In pratica l'iterazione  $(t+1)$  non contribuisce all'evoluzione del sistema. Più in generale, si considerino due iterazioni,  $t$  e  $(t+k)$ , che coinvolgono la stessa coppia di videocamere,  $V_i$  e  $V_{i+1}$ , e si assuma che nessuna iterazione tra  $t$  e  $(t+k)$  interessi una delle due: le uniche PA a poter cambiare a seguito di  $(t+k)$  sono  $A_i(t+k)$  e  $A_{i+1}(t+k)$  che si trovano già nel

loro valore ottimo relativo <sup>1</sup>, quindi  $A_j(t+k+1) = A_j(t+k)$ ,  $\forall j \in \{1, 2, \dots, N\}$  e l'iterazione  $(t+k)$  non contribuisce all'evoluzione del sistema. Risulta ora più chiaro perché il numero di iterazioni per la convergenza richieste da un algoritmo di partizionamento che utilizza il Gossip Simmetrico, sia sicuramente maggiore di  $N$  volte quello richiesto utilizzando il protocollo di comunicazione basato su Gossip Sincrono.

### 3.2.3 Gossip Asimmetrico

La quantità di dati trasmessi sulla rete è la minore tra i metodi considerati, infatti ad ogni iterazione solo una coppia di videocamere vicine,  $V_i$  e  $V_{i+1}$ , comunica in maniera unidirezionale. Sarebbe a dire che solo una delle due trasmette i suoi dati alla rimanente, che al contrario può solo ricevere. La scelta della coppia di videocamere che comunicano durante un'iterazione è randomica, come nel caso del Gossip Simmetrico. Se, ad esempio, è  $V_{i+1}$  che trasmette il valore degli estremi della sua attuale PA  $A_{i+1}(t)$  a  $V_i$ , allora l'unica videocamera abilitata ad aggiornare la sua PA sarà quest'ultima, e in particolare aggiornerà il suo estremo destro  $r_i(t)$ . Viceversa, se è  $V_i$  a trasmettere, allora  $V_{i+1}$  aggiornerà il suo estremo sinistro  $\ell_i(t)$ . In Fig. 3.4 si rappresentano i link di comunicazione attivi e le PA aggiornate durante una generica iterazione  $t$  di un algoritmo di partizionamento che utilizza Gossip Asimmetrico.

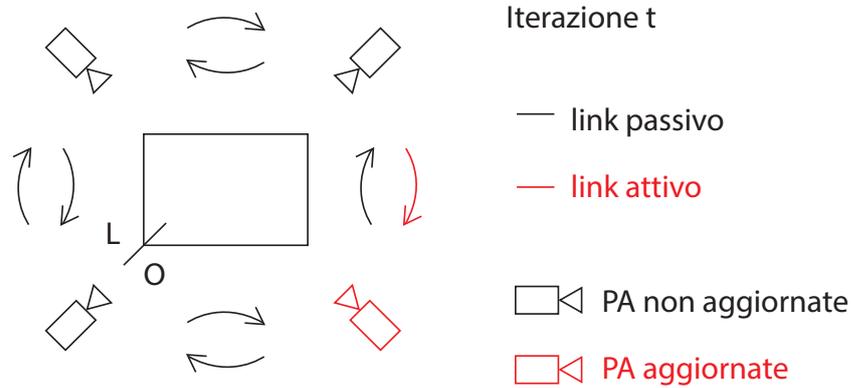


Fig. 3.4: Trasmissioni e aggiornamenti delle PA durante un'iterazione del Gossip Asimmetrico.

I dati trasmessi sulla rete ad ogni iterazione sono  $\frac{1}{2N}$  volte quelli trasmessi

<sup>1</sup>  $A_i$  e  $A_{i+1}$  si trovano nel loro valore ottimo relativo a seguito dell'iterazione  $t$ , quindi da  $k-1$  iterazioni.

---

nel Gossip Sincrono e  $\frac{1}{2}$  di quelli trasmessi nel Gossip Simmetrico. Rispetto a quest'ultimo, per scambiare la stessa quantità di dati viene richiesto il doppio delle iterazioni. Utilizzando un ragionamento analogo a quello fatto nella sezione precedente, si può affermare che il numero di iterazioni necessarie per raggiungere la configurazione finale è sicuramente maggiore di  $2N$  volte quello necessario utilizzando il Gossip Sincrono come protocollo di comunicazione. Risulta invece più complicato un confronto col numero di iterazioni necessarie utilizzando il Gossip Asimmetrico, poiché anche quest'ultimo si basa su una scelta casuale delle coppie di videocamere comunicanti. Per un'analisi quantitativa e un confronto più accurato dell'efficienza dei tre metodi di comunicazione descritti si rimanda al capitolo 5.



## 4. ALGORITMI DI PARTIZIONAMENTO

In questo capitolo vengono descritti nel dettaglio gli algoritmi di partizionamento ideati, ponendo particolare attenzione al criterio con cui le videocamere scelgono le nuove PA a seguito di una comunicazione. Il primo algoritmo proposto, *Algoritmo Perimetrico* (AP), è stato pensato per trovare una soluzione ottima del problema  $\mathcal{P}'_3$ . Il suo funzionamento, che utilizza un metodo grafico, non richiede grossi calcoli da parte delle videocamere e risulta molto semplice da analizzare. Il secondo algoritmo, *Algoritmo Perimetrico - Angolo d'Incidenza* (AP $\Psi$ ), trova la soluzione ottima di  $\mathcal{P}_4$ . Seppur anch'esso basi la sua regola di aggiornamento su un metodo grafico, presenta dinamiche decisamente più complesse e maggiori oneri computazionali rispetto al precedente. Se si combina una delle regole di aggiornamento proposte (AP, AP $\Psi$ ) con uno dei tre protocolli di comunicazione considerati (Gossip Sincrono, Gossip Simmetrico, Gossip Asimmetrico) si definisce completamente un algoritmo di partizionamento: infatti si determina ad ogni iterazione sia quali videocamere debbano comunicare e aggiornare la propria PA, sia il criterio secondo il quale viene scelta la nuova area di pattugliamento. Nel seguito si descrivono nel dettaglio AP e AP $\Psi$ .

### 4.1 *Algoritmo Perimetrico (AP)*

L'Algoritmo Perimetrico (AP) è un adattamento al problema di partizionamento su un perimetro reale  $\mathcal{P}'_3$ , degli algoritmi proposti in [1] e [3]. Con questi ultimi due condivide la forma della partizione ottima, nella quale (quando possibile) ciascuna PA è proporzionale alla velocità massima della rispettiva videocamera. Dal momento che  $\mathcal{P}'_3$ , in generale, ha più di una soluzione ottima, non si è in grado di individuare una configurazione precisa verso la quale il sistema tende applicando AP. Il sistema può convergere verso una qualsiasi delle partizioni ottime secondo  $\mathcal{P}'_3$ , che potenzialmente sono infinite. Ciò che le accomuna tutte è la lunghezza delle PA, proporzionali alle velocità massima delle rispettive videocamere. Il funzionamento di AP viene descritto dettagliatamente nel seguito.

Consideriamo una generica comunicazione all'istante  $t$  tra due videocamere vicine,  $V_i$  e  $V_{i+1}$ , e assumiamo che ci sia una sovrapposizione tra i rispet-

tivi PR,  $D_i$  e  $D_{i+1}$  (sempre vero se si fa riferimento al caso non banale di pattugliamento sul perimetro). Se ipotizziamo che il protocollo di comunicazione utilizzato è il Gossip Asimmetrico, eventualmente solo una delle due videocamere aggiornerà la sua PA:

- se  $V_i$  è la videocamera che trasmette e  $V_{i+1}$  quella che riceve, allora  $V_{i+1}$  aggiornerà l'estremo sinistro  $\ell_{i+1}(t)$  della sua PA;
- se, viceversa,  $V_{i+1}$  trasmette e  $V_i$  riceve, allora  $V_i$  aggiornerà l'estremo destro  $r_i(t)$  della sua PA.

Se si utilizza il Gossip Simmetrico entrambi i punti precedenti verranno eseguiti durante la stessa iterazione. In entrambi i casi si vuole che il valore aggiornato  $p^*$  dell'estremo della nuova area di pattugliamento, soddisfi il criterio dell'egual tempo di percorrenza (da ora in avanti criterio *etp*), introdotto in [1]:

*“ il tempo richiesto da  $V_i$  per viaggiare alla velocità  $\bar{v}_i$  da  $p^*$  a  $\ell_i(t)$  è uguale al tempo richiesto da  $V_{i+1}$  per viaggiare alla velocità  $\bar{v}_{i+1}$  da  $p^*$  a  $r_{i+1}(t)$  ”*

Formalmente  $p^*$  deve soddisfare la condizione:

$$\frac{p^* - \ell_i(t)}{\bar{v}_i} = \frac{r_{i+1} - p^*}{\bar{v}_{i+1}}, \quad (4.1)$$

da cui si ricava:

$$p^* = \frac{\ell_i(t) \cdot \bar{v}_{i+1} + r_{i+1} \cdot \bar{v}_i}{\bar{v}_i + \bar{v}_{i+1}} \quad (4.2)$$

Sfortunatamente (4.1) e (4.2) non sono valide in  $\mathcal{P}'_3$  per  $V_1$ , a causa della forma del suo PR (2.14), che implica  $\ell_1(t) > r_i(t)$ ,  $\forall t$ . Si consideri, ad esempio, il caso in cui  $V_1$  e  $V_2$  vogliono aggiornare le rispettive PA, seguendo il criterio *etp* si identifica un  $p^*$  tale che  $p^* < \ell_i(t)$ . Il primo membro di (4.1) sarebbe negativo,  $\frac{p^* - \ell_1(t)}{\bar{v}_1} < 0$ , e non avrebbe più significato di tempo di percorrenza. Un ragionamento analogo si può fare per il secondo membro di (4.1), considerando l'aggiornamento di  $V_1$  e  $V_N$ . Il criterio *etp* viene formalizzato correttamente per un perimetro reale tramite la seguente espressione:

$$\frac{(p^* - \ell_i(t)) \bmod L}{\bar{v}_i} = \frac{(r_{i+1}(t) - p^*) \bmod L}{\bar{v}_{i+1}}, \quad (4.3)$$

dove  $(x) \bmod L$  è la funzione resto della divisione di  $x$  per  $L$ . A causa dell'introduzione della funzione resto, da (4.3) non è possibile ricavare un'espressione esplicita di  $p^*$  del tipo (4.2). Per aggirare quest'ostacolo si è sviluppato un metodo grafico per trovare il valore del punto  $p^*$ , prendendo spunto dall'algoritmo proposto in [4].

## 4.1.1 Metodo grafico

Per ogni videocamera  $V_i$ , con  $i = 1, \dots, N$ , si definisce la funzione  $\tau_i$ :

$$\tau_i : \mathbb{R}^+ \longrightarrow \mathbb{R}^+ : x \in D_i \longmapsto t \in \mathbb{R}^+ \quad (4.4)$$

che associa ad ogni punto  $x$  di  $D_i$ , un tempo  $t$  che ha significato di “tempo che impiega  $V_i$ , muovendosi alla velocità  $\bar{v}_i$  in senso antiorario lungo  $\mathcal{L}$ , per arrivare da  $\underline{d}_i$  a  $x$ ”. Poiché le videocamere per ipotesi si muovono sempre alla loro velocità massima, costante lungo il perimetro, allora le  $\tau_i$  sono funzioni strettamente crescenti, per  $i = 2, 3, \dots, N$ , e strettamente crescenti a tratti per  $i = 1$ . Nel piano distanza-tempo (d-t) ciascuna  $\tau_i$  è un segmento di retta di pendenza  $\bar{v}_i$ , compreso tra  $\underline{d}_i$  e  $\bar{d}_i$ , con  $t = 0$  per  $x = \underline{d}_i$  e  $t = \bar{v}_i \cdot |D_i|$  per  $x = \bar{d}_i$ . Da (2.14) si trova che  $\tau_1$  sul piano d-t è una coppia di segmenti di rette, entrambi di pendenza  $\bar{v}_1$ , compresi tra  $\underline{d}_1$  e  $L$  e tra  $0$  e  $\bar{d}_1$ , dove  $t = 0$  per  $x = \underline{d}_1$  e  $t = \bar{v}_1 \cdot (L - \underline{d}_1)$  per  $x = 0$ .

Si definiscono inoltre per ogni videocamera  $V_i$ , con  $i = 1, \dots, N$ , le funzioni  $\tau_{inv,i}$ :

$$\tau_{inv,i} : \mathbb{R}^+ \longrightarrow \mathbb{R}^+ : x \in D_i \longmapsto t \in \mathbb{R}^+ \quad (4.5)$$

che associa ad ogni punto  $x$  del PR  $D_i$ , un tempo  $t$  che ha significato di “tempo che impiega  $V_i$ , muovendosi alla velocità  $\bar{v}_i$  in senso orario lungo  $\mathcal{L}$ , per arrivare da  $\bar{d}_i$  a  $x$ ”. Le funzioni  $\tau_{inv,i}$  sono strettamente decrescenti, per  $i = 2, 3, \dots, N$ , strettamente decrescente a tratti per  $i = 1$ . Sono definite all'interno dei PR delle proprie videocamere e nel piano d-t si rappresentano sempre come segmenti di retta, ma con coefficienti angolari negativi e valori iniziali e finali invertiti rispetto alle  $\tau_i$ .

Per maggior chiarezza sulla forma delle funzioni si introduce l'esempio di Fig. 4.1. Nell'immagine (b) vengono rappresentate nel piano d-t le funzioni  $\tau_1$ ,  $\tau_2$ ,  $\tau_3$  e  $\tau_4$ , per le videocamere  $V_1$ ,  $V_2$ ,  $V_3$  e  $V_4$  di figura (a): la simmetria dello scenario porta le funzioni ad avere la stessa forma, di fatto sono la stessa funzione traslata di multipli di 2 sull'asse delle ascisse. La funzione  $\tau_1$ , consiste in due segmenti di retta separati, poiché è relativa a  $V_1$  che si trova a cavallo del punto iniziale di  $\mathcal{L}$ . Nell'immagine (b) vengono invece rappresentate, sempre nel piano d-t, le funzioni  $\tau_{inv,1}$ ,  $\tau_{inv,2}$ ,  $\tau_{inv,3}$ , e  $\tau_{inv,4}$  per le videocamere  $V_1$ ,  $V_2$ ,  $V_3$  e  $V_4$  di figura (a): presentano caratteristiche analoghe alle precedenti.

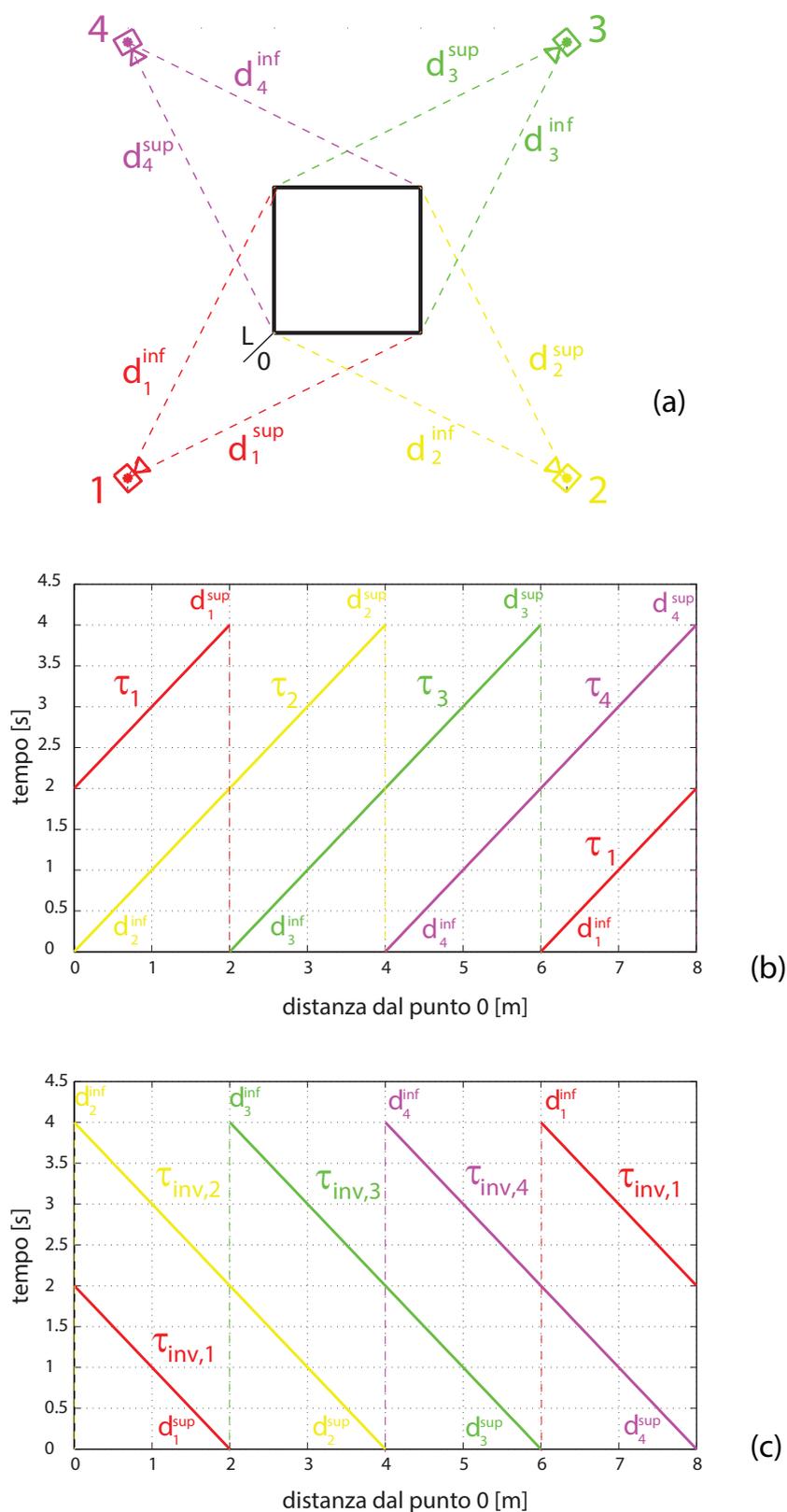


Fig. 4.1: (a): perimetro quadrato pattugliato da 4 videocamere PTZ con velocità massima uguale; (b): funzioni  $\tau_i$  per le videocamere in figura(a); (c): funzioni  $\tau_{inv,i}$  per le videocamere in figura(a);

Per individuare il punto  $p^*$  che soddisfa il criterio *etp* tra  $V_i$  e  $V_{i+1}$ , il metodo grafico di AP esegue i seguenti step:

1. traslazione verso il basso  $\tau_i$  di  $t_i$ , con  $t_i = (\ell_i(t) - \underline{d}_i) \bmod L \cdot \bar{v}_i$ , ottenendo la funzione  $\tau'_i$ ;
2. traslazione verso il basso  $\tau_{inv,i+1}$  di  $t_{i+1}$ , con  $t_{i+1} = (\bar{d}_i - r_{i+1}(t)) \bmod L \cdot \bar{v}_{i+1}$ , ottenendo la funzione  $\tau'_{inv,i+1}$ ;
3. calcolo del valore dell'ascissa  $x^*$  del punto d'intersezione tra  $\tau'_i$  e  $\tau'_{inv,i+1}$  (se esiste), ossia il punto  $x^* \in D_i \cap D_{i+1}$  tale che  $\tau'_i(x^*) = \tau'_{inv,i+1}(x^*)$ ; tale punto soddisfa il criterio *etp*:  $x^* = p^*$ .

