



*Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni*

Smart Object e forme di cooperazione

Andrea Giordano,
Giandomenico Spezzano,
Andrea Vinci

RT-ICAR-CS-14-01

Febbraio 2014



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: www.icar.cnr.it
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: www.icar.cnr.it
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: www.icar.cnr.it

Sommario

Gli Smart Object costituiscono la base fondamentale per la realizzazione dell'Internet Delle Cose (Internet of Things). Essi sono oggetti fisici aumentati con capacità di computazione e di rete, e che sono quindi in grado di comunicare attraverso Internet con servizi remoti o altri oggetti. Nel documento sono dettagliate le loro caratteristiche e presentati differenti modelli di cooperazione. Sono inoltre descritti possibili casi d'uso e scenari applicativi.

Indice

1. Introduzione	5
2. Smart Object.....	6
2.1 Tipologie di Smart Object	7
2.2 Sfide.....	10
3. Smart Object e Internet delle Cose	10
4. Sistemi di smart objects	13
5. Modello smart object	15
6. Forme di comunicazione e cooperazione.....	18
6.1 Event-based smart object.....	18
6.2 Self-adaptive smart object.....	18
6.3 Tecniche ed Algoritmi di Swarm Intelligence	20
a) Ant Colony Optimization	20
b) Artificial Bee Colony	21
c) Artificial Immune System.....	22
d) Negative Selection Algorithm	22
e) Clonal Selection Algorithm	22
f) Particle Swarm Optimization	22
g) Firefly-inspired Synchronicity	23
h) River Formation Dynamics e Intelligent Water Drop	23
i) Digital Infochemical	23
j) Division of labour.....	24
k) Gossip based algorithms.....	25
l) Flocking.....	25
7. Casi di studio.....	26

7.1 Smart Street.....	27
7.1.1 Smart Basket Network.....	28
7.1.2 Rete di gestione code	28
7.1.3 Open-air Museum.....	29
7.1.4 Eco-shop	29
7.1.5 Smart Advertising System.....	29
7.1.6 Sistema di sorveglianza intelligente	30
7.2 Water grid.....	30
7.2.1 Water Distribution Networks (WDN)	30
7.2.2 Indirect defense strategy.....	32
7.2.3 Algoritmo di indirect defense strategy applicato alle WDN	33
7.3 Monitoraggio dell'inquinamento acustico	33
7.3.1 L'approccio a "Macro sensori virtuali"	35
7.3.2 Algoritmo per la formazione delle regioni	36
7.3.3 Aggregazione	38
7.3.4 Situazioni dinamiche e transitorie.....	39
7.3.5 Simulazione del monitoraggio dell'inquinamento acustico	40
8 Bibliografia.....	45

1. Introduzione

Il termine internet of things (IoT) è recentemente diventato popolare per l'enfatizzata visione di un'infrastruttura globale di oggetti interconnessi tra loro, in grado di fornire connettività *anytime* ed *anyplace*, non soltanto per gli utenti umani ma anche per le cose.

Internet of things è stata parzialmente ispirata dal successo della tecnologia RFID, largamente usata per tracciare oggetti, persone e animali. L'architettura del sistema RFID è caratterizzata da una netta dicotomia tra semplici tag RFID e, un'estesa, infrastruttura di rete di lettori RFID. Questo approccio supporta in modo ottimale il monitoraggio di oggetti fisici all'interno di confini ben stabiliti (come un magazzino), ma limita le capacità di rilevamento e la flessibilità d'implementazione in scenari applicativi più impegnativi. Per queste ragioni si sta andando verso un modello, di l'IoT, alternativo che sia *loosely coupled*. Un sistema decentralizzato di oggetti intelligenti (*smart object*); oggetti fisici/digitali, estesi con capacità di rilevamento (*sensing*), elaborazione (*processing*) e capacità d'interconnessione (*network capabilities*). In contrasto ai tag RFID, gli *smat object* inglobano parti di una logica applicativa, che consente loro di percepire la situazione locale e interagire con gli utenti umani. Essi sono in grado di *percepire, memorizzare e interpretare*, cosa accade nel mondo, sono *proattivi*, comunicano tra loro e scambiano informazioni con le persone. La visione di un'IoT costituita da smart object solleva molte questioni importanti sul tipo di: architetture, progettazione e sviluppo di sistemi e coinvolgimento umano.

L'internet of things potrebbe diventare il livello base di una nuova infrastruttura di rete, in grado di cambiare il concetto di internet. Molte sono le analogie con quella che è stata Internet agli albori, e proprio come l'Internet di due decenni fa, ancora non si conoscono quali saranno gli scenari che interesseranno questo sistema.

Nel seguito sarà presentato un modello di smart object e le varie forme di cooperazione definite. Il modello è visto in un'ottica di Internet delle cose per favorire la comunicazione degli smart object mentre un sistema multi-agente è adottato per realizzare la fase di "smartness" degli oggetti rappresentati come un ecosistema di oggetti cooperanti attraverso meccanismi basati sugli eventi e forme avanzate di cooperazione derivanti dall'utilizzo di algoritmi di intelligenza collettiva distribuita.

Infine, a dimostrazione della validità del modello sono presentati alcuni casi studio riguardanti varie applicazioni per rendere una strada smart, un modello di distribuzione dell'acqua dove è utilizzato un algoritmo di swarm intelligence basato su *digital infochemicals* per effettuare un controllo automatico del sistema e il monitoraggio dell'inquinamento acustico in una zona della città dove si fa uso di un modello auto-organizzante di macro sensori virtuali per suddividere le zone rumorose in regioni che evolvono dinamicamente.

2. Smart Object

La tecnologia smart object, frutto della convergenza, come mostrato in figura 1, di diversi campi applicativi, è conosciuta con molti nomi, *web of things*, *cooperation objects*, *web of objects*, spesso indicata con il termine più generale di Internet of things. Lo smart object, è un oggetto (può essere un oggetto di uso quotidiano, come una sedia, un ombrello, una penna) equipaggiato con sensori e/o attuatori, una piccola unità di elaborazione, un dispositivo per la comunicazione (nella maggior parte dei casi wireless) e una fonte di alimentazione. Gli attuatori e i sensori forniscono agli smart object la possibilità di interagire con il mondo fisico. Il microprocessore consente di manipolare i dati catturati dai sensori. Il dispositivo di comunicazione consente di comunicare con il resto del mondo (es. scambio di dati con altri smart object). L'alimentazione fornisce energia elettrica allo smart object. Lo smart object è molto più piccolo sia dei laptop che dei cellulari, perché deve poter essere integrato sugli oggetti di uso comune, per cui le sue dimensioni non devono essere superiori a pochi centimetri cubici.

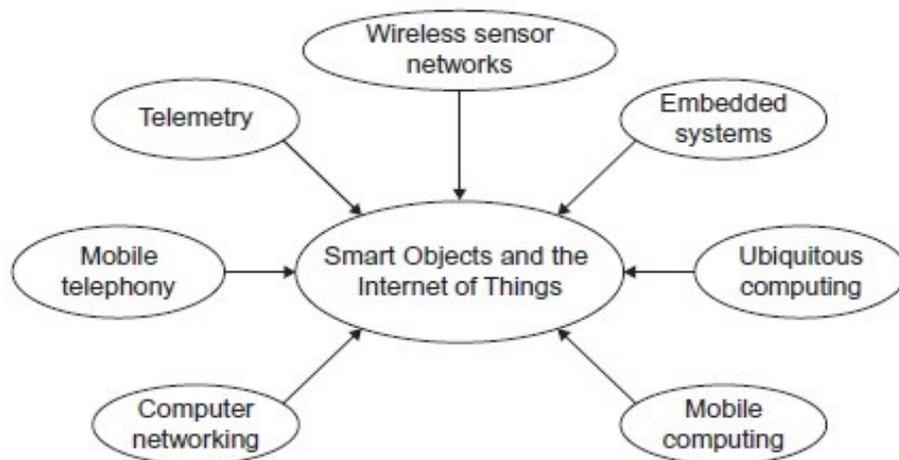


Figura 1. I diversi campi applicativi interessati dall'IoT

Questa descrizione non mette in risalto le possibilità e le capacità degli smart object, anche perché il comportamento di uno smart object è strettamente dipendente dal contesto e/o ambiente in cui opera. Uno smart object in un frigo avrebbe compiti differenti rispetto ad uno smart object in una tv. Ma pensare ad un singolo, o ad una comunità ristretta di smart object è molto riduttivo; ciò che rende davvero interessante questi oggetti è la loro capacità di comunicare con altri smart object. Lo scambio di informazioni rende possibile l'adattamento alla dinamicità del mondo fisico; inoltre uno smart object ha due identità, una appartenente al mondo reale, ed una immersa nel mondo virtuale. Anche se non si conoscono quali potranno essere i comportamenti di uno smart object, si sa che esistono due caratteristiche che

accomunano tutti questi oggetti: la capacità di relazionarsi con il mondo reale e la capacità di comunicare con il mondo virtuale.