In pratica, durante lo step 1, la quantità  $t_i$  viene ricavata prendendo l'ordinata del punto d'intersezione tra  $\tau_i$  e la retta verticale di equazione  $x = \ell_i(t)$ . Analogamente, durante lo step 2,  $t_{i+1}$  si ricava dall'intersezione tra  $\tau'_{inv,i+1}$  e la retta verticale di equazione  $x = r_{i+1}(t)$ . Nel punto  $x = \ell_i(t)$  la retta strettamente crescente  $\tau'_i$  interseca l'asse delle ascisse, quindi per  $x \geq \ell_i(t)$  i valori  $\tau'_i(x)$  hanno significato fisico di “tempo che impiega  $V_i$ , muovendosi alla velocità  $\bar{v}_i$  in senso antiorario lungo  $\mathcal{L}$ , per arrivare da  $\ell_i(t)$  a  $x$ ”. La retta  $\tau'_{inv,i+1}$  interseca l'asse delle ascisse in  $x = r_{i+1}(t)$  ma è strettamente decrescente, quindi per  $x \leq r_{i+1}(t)$  i valori della funzione  $\tau'_{inv,i+1}$  hanno significato fisico di “tempo che impiega  $V_{i+1}$ , muovendosi alla velocità  $\bar{v}_{i+1}$  in senso orario lungo  $\mathcal{L}$ , per arrivare da  $r_{i+1}(t)$  a  $x$ ”. Il punto  $p^*$  che soddisfa il criterio *etp* si trova eguagliando i tempi di percorrenza delle videocamere, quindi deve verificare  $\tau'_i(p^*) = \tau'_{inv,i+1}(p^*)$ , si tratta dello stesso punto  $x^*$  calcolato allo step 3, attraverso l'intersezione tra  $\tau'_i$  e  $\tau'_{inv,i+1}$ .

A seguito dei tre step del metodo grafico possono verificarsi due situazioni:

- (a) viene individuato il valore di  $p^*$ ;
- (b) il punto d'intersezione  $p^*$  non esiste;

Nel caso (a), va notato innanzitutto che il valore  $p^*$  trovato soddisfa:  $p^* \in D_i \cap D_{i+1}$ , poiché le funzioni  $\tau'_i$  e  $\tau'_{inv,i+1}$  sono sempre definite solo all'interno dei PR delle rispettive videocamere. Si procede come segue:

- $V_i$  pone  $r_i(t+1) = p^*$ ,  $\ell_i(t+1) = \ell_i(t)$  e aggiorna la sua PA a  $A_i(t+1) = [\ell_i(t) \ p^*] \in D_i$ ;
- $V_{i+1}$  pone  $\ell_{i+1}(t+1) = p^*$ ,  $r_{i+1}(t+1) = r_{i+1}(t)$  e aggiorna la sua PA a  $A_i(t+1) = [p^* \ r_{i+1}(t)] \in D_{i+1}$ ;

Se si usa il Gossip Simmetrico come protocollo di comunicazione, entrambi gli aggiornamenti vengono effettuati nella stessa iterazione: il valore dell'estremo destro di  $V_i$  e dell'estremo sinistro di  $V_{i+1}$  al tempo  $t+1$  coincidono:  $r_i(t+1) = \ell_{i+1}(t+1) = p^*$ . In questo caso si può assicurare che il nuovo partizionamento del perimetro soddisfa i vincoli fisici (3.2), i vincoli di allacciamento (3.5) e quindi anche il vincolo di copertura (3.6). Se invece si usa il Gossip Asimmetrico, uno solo dei due punti precedenti viene eseguito: una PA viene aggiornata mentre l'altra resta invariata. Il nuovo partizionamento soddisfa sempre i vincoli fisici (3.2), mentre non è detto che soddisfi quelli di allacciamento (3.5).

Nel caso **(b)** non esiste alcun punto appartenente all'intersezione  $D_i \cap D_{i+1}$  che soddisfa il criterio *etp*. Si procede calcolando  $T_{0,i}(t)$  e  $T_{0,i+1}(t)$ , introdotti in (2.11), che rappresentano il tempo necessario a  $V_i$  e  $V_{i+1}$  per muoversi da un estremo all'altro delle rispettive PA:

$$T_{0,i}(t) = \frac{(r_i(t) - \ell_i(t)) \bmod L}{\bar{v}_i}, \quad T_{0,i+1}(t) = \frac{(r_{i+1}(t) - \ell_{i+1}(t)) \bmod L}{\bar{v}_{i+1}}$$

Se  $T_{0,i}(t) < T_{0,i+1}$  significa che non c'è stata intersezione perché il punto  $p^*$  si trova fuori dal PR  $D_i$ , oltre il suo estremo destro  $\bar{d}_i$ . Viceversa, se  $T_{0,i}(t) > T_{0,i+1}$ ,  $p^*$  si trova fuori dal PR  $D_{i+1}$ , prima del suo estremo sinistro  $\underline{d}_{i+1}$ . Quindi si procede come segue:

- $V_i$  pone:  $r_i(t+1) = \begin{cases} \bar{d}_i, & \text{se } T_{0,i}(t) < T_{0,i+1}(t) \\ \underline{d}_{i+1}, & \text{se } T_{0,i}(t) > T_{0,i+1}(t) \end{cases}$   
e aggiorna la sua PA:  $A_i(t+1) = [\ell_i(t) \ r_i(t+1)]$ ;
- $V_{i+1}$  pone:  $\ell_{i+1}(t+1) = \begin{cases} \bar{d}_i, & \text{se } T_{0,i+1}(t) > T_{0,i}(t) \\ \underline{d}_{i+1}, & \text{se } T_{0,i+1}(t) < T_{0,i}(t) \end{cases}$   
e aggiorna la sua PA:  $A_{i+1}(t+1) = [\ell_{i+1}(t) \ r_{i+1}(t)]$ ;

Anche per questo caso valgono le stesse considerazioni fatte prima riguardo i protocolli di comunicazione considerati: se si utilizza Gossip Simmetrico entrambi i punti precedenti vengono realizzati durante la stessa iterazione, se invece si usa Gossip Asimmetrico uno solo tra i due avrà luogo.

L'applicazione iterativa delle precedenti procedure, qualunque sia il protocollo di comunicazione considerato, porta il sistema a raggiungere una partizione associata a una delle soluzioni ottime di  $\mathcal{P}'_3$ . Quindi, ove possibile, le PA individuate sono proporzionali alle velocità massime delle corrispondenti videocamere. Se questo non è vero significa che una tale situazione porterebbe a violare i vincoli fisici di qualche videocamera.

## 4.1.2 AP nella pratica

Di seguito verranno proposti alcuni esempi dell'applicazione di AP su scenari molto semplici al fine di chiarirne il funzionamento, descritto formalmente nella sezione precedente. Si consideri lo scenario di Fig. 2.3: un perimetro di forma quadrata con quattro videocamere, ciascuna posta in corrispondenza di un vertice del quadrato. Si mostrerà l'evoluzione di questo scenario durante due iterazioni successive di AP con Gossip Asimmetrico, concentrando l'analisi sulle elaborazioni grafiche che esegue l'algoritmo per trovare  $p^*$ . All'istante iniziale il sistema viene inizializzato ponendo le PA delle videocamere uguali ai rispettivi PR (Fig. 4.2).

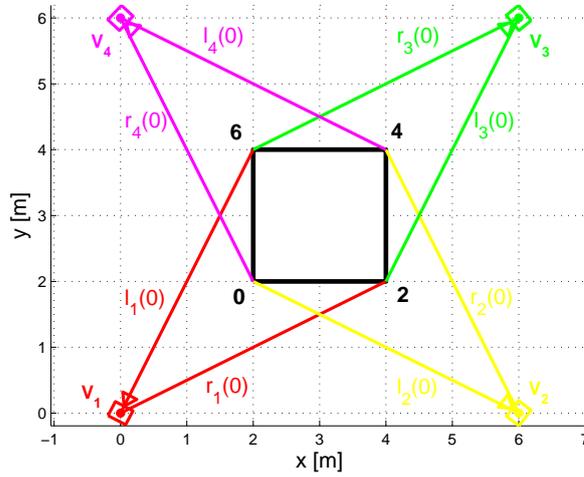


Fig. 4.2: Configurazione iniziale del sistema: la PA di ciascuna videocamera viene posta uguale al rispettivo PR. Le linee continue che partono dalle videocamere rappresentano gli estremi delle PA al tempo  $t = 0$ .

All'istante  $t = 0$  avviene una comunicazione tra  $V_3$  e  $V_2$ , in cui  $V_2$  trasmette i valori dei suoi estremi di pattugliamento a  $V_3$ , che invece aggiorna la sua PA elaborando le informazioni ricevute. Facendo riferimento a Fig. 4.3, si analizza il funzionamento del metodo grafico per la ricerca di  $p^*$ :  $\tau_2$  e  $\tau_{inv,3}$  vengono traslati verso il basso di  $t_2 = t_3 = 0$ , dato che  $\ell_2(0) = \underline{d}_2$  e  $r_3(0) = \bar{d}_3$ , quindi  $\tau_2 = \tau'_2$  e  $\tau_{inv,3} = \tau'_{inv,3}$ . Siccome le videocamere viaggiano alla stessa velocità,  $\bar{v}_2 = \bar{v}_3$ , l'intersezione avviene in  $d = 3$ , punto intermedio tra  $\ell_2(0)$  e  $r_3(0)$ . La configurazione del sistema a  $t = 1$  (Fig. 4.3) è identica a quella precedente, eccezion fatta per la PA di  $V_3$  che è stata aggiornata:  $A_i(1) = A_i(0)$  per  $i = 1, 2, 4$ , mentre  $A_3(1) = [\ell_3(0) p^*]$ .

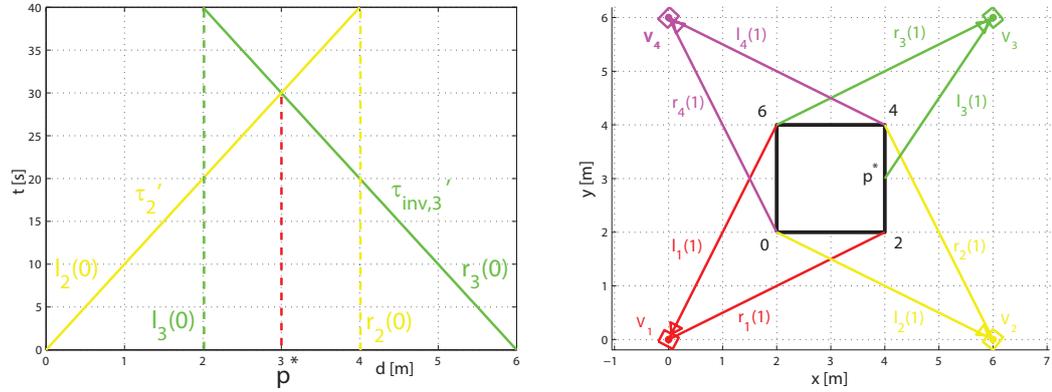


Fig. 4.3: La figura a sinistra mostra l'applicazione degli step del metodo grafico per la ricerca di  $p^*$ . La figura a destra rappresenta la configurazione a del sistema per  $t = 1$ , l'unica PA cambiata rispetto a prima è  $A_3$ .

All'istante  $t = 1$ ,  $V_4$  trasmette i valori dei suoi estremi a  $V_3$ , che aggiorna la sua PA. In figura Fig. 4.4 sono rappresentati i vari step del metodo grafico: la funzione  $\tau_3$  viene traslata verso il basso di  $t_3 = 10$ , mentre  $\tau_{inv,4}$  non subisce spostamenti poiché  $t_4 = 0$ ; l'intersezione tra  $\tau_3'$  e  $\tau_4'$  avviene in  $d = 5.5$ , che soddisfa il criterio *etp* per  $V_3$  e  $V_4$ . Lo scenario a  $t = 2$ , dopo l'ulteriore aggiornamento di  $A_3$  è rappresentato Fig4.4.

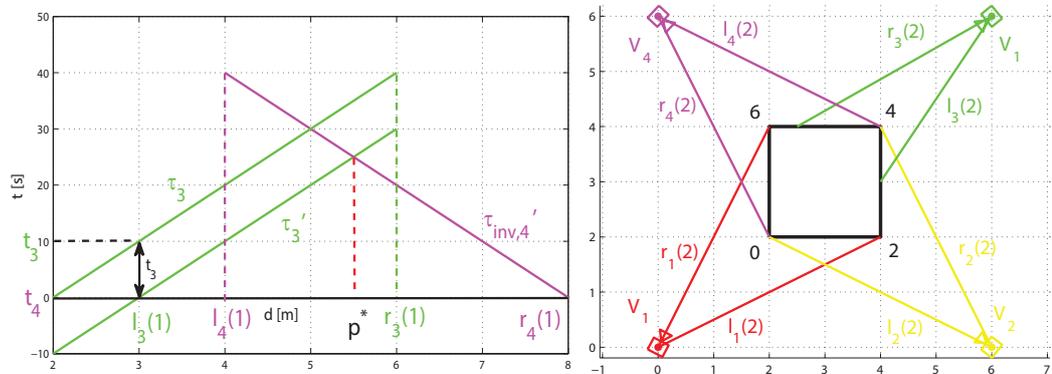


Fig. 4.4: La figura a sinistra mostra l'applicazione degli step del metodo grafico per la ricerca di  $p^*$ . La figura a destra rappresenta la configurazione a del sistema per  $t = 2$ , l'unica PA cambiata rispetto a prima è  $A_3$ .

Entrambe le iterazioni precedenti fanno riferimento al caso (a) descritto nella sezione precedente, in cui il metodo grafico individua un'intersezione. Nel seguito si propone un esempio del comportamento di AP nel caso (b), in cui nessun punto di intersezione  $p^*$  viene individuato. A tal fine si introduce un nuovo scenario, composto da un perimetro di forma rettangolare pattugliato

da quattro videocamere che hanno velocità massima  $\bar{v}$  uguale. La disposizione delle videocamere si può vedere in Fig. 4.5, dove viene rappresentato il sistema all'istante  $t = 0$ : le PA delle videocamere sono inizializzate ponendole pari ai rispettivi PR. Si assuma che il protocollo di comunicazione adottato dalle videocamere sia sempre il Gossip Asimmetrico.

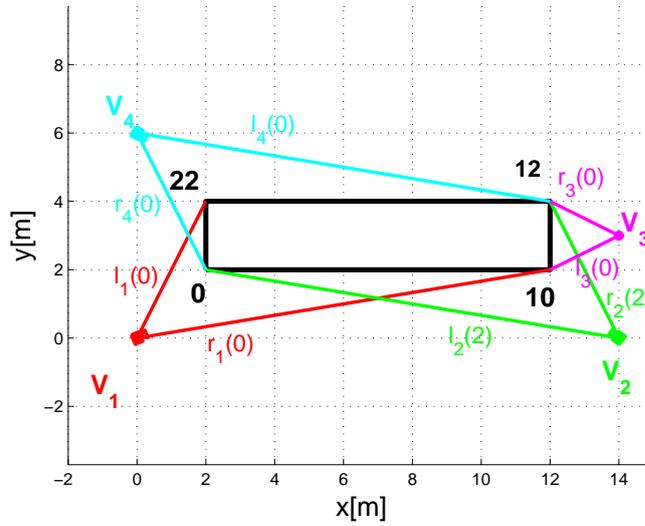


Fig. 4.5: Configurazione del sistema al tempo  $t = 0$ : le PA delle videocamere sono inizializzate ponendole pari ai rispettivi PR. Le linee continue che partono da ciascuna videocamera rappresentano gli estremi delle rispettive PA .

All'istante  $t = 0$  avviene una comunicazione tra  $V_3$  e  $V_2$ , in cui  $V_3$  trasmette i valori dei suoi estremi di pattugliamento a  $V_2$ . Applicando il metodo grafico,  $V_2$  cerca il punto che soddisfa il criterio *etp*: sia  $\tau_2$  che  $\tau_{inv,3}$  non vengono traslate verso il basso poiché  $\ell_2(0) = \underline{d}_2$  e  $r_3(0) = \bar{d}_3$ , quindi  $\tau_2 = \tau'_2$  e  $\tau_{inv,3} = \tau'_{inv,3}$ . Come si vede in Fig. 4.6  $\tau'_2$  e  $\tau'_{inv,3}$  non si intersecano, perché non c'è nessun punto appartenente a  $D_2 \cap D_3$  che verifica il criterio *etp*: si cade quindi nel caso (b). Come da prassi  $V_2$  confronta il suo periodo di pattugliamento  $T_{0,2} = 12s$ , con quello di  $V_3$ , pari a  $T_{0,3} = 2s$ . Dato che risulta  $T_{0,2} > T_{0,3}$  l'intersezione non ha avuto luogo perché il punto  $p^*$  che verifica il criterio *etp* appartiene a  $D_2$  ma non a  $D_3$ . Di conseguenza  $V_2$  pone  $r_2(1) = \underline{d}_3 = 10m$  e aggiorna la sua PA a  $A_3(1) = [\ell_2(0) \ \underline{d}_3] = [0 \ 10]$ .

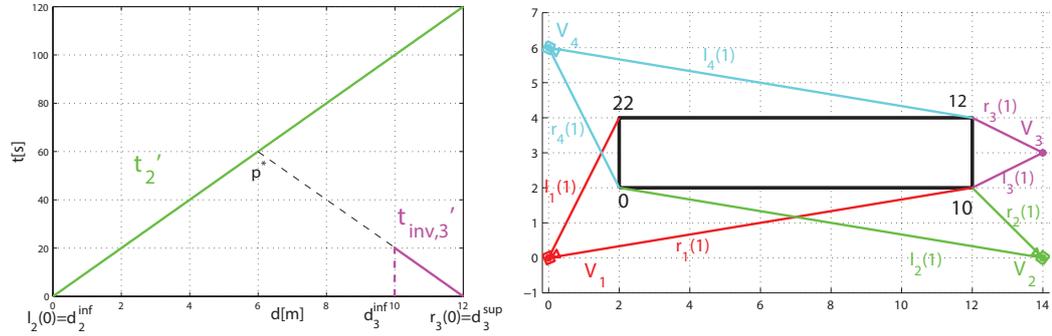


Fig. 4.6: La figura a sinistra mostra l'applicazione degli step del metodo grafico per la ricerca di  $p^*$ . La figura a destra rappresenta la configurazione a del sistema per  $t = 1$ , l'unica PA cambiata rispetto a prima è  $A_2$ .

## 4.2 Algoritmo Perimetrico - Angolo d'Incidenza ( $AP\Psi$ )

$AP\Psi$  è stato ideato per realizzare la partizione ottima di un perimetro  $\mathcal{L}$  secondo il problema  $\mathcal{P}_4$ . In essa le PA dipendono non solo dalle velocità delle rispettive videocamere, ma anche dalla perpendicolarità della loro visuale, più formalmente dal valore dell'area sottesa alla funzione angolo d'incidenza  $\psi$ . Dato che l'obiettivo principale del partizionamento rimane la minimizzazione di  $T_{lag}$ , la prima dipendenza è dominante mentre la seconda assume un'importanza secondaria, regolabile tramite il parametro  $C$  in (2.23). Diversamente da  $AP$ , quest'algoritmo converge sempre verso una sola partizione, corrispondente all'unica soluzione ottima del problema  $\mathcal{P}_4$ . Nel seguito se ne descrive dettagliatamente il funzionamento.

Consideriamo una generica comunicazione all'istante  $t$  tra due videocamere vicine,  $V_i$  e  $V_{i+1}$ , e assumiamo che ci sia una sovrapposizione tra i rispettivi PR,  $D_i$  e  $D_{i+1}$ . Se ipotizziamo che il protocollo di comunicazione utilizzato è il Gossip Asimmetrico, eventualmente solo una delle due videocamere aggiornerà la sua PA:

- se  $V_i$  è la videocamera che trasmette e  $V_{i+1}$  quella che riceve, allora  $V_{i+1}$  aggiornerà l'estremo sinistro  $\ell_{i+1}(t)$  della sua PA;
- se, viceversa,  $V_{i+1}$  trasmette e  $V_i$  riceve, allora  $V_i$  aggiornerà l'estremo destro  $r_i(t)$  della sua PA.

Se si utilizza il Gossip Simmetrico entrambi i punti precedenti verranno eseguiti durante la stessa iterazione. In entrambi i casi si vuole che il valore aggiornato  $p^*$  dell'estremo della nuova area di pattugliamento soddisfi tre criteri:

1. criterio *etp*,
2. criterio *Centramento*,
3. criterio *Area*.

Il primo criterio è già stato esaminato: si tratta dell'unico su cui si basa l'aggiornamento delle PA in AP e matematicamente si esprime tramite (4.3). In APΨ assume carattere dominante rispetto agli altri due, poiché è responsabile della convergenza del sistema verso partizionamenti in cui le PA sono proporzionali alle velocità delle rispettive videocamere, in altri termini della minimizzazione di  $T_{lag}$ . I rimanenti due criteri sono responsabili della convergenza del sistema verso il partizionamento in cui le PA minimizzano  $P_{\Psi}$  in (2.23). Ricordando quanto osservato nel capitolo 2 sulla forma delle funzioni  $|\psi_i|$ , sappiamo che ciascuna PA in questione presenta le seguenti caratteristiche:

- è centrata attorno a  $d_{min}$ , in cui si ha il minimo di  $|\psi_i|$ ;
- ha una lunghezza proporzionale all'area sottesa alla curva  $|\psi_i|$  e compresa tra i suoi estremi.

Come si può facilmente intuire, il criterio *Centramento* e quello di *Area* sono legati rispettivamente alla prima e alla seconda delle caratteristiche elencate. Nel seguito si fornisce una descrizione formale per entrambi.

#### 4.2.1 Criterio Centramento

“La differenza tra l'angolo visuale d'incidenza di  $V_i$  in  $p_{centr}^*$  e quello in  $\ell_i(t)$  è uguale alla differenza tra l'angolo visuale d'incidenza di  $V_{i+1}$  in  $p_{centr}^*$  e quello in  $r_{i+1}(t)$ ”

Formalmente  $p^*$  deve soddisfare la condizione:

$$|\psi_i(p_{centr}^*)| - |\psi_i(\ell_i(t))| = |\psi_{i+1}(p_{centr}^*)| - |\psi_{i+1}(r_{i+1}(t))| \quad (4.6)$$

Le funzioni angolo visuale d'incidenza  $|\psi_i|$  sono molto eterogenee poiché dipendono sia dalla forma del perimetro sia dalla topologia degli agenti della rete, e non se ne conosce l'espressione analitica. Perciò il modo più semplice per individuare  $p_{centr}^*$  è attraverso elaborazioni grafiche analoghe a quelle utilizzate per trovare  $p^*$  del criterio *etp*. Per determinare  $p_{centr}^*$  al tempo  $t$  tra le videocamere  $V_i$  e  $V_{i+1}$  si seguono i seguenti step:

1. traslazione in basso  $|\psi_i|$  di  $|\psi_i(\ell_i(t))|$ , ottenendo  $|\psi_i|'$ ;
2. traslazione in basso  $|\psi_{i+1}|$  di  $|\psi_{i+1}(r_{i+1}(t))|$ , ottenendo  $|\psi_{i+1}|'$ ;
3. trovo l'ascissa  $x^*$  del punto d'intersezione tra  $|\psi_i|'$  e  $|\psi_{i+1}|'$ , ossia il valore  $x^* \in D_i \cap D_{i+1}$  tale che  $|\psi_i(x^*)|' = |\psi_{i+1}(x^*)|'$ ;  $x^*$  soddisfa il criterio *Centramento*:  $x^* = p_{centr}^*$ .

Dato che l'espressione esplicita di  $|\psi_i|$  e  $|\psi_{i+1}|$  non è nota,  $|\psi_i(\ell_i(t))|$  si determina prendendo l'ordinata del punto d'intersezione tra la funzione  $|\psi_i|$  e la retta verticale di equazione  $x = \ell_i(t)$ . Analogamente,  $|\psi_{i+1}(r_{i+1}(t))|$  si determina prendendo l'ordinata del punto d'intersezione tra  $|\psi_{i+1}|$  e la retta  $x = r_{i+1}(t)$ . Va osservato che potrebbe non esserci alcun punto di intersezione tra  $|\psi_i|'$  e  $|\psi_{i+1}|'$ . Per il momento non ci si preoccupa di questo fatto, perché lo scopo di questa sezione è solo di mostrare come si procede per individuare  $p_{centr}^*$ . La gestione dell'aggiornamento delle PA viene definita successivamente all'unione dei tre criteri.

Si consideri ad esempio, l'evoluzione dello scenario di Fig. 4.2 proposta in sezione 1.1.2. Si vogliono confrontare, nelle due iterazioni ipotizzate, i  $p^*$  individuati dal criterio *etp* con quelli del criterio *Centramento*. Il valore di  $p_{centr}^*$  tra  $V_1$  e  $V_2$  al tempo  $t = 0$ , si determina applicando il procedimento grafico descritto precedentemente, rappresentato in Fig. 4.7. Risulta  $p_{centr}^* = p_{etp}^* = 3$ , infatti tale valore soddisfa sia (4.3) che (4.6).

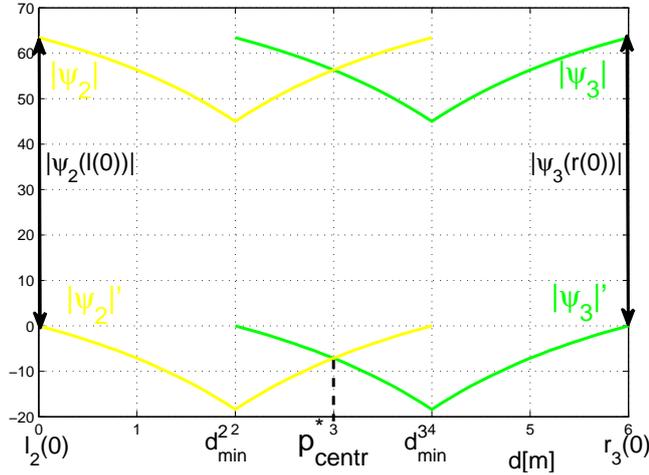


Fig. 4.7: Elaborazioni grafiche per individuare  $p_{centr}^*$  tra  $V_2$  e  $V_3$  di Fig. 4.2: il valore individuato è uguale a quello che soddisfa il criterio *etp*, per confrontare i due criteri si guardi Fig. 4.3.

Se però si considera la seconda iterazione, che coinvolge  $V_3$  e  $V_4$  al tempo  $t = 1$ , si verifica che  $p_{centr}^* = 4.55 \neq p_{etp}^* = 5.5$ , come mostra Fig. 4.8.

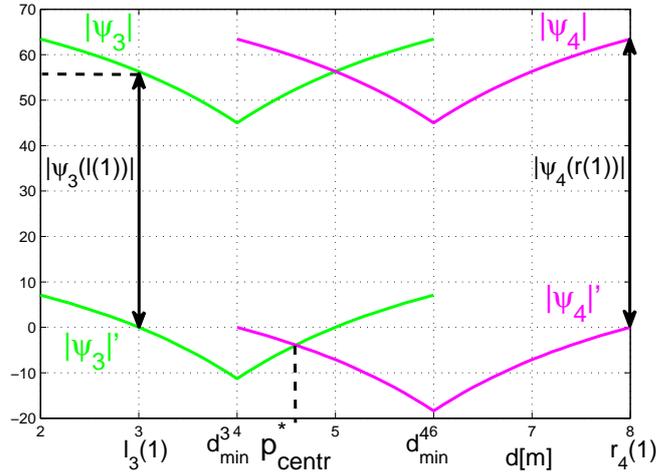


Fig. 4.8: Elaborazioni grafiche per individuare  $p_{centr}^*$  tra  $V_4$  e  $V_3$  di Fig. 4.3: in questo caso il valore trovato differisce da quello del criterio  $etp$ , si confronti con Fig. 4.4.

I criteri  $etp$  e  $centramento$  spingono il sistema a evolversi in direzioni diverse: il primo verso partizionamenti con PA proporzionali alle velocità delle rispettive videocamere, il secondo verso partizionamenti in cui le PA sono il più possibile centrate attorno a  $d_{min}$ , senza alcun controllo sulla loro dimensione. L'idea che sta alla base di AP $\Psi$  è di unire questi due criteri di aggiornamento in modo da far convergere il sistema verso un partizionamento con PA che siano sia centrate che proporzionali alle velocità delle rispettive videocamere.

#### 4.2.2 Criterio Area

“ L'area sottesa alla curva  $|\psi_i|$  da  $l_i(t)$  a  $p^*$  è uguale all'area sottesa alla curva  $|\psi_{i+1}|$  da  $p^*$  a  $r_{i+1}(t)$  ”

Formalmente  $p_{area}^*$  deve soddisfare:

$$\int_{l_i(t)}^{p_{area}^*} |\psi_i(s)| ds = \int_{p_{area}^*}^{r_{i+1}(t)} |\psi_{i+1}(s)| ds \quad (4.7)$$

Per poter disporre di un metodo per individuare semplicemente  $p_{area}^*$  si introducono per ogni videocamera della rete la funzione area  $\alpha_i$ ,

$$\alpha_i : D_i \times \mathbb{N} \longrightarrow \mathcal{R}^+ : (x, t) \longmapsto \alpha_i(x, t) = \begin{cases} 0, & x \bmod L < \ell_i(t) \\ \int_{\ell_i(t)}^x |\psi_i(s)| ds, & x \bmod L \geq \ell_i(t) \end{cases} \quad (4.8)$$

che associa ad ogni punto  $x$  di  $D_i$  l'area sottesa alla curva  $|\psi_i|$  compresa tra  $\ell_i(t)$  e  $x$  stesso, se  $x$  è prima di  $\ell_i(t)$  la funzione assume valore nullo. Si definisce inoltre per ciascuna videocamera  $V_i$  la funzione area inversa  $\alpha_{inv,i}$ :

$$\alpha_{i,inv} : D_i \times \mathbb{N} \longrightarrow \mathcal{R}^+ : (x, t) \longmapsto \alpha_{i,inv}(x, t) = \begin{cases} \int_x^{r_i(t)} |\psi_i(s)| ds, & x \bmod L \leq r_i(t) \\ 0, & x \bmod L > r_i(t) \end{cases} \quad (4.9)$$

che associa ad ogni punto  $x$  di  $D_i$  l'area sottesa alla curva  $|\psi_i|$  compresa tra  $x$  stesso e  $r_i(t)$ , se  $x$  è successivo a  $r_i(t)$  assume valore nullo. Entrambe le funzioni non restano costanti durante le iterazioni dell'algoritmo, poiché dipendono dagli estremi delle PA, che possono cambiare a seguito di un aggiornamento. Ogni volta che una PA viene aggiornata è necessario ricalcolare almeno una delle funzioni area: se cambia  $\ell_i$  bisogna ricalcolare  $\alpha_i$ , se invece cambia  $r_i$  bisogna ricalcolare  $\alpha_{i,inv}$ .

La (4.7) si può riscrivere come:

$$\alpha_i(p_{area}^*, t) = \alpha_{i+1,inv}(p_{area}^*, t) \quad (4.10)$$

Il metodo grafico per calcolare  $p_{area}^*$  è più semplice rispetto agli altri due analizzati: è sufficiente trovare il valore dell'ascissa del punto d'intersezione tra  $\alpha_i$  ed  $\alpha_{i+1}$ . Tale punto assicura (4.7), ossia che le aree sottese a  $|\psi_i|$  e  $|\psi_{i+1}|$  calcolate entro le rispettive PA siano identiche.

Utilizzando solo questo criterio per l'aggiornamento degli estremi delle PA, il sistema converge verso partizionamenti in cui  $\int_{\ell_i}^{r_i} |\psi_i(s)| ds$  sono uguali  $\forall i \in \{1, 2, \dots, N\}$ . Il fatto che queste aree siano uguali per ogni videocamera non assicura la minimizzazione di  $P_\Psi$ , che invece si ottiene combinando gli effetti di questo criterio con quelli del criterio *Centramento*: quest'ultimo viene utilizzato per centrare le PA attorno al loro punto  $d_{min}$  con visuale più perpendicolare su  $\mathcal{L}$  ( $|\psi|$  minimo), il primo invece opera una modulazione dell'ampiezza delle PA, espandendo quelle con visuali migliori (piccole aree sottese) e riducendo quelle con visuali oblique (grandi aree sottese).

Nel seguito, facendo riferimento all'evoluzione dello scenario di Fig. 4.2 descritto in sezione 1.1.2, si mostrano quali siano i punti  $p_{area}^*$  individuati dal criterio *area*, nei due aggiornamenti ipotizzati.

Il valore di  $p_{area}^*$  tra  $V_2$  e  $V_3$  a  $t = 0$ , si ottiene tramite l'intersezione delle

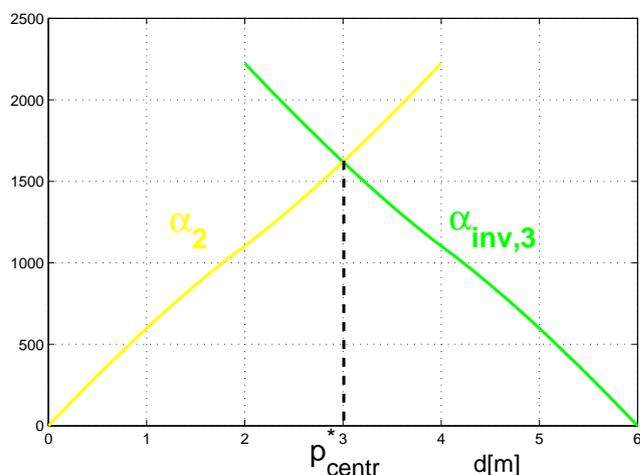


Fig. 4.9: Elaborazioni grafiche per individuare  $p_{area}^*$  tra  $V_2$  e  $V_3$  di Fig. 4.2: il valore individuato è uguale a quello che soddisfa il criterio  $etp$  e di *Centramento*.

curve  $\alpha_2(\cdot, 0)$  e  $\alpha_{inv,3}(\cdot, 0)$ , rappresentata in Fig. 4.9.

L'intersezione delle curve  $\alpha_3(\cdot, 1)$  e  $\alpha_{inv,4}(\cdot, 1)$  per individuare  $p_{area}^*$  tra  $V_3$  e  $V_4$  al tempo  $t = 1$ , è rappresentata in Fig. 4.10. Il fatto che si verifichi  $p_{area}^* \neq p_{etp}^* \neq p_{centr}^*$  è una prova che i tre criteri spingono il sistema verso equilibri in generale diversi.

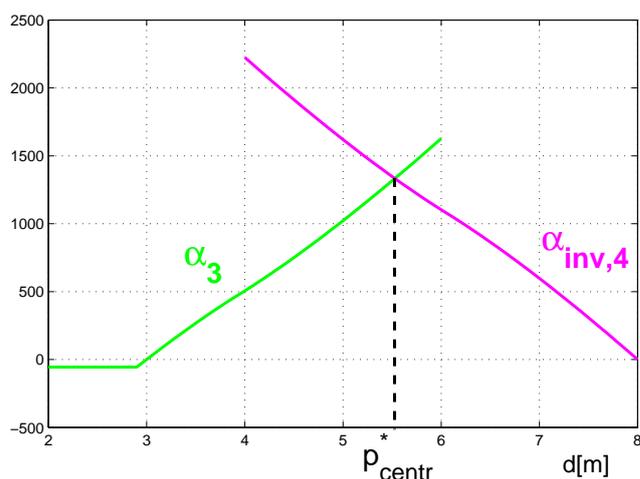


Fig. 4.10: Elaborazioni grafiche per individuare  $p_{area}^*$  tra  $V_3$  e  $V_4$  di Fig. 4.3: il valore individuato differisce sia da quello che soddisfa il criterio  $etp$  sia da quello che soddisfa il criterio *Centramento*.

## 4.2.3 Metodo Grafico

Adesso che tutti i criteri di aggiornamento sono stati analizzati, si ricapitolano in breve gli effetti di ciascuno sull'evoluzione del sistema:

- criterio *etp*: fa convergere il sistema verso partizionamenti che sono soluzioni di  $\mathcal{P}'_3$ , in cui le PA sono proporzionali alle velocità delle rispettive videocamere e  $T_{lag}$  viene minimizzato;
- criterio *Centramento*: fa convergere il sistema verso partizionamenti in cui le PA sono centrate attorno ai loro punti con visuale più perpendicolare sul perimetro; non esercita alcun controllo sulla lunghezza delle PA;
- criterio *Area*: fa convergere il sistema verso partizionamenti in cui l'area  $\int_{\ell_i}^{r_i} |\psi_i(s)| ds$  è identica per ogni videocamera  $V_i$ ; non centra le PA né le modula secondo le rispettive velocità.