2.1 Tipologie di Smart Object

Nella figura 2 vengono descritti vari scenari alternativi per la progettazione di smart object. Anche se, per andare incontro alle richieste computazionali e all'emergere di nuovi standard, sono state sviluppate molte piattaforme hardware, le tecniche di progettazione dell'hardware sono rimaste sostanzialmente le stesse. Le differenze tra le loro progettazioni siano riconducibili a tre aspetti chiave:

- Consapevolezza (*Awareness*) è l'abilità di uno smart object di capire (nel senso di percepire interpretare e reagire) gli eventi e le attività umane che si verificano nel mondo fisico,
- Rappresentazione (*Rappresentation*) è l'abilità di rappresentazione, riferita all'applicazione di uno smart object e al modello di programmazione, in particolare alla programmazione astratta,
- Interazione (*Interaction*) denota l'abilità che uno smart object ha, di conversare con un utente in termini di input, output e feedback.

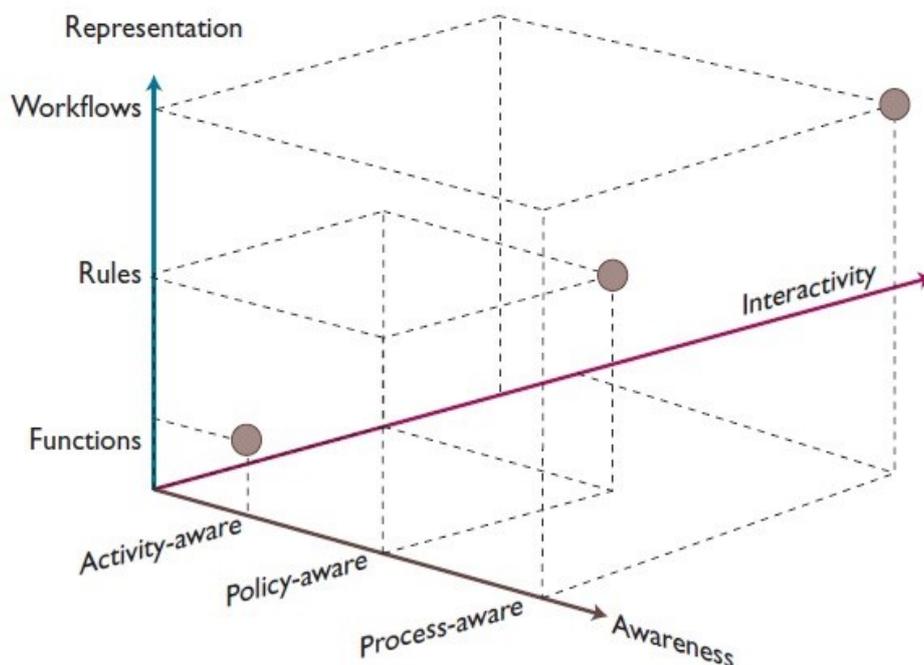


Figura 2. Le tre dimensioni di uno smart object. Si possono vedere anche i tre tipi di oggetti *activity-aware*, *policy-aware*, e *process-aware*.

Attraverso un'analisi di vari progetti, è stato notato come, la maggior parte dei progetti utili ruota attorno a tre tipi di oggetti chiavi. La combinazione delle tre dimensioni, descritte in precedenza, con le tre tipologie di oggetti è rappresentata in figura 3.

	Awareness	Representation	Interaction	Augmentation	Example application
Activity-aware object	Activities and usage	Aggregation function	None	Time, state (on/off), vibration	Pay-per-use
Policy-aware object	Domain-specific policies	Rules	Accumulated historical data, threshold warnings	Time, vibration, state, proximity	Health and safety
Process-aware object	Work processes (that is, sequence and timing of activities and events)	Context-driven workflow model	Context-aware task guidance and alerts	Time, location, proximity, vibration, state	Active work guidance

Figura 3. Le possibili combinazioni tra tipi di oggetti e le tre dimensioni d'interesse.

Activity-Aware Smart Object

Un *activity-aware* object può memorizzare informazioni sulle attività e sull'uso che ne viene fatto, ed è così caratterizzato:

- *Awareness*. Un *activity-aware* object percepisce il mondo in termini di flussi di eventi e attività, dove ogni evento o attività è direttamente relazionato all'uso e alla gestione di oggetti (accendere, mettere in funzione, etc).
- *Representation*. Il suo modello applicativo è costituito di funzioni di aggregazione, per accumulare attività nel tempo.
- *Interaction*. L'*activity-aware* object mantiene principalmente un log dei dati e non è dotato di capacità interattive.

Gli oggetti *activity-aware* sono i più semplici, ma possono supportare interessanti applicazioni, come un tool per calcolare il costo del noleggio di strumenti: i sensori memorizzano i dati sul tempo e la durata e il tipo di utilizzo degli strumenti.

Tecnicamente uno smart object di questo tipo analizza il flusso di dati proveniente dai propri sensori, usando algoritmi di riconoscimento per individuare attività ed eventi, applicando funzioni di aggregazione *application-specific*.

Policy-Aware Smart Objects

Un *Policy-Aware* Smart Object è un *activity-aware* object che è in grado di interpretare eventi e attività, rispettando politiche organizzative predefinite.

- *Awareness.* Un *policy-aware object* comprende fino a che punto le attività e gli eventi del mondo reale siano conformi con le politiche dell'organizzazione.
- *Representation.* Il modello applicativo è costituito da un insieme di regole che operano su flussi di eventi o attività per creare azioni.
- *Interaction.* L'*activity-aware object* fornisce informazioni context-sensitive sulla gestione dell'oggetto e sulle performance dell'attività lavorativa. In particolare, può emettere warning e alert nel caso i lavoratori ne stiano violando le politiche.

Questo tipo di smart object può essere utilizzato, ad esempio, nel campo della salute e della sicurezza, per lo sviluppo di applicazioni che tengano sotto controllo depositi chimici, o il funzionamento di macchinari, prevedendo scenari di rottura, prevenendo così danni e incidenti.

Tecnicamente un *policy-aware object* è un *activity-aware object* con aggiunto un modello integrato di *policy*. L'interfaccia utente è un'importante aspetto dei *policy-aware object*; questi oggetti non solo memorizzano e interpretano i dati, ma forniscono anche informazioni tempestive. In questo senso, i *policy-aware object* si possono considerare come sistemi interattivi.

Process-Aware Smart Objects

I processi giocano un ruolo fondamentale nella gestione e nell'organizzazione del lavoro industriale. Un processo è un insieme di attività, o *task*, correlate, ordinate in base alla loro posizione nel tempo e nello spazio. Un *process-aware object* rappresenta il più completo tra i tre tipi descritti, ed è caratterizzato come segue:

- *Awareness.* Un *process-aware object* comprende i processi organizzativi di cui è parte, e può comunicare, a questi processi, gli eventi e le attività che accadono nel mondo reale.
- *Representation.* Il suo modello applicativo è costituito da un modello di *workflow context-aware* che definisce la tempistica e l'ordine delle attività lavorative.
- *Interaction.* Un *process-aware object* fornisce ai lavoratori indicazioni *context-aware* su processi, deadline e decisioni.