L'idea che sta alla base  $\mathbf{AP}\Psi$  è di utilizzare tutti e tre gli effetti di convergenza elencati per raggiungere il partizionamento ottimo del problema  $\mathcal{P}_4$ . Infatti il primo criterio assicura la minimizzazione di  $f_3$  e quindi di  $T_{lag}$ , che rimane l'obiettivo primario anche per  $\mathcal{P}_4$ . Gli altri due criteri, presi assieme, garantiscono la minimizzazione di  $P_\Psi$  attraverso due effetti: centramento delle PA attorno al punto in cui hanno la visuale migliore di  $\mathcal{L}$  e modulazione della loro lunghezza (ampliamento delle PA con visuali complessivamente migliori e riduzione di quelle con visuali oblique).

Per ottenere tutti gli effetti descritti all'unisono si prende come valore  $p^*$  tra  $V_i$  e  $V_{i+1}$ , una combinazione di  $p_{etp}^*$ ,  $p_{centr}^*$  e  $p_{area}^*$ . Formalmente si prende  $p^*$  che soddisfa:

$$\begin{aligned} & \frac{(p^* - \ell_i(t)) \bmod L}{C_{etp} \cdot \bar{v}_i} + \frac{|\psi_i(p^*)| - |\psi_i(\ell_i(t))|}{C_{centr}} + \frac{\int_{\ell_i(t)}^{p^*} |\psi_i(s)| ds}{C_{area}} = \\ & \frac{(r_{i+1}(t) - p^*) \bmod L}{C_{etp} \cdot \bar{v}_{i+1}} + \frac{|\psi_{i+1}(p^*)| - |\psi_{i+1}(r_{i+1}(t))|}{C_{centr}} + \frac{\int_{p^*}^{r_{i+1}} |\psi_{i+1}(s)| ds}{C_{area}} \end{aligned} \quad (4.11)$$

dove  $C_{etp}$ ,  $C_{centr}$  e  $C_{area}$  sono delle costanti definite arbitrariamente. Ciascuno dei tre termini in (4.11) si misura con una grandezza fisica diversa: il primo termine rappresenta un tempo e si misura in secondi, il secondo rappresenta un angolo e si misura in gradi, l'ultimo invece rappresenta un'area. Di conseguenza la scala di valori su cui variano è molto eterogenea; per rendersene conto è sufficiente confrontare gli ordini di grandezza degli assi  $y$  delle Fig 4.3, 4.7 e 4.9. Le costanti  $C_{centr}$  e  $C_{area}$  vengono introdotte per

normalizzare il termine di *Centramento* e quello di *Area* allo stesso ordine di grandezza del termine *etp*. Siccome si desidera che il termine *etp* sia quello dominante nel sistema, si scelgono  $C_{centr}$  e  $C_{area}$  in modo che i rimanenti siano circa il 10% del primo.

Ricavare un'espressione esplicita di  $p^*$  da (4.11) non è possibile. Perciò lo si individua, come oramai si è abituati, attraverso un procedimento grafico, il cui funzionamento è descritto nel seguito:

1. somma delle le funzioni  $\tau_i(\cdot)$ ,  $|\psi_i(\cdot)|$  e  $\alpha_i(\cdot, t)$ , tutte definite in  $D_i$ , ottenendo:

$$\gamma_{i,t}(\cdot) = \frac{\tau_i(\cdot)}{C_{etp}} + \frac{|\psi_i(\cdot)|}{C_{centr}} + \frac{\alpha_i(\cdot, t)}{C_{area}} ;$$

2. somma delle funzioni  $\tau_{inv,i+1}(\cdot)$ ,  $|\psi_{i+1}(\cdot)|$  e  $\alpha_{inv,i+1}(\cdot, t)$ , tutte definite in  $D_i$ , ottenendo:

$$\gamma_{inv,i,t}(\cdot) = \frac{\tau_{inv,i+1}(\cdot)}{C_{etp}} + \frac{|\psi_{i+1}(\cdot)|}{C_{centr}} + \frac{\alpha_{inv,i+1}(\cdot, t)}{C_{area}} ;$$

3. traslazione in basso  $\gamma_{i,t}$  di  $g_i = \gamma_{i,t}(\ell_i(t))$ , ottenendo  $\gamma'_{i,t}$  ;
4. traslazione in basso  $\gamma_{inv,i,t}$  di  $g_{i+1} = \gamma_{inv,i,t}(r_{i+1}(t))$ , ottenendo  $\gamma'_{inv,i,t}$  ;
5. calcolo dell'ascissa  $x^*$  del punto d'intersezione tra  $\gamma'_{i,t}$  e  $\gamma'_{inv,i,t}$ ; tale valore verifica (4.11):  $x^* = p^*$ .

A seguito delle elaborazioni grafiche possono verificarsi due casi:

- (a) viene individuato il valore di  $p^*$ ;
- (b) il punto d'intersezione  $p^*$  non esiste;

A questo punto la gestione dell'aggiornamento delle PA nei due casi considerati è in tutto e per tutto identica a quella di AP. Si rimanda quindi alla sezione 1.1.1 per una descrizione esaustiva della gestione dell'aggiornamento nei due casi.



## 5. SIMULAZIONI IN MATLAB

Una parte consistente del lavoro di tesi è stato lo sviluppo di un programma Matlab per simulare il problema di pattugliamento di un perimetro generico con una rete di videocamere PTZ. In quest'ambiente virtuale si può simulare l'applicazione di un algoritmo di partizionamento su un qualsiasi scenario. Le informazioni ricavate dal test degli algoritmi di partizionamento progettati assumono un'importanza rilevante poiché:

- supportano le ipotesi di convergenza degli algoritmi;
- forniscono una prova che la configurazione finale raggiunta dagli algoritmi sia una soluzione del problema di ottimizzazione considerato ( $\mathcal{P}'_3$  o  $\mathcal{P}_4$ );
- danno indicazioni riguardo le performance degli algoritmi, ossia la rapidità di convergenza alla soluzione finale.

Nella prima sezione si presentano i risultati dell'applicazione di AP su alcuni semplici scenari, con il triplice obiettivo di comprenderne meglio il funzionamento, il tipo di partizionamento ottimo trovato e di individuarne le performance di convergenza con i vari protocolli di comunicazione. In maniera analoga, nella seconda sezione vengono esaminate nel dettaglio alcune simulazioni di AP $\Psi$ , sugli stessi scenari precedentemente considerati, in modo da poter confrontare i risultati dei due algoritmi. Infine nell'ultima sezione si introduce uno scenario realistico, caratterizzato da un perimetro esteso e un gran numero di videocamere, su cui vengono testati AP e AP $\Psi$ , per ricavare indicazioni più affidabili sia sulla convergenza che sulle performance.

### 5.1 Simulazioni di AP

Per meglio comprendere come funziona AP si presentano i risultati di alcune simulazioni dell'algoritmo in Matlab. Per iniziare si prende in considerazione lo scenario di Fig. 2.3 in cui un perimetro di forma quadrata deve essere pattugliato da 4 videocamere PTZ, disposte in corrispondenza dei vertici. Si assume che tutte le videocamere abbiano velocità massima uguale:

$\bar{v}_1 = \bar{v}_2 = \bar{v}_3 = \bar{v}_4 = 10 \text{ m/s}$ , e che il protocollo di comunicazione utilizzato sia Gossip Simmetrico. Il sistema viene inizializzato secondo le regole stabilite nel capitolo 2, portando le PA delle videocamere a coincidere con i rispettivi PR, come in Fig. 4.2. Poi viene applicato AP sulla struttura, il partizionamento ottimo  $\xi_{AP1}^*$  individuato è rappresentato in Fig. 5.1.

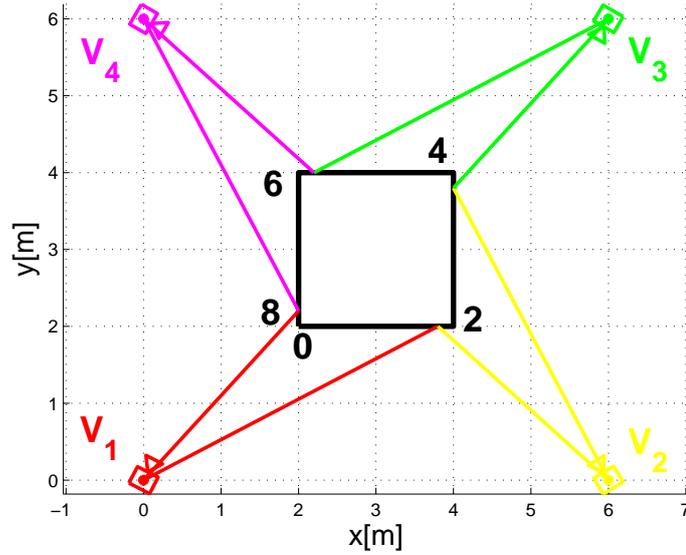


Fig. 5.1: Partizione ottima trovata dalla prima simulazione di AP sullo scenario di Fig. 2.3.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell) - \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	2	0.4	3.629	283.5965
2	2	0.4	3.629	283.5965
3	2	0.4	3.629	283.5965
4	2	0.4	3.629	283.5965

Tab. 5.1: Dati relativi alla partizione ottima di Fig. 5.1.

In tabella 5.1 vengono riportati alcuni dati relativi al partizionamento ottimo di Fig. 5.1, per ogni PA  $A_i$  vengono calcolati: lunghezza, periodo  $T_0$  di pattugliamento della relativa videocamera<sup>1</sup>, valore assoluto della differenza tra gli angoli d'incidenza visuali  $\psi$  ai suoi estremi, infine area del tipo 4.7,

<sup>1</sup> ottenuto da  $T_{0,i} = \frac{2(\ell_i - r_i) \bmod L}{\bar{v}_i}$ .

sottesa alla funzione  $|\psi_i|$ . Nel capitolo precedente si è detto che le partizioni ottime individuate da AP presentano PA di lunghezza proporzionale alle velocità delle rispettive videocamere. In questo caso, dato che le velocità delle videocamere sono uguali, anche le PA hanno lunghezza uguale, come si può notare osservando la seconda colonna di 5.1. La somma delle PA è 8, pari alla lunghezza totale del perimetro, inoltre i vincoli fisici (3.2), di allacciamento (3.5) e di copertura (3.6) sono soddisfatti. D'altronde l'utilizzo del Gossip Simmetrico come protocollo di comunicazione assicura ciò non solo per la partizione finale ma per ogni configurazione assunta dal sistema durante le iterazioni dell'algoritmo. Ricordando che AP individua una soluzione ottima del problema  $\mathcal{P}'_3$ , si scrive il valore della funzione costo  $f_3$  per la partizione  $\xi_{AP1}^*$  in Fig. 5.1:

$$f_3(\xi_{AP1}^*) = \sum_{i=1}^N \frac{((\ell_i^* - r_i^*) \bmod L)^2}{\bar{v}_i} = 1.6 \quad (5.1)$$

Ogni altro partizionamento del sistema ha valore della funzione costo  $f_3$  maggiore o uguale a (5.1) e quindi anche  $T_{lag}$  maggiore o uguale. Per sapere il numero di soluzioni ottime secondo  $\mathcal{P}'_3$  per questo scenario, si applica la *Proposizione I* (definita nel capitolo 2): dato che in  $\xi_{AP1}^*$  nessuna PA ha un estremo corrispondente con l'estremo del relativo PR allora esistono infinite partizioni ottime che minimizzano  $f_3$ . Infatti effettuando una seconda simulazione di AP si arriva ad una configurazione finale diversa, rappresentata in Fig. 5.2.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	2	0.4	2.229	275.0295
2	2	0.4	2.229	275.0295
3	2	0.4	2.229	275.0295
4	2	0.4	2.229	275.0295

Tab. 5.2: Dati relativi alla partizione ottima di Fig. 5.2.

La partizione  $\xi_{AP2}^*$  trovata presenta caratteristiche analoghe alla precedente: le PA hanno tutte lunghezza  $2m$  e il valore della funzione costo rimane invariato rispetto a prima,  $f_3(\xi_{AP2}) = f_3(\xi_{AP1}^*) = 1.6$ , si tratta quindi di un'altra soluzione ottima di  $\mathcal{P}'_3$ . In generale ad ogni simulazione di AP si individua una partizione in cui gli estremi delle PA si trovano in una posizione diversa, ciò che le accomuna tutte è la lunghezza delle PA che assicura la minimizzazione di  $f_3$  e quindi di  $T_{lag}$ . La casualità della partizione ottima individuata è legata

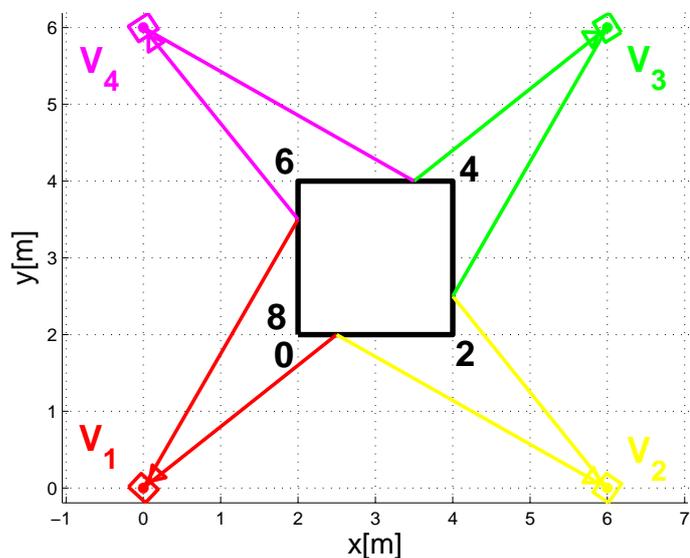


Fig. 5.2: Partizione ottima trovata dalla seconda simulazione di AP sullo scenario di Fig. 2.3.

al protocollo di comunicazione che ad ogni iterazione sceglie casualmente la coppia di videocamere che aggiornano la propria PA.

Su un perimetro di forma regolare e con un numero ridotto di videocamere che lo pattugliano, come nel caso esaminato, la convergenza dell'algoritmo AP è piuttosto rapida con qualsiasi protocollo di comunicazione. In Fig. 5.3 e Fig. 5.4 si mostra l'evoluzione temporale delle lunghezze delle PA delle videocamere durante due diverse simulazioni di AP sul solito scenario: nella prima si è utilizzato il Gossip Simmetrico come protocollo di comunicazione, nella seconda si è utilizzato il Gossip Asimmetrico. Ogni linea in Fig. 5.3 rappresenta l'ampiezza di una PA durante le iterazioni dell'algoritmo AP con Gossip Simmetrico: all'iterazione 0, quando il sistema viene inizializzato, hanno tutte ampiezza  $4m$  pari quella dei rispettivi PR, dopo 40 iterazioni, quando il sistema raggiunge la partizione ottima, convergono tutte al valore  $2m$ . Una volta raggiunta questa configurazione il sistema entra in una fase statica in cui continuano ad esserci comunicazioni tra le videocamere ma nessuna di esse parta all'aggiornamento di una PA.

Utilizzando il Gossip Asimmetrico la convergenza risulta decisamente più lenta, come si può vedere in Fig. 5.4 : la configurazione finale viene raggiunta dopo 150 iterazioni dell'algoritmo. I dati ottenuti non sono assoluti ma dipendono dalla specifica simulazione cui si fa riferimento. In generale diverse simulazioni di AP fanno convergere il sistema verso partizioni ottime

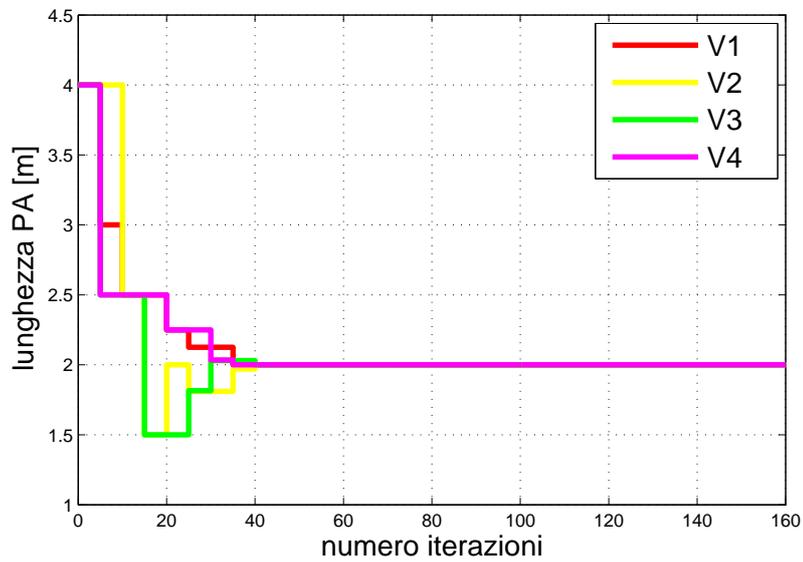


Fig. 5.3: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Simmetrico sullo scenario di Fig. 2.3.

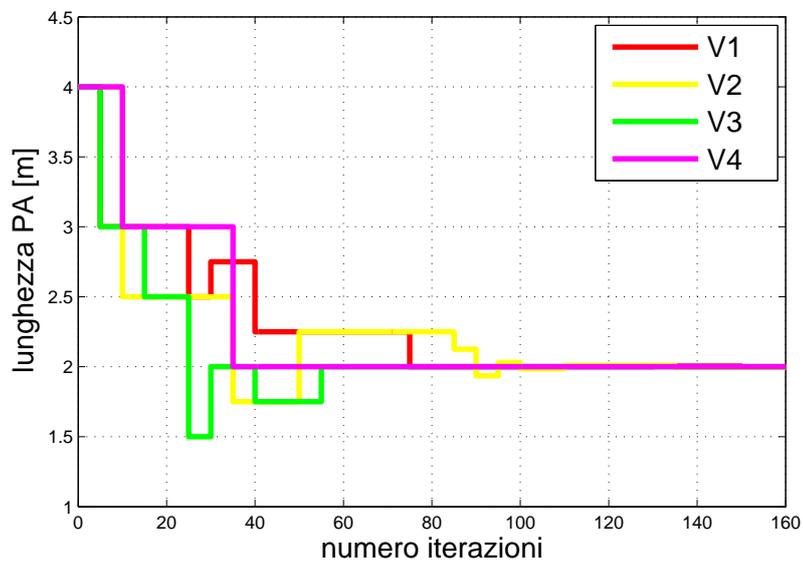


Fig. 5.4: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Asimmetrico sullo scenario di Fig. 2.3.

differenti in tempi differenti, sebbene il protocollo di comunicazione utilizzato rimanga invariato.

Si consideri ora uno scenario più complesso, in cui il solito perimetro di forma quadrata viene pattugliato da 8 videocamere PTZ, il doppio dell'esempio precedente. Si assuma che tutte le videocamere abbiano la stessa velocità massima,  $\bar{v}_i = 10 \text{ m/s}$ , e che il protocollo di comunicazione utilizzato sia il Gossip Simmetrico. Si vuole analizzare il comportamento di AP su questa nuova struttura, per vedere come cambiano i parametri della soluzione ottima individuata e come cambia la velocità di convergenza al raddoppiare degli agenti della rete. Simulando il funzionamento di AP sullo scenario descritto si ottiene la partizione ottima  $\xi_{AP3}^*$ , rappresentata in Fig. 5.5. I dati relativi alle PA di questa partizione vengono riportati nella tabella 5.3.

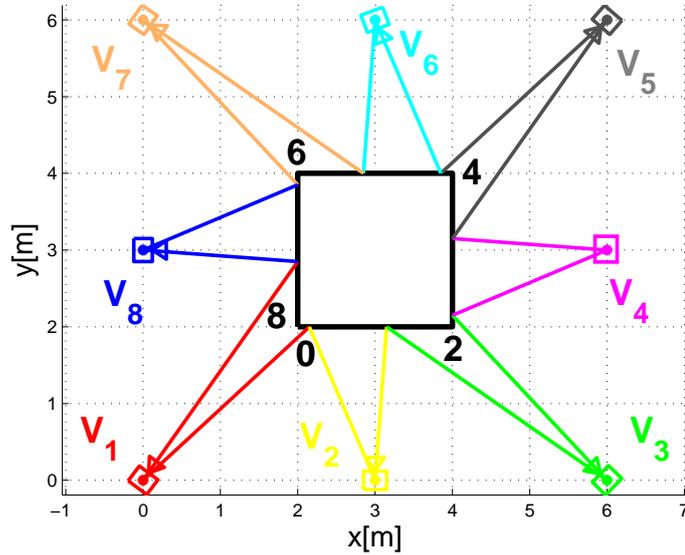


Fig. 5.5: Partizione ottima trovata dalla simulazione di AP sul nuovo scenario descritto.

Le PA di  $\xi_{AP3}^*$  hanno tutte lunghezza  $1\text{m}$  e minimizzano la funzione costo  $2.8$ ,  $f_3 = 0.8$ . Il periodo di pattugliamento delle videocamere diminuisce rispetto a prima, come era logico aspettarsi, diventando  $T_{0,i} = T_{lag} = 0.2\text{s}$ ,  $i = 1, 2, \dots, 8$ . Questo valore è la metà di quelli in tabella 5.1 e tabella 5.2: quindi i tempi di pattugliamento e  $T_{lag}$  vengono dimezzati al raddoppiare delle videocamere. Partendo dalla forma  $\xi_{AP3}^*$ , con un ragionamento analogo a prima si prova l'esistenza di infinite soluzioni ottime di  $\mathcal{P}'_3$  che minimizzano  $f_3$ , esse corrispondono a tutte le possibili partizioni in cui tutte le videocamere hanno PA di lunghezza  $1\text{m}$  e rispettano vincoli fisici, di allacciamento e di copertura. Quindi in generale ogni simulazione di AP su questa struttura porta a individuare un partizionamento ottimo differente.

Videocamera	lunghezza PA [ $m$ ]	Periodo $T_0$ [ $s$ ]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	1	0.2	1.684	136.0343
2	1	0.2	4.023	27.2034
3	1	0.2	1.684	136.0343
4	1	0.2	4.023	27.2034
5	1	0.2	1.684	136.0343
6	1	0.2	4.023	27.2034
7	1	0.2	1.684	136.0343
8	1	0.2	4.023	27.2034

Tab. 5.3: Dati relativi alla partizione ottima di Fig. 5.5.

Si mostra ora come varia la velocità di convergenza del sistema al raddoppiare delle videocamere. A tale scopo si effettuano due ulteriori simulazioni di AP: nella prima viene utilizzato il Gossip Simmetrico come protocollo di comunicazione, mentre nella seconda si utilizza il Gossip Asimmetrico.

In Fig. 5.6 viene rappresentata l'evoluzione delle ampiezza delle PA durante le iterazioni dell'algorithmo nella prima simulazione: all'istante iniziale le PA hanno ampiezza  $4m$  (per le videocamere di indice dispari) o  $2m$  (per le videocamere di indice pari), uguale a quella dei rispettivi PR, la convergenza al valore  $1m$  avviene dopo 430 iterazioni dell'algorithmo. Dal punto di vista pratico la configurazione finale può dirsi raggiunta quando le ampiezze delle PA raggiungono il valore ottimo con certo margine d'errore che dipende dalla precisione richiesta per il partizionamento. In questo caso si considera più che accettabile un margine d'errore di  $5cm$ , quindi la partizione finale può dirsi raggiunta quando le PA entrano nell'intervallo  $1m \pm 0.05m$ . Il numero di iterazioni necessarie alle PA per entrare all'interno dell'intervallo è 190, meno della metà di quelle richieste per convergere al valore esatto  $1m$ .

La Fig. 5.7 mostra invece l'evoluzione delle ampiezze delle PA nella seconda simulazione, nella quale è stato utilizzato il Gossip Asimmetrico come protocollo di comunicazione. In questo caso la convergenza al valore esatto  $1m$  avviene dopo oltre 700 iterazioni, mentre ne bastano 460 affinché tutte le ampiezze entrino nel range  $1m \pm 0.05m$ .

Si conclude la sezione ricordando che il criterio *etp*, l'unico su cui si basa AP, non esercita alcun controllo sulla qualità della visuale delle videocamere. Le partizioni individuate in tutte le simulazioni effettuate sono ben lontane dall'offrire una visuale perpendicolare alle videocamere, come si può notare intuitivamente da Fig. 5.1, Fig. 5.2 e Fig. 5.5 oppure più formalmente osservando i valori nelle ultime due colonne di tabella 5.1, 5.2 e 5.3.

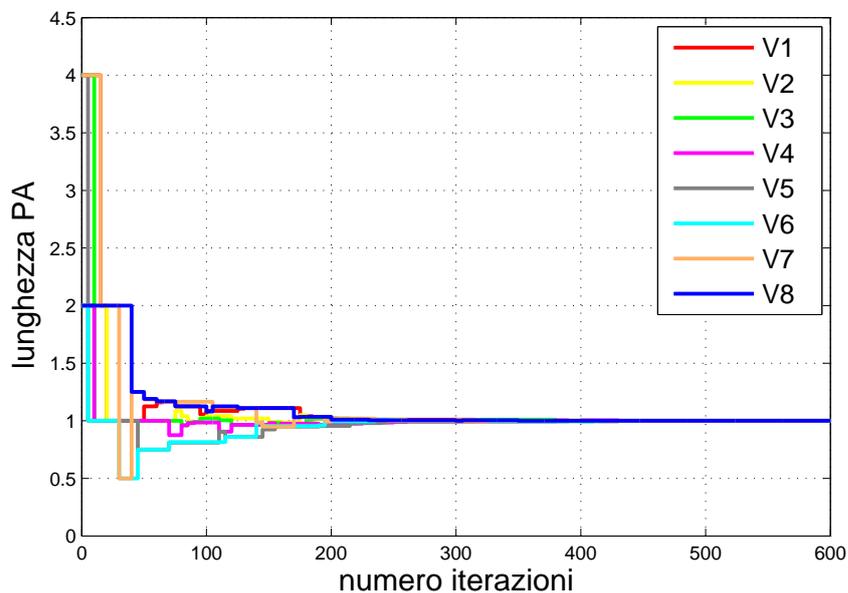


Fig. 5.6: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Simmetrico sullo scenario di Fig. 5.5 .

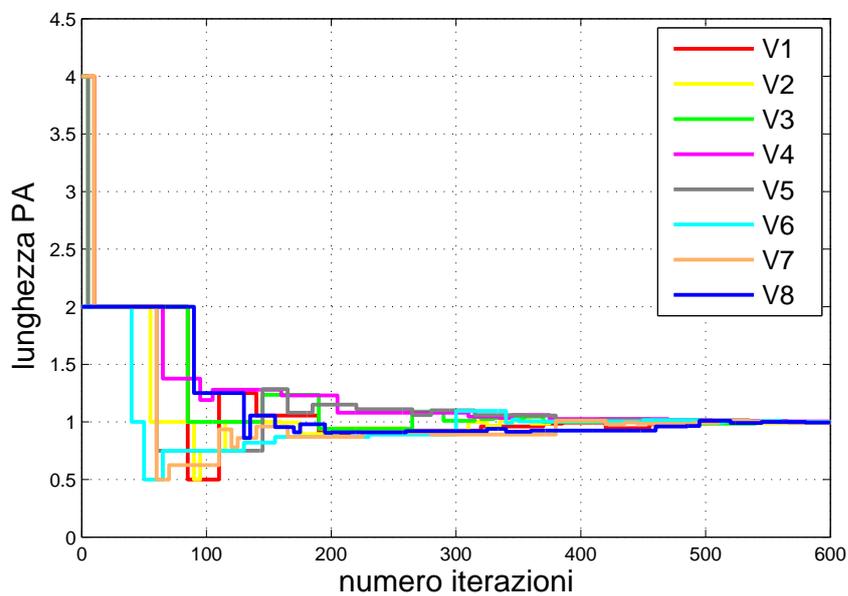


Fig. 5.7: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Asimmetrico sullo scenario di Fig. 5.5 .

## 5.2 Simulazioni di AP $\Psi$

Prima di presentare i risultati delle simulazioni di AP $\Psi$  sul perimetro quadrato con 4 ed 8 videocamere PTZ e confrontare le partizioni ottime individuate con quelle trovate da AP, si vogliono esaminare separatamente gli effetti sull'evoluzione del sistema dati dall'introduzione dei criteri *Centramento* e *Area*. A tal fine si definiscono due nuovi algoritmi di partizionamento AC ed AA, nei quali per l'aggiornamento delle PA vengono considerati solo i criteri rispettivamente *Centramento* e *Area*. Successivamente ne viene simulato il funzionamento sugli scenari già introdotti e si analizzano le partizioni individuate per vedere se effettivamente rispecchiano le ipotesi del capitolo precedente sugli effetti di convergenza dei due criteri introdotti.

### 5.2.1 Algoritmo di Centramento AC

Nel capitolo 4 si è scritto che utilizzando il criterio di *Centramento* come unico criterio per gli aggiornamenti, il sistema tende verso una configurazione in cui ogni PA  $A_i$  è centrata il più possibile su  $d_{i,min}$ , punto del perimetro in cui la rispettiva videocamera ha la visuale migliore. Più formalmente il sistema tende verso un partizionamento in cui sono minime le differenze tra gli angoli d'incidenza visuale degli estremi delle PA,  $||\psi_i(\ell_i) - \psi_i(r_i)||$ . Si consideri ora un algoritmo di partizionamento in cui l'unico criterio utilizzato per l'aggiornamento delle PA sia appunto quello di *Centramento*, chiamiamolo Algoritmo di Centramento, in breve AC. Per ottenerlo, in pratica è sufficiente prendere una versione di AP $\Psi$  in cui le costanti  $C_{etp}$  e  $C_{area}$  in (4.11) vengono settate a valori molto maggiori di  $C_{centr}$ , in modo che durante le elaborazioni del metodo grafico il termine *etp* e quello *Area* risultino trascurabili.

Si consideri ora di simularne il funzionamento sul solito scenario di Fig. 2.3, in cui un perimetro quadrato viene pattugliato da 4 videocamere PTZ. La partizione  $\xi_{AC}^*$  individuata da AC è rappresentata in Fig. 5.8, mentre i dati relativi alle PA vengono riportati in tabella 5.4. Le PA di  $\xi_{AC}^*$  sono centrate attorno ai propri  $d_{i,min}$ , punti in cui le rispettive videocamere hanno la visuale più perpendicolare del perimetro, che in questo caso corrispondono ai vertici del quadrato. Di conseguenza la differenza tra gli angoli visuali d'incidenza agli estremi è zero per ogni videocamera (si veda la quarta colonna di tabella 5.4). Al contrario le ampiezze delle PA sono variabili, nonostante le velocità massime siano le stesse per ogni agente della rete, ciò accade perché il criterio di *Centramento* non fornisce alcun controllo in questa direzione.

In generale AC potrebbe convergere verso uno qualsiasi tra i partizionamenti in cui le PA sono centrate, che potenzialmente sono infiniti.

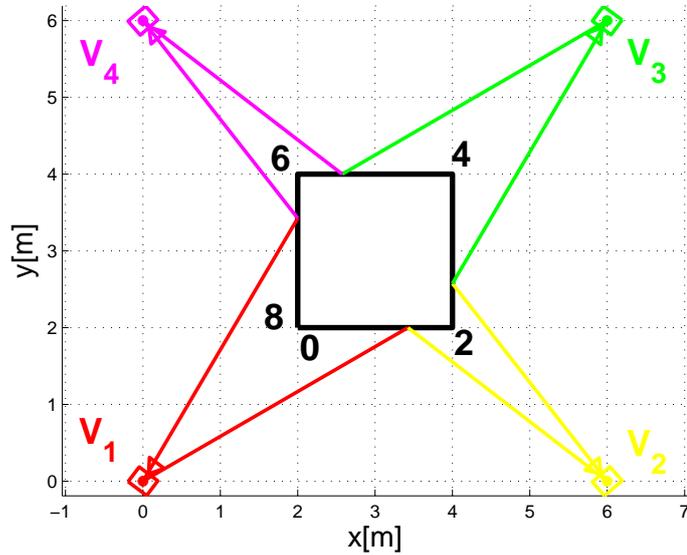


Fig. 5.8: Partizione ottima trovata dalla simulazione di AC sullo scenario di Fig. 2.3.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell) - \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	2.85	0.57	0	386.3568
2	1.15	0.23	0	159.7902
3	2.85	0.57	0	386.3568
4	1.15	0.23	0	159.7902

Tab. 5.4: Dati relativi alla partizione ottima di Fig. 5.8.

### 5.2.2 Algoritmo Area AA

Nel capitolo 4 si è scritto che il Criterio *Area* fa convergere il sistema verso configurazioni in cui per ogni videocamera il valore delle aree sottese alle funzioni angolo visuale d'incidenza  $\psi_i$ , comprese tra gli estremi delle rispettive PA sono uguali:

$$\int_{\ell_i^*}^{r_i^*} |\psi_i(s)| ds = \text{cost. } i = 1, \dots, N \quad (5.2)$$

Per provarlo si consideri un algoritmo di partizionamento in cui l'unico criterio utilizzato per l'aggiornamento delle PA sia il criterio *Area*, chiamiamolo Algoritmo Area, in breve AA. Nella pratica per ottenere un tale algoritmo si procede analogamente a come si è fatto per AC: si utilizza una versione di AP $\Psi$

in cui le costanti  $C_{etp}$   $C_{centr}$  in (4.11) vengono settate a valori molto maggiori di  $C_{area}$ , in modo che durante le elaborazioni del metodo grafico il termine  $etp$  e quello  $Centramento$  risultino trascurabili. Per ottenere dei risultati più significativi dal punto di vista espositivo, AA viene simulato sullo scenario di Fig. 5.5, dove il perimetro quadrato viene pattugliato da 8 videocamere PTZ. La partizione ottima  $\xi_{AA}^*$  trovata da AA viene rappresentata in Fig. 5.9, mentre i dati delle PA relative vengono riportati nella tabella 5.5.

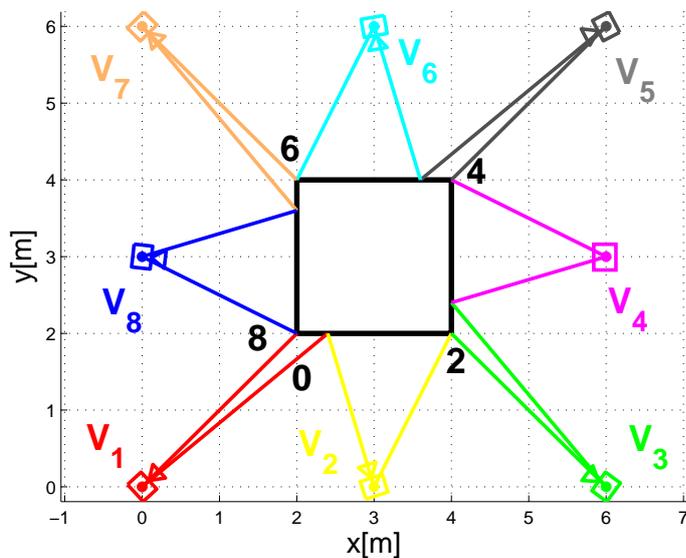


Fig. 5.9: Partizione ottima trovata dalla simulazione di AA sul quadrato con 8 videocamere PTZ ( $C_{etp} = C_{centr} = 10^4$ ,  $C_{area} = 1$ ).

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	0.4	0.08	5.194	230.3088
2	1.6	0.32	9.866	230.3088
3	0.4	0.08	5.194	230.3088
4	1.6	0.32	9.866	230.3088
5	0.4	0.08	5.194	230.3088
6	1.6	0.32	9.866	230.3088
7	0.4	0.08	5.194	230.3088
8	1.6	0.32	9.866	230.3088

Tab. 5.5: Dati relativi alla partizione ottima di Fig. 5.9.

Come previsto, l'algoritmo ha diviso il perimetro in modo da eguagliare le aree (5.2), che in  $\xi_{AA}^*$  assumono tutte lo stesso valore (si veda la tabella 5.5). Per ottenere questo stato le ampiezze delle zone di pattugliamento delle videocamere sono state ampliate a seconda della qualità della loro vista sul perimetro: a quelle con visuale più perpendicolare, perché poste di fronte a un lato del quadrato ( $V_2, V_4, V_6$  e  $V_8$ ), vengono assegnate delle PA di  $1.6m$ , mentre a quelle con visuale peggiore, perché poste in corrispondenza dei vertici del perimetro ( $V_1, V_3, V_4$  e  $V_4$ ), vengono assegnate delle PA di  $0.4m$ .

In Fig. 5.10 sono rappresentate le funzioni  $|\psi_i|$  per ogni videocamera della rete, le zone colorate al di sotto sono le aree (5.2) relative a  $\xi_{AA}^*$ , ora si comprende più intuitivamente perché per eguagliarle sia necessario modulare le ampiezze delle PA nel modo descritto. Una configurazione in cui l'ampiezza delle PA è varia e non tiene conto delle velocità delle rispettive videocamere, come  $\xi_{AA}^*$ , è indesiderabile in termini di ottimizzazione del pattugliamento, in quanto  $T_{lag}(\xi_{AA}^*) = 0.32s$  risulta più del 50% maggiore di quello minimo  $T_{lag}(\xi_{AP3}^*) = 0.2s$ , ottenuto applicando AP. Va osservato inoltre che le PA di  $\xi_{AA}^*$  non vengono nemmeno centrate poiché il criterio *Area* non fornisce alcun controllo in questa direzione.