Questo tipo di smart object consente quindi la creazione di tool che possano aiutare i lavoratori, attraverso la comunicazione di informazioni just-in-time sulle attività di lavoro necessarie.

Le tre dimensioni sviluppate per la progettazione di smart object forniscono un'approccio strutturato. Il giusto bilanciamento tra rappresentazione, consapevolezza e interattività dipende da ciò che lo scenario applicativo richiede; progettazioni più complesse e astratte non sono sempre le migliori.

2.2 Sfide

Ogni nuova tecnologia introduce sfide tecniche e non, che vanno affrontate, nello sviluppo degli smart object. Alcune di queste sfide sono nuove, proprie dell'area di ricerca sugli smart object, ma molte sono le stesse che affliggono sistemi esistenti. Le sfide tecniche includono i node-level di ogni smart object, le dimensioni, il consumo di energia, così come i meccanismi a livello di rete e le strutture formate dagli smart object. A rendere le cose più complesse contribuisce il fatto che i vari problemi si influenzano tra loro. Ad esempio il problema energetico di uno smart object è reso più complicato dai pattern di comunicazione della rete, alla quale gli smart object partecipano, questo perché alcuni protocolli consumano molta energia. Allo stesso modo, progettare nuovi protocolli di rete per smart object, significa anche, tener conto del consumo di energia, che l'invio dei dati comporta, inoltre, lo sviluppo di protocolli proprietari ha compromesso l'interoperabilità tra smart object e il mondo esterno (a meno di usare gateway tra le reti).

Ci sono anche problemi non tecnici, che vanno affrontati prima che la tecnologia degli smart object possa essere adottata. I problemi non tecnici sono più ostici di quelli tecnici, perché quest'ultimi possono essere risolti dalla progettazione di un protocollo, ma le sfide che riguardano l'area non strettamente tecnica riguardano, invece, la diffusione e la consapevolezza della tecnologia. Senza la consapevolezza anche la tecnologia meglio progettata non riuscirà mai ad avere un grande impatto su larga scala. L'Internet Protocol for Smart Object (IPSO) Alliance è stata fondata con l'obiettivo di diffondere la consapevolezza della tecnologia che orbitano intorno agli smart object.

3. Smart Object e Internet delle Cose

Come noto, l'Internet delle Cose (IOT) è un paradigma di sviluppo tecnologico in base al quale potenzialmente ogni oggetto della nostra vita quotidiana assume una sua identità digitale, attraverso la rete Internet. Alla base dell'IOT vi sono quindi gli oggetti "intelligenti". Non ha senso parlare di IOT, se gli oggetti non hanno al loro interno un certo livello di intelligenza. Ma come si struttura questa intelligenza?

Secondo la School of Management del Politecnico di Milano, che ha sviluppato l'Osservatorio Internet of Things, l'intelligenza degli oggetti si declina lungo tre direzioni: *funzionalità di self-awareness*, *funzionalità di interazione* con l'ambiente circostante e *funzionalità di elaborazione dati*. Nel dettaglio le tre funzionalità sono:

a. Funzionalità di self-awareness

Questa particolare funzionalità può essere a sua volta vista lungo tre direzioni:

- Identificazione. L'oggetto deve essere in possesso di un identificativo digitale.

- Univoco, cioè di una funzionalità di base riconosciuta in tutte le applicazioni IOT.
- Localizzazione. Misura la capacità di conoscere la propria posizione, o in tempo reale (funzione di real time location) oppure elaborando informazioni di tracciabilità nel processo produttivo o logistico.
- Diagnosi stato. Misura la capacità di monitorare parametri interni all'oggetto (per comprenderne il corretto funzionamento e le necessità di assistenza).

b. Funzionalità di interazione con l'ambiente circostante

La capacità di interagire con l'ambiente si declina lungo 2 direzioni:

- Acquisizione dati (Sensing e Metering). Le misure di sensing indicano le variabili di stato che descrivono il sistema fisico circostante come ad esempio temperature, pressione, accelerazione, rumore. Quelle di metering misurano le variabili di flusso (consumo di energia elettrica, calore, gas acqua).
- Attuazione. Misura la capacità di eseguire comandi impartiti da remoto o derivati dall'elaborazione dei dati in loco.

c. Funzionalità di elaborazione dati

Rappresenta la capacità di trattare le informazioni, vista su due livelli:

- Elaborazione base, cioè il trattamento del dato primitivo raccolto, ad esempio tramite filtraggio, aggregazione algebrica, conversione, crittografia.
- Elaborazione avanzata, cioè l'analisi di informazioni dal dato primitivo, ad esempio tramite analisi statistiche, inferenze, previsioni.

Queste tre funzionalità rappresentano la base per stimare il livello di intelligenza di un oggetto che possa definirsi "smart object". A ciò si deve aggiungere una quarta funzionalità, ma che in realtà è una vera e propria "conditio sine qua non": la capacità di connessione, tramite cavo o wireless. Alcuni SO possono anche interagire con altri SO o con gli utenti per raggiungere il proprio goal.

Uno smart object può essere definito, come mostrato in figura 4, come l'estensione di una *Physical Entity* con un *Digital Proxy* associato. Una *Physical Entity* rappresenta un qualsiasi oggetto del mondo fisico di interesse per l'utente può essere utilizzato nel mondo digitale da una *Digital Entity* attraverso un *Digital Proxy*. La *Digital Entity* è un'entità software che può essere un agente con un comportamento autonomo.

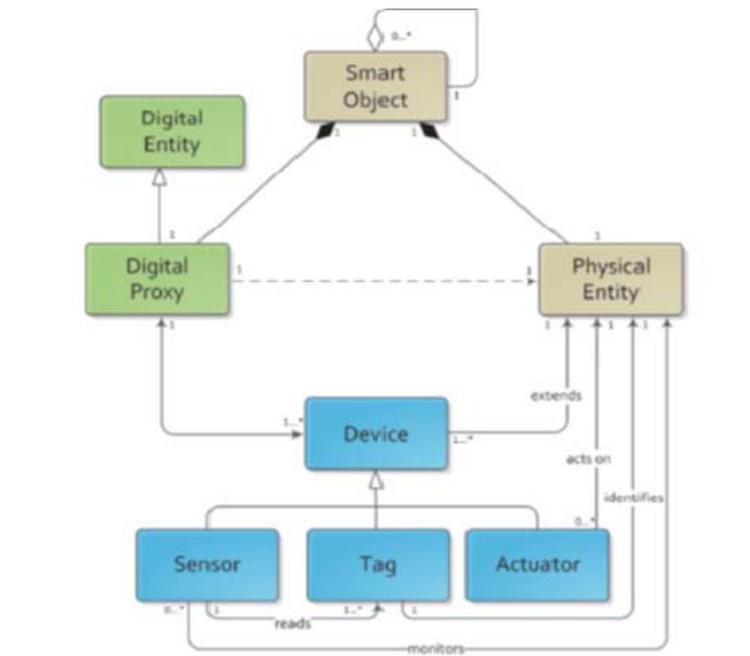


Figura 4. Modello concettuale di uno Smart Object.

Nel contesto dell'IOT, i *Digital Proxy* hanno due proprietà fondamentali:

- essi sono *Digital Entity* che sono biunivocamente associati alle entità fisiche che essi rappresentano. Ogni *Digital Proxy* deve avere uno e un solo ID che identifica l'oggetto rappresentato. L'associazione fra il Digital Proxy e l'entità fisica deve essere stabilita automaticamente.
- essi sono una rappresentazione sincronizzata di un dato set di aspetti (o proprietà) della *Physical Entity*. Questo significa che i parametri fisici rappresentanti le caratteristiche dell'entità fisica possono essere aggiornati sopra ad ogni cambiamento del primo. Allo stesso modo, cambiamenti che influenzano il *Digital Proxy* potrebbero manifestarsi sull'entità fisica nel mondo fisico.