Per finire si ricorda che anche in questo caso la partizione  $\xi_{AA}^*$  è solo una tra tutte le possibili configurazioni ottime in cui le aree sono uguali, in generale ogni simulazione di AA individua una partizione diversa.

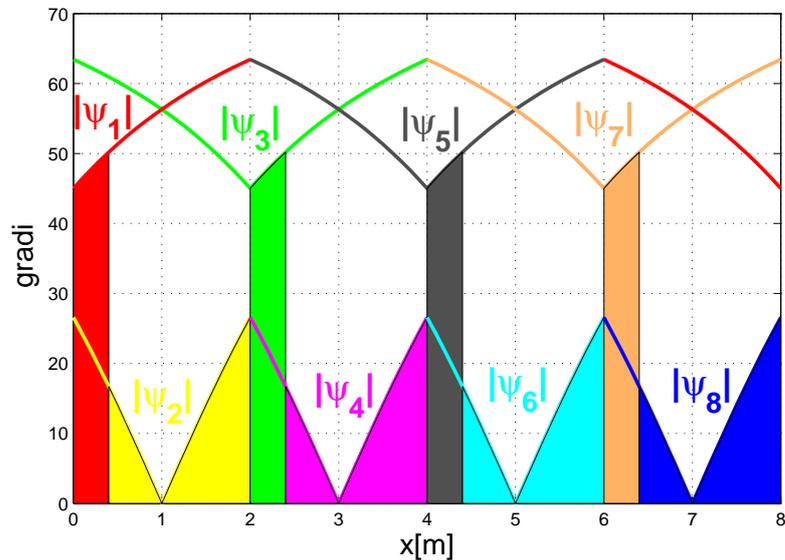


Fig. 5.10: Le zone colorate sono le aree sottese alle funzioni  $|\psi_i|$  relative alla partizione ottima  $\xi_{AA}^*$  di Fig. 5.9.

5.2.3 Simulazioni di AP $\Psi$ 

Si consideri ora AP $\Psi$  come descritto nel capitolo 4, in cui l'aggiornamento delle PA avviene tramite la collaborazione dei tre termini legati ai criteri *etp* *Centramento* ed *Area*. Per meglio comprendere il funzionamento dell'algoritmo e per provarne la convergenza alla soluzione ottima di  $\mathcal{P}_4$ , se ne effettuano alcune simulazioni sugli scenari precedentemente introdotti. Si consideri la struttura di Fig. 2.3, dove il solito quadrato viene pattugliato da 4 videocamere PTZ. La partizione ottima  $\xi_{AP\Psi}^*$  trovata da AP $\Psi$  viene rappresentata in Fig. 5.11, i dati relativi alle sue PA vengono riportati nella tabella 5.6.

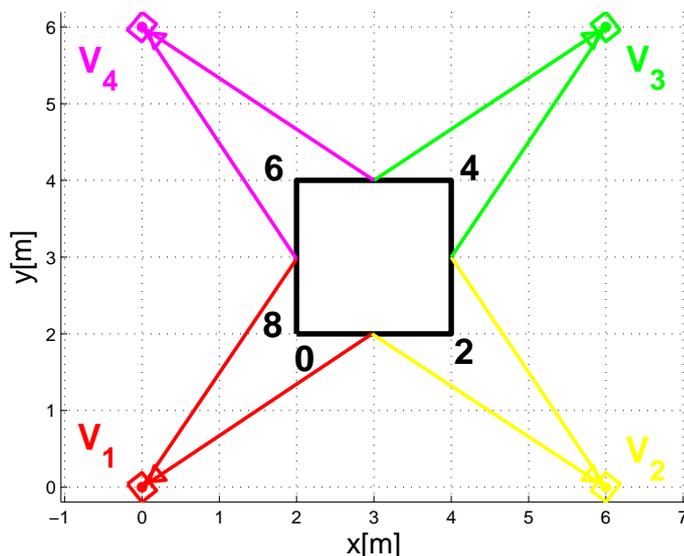


Fig. 5.11: Partizione ottima trovata dalla prima simulazione di AP $\Psi$  sullo scenario di Fig. 2.3.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	2	0.4	0	215.6926
2	2	0.4	0	215.6926
3	2	0.4	0	215.6926
4	2	0.4	0	215.6926

Tab. 5.6: Dati relativi alla partizione ottima di Fig. 5.11.

Grazie alla cooperazione dei termini *etp*, *Centramento* e *Area*, la partizione  $\xi_{AP\Psi}^*$  presenta PA:

- della stessa lunghezza,  $2m$ ;
- centrate attorno ai rispettivi  $d_{min}$ ;
- le cui aree (5.2) hanno tutte lo stesso valore.

Il verificarsi di queste tre condizioni garantisce il raggiungimento dei due obiettivi desiderati:

1. ottimizzazione del pattugliamento, formalmente minimizzazione di  $T_{lag}$ ;
2. ottimizzazione della visuale delle videocamere, formalmente minimizzazione di  $P_{\Psi}$ .

In questo caso particolarmente fortunato l'ottimizzazione della visuale delle videocamere non ha portato ad un peggioramento delle performance di pattugliamento, in termini di  $T_{lag}$ . La particolare simmetria dello scenario permette di raggiungere entrambi gli obiettivi contemporaneamente.

In tabella 5.7 vengono riportati i valori delle funzioni costo di  $f_3$  e  $f_4$  e della funzione penalità  $P_{\Psi}$  di  $\mathcal{P}_4$ , per le varie simulazioni effettuate con AP e AP\Psi sulla struttura di Fig. 2.3. Tutte le partizioni individuate sono soluzioni ottime di  $\mathcal{P}'_3$ , infatti minimizzano la funzione costo  $f_3$  e di conseguenza  $T_{lag}$ . Invece l'unica partizione soluzione ottima di  $\mathcal{P}_4$  che minimizza  $f_4$ , garantendo alle videocamere la vista più perpendicolare del perimetro, è  $\xi_{AP\Psi}^*$ , trovata da AP\Psi. Le partizioni  $\xi_{AP1}^*$  e  $\xi_{APe}^*$ , ricavate dalle simulazioni di AP, pur garantendo il  $T_{lag}$  ottimo, individuano delle PA in cui le videocamere non hanno la visuale ottimale, formalmente lo si può vedere confrontando i valori delle aree nelle ultime colonne delle tabelle 5.1, 5.2 e 5.6.

Si osserva inoltre che tutte le simulazioni di AP\Psi fanno convergere il sistema in  $\xi_{AP\Psi}^*$ , poiché rappresenta l'unica soluzione ottima secondo  $\mathcal{P}_4$ .

Partizione ottima	funzione costo $f_3$	funzione costo $f_4$	$T_{lag}$	penalità $\frac{P_{\Psi}}{C}$
$\xi_{AP1}^*$	1.6	2.5075	0.4	0.90751
$\xi_{AP2}^*$	1.6	2.4801	0.4	0.88009
$\xi_{AP\Psi}^*$	1.6	2.4631	0.4	0.86277

Tab. 5.7: Funzioni costo e penalità per le partizioni individuate dalle simulazioni di AP e AP\Psi sullo scenario di Fig. 2.3. (C=250)

---

L'aggiunta dei termini di *Centramento* e di *Area* lascia sostanzialmente invariate le performance di convergenza dell'algoritmo. Fig. 5.12 e Fig. 5.13 mostrano come evolvono le lunghezze delle PA durante due diverse simulazioni di AP $\Psi$ : nella prima viene utilizzato il Gossip Simmetrico come protocollo di comunicazione, nella seconda invece il Gossip Asimmetrico.

Nel primo caso il sistema impiega 40 iterazioni per convergere alla configurazione finale, che si considera raggiunta quando le lunghezze di tutte le PA entrano nell'intervallo  $2m \pm 0.05m$ . Per la convergenza al valore esatto  $2m$  bisogna aspettare 70 iterazioni.

Nella simulazione in cui si utilizza il Gossip Asimmetrico la convergenza verso la partizione ottima impiega il doppio del tempo: le PA entrano nell'intervallo  $2m \pm 0.05m$  dopo 80 iterazioni dell'algoritmo. La convergenza al valore esatto  $2m$  impiega 150 iterazioni, circa il doppio di quelle richieste considerando il margine d'errore. I dati riportati sono relativi alle specifiche simulazioni analizzate, in generale altre simulazioni possono portare a una convergenza più rapida o più lenta, i due casi valutati rientrano nella media delle possibili occorrenze.

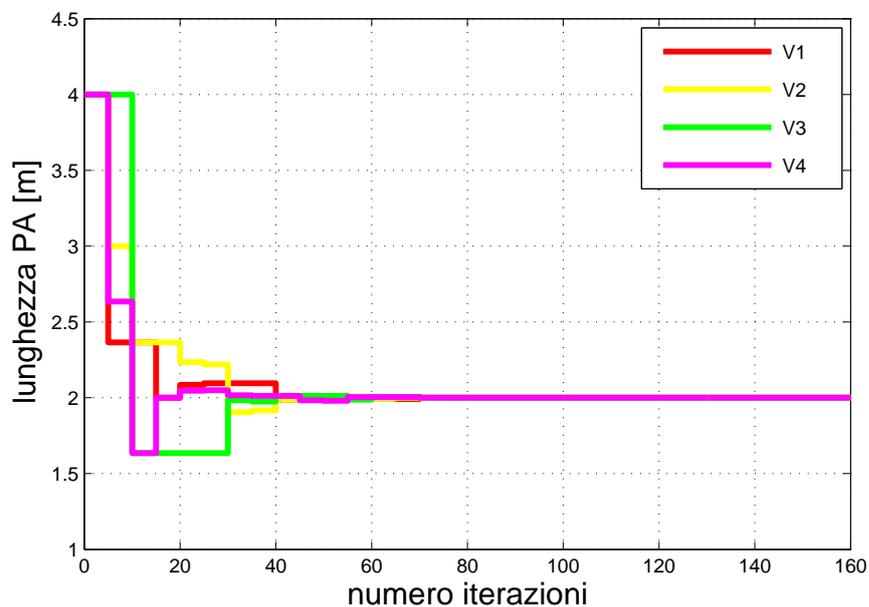


Fig. 5.12: Evoluzione dell'ampiezza delle PA in  $AP\Psi$  con Gossip Simmetrico.

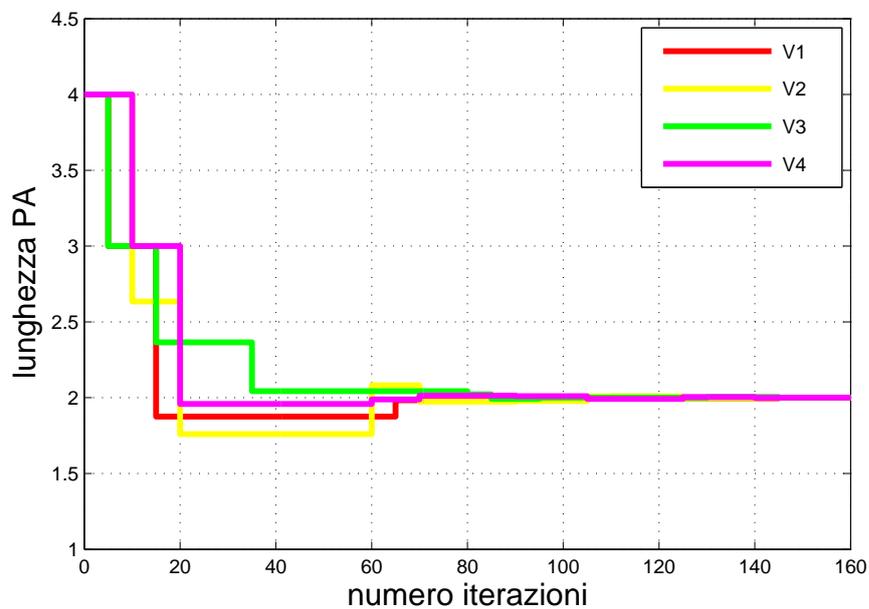


Fig. 5.13: Evoluzione dell'ampiezza delle PA in  $AP\Psi$  con Gossip Asimmetrico.

Per analizzare il comportamento di AP $\Psi$  con un numero maggiore di videocamere, si esegue una simulazione dell'algoritmo sullo stesso scenario di figura 5.5, in cui il perimetro di forma quadrata viene pattugliato da 8 videocamere PTZ. La partizione ottima  $\xi_{AP\Psi}^*$  individuata è rappresentata in Fig. 5.14, i dati relativi alle sue PA sono riportati in tabella 5.8.

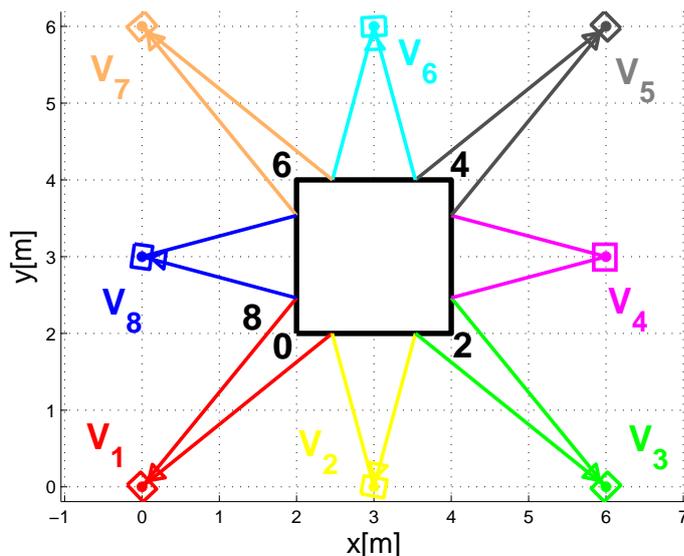


Fig. 5.14: Partizione ottima trovata dalla simulazione di AP $\Psi$  sul quadrato con 8 videocamere PTZ.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	0.9	0.18	0	86.3235
2	1.1	0.22	0	23.6597
3	0.9	0.18	0	86.3235
4	1.1	0.22	0	23.6597
5	0.9	0.18	0	86.3235
6	1.1	0.22	0	23.6597
7	0.9	0.18	0	86.3235
8	1.1	0.22	0	23.6597

Tab. 5.8: Dati relativi alla partizione ottima di Fig. 5.14.

La differenza tra gli angoli d'incidenza visuale agli estremi è zero per ogni PA, ciò significa che sono tutte centrate attorno ai rispettivi  $d_{min}$ . Al contrario

l'ampiezza non è uguale per tutte: quelle di indice dispari, che si trovano in corrispondenza dei vertici del quadrato, sono lunghe  $1.1m$ , quelle di indice pari, posizionate in corrispondenza dei lati, sono lunghe  $0.9m$ . Anche i valori delle aree dividono le videocamere in due gruppi analoghi ai precedenti: per quelle di indice dispari risultano 106.8596, per quelle di indice pari invece 16.98. Inoltre  $T_{lag}(\xi_{AP\Psi 2}^*) = 0.22s$  è maggiore del minimo possibile per la struttura in esame ( $0.2s$ ), relativo alle partizioni identificate con AP.

In tabella 5.9 si riportano i valori delle funzioni costo e penalità di  $\mathcal{P}'_3$  e  $\mathcal{P}_4$ , per le partizioni  $\xi_{AP3}^*$  e  $\xi_{AP\Psi 2}^*$ , ottenute dalle simulazioni, rispettivamente di AP e AP\Psi, sul perimetro quadrato con 8 videocamere PTZ. La prima partizione è una soluzione ottima di  $\mathcal{P}'_3$ , minimizza  $f_3$  e  $T_{lag}$  per la struttura in esame, ma non offre delle visuali perpendicolari alle videocamere, come si vede in Fig. 5.14. La seconda partizione invece è soluzione ottima di  $\mathcal{P}_4$ , minimizza  $f_4$  e garantisce alle videocamere delle visuali perpendicolari, ma non minimizza né  $f_3$  né  $T_{lag}$ , perciò non è una soluzione ottima di  $\mathcal{P}'_3$ .

Partizione ottima	funzione costo $f_3$	funzione costo $f_4$	$T_{lag}$	penalità $\frac{P_\Psi}{C}$
$\xi_{AP3}^*$	0.8	1.3433	0.2	0.54332
$\xi_{AP\Psi 2}^*$	0.80444	1.2998	0.22	0.49536

Tab. 5.9: Funzioni costo e penalità per le partizioni individuate dalle simulazioni di AP e AP\Psi sullo scenario di Fig. 5.5. ( $C=250$ )

Per questo scenario esiste un trade-off tra minimizzazione di  $T_{lag}$  e ottimizzazione della visuale delle videocamere: da una parte il criterio *etp* spinge il sistema verso una configurazione in cui le PA hanno tutte la stessa lunghezza, dall'altra il criterio *Area*, per pareggiare il valore delle aree (5.2), amplia le PA delle videocamere di indice pari e riduce quelle di indice dispari, dato che le prime possiedono una visuale più perpendicolare delle seconde. L'equilibrio raggiunto viene determinato dai valori di  $C_{etp}$ ,  $C_{centr}$  e  $C_{area}$  impostati. Nella simulazione analizzata si sono presi  $C_{etp} = 1$ ,  $C_{centr} = 5$  e  $C_{area} = 250$ . Il termine *etp* è dominante nell'identificazione della partizione ottima  $\xi_{AP\Psi 2}^*$ , infatti mentre la differenza tra le lunghezze delle PA è minima ( $1.1m - 0.9m$ ), quella tra le aree è significativa ( $86 - 23$ ). Modificando il valore di  $C_{area}$  si cambia il tipo di configurazione finale individuata dall'algoritmo: se una buona visuale è un requisito fondamentale per il problema di pattugliamento, si sceglie un  $C_{area}$  basso in modo da ridurre la lunghezza delle PA delle videocamere con una visuale intrinsecamente peggiore; se invece il requisito fondamentale è ridurre al minimo la probabilità di non rilevare un'intrusione, si sceglie un  $C_{area}$  molto elevato in modo che le PA abbiano tutte la stessa

lunghezza e  $T_{lag}$  sia il minimo possibile.

In tabella 5.10 si riportano i dati relativi alla partizione ottima individuata da AP $\Psi$  con  $C_{area} = 25$ : la differenza tra le lunghezze delle PA aumenta rispetto a prima mentre quella tra le aree diminuisce, di conseguenza  $T_{lag}$  diventa  $0.28s$ , quasi il 50% maggiore di quello ottimo. In tabella 5.11 si riportano i dati relativi alla partizione ottima individuata da AP $\Psi$  con  $C_{area} = 2500$ : il termine  $Area$  diventa trascurabile, infatti le PA hanno praticamente la stessa lunghezza mentre aumenta la differenza tra le aree,  $T_{lag} = 0.202s$  è approssimabile a quello ottimo per questa struttura.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	0.6	0.12	0	66.2458
2	1.4	0.28	0	31.3757
3	0.6	0.12	0	66.2458
4	1.4	0.28	0	31.3757
5	0.6	0.12	0	66.2458
6	1.4	0.28	0	31.3757
7	0.6	0.12	0	66.2458
8	1.4	0.28	0	31.3757

Tab. 5.10: Partizione ottima con  $C_{area} = 25$ .