Da un punto di vista funzionale il *Device* ha tre sottotipi:

- *Sensori* che forniscono informazioni circa la *Physical Entity* che essi monitorano.
- *Tags* sono usati da sensori specializzati usualmente chiamati reader per supportare l'identificazione del processo. Il processo può essere ottico come nel caso di barcode e Qr-code può essere quello usato dai sensori tipo RFID.
- *Attuatori* che possono modificare lo stato della *Physical Entity*.

E' interessante notare che, poiché gli oggetti quotidiani possono essere logicamente raggruppati insieme per formare un oggetto composto e come oggetti complessi possono essere divisi in componenti, lo stesso è anche vero per le *Digital Entity* e gli Smart Object che possono essere logicamente raggruppati in maniera strutturata, spesso gerarchica.

Gli utenti che hanno necessità di interagire con loro lo devono fare attraverso l'uso di *Risorse*. Le *Risorse* sono digitali, componenti identificabili che implementano differenti capability e sono associate a entità digitali, specificatamente a Digital Proxy nel caso di IOT. Più di una risorsa può essere associata a un proxy digitale e perciò a uno smart Object. Per ottenere l'interoperabilità, poiché essi possono essere eterogenei e le implementazioni possono essere altamente dipendenti dall'hardware dei device, l'accesso alle Risorse è previsto come servizi.

L'associazione fra gli Smart Object e le Risorse (cioè la loro identità) e le locazioni (cioè l'indirizzo di rete) dei Servizi relativi è o registrato negli Smart Object stessi o può essere memorizzato (insieme con l'aggiunta di piccolo ammontare di informazione ausiliaria) in quello che è il servizio di risoluzione un componente infrastrutturale dell'Internet delle Cose. Il servizio di risoluzione è concepito come un registro. Il suo compito è molto simile a quello del corrente servizio DNS: esso prende come input l'ID di uno Smart Object o Risorsa e fornisce come output gli indirizzi di rete del servizio associato a esso.

4. Sistemi di smart objects

In generale, gli smart object operano individualmente, o sono collettivamente integrati da applicazioni o collaborano con *peer* per ottenere uno specifico proposito. Quando lavorano collettivamente, una rete di smart object è spesso riferita come un sistema di smart object. In genere i sistemi di smart object possono essere osservati da tre prospettive:

1. Oggetti stand-alone e auto contenuti che sono indipendenti da ogni infrastruttura e sono capaci di percezione, ragionamento e possono prendere decisioni autonomamente con appropriati feedback percettivi (audio o visuali) vedi figura 5.

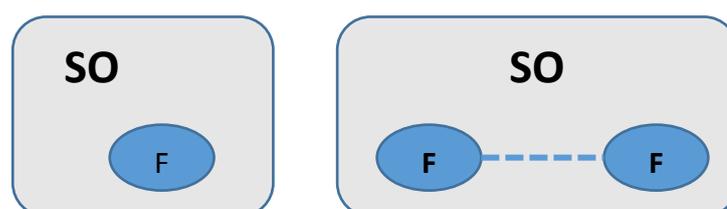


Figura 5. Oggetti stand-alone con una o multiple funzionalità.

2. Applicazioni che integrano multipli smart object, specificando l'interazione fra smart object in modo da fornire alcuni servizi proattivi (vedi figura 6). Questa applicazione è eseguita da un'altra entità che orchestra gli smart object. Tipicamente un'infrastruttura back end è utilizzata dalle applicazioni per l'integrazione degli smart object. Esempi includono smart space con multipli smart object, etc.

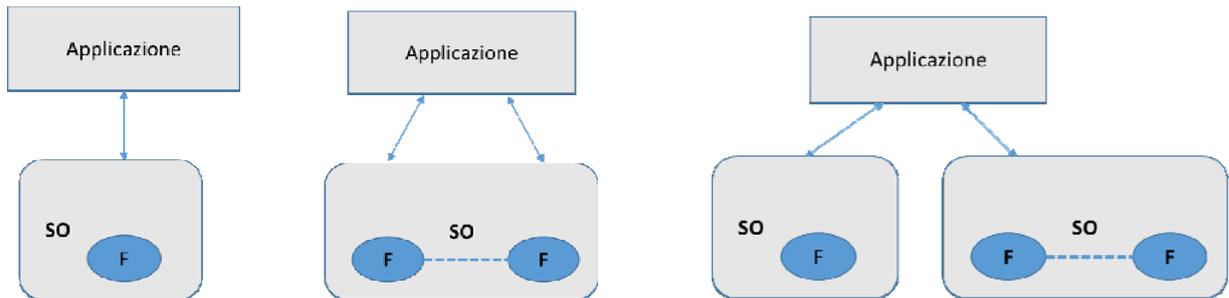


Figura 6. Un'applicazione usa uno o più oggetti con una o multiple funzionalità.

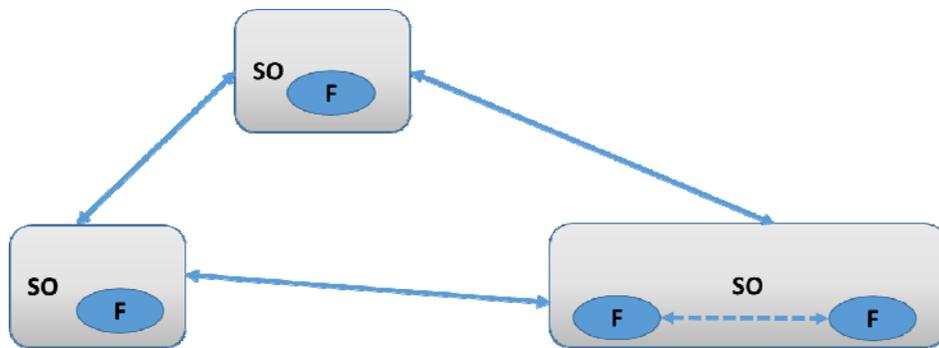


Figura 7. Oggetti cooperativi con una o multiple funzionalità.

3. La terza prospettiva, mostrata in figura 7 si pone in mezzo, gli smart object stand-alone sono anche capaci di comunicare con peer per prendere azioni autonome perciò creando una ecologia cooperativa di smart object.

Al momento noi siamo interessati a progettare sistemi di smart object secondo la modalità illustrata in figura 6 e 7.

L'Internet delle Cose è considerato un paradigma del futuro dove la principale sfida è dare un accesso trasparente di tipo IP a un gran numero di servizi disponibili come risorse IOT. Dovuto al grande numero di device e alla natura dinamica degli ambienti IoT, questo problema di integrazione è piuttosto intricato. Lo stato corrente è principalmente focalizzato sull'integrazione di smart object dotati di IP sulla base dell'architettura SOA e lo stile architetturale REST. Tuttavia, oltre questi approcci è interessante considerare la composizione adattativa di servizi e smart oggetti esplorando i concetti di orchestrazione dei servizi e coreografia.

Nelle architetture SOA, la composizione dei servizi è raggiunta o attraverso un controllore centralizzato o in maniera decentralizzata. Il primo approccio (conosciuto come orchestrazione), ipotizza l'esistenza di un singolo servizio, chiamato mediatore, che ha il ruolo specializzato di interagire con i servizi disponibili sul web e controllare l'esecuzione di altri servizi che partecipano nella composizione, mentre l'altro approccio (conosciuto come coreografia) è privo del concetto di servizio mediatore. Ogni servizio partecipante può eseguire una composizione (parziale) dei servizi in maniera peer-to-peer. Il modello a coreografia è più scalabile e esibisce un comportamento efficiente rispetto ai costi di elaborazione e comunicazione inviando messaggi direttamente dal servizio produttore al consumatore. In questo contesto, lo sforzo è quello di indagare nuovi approcci per la composizione di servizi e smart objects. Recenti sforzi di ricerca stanno indagando su come integrare reti smart object utilizzando architetture SOA e tecniche basate su REST. Con l'emergenza del protocollo IPv6 è ora possibile equipaggiare ogni nodo sensore con un unico indirizzo IP.

Oltre questi approcci è possibile indagare anche modelli basati su sistemi multi-agente. Noi siamo interessati alla composizione adattativa di servizi e smart object esplorando l'utilizzo di sistemi multi-agente dotati di intelligenza collettiva che possano rappresentare in un sistema cyber-physical il collante fra il mondo fisico e quello digitale fornendo al contempo la "smartness" degli oggetti.

5. Modello smart object

Il modello di smart object proposto è strutturato su più livelli. Al livello più basso uno smart object è costituito da uno o più dispositivi che insieme sono in grado di fornire comportamenti complessi.

Ogni smart object comprende le "funzionalità" fornite direttamente dalla parte fisica. Per fornire una rappresentazione standard ogni oggetto del mondo reale è rappresentato come un oggetto virtuale. Un oggetto virtuale è la rappresentazione software di un oggetto reale fisico (sensore, attuatore, etc.) che deve diventare virtualmente "sempre attivo". Gli oggetti virtuali possono essere creati e distrutti dinamicamente. Essenzialmente un oggetto virtuale fornisce una rappresentazione astratta delle proprietà, caratteristiche e funzionalità degli oggetti fisici presenti in uno smart environment. Le funzionalità/servizi offerte da differenti tipi di oggetti virtuali possono essere combinate in una maniera più sofisticata sulla base di regole event-driven. Ogni oggetto virtuale è in grado di eseguire semplici azioni (illuminarsi, registrare, etc.) in risposta a semplici eventi (rilevare la presenza di una persona o

una persona che è seduta su una sedia). Ogni oggetto virtuale può cambiare dinamicamente il proprio comportamento da remoto riprogrammando gli oggetti fisici associati. Inoltre ogni oggetto virtuale è in grado di fornire informazioni sullo stato, fornire misure e effettuare azioni.

Ad ogni oggetto virtuale è associato, al livello più alto, uno smart object agent (SOAg) che realizza la "smartness" dell'oggetto. L'agente utilizza un meccanismo publish/subscribe per interagire con gli oggetti virtuali. Ogni agente riceve dall'applicazione una o più regole costituite da eventi complesse che scompone e associa ai vari oggetti virtuali che sono in grado di comunicare gli eventi elementari. Gli eventi generati dai vari oggetti virtuali vengono pubblicati nel sistema publish/subscribe che provvede, sulla base delle regole definite nell'applicazione, a notificarli agli agenti che si sono sottoscritti. In questo modo un agente interagisce con un ben definito numero di oggetti virtuali.

Lo SOAg può anche ricevere dati dagli oggetti virtuali. Tali dati sono analizzati, filtrati e aggregati, e quelli di interesse per le analisi storiche sono compressi e inviati al Cloud per essere utilizzati dagli strumenti di Analytics e Forecasting.

La piattaforma per l'esecuzione degli SOAg per motivi di efficienza è posta in esecuzione su un computer (*raspberry*) che è posto nelle vicinanze degli oggetti che è necessario monitorare o controllare. Lo SOAg può attraverso l'API definita nella piattaforma interagire con gli oggetti virtuali la cui rappresentazione software è presente sul *raspberry*. Attraverso opportune operazioni, che standardizzano l'accesso, è possibile conoscere lo stato dell'oggetto virtuale, analizzarne il contesto, modificarne i parametri di funzionamento ed effettuare attuazioni. Naturalmente per effettuare operazioni di controllo o di monitoring lo SOAg può coordinarsi ad altri SOAg presenti sulla rete che rappresentano altri oggetti virtuali non disponibili localmente. Il coordinamento può avvenire attraverso un modello event-based o attraverso forme più avanzate di cooperazione come l'utilizzo di forme di coordinamento basate su tecniche di intelligenza distribuita che si basi su agenti bio-ispirati.

Per realizzare forme di funzionamento proattive lo SOAg può avvalersi di servizi disponibili sul Web che possono indicargli ad esempio le previsioni metereologiche per definire opportune politiche che gli permettano di anticipare o suggerire comportamenti.

Uno SOAg dispone anche di forme di visualizzazione avanzate, ad esempio per riportare i dati di un monitoraggio su google map o fornire grafici che indichino il comportamento temporale o spaziale di un fenomeno. Tali dati possono essere trasferiti ad uno SOAg che gestisce un display intelligente dove i dati possono anche essere adattati in funzione dell'utenza (maggioranza di persone di sesso femminile o bambini).

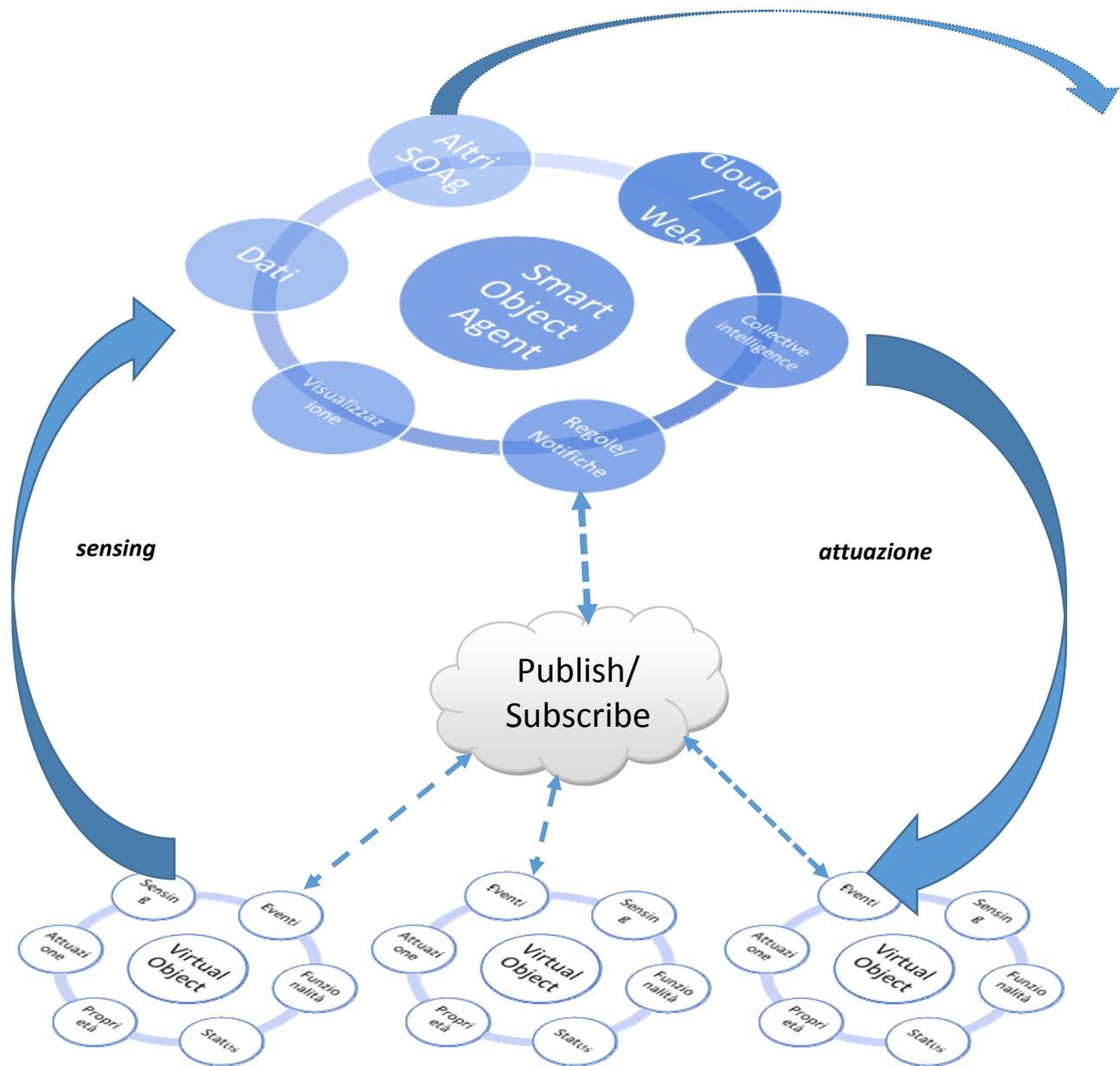


Figura 8. Modello di Smart Object.