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	0.99	0.198	0	106.8596
2	1.01	0.202	0	16.98
3	0.99	0.198	0	106.8596
4	1.01	0.202	0	16.98
5	0.99	0.198	0	106.8596
6	1.01	0.202	0	16.98
7	0.99	0.198	0	106.8596
8	1.01	0.202	0	16.98

Tab. 5.11: Partizione ottima con  $C_{area} = 2500$ .

Per analizzare la velocità di convergenza di AP $\Psi$  su questa struttura, come di consueto si effettuano due simulazioni dell'algoritmo utilizzando nella

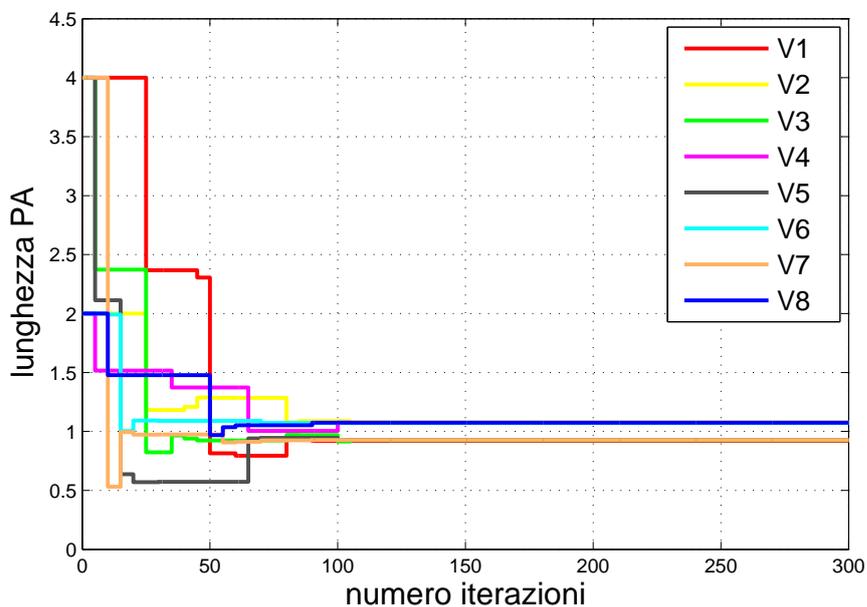


Fig. 5.15: Evoluzione dell'ampiezza delle PA in  $AP\Psi$  con Gossip Simmetrico.

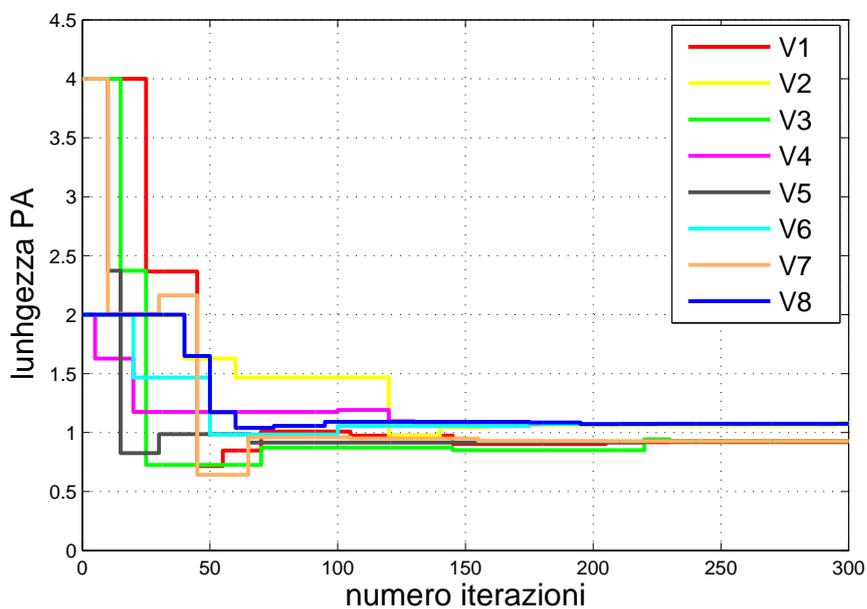


Fig. 5.16: Evoluzione dell'ampiezza delle PA in  $AP\Psi$  con Gossip Asimmetrico.

prima il Gossip Simmetrico e nella seconda il Gossip Asimmetrico come protocollo di comunicazione e settando i parametri sugli stessi valori utilizzati per  $\xi_{AP\Psi}^*$ :  $C_{etp} = 1$ ,  $C_{centr} = 5$  e  $C_{area} = 250$ , in modo da ottenere la stessa partizione ottima da entrambe le simulazioni. L'evoluzione delle lunghezze delle PA nel primo caso è rappresentata in Fig. 5.15: la convergenza alla configurazione finale impiega 100 iterazioni. In Fig. 5.16 invece è rappresentata l'evoluzione delle PA nel secondo caso: la convergenza impiega 220 iterazioni. In entrambi i casi la partizione ottima si considera raggiunta quando le PA entrano negli intervalli  $1.1m \pm 0.05m$  e  $0.9m \pm 0.05m$ .

Il numero di iterazioni richieste per la convergenza non è costante ma dipende dalla simulazione effettuata e le due analizzate rientrano nella media delle possibili occorrenze.

Confrontando Fig. 5.6 e Fig. 5.7 con Fig. 5.15 e Fig. 5.16 si nota che AP $\Psi$  in media raggiunge la configurazione finale più rapidamente rispetto a AP, sia utilizzando il Gossip Simmetrico (100 iterazioni per AP $\Psi$  contro 190 per AP) sia utilizzando il Gossip Asimmetrico (220 iterazioni per AP $\Psi$  contro 460 per AP).

### 5.3 Scenario Reale

Nelle sezioni precedenti si sono analizzati i risultati di alcune simulazioni di AP e AP $\Psi$  su scenari molto semplici: perimetri regolari con un numero ridotto di videocamere per pattugliarli che portano una certa simmetria nelle soluzioni trovate. Lo scopo principale delle simulazioni presentate è descrivere il funzionamento dei due algoritmi progettati: mostrare le caratteristiche delle partizioni ottime che individuano e il modo di operare sui parametri  $C_{etp}$ ,  $C_{centr}$  e  $C_{area}$  per cambiarle a secondo delle necessità.

Nella pratica implementare un sistema di sorveglianza automatizzato su strutture così semplici risulta poco conveniente, sarebbe sufficiente posizionare un numero accettabile di videocamere fisse per monitorare l'intero perimetro. Per avere dei dati più significativi sulle partizioni ottime e le performance di convergenza di AP e AP $\Psi$  bisogna prendere in considerazione situazioni più complesse. Si vuole quindi analizzare il comportamento dei due algoritmi proposti su uno scenario più realistico, in cui il perimetro da pattugliare abbia un'estensione considerevole e il numero di videocamere sia rilevante.

Si introduce a tal fine un perimetro di forma irregolare e lunghezza  $L = 59m$ , con rientranze ed occlusioni visuali (rappresentato in Fig. 5.17 e Fig. 5.18), e una rete di 22 videocamere disposte lungo il perimetro, tutte con velocità massima  $\bar{v} = 1 m/s$ .

La partizione ottima  $\xi_{AP\Psi,r}^*$  trovata simulando AP sulla struttura descritta

è rappresentata in Fig. 5.17, i dati relativi alle sue PA sono riportati nella tabella 5.17. Come ci si aspettava tutte le PA hanno la stessa lunghezza, che si appropia al valore ottimale  $L/N = 59m/22 = 2.68m$ . La partizione  $\xi_{AP\Psi,r}^*$  è quindi una soluzione ottima di  $\mathcal{P}'_3$  e garantisce la minimizzazione di  $T_{lag}$ . In pratica la lunghezza non è esattamente  $2.68m$  per ognuna delle PA, infatti con un numero così elevato di videocamere nella rete, la convergenza al valore esatto richiede un numero elevatissimo di iterazioni dell'algoritmo di partizionamento. Fortunatamente, come si è già visto, nella pratica si accetta un certo margine d'errore, in questo caso  $5cm$ , quindi la configurazione finale può considerarsi raggiunta quando le PA entrano nell'intervallo  $2.68m \pm 0.05m$ . Sotto quest'ipotesi la partizione ottima viene individuata in un numero ragionevole di iterazioni, come mostrato nel seguito.

Per analizzare nello specifico le performance di convergenza si eseguono due simulazioni di AP, nella prima si utilizza Gossip Simmetrico come protocollo di comunicazione, nella seconda invece si utilizza Gossip Asimmetrico. Fig. 5.19 e Fig. 5.20 rappresentano l'evoluzione delle PA delle videocamere durante le due simulazioni di AP: nel primo caso la convergenza alla configurazione finale avviene dopo 1180 iterazioni, nel secondo caso invece ne impiega più di 2000.

La partizione  $\xi_{AP\Psi,r}^*$  trovata dalla simulazione di AP\Psi sulla struttura è rappresentata in Fig. 5.18 mentre i dati relativi alle sue PA sono riportati nella tabella 5.14. Contrariamente alla partizione precedentemente analizzata, le PA di questa soluzione hanno lunghezza variabile, che spazia da  $1.97m$  di  $A_1$  ai  $3.05m$  di  $V_5$  e  $V_6$ , il resto delle PA ha lunghezze comprese tra questi due valori. Come conseguenza vi è un peggioramento delle performance di pattugliamento: infatti  $T_{lag}(\xi_{AP\Psi,r}^*) = 6.1s$  mentre  $T_{lag}(\xi_{AP,r}^*) = 5.39s$ , c'è quindi un incremento di  $0.7s$  (12,8%) rispetto al valore ottenuto dalla partizione individuata da AP. Il minor valore possibile di  $T_{lag}$  per la struttura che si sta analizzando è  $2.36s$ , si ottiene solo se tutte le PA sono ampie esattamente  $2.68m$ . D'altro canto in  $\xi_{AP\Psi,r}^*$  la visuale delle videocamere sulle proprie PA risulta notevolmente migliorata rispetto a  $\xi_{AP,r}^*$ .

Si può notare facilmente confrontando le zone di pattugliamento delle prime 5 videocamere della rete in Fig. 5.17 con quelle in Fig. 5.18: nel primo sono spostate alla destra delle rispettive videocamere, costringendole ad una visuale obliqua, nel secondo caso risultano centrate offrendo alle videocamere una visuale perpendicolare.

Per provarlo in maniera più formale si confronta la quarta colonna di tabella 5.13 con quella di tabella 5.14: i valori della differenza tra gli angoli d'incidenza visuale agli estremi delle PA nella prima sono sempre inferiori nel secondo caso, ciò significa che in  $\xi_{AP\Psi,r}^*$  tutte le videocamere hanno una

visuale più perpendicolare delle loro PA rispetto a  $\xi_{AP,r}^*$ . Inoltre confrontando le ultime colonne delle stesse tabelle si scopre che anche i valori delle aree (5.2) sono quasi sempre inferiori nella partizione relativa ad  $\mathbf{AP}\Psi$ , infatti è proprio per minimizzare la somma di queste aree che si crea la disparità tra le lunghezze delle PA: vengono ampliate quelle relative a videocamere con una visuale intrinsecamente perpendicolare sul perimetro, mentre vengono ridotte quelle relative a videocamere con una visuale intrinsecamente obliqua ( $V_1, V_{14}, V_{19}$ ). Per un confronto più formale tra le due partizioni individuate si veda tabella 5.12, in cui vengono riportati valori di  $f_3, f_4$ , di  $T_{lag}$  e della funzione penalità  $P_\Psi$  per entrambe.

Partizione ottima	funzione costo $f_3$	funzione costo $f_4$	$T_{lag}$	penalità $\frac{P_\Psi}{C}$
$\xi_{AP,r}^*$	15.8737	18.2016	5.36	2.3279
$\xi_{AP\Psi,r}^*$	16.2268	18.1075	6.1	1.8807

Tab. 5.12: Funzioni costo e penalità per le partizioni individuate dalle simulazioni di AP e  $\mathbf{AP}\Psi$  sullo scenario realistico ( $C=2500$ ).

Per studiare le performance di convergenza di  $\mathbf{AP}\Psi$  come di consueto si eseguono due simulazioni dell’algoritmo, una utilizzando Gossip Simmetrico come protocollo di comunicazione, l’altra invece utilizzando Gossip Asimmetrico. L’evoluzione delle PA durante le iterazioni degli algoritmi sono rappresentate rispettivamente in Fig. 5.21 e Fig. 5.21. Dato che la configurazione in questo caso non prevede PA della stessa lunghezza le linee nelle figure citate non convergono verso una singola retta, ma ognuna sulla propria determinata dalla lunghezza ottimale. Per capire dopo quante iterazioni la configurazione finale viene raggiunta si cerca il valore nelle ascisse dei grafici dopo i quali la situazione diventa statica, formalmente non si hanno più cambiamenti nella lunghezza delle PA superiori a  $0.05m$ . Per il Gossip Simmetrico la partizione ottima può dirsi raggiunta dopo circa 1000 iterazioni, invece nella simulazione con Gossip Asimmetrico la convergenza impiega più di 2000 iterazioni.

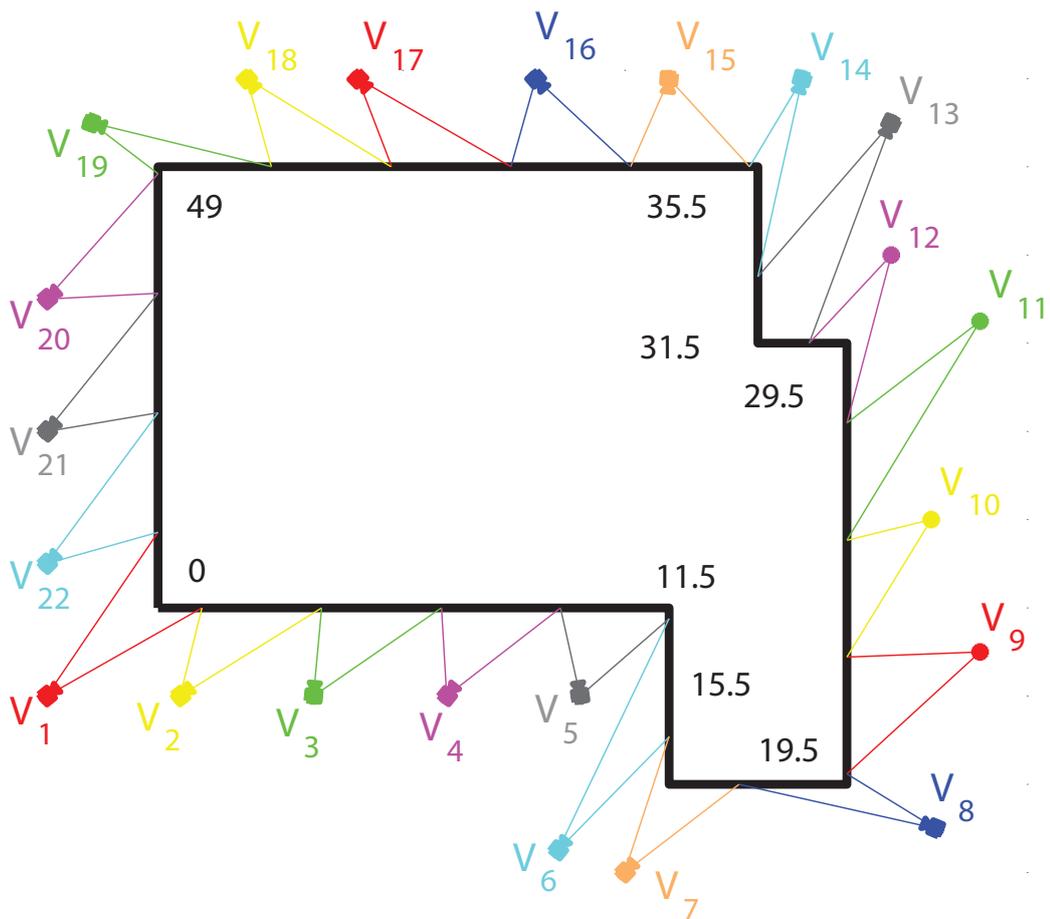


Fig. 5.17: Partizione ottima trovata dalla simulazione di AP sullo scenario realistico.

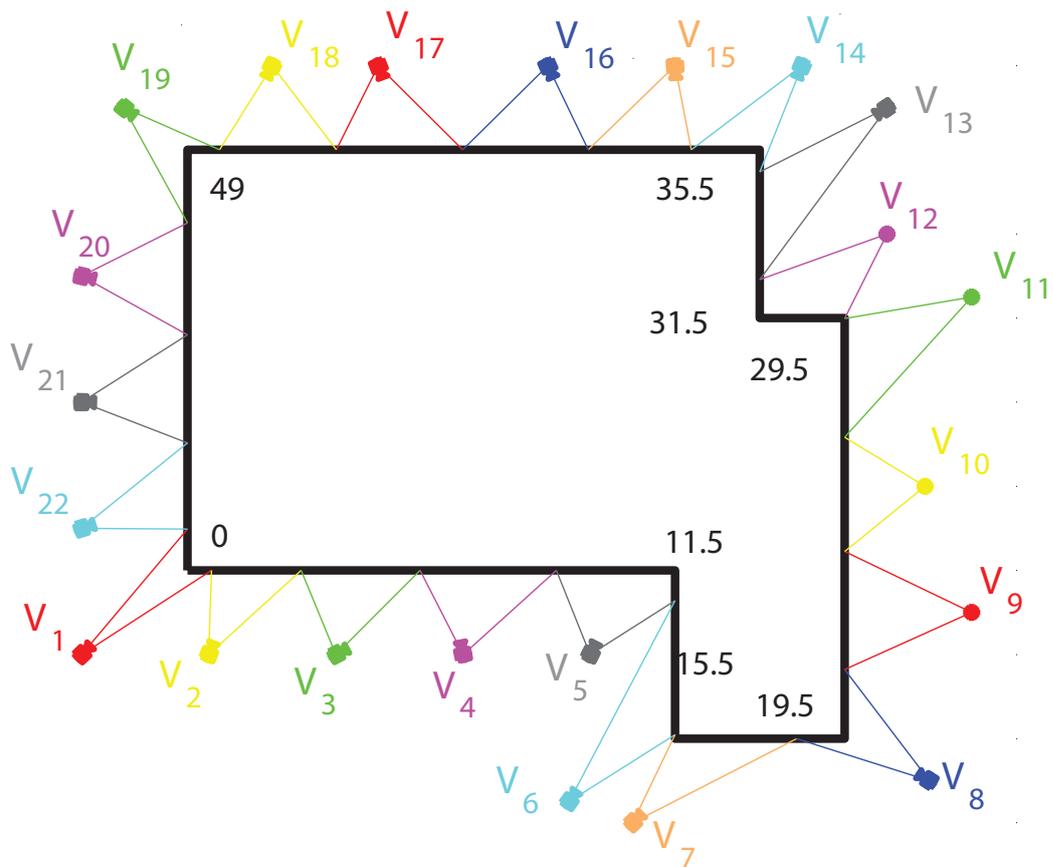


Fig. 5.18: Partizione ottima trovata dalla simulazione di  $AP\Psi$  sullo scenario realistico.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	2.675	5.35	2.15	152.0066
2	2.68	5.36	2.64	152.468
3	2.69	5.38	3.041	143.5976
4	2.695	5.39	3.504	133.3789
5	2.7	5.34	0.259	90.3874
6	2.695	5.39	0.139	135.7259
7	2.7	5.34	3.343	136.9466
8	2.7	5.34	2.131	147.3882
9	2.7	5.34	0.476	36.4887
10	2.685	5.37	1.811	58.5644
11	2.685	5.37	3.35	99.7521
12	2.68	5.36	3.784	125.1704
13	2.68	5.36	2.246	128.879
14	2.68	5.36	1.969	145.3668
15	2.68	5.36	3.768	69.6765
16	2.665	5.33	2.571	60.3676
17	2.665	5.33	2.83	58.4386
18	2.665	5.33	2.173	58.0477
19	2.67	5.34	1.064	154.706
20	2.665	5.33	0.721	40.6008
21	2.67	5.34	1.949	48.0482
22	2.675	5.35	3.004	55.3637

Tab. 5.13: Dati relativi alla partizione ottima di Fig. 5.17 trovata da AP.