Infine, uno SOAg può utilizzare il Cloud per farsi assistere nel suo funzionamento. Grazie alle capacità di calcolo scalabile è possibile che vari algoritmi da quelli di predizione a quelli di ottimizzazione o di analisi dei dati che utilizzino tecniche di data mining possano essere in esecuzione sul Cloud e fornire all'occorrenza dati ai vari SOAg che svolgono operazioni di controllo in real-time. Tale assistenza è molto importante per le operazioni di ottimizzazione e predizione che devono essere svolte in molti domini come quelli delle smart grid e più in generale di tutte le infrastrutture fisiche che sono presenti sul territorio di una città.

6. Forme di comunicazione e cooperazione

6.1 Event-based smart object

Come già specificato gli smart object possono utilizzare il meccanismo event-driven per coordinare il loro comportamento e descrivere situazioni prettamente reattive. E' possibile strutturare una computazione utilizzando il modello basato su eventi che attraverso il meccanismo publish/subscribe. Un'illustrazione completa di questa modalità di funzionamento è illustrata nel deliverable "Definizione dell'architettura software del sistema Cyber-physical per la gestione di infrastrutture energetiche".

6.2 Self-adaptive smart object

Un mondo fortemente instrumentato, interconnesso ed intelligente, necessita di strategie di coordinazione e cooperazione tra le numerose entità coinvolte.

Le problematiche di cooperazione e coordinazione tra entità multiple sono bene affrontate da strategie ed algoritmi basati su tecniche di swarm-intelligence (intelligenza di sciame). Tali algoritmi partono dall'assunto che sia possibile ottenere comportamenti complessi dalla cooperazione di una moltitudine di entità che attua comportamenti molto semplici. Si pensi, ad esempio, ad insetti come le formiche: il singolo elemento, la formica, non è certamente capace di ragionamenti profondi, ma, nell'insieme, un formicaio è da sempre riconosciuto come un esempio di operosità, efficienza e coordinazione. Si osserva, insomma, l'emergere di un comportamento complesso dalla somma delle attività di singole entità semplici.

Comportamenti emergenti, osservati in natura, sono caratteristici di numerose specie differenti, la cui evoluzione ha insegnato a trarre vantaggio dalla coordinazione e dalla aggregazione in gruppi più o meno numerosi, dalle colonie di insetti (formiche, api, termiti), ai banchi di pesci, agli stormi di uccelli.

Le principali caratteristiche di un algoritmo di swarm intelligence sono:

Località e conoscenza incompleta. Ogni individuo agisce in base ad informazioni ad esso direttamente disponibili, senza possedere una conoscenza completa della struttura del collettivo e dell'intero ambiente nel quale si muove. Nemmeno è presente un "supervisore" che possieda questo tipo di conoscenza. Ogni entità coinvolta agisce, quindi, secondo le sole informazioni locali che è in grado di percepire e collezionare, momento per momento.

Semplicità ed autonomia. Ogni individuo del gruppo, inoltre, segue un piccolo insieme di regole semplici. Il rispetto di tali regole consente la coordinazione delle attività di tutto il collettivo e la costruzione di strutture complesse.

Flessibilità e robustezza. Il collettivo riesce ad agire per risolvere problemi complessi, riuscendo a tollerare bene i "fallimenti" di individui singoli (es. morte o

ferimento di un individuo), e riuscendo ad adattarsi bene a nuove situazioni impreviste che possano crearsi nell'ambiente.

Comportamenti emergenti e self-organization. Comportamenti emergenti e sistemi auto-organizzanti possono essere, nel nostro caso, trattati come sinonimi. La capacità di auto-organizzazione di un sistema è definita come un insieme di meccanismi dinamici dove una struttura appare al livello globale del sistema a partire da interazioni locali delle componenti al livello più basso [bon99][kar08]. Fondamentali per descrivere un sistema auto-organizzante sono i concetti di struttura, multi-stabilità e transizioni di stato.

- **Struttura.** In un sistema auto-organizzante, una struttura emerge a partire da uno stato iniziale omogeneo dell'ambiente nel quale il collettivo opera (es. architettura dei nidi di termiti o api);
- **Multi-stabilità.** Il sistema può raggiungere uno tra più stati stabili a seconda delle condizioni iniziali e di parametri casuali (es. le formiche utilizzano una tra più fonti di cibo disponibili);
- **Transizioni di stato.** L'insieme di regole che ogni individuo segue, e quindi il comportamento del collettivo, possono variare drasticamente in base ad alcuni fattori (es. le termiti passano da un fase non coordinata ad una coordinata in base alla densità delle termiti che ogni individuo percepisce).

Come accennato in precedenza, le regole che specificano le interazioni e la coordinazione tra gli individui del sistema sono eseguite sulla base di una conoscenza locale. I meccanismi attraverso i quali gli individui si coordinano sono essenzialmente quattro:

- *Feedback positivi*, semplici regole che promuovono la costruzione di alcune strutture o l'attuazione di certi comportamenti.
- *Feedback negativi*, che controbilanciano quelli positivi e portano a stabilizzare il comportamento del collettivo, focalizzandolo su un certo stato stabile.
- *Fluttuazioni*, quali scelta casuale di percorsi, errori, cambiamento casuale dei task del singolo individuo, sono importanti per garantire l'esplorazione delle diverse possibilità e la scoperta di nuove e migliori soluzioni.
- *Interazioni Multiple*, che si hanno quando un individuo utilizza informazioni provenienti da altri individui, consentono alle informazioni di spargersi per tutto il sistema.

Le comunicazioni tra individui, che consentono la coordinazione, possono utilizzare due tipi di canali:

- *Canali diretti*, quando la comunicazione avviene tra individui che possono percepirsi direttamente (es. attraverso antenne, contatto visivo, contatto fisico)

- *Stigmergia*: gli individui comunicano indirettamente. Un individuo modifica l'ambiente e gli altri agiscono, in un secondo tempo, in funzione della percezione dell'ambiente modificato. (es. emissione di feromoni negli insetti, graffi sugli alberi nei felini, ecc.).

Stigmergia. In particolare la stigmergia costituisce un meccanismo interessante, in quanto utilizza l'ambiente come memoria per l'esecuzione di un lavoro laddove il lavoro stesso non dipende da un agente specifico. Il coordinamento tra gli individui coinvolti nell'esecuzione di un certa attività non dipende direttamente dagli individui, ma piuttosto dallo stato di avanzamento dell'attività e dell'ambiente nel quale essi operano. Un individuo non dirige una certa attività, piuttosto è guidato dall'attività stessa.

Stigmergia è, letteralmente, la contrazione delle parole "stigma" (segno, traccia) ed "ergon" (lavoro): è il lavoro stesso a costituire il canale di comunicazione e coordinazione tra gli individui. Prendiamo ad esempio le formiche: mentre si muove, una formica segna il suo passaggio rilasciando una certa quantità di feromone. Ogni formica sceglie una direzione in base alla quantità di feromone percepito e rilascia altro feromone. Elevate quantità di feromone tendono a concentrarsi sui percorsi più battuti, cioè quelli più brevi tra la tana ed una fonte di cibo.

6.3 Tecniche ed Algoritmi di Swarm Intelligence

Di seguito presentiamo alcuni tra i principali algoritmi di swarm intelligence attualmente utilizzati in diversi campi applicativi.

a) **Ant Colony Optimization**

Le formiche sono una famiglia di insetti in grado di risolvere in modo completamente decentralizzato il problema della ricerca dei cammini minimi. Abbiamo già descritto in precedenza come queste usino la meccanica della stigmergia:

- ogni formica, mentre si muove, rilascia una quantità di feromone;
- ogni formica sceglie con una probabilità maggiore la direzione dove sente più feromone;
- elevate quantità di feromone tendono a concentrarsi sui percorsi più battuti, ovvero quelli più brevi tra la tana ed una fonte di cibo.