Videocamera	lunghezza PA [m]	Periodo $T_0$ [s]	Centramento $  \psi(\ell)  -  \psi(r)  $	Area sottesa $\int_{\ell}^r  \psi(s)  ds$
1	1.97	3.94	0.551	82.1733
2	2.22	4.44	1.972	61.936
3	2.63	5.23	2.083	62.5477
4	2.995	5.99	1.675	77.0393
5	3.05	6.1	0.915	89.4906
6	3.05	6.1	1.438	166.4565
7	2.99	5.98	0.166	152.3418
8	2.88	5.76	1.011	145.3668
9	2.795	5.79	0	41.3332
10	2.75	5.5	0.391	55.0374
11	2.78	5.56	1.641	96.9264
12	2.78	5.56	0.238	113.2751
13	2.58	5.16	1.246	113.4046
14	2.37	4.74	0.81	103.0021
15	2.51	5.02	1.96	55.365
16	2.83	5.66	1.23	67.627
17	2.85	5.70	0.965	67.627
18	2.7	5.4	0.492	52.6502
19	2.6	5.2	0.322	139.3805
20	2.6	5.2	0	42.816
21	2.52	5.04	0.824	41.4878
22	2.22	4.44	1.953	43.8151

Tab. 5.14: Dati relativi alla partizione ottima di Fig. 5.18 trovata da AP $\Psi$ .

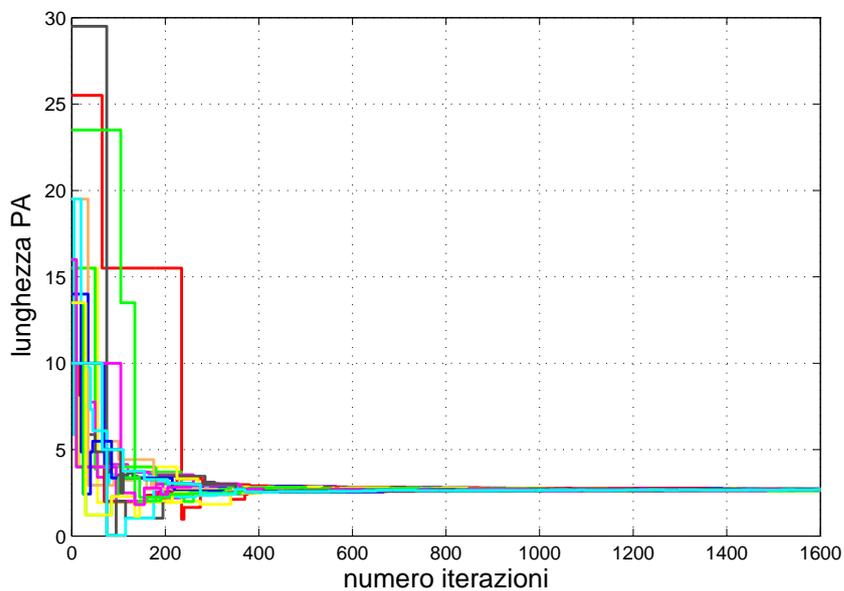


Fig. 5.19: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Simmetrico.

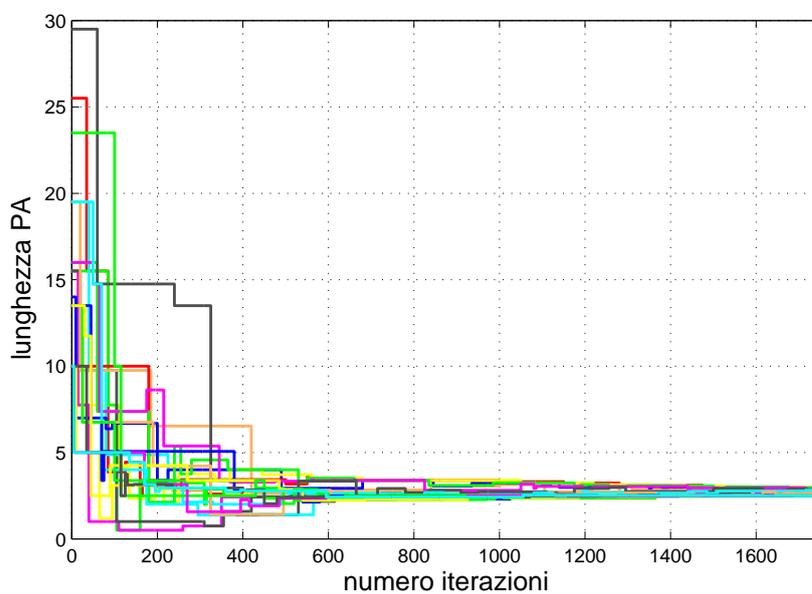


Fig. 5.20: Evoluzione dell'ampiezza delle PA nella simulazione di AP con Gossip Asimmetrico.

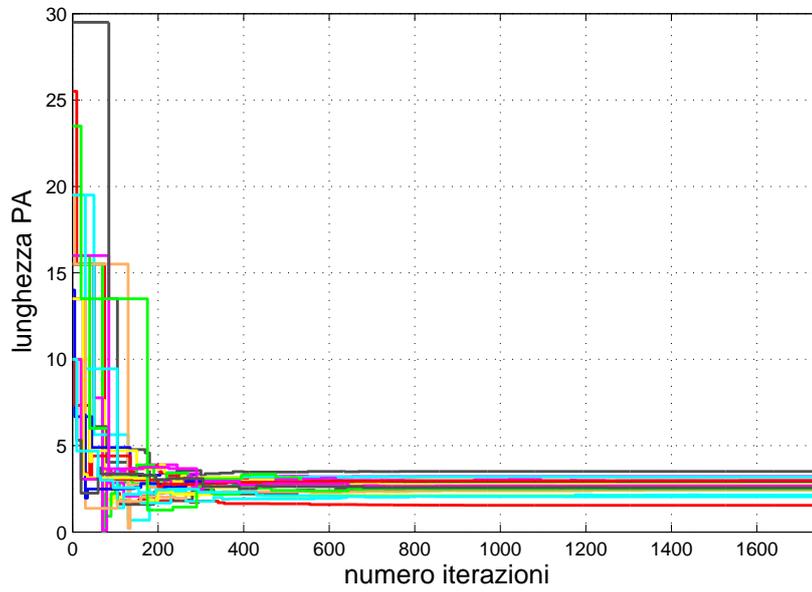


Fig. 5.21: Evoluzione dell'ampiezza delle PA nella simulazione di  $AP\Psi$  con Gossip Simmetrico.

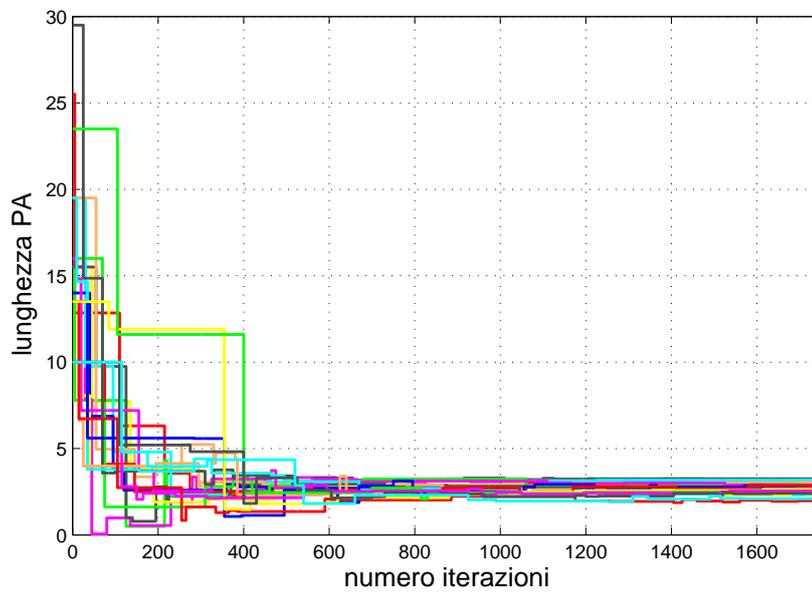


Fig. 5.22: Evoluzione dell'ampiezza delle PA nella simulazione di  $AP\Psi$  con Gossip Asimmetrico.



## 6. CONCLUSIONE

Nell'ambito del problema di pattugliamento di perimetro con una rete di videocamere PTZ sono state già state proposte in letteratura varie strategie iterative e distribuite che, basandosi solo su comunicazioni locali tra gli agenti della rete, trovano il partizionamento che garantisce la minima probabilità di non individuare un'eventuale intrusione, o più formalmente che minimizza il tempo massimo che intercorre tra due ispezioni consecutive di uno stesso punto del perimetro. Questa tesi si inserisce sulla scia dei lavori precedenti con l'obiettivo di superare alcune limitazioni che li affliggono:

1. tutti gli algoritmi proposti in letteratura nella pratica risolvono il problema di partizionamento su un segmento e non su un perimetro;
2. la partizione ottima individuata minimizzando  $T_{lag}$  non tiene conto in alcun modo della qualità della visuale delle videocamere sulla propria zona di pattugliamento.

Il partizionamento su un perimetro reale è stato formalizzato matematicamente introducendo il problema di ottimizzazione  $\mathcal{P}'_3$ , per il quale si è visto esistere una quantità potenzialmente infinita di soluzioni ottime che minimizzano  $T_{lag}$ . Una misura della qualità della visuale delle videocamere in un partizionamento è stata fornita attraverso l'indice  $P_\Psi$  che viene aggiunto come funzione penalità alla funzione costo  $f_3$  di  $\mathcal{P}'_3$  per costituire il problema di ottimizzazione  $\mathcal{P}_4$ , nel quale vengono penalizzate PA in cui le rispettive videocamere hanno visuali oblique.

Per ciascuno dei due nuovi problemi introdotti è stata proposta una strategia di partizionamento, iterativa e distribuita, che ne individua la soluzione ottima. Per risolvere il problema di pattugliamento su un perimetro reale è stato progettato l'algoritmo AP, che converge verso una delle partizioni ottime di  $\mathcal{P}'_3$ . A causa della natura aleatoria dei protocolli di comunicazione utilizzati, non esiste in generale una configurazione preferenziale verso la quale converge, al contrario l'algoritmo può convergere verso qualsiasi partizionamento soluzione di  $\mathcal{P}'_3$  in maniera statica: una volta raggiunto il sistema rimane in esso. La velocità di convergenza di AP dipende da vari fattori: configurazione del perimetro, numero di videocamere nella rete, protocollo di comunicazione

utilizzato. La tabella 6.1 mostra il numero di iterazioni per il raggiungimento della configurazione finale in varie simulazioni effettuate, su diversi perimetri al variare del numero di videocamere della rete.

Perimetro	numero videocamere	iterazioni Gossip Simmetrico	iterazioni Gossip Asimmetrico
quadrato	4	40	110
quadrato	8	190	460
realistico	22	1100	3000

Tab. 6.1: Numero di iterazioni per raggiungere la configurazione finale in varie simulazioni di AP effettuate.

La seconda strategia di partizionamento proposta trova la soluzione ottima di  $\mathcal{P}_4$ : la partizione che individua oltre a fornire  $T_{lag}$  prossimi al minimo, garantisce alle videocamere della rete una visuale perpendicolare sulle proprie PA. In generale esiste un trade-off fra ottimizzazione del pattugliamento e ottimizzazione della visuale:

- (a) la minimizzazione di  $T_{lag}$  richiede PA proporzionali alle velocità delle rispettive videocamere;
- (b) la minimizzazione di  $P_\Psi$  richiede PA proporzionali alle aree (5.2) sottese alle funzioni angolo d'incidenza 2.18: ampie per videocamere con una vista intrinsecamente perpendicolare del perimetro e ridotte per quelle con vista obliqua.

L'importanza della qualità della visuale delle videocamere nella determinazione della partizione ottima trovata da  $\text{AP}\Psi$ , può essere variata agendo sul parametro  $C_{area}$ : per valori elevati la configurazione ha caratteristiche di tipo (a), per valori ridotti di tipo (b). Nella pratica si vuole che la minimizzazione di  $T_{lag}$  sia sempre dominante nella determinazione della configurazione ottima, poiché l'obiettivo principale resta evitare che intrusioni del perimetro non vengano rilevate, quindi si sceglie un  $C_{area}$  che non incrementi il  $T_{lag}^*$  minimo della struttura di più del 20%. La velocità di convergenza di  $\text{AP}\Psi$  dipende da vari fattori: numero di videocamere nella rete e tipo di protocollo di comunicazione utilizzato. In tabella 6.2 si mostra il numero di iterazioni per il raggiungimento della configurazione finale in varie simulazioni effettuate, su diversi perimetri, al variare del numero di videocamere della rete.

Perimetro	numero videocamere	iterazioni Gossip Simmetrico	iterazioni Gossip Asimmetrico
quadrato	4	40	150
quadrato	8	100	220
realistico	22	1000	3000

Tab. 6.2: Numero di iterazioni per raggiungere la configurazione finale in varie simulazioni di  $\text{AP}\Psi$  effettuate.

### 6.1 *Sviluppi futuri*

Per rendere il pattugliamento del perimetro efficace, una volta individuato il partizionamento ottimo occorre sincronizzare il movimento delle videocamere in modo che:

1. abbiano tutte lo stesso periodo di pattugliamento  $\bar{T}_0$

$$\bar{T}_0 = \max_{i=1,\dots,N} T_{0,i} = T_{lag} ;$$

2. si muovano in opposizione di fase.

In questo modo qualsiasi intruso intelligente, cioè in grado di spostarsi lungo il perimetro per evitare di entrare nel campo visivo delle videocamere, viene rilevato al più dopo un tempo pari a  $\bar{T}_0$ . Si lascia a lavori futuri lo sviluppo di una tecnica per raggiungere gli obiettivi 1. e 2. agendo durante le iterazioni di  $\text{AP}$  e  $\text{AP}\Psi$ .

Uno dei punti deboli di  $\text{AP}\Psi$  è la complessità: per l'aggiornamento delle PA viene richiesto di calcolare diverse funzioni tra cui  $\alpha_i(\cdot, t)$  e  $\alpha_{i,inv}(\cdot, t)$  che non rimangono costanti, inoltre è necessaria un'accurata progettazione dei parametri  $C_{centr}$  e  $C_{area}$  per fornire alla configurazione ottima le caratteristiche volute. Apparentemente è possibile ottenere gli stessi risultati di  $\text{AP}\Psi$ , in particolare garantire una visuale perpendicolare alle videocamere, utilizzando un algoritmo frutto di un'intuizione, in cui l'aggiornamento delle PA unisce l'approccio angolare proposto in [4] con il concetto di angolo visuale d'incidenza. Senza soffermarsi troppo sul funzionamento dell'algoritmo si espongono i suoi principali punti di forza:

- semplicità della regola di aggiornamento;
- usa solo angoli, risulta più semplice integrare il termine relativo alle funzioni  $\psi_i$ ;

- converge a una soluzione ottima del tipo  $\mathcal{P}_4$ , in cui le videocamere hanno una buona visuale sulle proprie PA.

La formalizzazione matematica del funzionamento dell'algoritmo assieme ad una prova della sua convergenza sembrano gli aspetti più interessanti su cui focalizzare future ricerche.

## BIBLIOGRAFIA

- [1] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato, “Distributed perimeter patrolling and tracking for camera networks,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 2093–2098, IEEE, 2010.
- [2] R. Carli, A. Cenedese, and L. Schenato, “Distributed partitioning strategies for perimeter patrolling,” in *American Control Conference (ACC), 2011*, pp. 4026–4031, IEEE, 2011.
- [3] R. Alberton, R. Carli, A. Cenedese, and L. Schenato, “Multi-agent perimeter patrolling subject to mobility constraints,” in *American Control Conference (ACC), 2012*, pp. 4498–4503, IEEE, 2012.
- [4] R. M. Gianluca Giorgini, Mattia Sessolo, “A partitioning strategy for a perimeter to be surveilled by a camera network, and development of coordination and synchronization algorithms for patrolling,”
- [5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [6] F. Bullo, R. Carli, and P. Frasca, “Gossip coverage control for robotic networks: dynamical systems on the space of partitions,” *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 419–447, 2012.
- [7] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, “Consensus-based distributed intrusion detection for multi-robot systems,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 120–127, IEEE, 2008.
- [8] I. Necoara, V. Nedelcu, and I. Dumitrache, “Parallel and distributed optimization methods for estimation and control in networks,” *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, 2011.

- [9] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications*. Cambridge university press, 2010.
- [10] F. Pasqualetti, A. Franchi, and F. Bullo, “On optimal cooperative patrolling,” in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 7153–7158, IEEE, 2010.
- [11] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo, “Equitable partitioning policies for robotic networks,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, pp. 2356–2361, IEEE, 2009.
- [12] M. M. Zavlanos, “Distributed control of robotic networks,” 2008.