Il feromone rilasciato garantisce un *feedback positivo* per la scelta di una direzione, e la formica sceglie dove muoversi in base ad una certa probabilità. La presenza di queste *fluttuazioni* casuali nella scelta dei percorsi garantisce l'esplorazione dell'ambiente. A mano a mano che il feromone si accumula su un certo cammino, ci sarà un momento nel quale avremo una *transizione di stato*: la quasi totalità delle formiche convergerà su un singolo cammino, ovvero su una singola soluzione del problema.

Approcci basati su Ant Colony Optimization, oltre ad essere usati per risolvere problemi di ottimizzazione, sono utilizzati anche per risolvere problemi di classificazione, routing distribuito [car98][car98b], data mining, image processing ed altri.

b) Artificial Bee Colony

Gli algoritmi ispirati al comportamento delle colonie di api sono relativamente nuovi nel panorama degli algoritmi di swarm intelligence. Essi si basano su meccanismi di ricerca di cibo e approvvigionamento esistenti negli alveari. Le componenti essenziali alla comprensione dell'algoritmo sono tre: sorgenti di cibo, raccoglitori occupati, raccoglitori non occupati.

- Sorgenti di cibo. Ad ogni sorgente di cibo è legato un valore che può dipendere da diversi fattori, quali la vicinanza all'alveare, la qualità del cibo, la facilità di estrazione. Per semplicità, possiamo però caratterizzare una sorgente con un unico parametro di *convenienza* che aggrega tutti gli altri.
- Raccoglitore occupato. Un raccoglitore è un individuo (ape) associato ad una certa sorgente di cibo, e che quindi sta raccogliendo presso quella sorgente. Un raccoglitore occupato porta all'alveare le informazioni sulla *convenienza* e sulla posizione della sorgente dalla quale raccoglie.
- Raccoglitore non occupato. Un raccoglitore non occupato è un individuo (ape) che non è associato a nessuna fonte di cibo. Può stazionare all'interno dell'alveare oppure, con probabilità minore (5-10%), esplorare l'ambiente circostante alla ricerca di una fonte. Se, esplorando, trova una fonte di cibo, il raccoglitore diventa occupato ed è associato a quella fonte.

Quando un raccoglitore occupato ritorna all'alveare, scambia informazioni sulla sorgente di cibo alla quale è associato con le altre api presenti nell'alveare (in natura, attraverso un meccanismo chiamato waggles dance). Ogni ape presente ha una probabilità di scegliere una certa fonte di cibo in base a due fattori:

- la qualità della sorgente,
- il numero di raccoglitori (occupati) già associati a quella sorgente: più numerosi sono i raccoglitori associati ad una sorgente, più essa sarà "pubblicizzata" all'interno dell'alveare.

Tali fattori rappresentano quindi una meccanica di *feedback positivo*. La presenza di una certa percentuale di esploratori garantisce la ricerca costante di sorgenti alternative di cibo.

Algoritmi basati su Artificial Bee Colony sono stati utilizzati in diversi contesti applicativi, dall'addestramento di reti neurali alla soluzione di problemi di ottimizzazione [kar12].

c) Artificial Immune System

Gli Artificial Immune System comprendono una serie di tecniche ed algoritmi ispirati ai sistemi immunitari degli organismi vertebrati [aisweb]. Ne citiamo due, di particolare rilievo, che prendono spunto dalle cellule T, più conosciute come anticorpi, e dai linfociti.

d) Negative Selection Algorithm

Nei sistemi immunitari, gli anticorpi sono in grado di attivarsi in presenza di cellule estranee all'organismo cui appartengono ed attaccarle.

Durante la maturazione, gli anticorpi devono superare un test negativo: vengono esposti ad antigeni dell'organismo ospite e, qualora si attivino, vengono soppresse. Superano la selezione le cellule che non si attivano in presenza di cellule del proprio organismo. Facendo evolvere casualmente un certo numero di anticorpi, ed attuando questa politica di selezione negativa, si rende disponibile al sistema un certo numero di rilevatori che possano attivarsi soltanto in presenza di cellule estranee, ed attaccarle.

Algoritmi [for94] basati su selezione negativa si utilizzano per task di classificazione o riconoscimento di pattern, laddove lo spazio delle soluzioni si modella come complementare alla conoscenza disponibile. Ad esempio, nel caso di task di outlier detection (o anomaly detection), algoritmi di questo tipo preparano un insieme di pattern detector addestrati su pattern normali (non anomali) presenti nel modello, essi riescono quindi a rilevare pattern sconosciuti o anomali.

e) Clonal Selection Algorithm

Il principio della selezione dei cloni è usato per spiegare l'adattatività della risposta immunitaria ad uno stimolo antigenico. Soltanto le cellule che riconoscono degli antigeni sono selezionate per la proliferazione. Le nuove cellule generate sono sottoposte ad un processo di maturazione che ne migliora l'affinità con gli antigeni esterni.

Gli algoritmi basati su Clonal Selection [dec02] sono utilizzati per task di machine-learning e pattern recognition, nonché per risolvere problemi di ottimizzazione multimodale e combinatoria.

f) Particle Swarm Optimization

Particle swarm optimization è un metodo di ricerca euristica che simula i movimenti che uno stormo di uccelli utilizza per trovare cibo.

Ogni individuo si muove all'interno dello spazio di ricerca, in base ad una semplice funzione che incorpora informazioni sulla propria miglior soluzione già trovata, sulla miglior soluzione trovata da tutto il gruppo, e da un elemento di casualità utile a non far convergere gli individui su un minimo locale.

Gli algoritmi di PSO sono utilizzati per lo più per risolvere problemi di ottimizzazione, mono-obiettivo o multi-obiettivo, utilizzando la nozione di ottimalità di Pareto.

g) Firefly-inspired Synchronicity

In natura, è stata osservata la capacità di alcune specie di insetti di sincronizzare azioni e comportamenti. È il caso, ad esempio, di alcune specie di lucciole, che riescono a sincronizzarsi in modo da emettere contemporaneamente impulsi luminosi [mir90]. In questo caso, con il concetto di sincronizzazione non si vuole intendere che gli elementi condividano un orario vero e proprio, ma soltanto che siano capaci di farsi per effettuare alcune operazioni contemporaneamente.

Il meccanismo è il seguente: in ogni individuo è biologicamente codificato un contatore di tempo t , che viene incrementato da 0 a T in base ad un "orologio" individuale. Ogni qual volta questo contatore arriva a T , l'individuo emette un impulso, e resetta il contatore. Quando osserva un altro individuo a lui vicino emettere un impulso, se questo non è sincronizzato con il proprio, immediatamente accelera momentaneamente il suo contatore.

Si dimostra che, se ogni individuo applica questa politica, il collettivo raggiungerà uno stato di sincronizzazione.

Algoritmi basati su questo principio sono utilizzati, ad esempio, per sincronizzare gli elementi di una rete di sensori [wer05].

h) River Formation Dynamics e Intelligent Water Drop

Queste tecniche si ispirano al processo di formazione dei fiumi ovvero ai modi in cui l'acqua piovana trova il percorso più breve (a energia minore) verso il mare.

River Formation Dynamics [rab07] ed Intelligent Water Drop [sha09] sono euristiche nate come varianti di Ant Colony Optimization. Mentre ACO associa delle quantità (feromone) agli archi di un grafo, RFD associa una quantità ai nodi. Ogni nodo è caratterizzato da una certa altezza, che è massima alla sorgente, e minima (0) alla destinazione. Man mano che le gocce fluiscono dalla sorgente, esse hanno una probabilità di muoversi tra i nodi che dipende dalla differenza di altezza tra il nodo in cui stazionano ed i nodi vicini. Ogni goccia preferisce il nodo con minore altezza.

Giunta su un nodo, una goccia può *eroderlo*, diminuendone l'altezza di una certa quantità, o, se i nodi vicini hanno tutti potenziale più basso, *depositare sedimenti*, aumentandone l'altezza.

Alla fine del processo, emergerà il percorso più breve tra la sorgente e la destinazione, che, al limite, verrà eroso tanto da far convergere ogni goccia d'acqua.

i) Digital Infochemical

Tipicamente gli algoritmi di swarm intelligence utilizzano l'ambiente come mezzo per trasportare informazioni (*stigmergia*). Tale meccanismo spesso è ispirato da comportamenti rilevabili in natura come avviene, ad esempio, per il meccanismo dei feromoni nelle formiche o durante l'impollinazione dei fiori da parte delle api. In questi meccanismi di stigmergia sono presenti delle sostanze chimiche che vengono emesse da organismi, propagati nell'ambiente e percepiti da altri organismi. In biologia tali sostanze chimiche vengono chiamate: *Infochemical*. Le infochemical si distinguono a

loro volta in 2 categorie: *feromoni* e *allochemical*. I primi coinvolgono organismi della stessa specie mentre i secondi vengono emessi e percepiti da organismi di specie differente. Gli *allochemical* si distinguono ulteriormente sulla base della loro "utilità". In particolare se l'organismo che emette è quello che trae beneficio dal processo adattivo si parla di *kairomone*. Viceversa se è solo il ricevente che trae beneficio si parla di *allomone*. In caso entrambe le specie coinvolte traggono beneficio si parla di *synomone*.

Sfruttare il meccanismo generale degli infochemical negli algoritmi di swarm intelligence richiede una modellazione matematica precisa: il digital infochemical. Un digital infochemical i è definito come segue:

$$i = \{ \gamma, T, \delta, \epsilon, \psi \}$$

dove γ è la concentrazione istante per istante di i , T è la soglia di concentrazione minima tale per cui un organismo percepisce i . δ è il coefficiente di diffusione e definisce come i si propaga e evapora. ϵ identifica l'organismo emettitore mentre ψ contiene informazioni specifiche del contesto di applicazione.

Oltre alla definizione di digital infochemical è opportuno formalizzare l'ambiente dentro il quale le sostanze si diffondono e gli agenti operano. In particolare l'ambiente definisce 3 funzioni che definiscono la diffusione dell'infochemical nello specifico ambiente: la funzione di *propagazione* che definisce come l'infochemical si propaga da una cella nelle celle contigue, la funzione di *aggregazione* che definisce come aggregare più infochemical e la funzione di *evaporazione* che definisce come l'infochemical evapora nel tempo. Oltre a ciò, l'ambiente deve fornire la possibilità agli agenti di poter percepire gli infochemical presenti, di muoversi nel territorio e di emetterne a loro volta. Un esempio di algoritmo di swarm intelligence che utilizza digital infochemical per la gestione di reti di distribuzione dell'acqua è descritto più avanti.

j) Division of labour

In molte specie di insetti, l'organizzazione delle colonie prevede lo svolgimento di più compiti, quali ricerca fonti di cibo, riproduzione, allevamento delle larve, costruzione e manutenzione dei nidi. Si osserva come una colonia mostri, nel suo insieme, capacità di elasticità e adattatività: è in grado, cioè, di rispondere a possibili perturbazioni ambientali e stimoli esterni variando il numero di individui allocati sui diversi compiti, e questo in base a decisioni che ogni individuo prende autonomamente.

È stato sviluppato [bon99] un modello semplice per la distribuzione dei compiti basato su soglie di risposta degli individui:

- In ogni individuo è internamente codificata una soglia per ogni specifico compito.
- Sono presenti degli stimoli stigmergici associati ad ogni compito, che ne indicano l'importanza, o la necessità, o lo "stato di esecuzione".

- Qualora un individuo sia esposto a una quantità di stimoli per un specifico compito superiore alla sua rispettiva soglia, l'individuo risponde iniziando a svolgere quel particolare compito con una certa probabilità.
- Le soglie possono variare da individuo ad individuo, a seconda le capacità di ognuno (una formica guerriera ha meno probabilità di accudire le larve rispetto ad una operaia, ad esempio), sono quindi consentiti individui specializzati per certi compiti, che però siano in grado di svolgere anche gli altri.

Gli individui specializzati in un dato compito hanno una soglia di risposta più bassa per quel compito, e soglie più alte per gli altri. Nei casi nei quali i soli specialisti presenti nella colonia non siano in grado di eseguire efficacemente un dato compito, gli stimoli per quest'ultimo tenderanno ad aumentare, ed anche gli individui non specializzati risponderanno e si attiveranno per eseguire quel compito.

Varianti di questo algoritmo introducono dinamiche anche nelle soglie, che diventano così variabili. Più un individuo svolge un certo compito, più la soglia corrispondente viene ridotta, mentre aumentano le soglie per i compiti che l'individuo non svolge. Questo porta ad una allocazione dei task differenziale ed alla specializzazione a partire da una popolazione omogenea di operai

k) Gossip based algorithms

Gli algoritmi gossip-based prevedono una rete di nodi (o agenti) che possiedono delle grandezze significative per le quali si vuole calcolare delle stime aggregate quali: media, varianza, massimo, minimo etc. La particolarità di questo tipo di algoritmi risiede nel fatto che tale stima viene effettuata tramite semplici scambi di messaggi tra agenti che si trovano in condizione di "vicinato". In pratica, ogni agente manterrà una versione locale del dato aggregato e istante per istante correggerà questo valore in base a quello dei suoi vicini. In [Jel05] è dimostrata (per diversi tipi di stime aggregate) la convergenza del metodo di gossiping, nel senso che, dopo un determinato numero di scambi di messaggi tra vicini, ogni stima locale interna al singolo agente convergerà al valore globale.

I metodi gossip-based sono utili nel caso in cui un calcolo di tipo centralizzato risulti molto oneroso o impraticabile, inoltre il gossip-based risulta tollerante ai guasti (es: se un agente smette di funzionare o se la comunicazione tra agenti è interrotta). I metodi gossip-based hanno dimostrato la loro efficacia in diversi ambiti applicativi quali: la stima di risorse computazionali (memoria, cpu etc.) in un sistema distribuito, load-balancing e applicazioni con sensori distribuiti in vasti ambienti.

l) Flocking

Gli individui di alcune specie animali, quando si spostano, tendono ad aggregarsi ed assumere formazioni organizzate. Esempi di questi comportamenti possono essere osservati nei banchi di sardine, negli stormi di uccelli, o anche nelle mandrie in movimento.

I vantaggi ottenuti da questo comportamento possono essere differenti, e variano a seconda le specie:

- Scoraggiare i predatori, facendo massa (mandrie di erbivori),
- Sfuggire dai predatori, attraverso politiche di movimento che evitino agli individui di intralciarsi tra loro (banchi di sardine),
- Aumentare l'efficacia locomotoria, ad esempio diminuendo il carico aerodinamico globale (stormi di uccelli in volo).

Gli individui coordinano autonomamente i loro movimenti in base a tre principali regole, che si attivano a seconda della distanza percepita da ogni individuo rispetto agli altri [rey87]:

- Repulsione. Quando un individuo è troppo vicino agli altri, tende ad allontanarsi, per evitare il contatto.
- Allineamento. Ogni individuo cerca di allinearsi, cambiando la sua direzione, mediando tra le direzioni dei suoi vicini.
- Attrazione. Se un individuo è troppo distante dal gruppo, cerca di avvicinarsi ed aggregarsi.

Questi meccanismi di coordinamento sono stati applicati in contesti di simulazione e per l'esecuzione di task di clustering di dati [fol02][fol09][for13].

7. Casi di studio

Nel seguito mostriamo alcuni casi studio nei quali vengono applicati i concetti e le metodologie fin qui descritte. Nel primo caso sono delineate alcune esemplificazioni che mostrano l'utilizzo degli smart object in un contesto urbano. In particolare, le applicazioni riguardano una strada pedonale su cui sono presenti esercizi commerciali e un museo all'aperto. L'utilizzo degli smart object ha il compito di realizzare una smart street in cui sia possibile coordinare il comportamento di persone e cose per migliorare la qualità della vita e attuare forme di risparmio energetico. Il secondo esempio riguarda l'infrastruttura fisica di distribuzione dell'acqua dove è possibile realizzare un CPS per supportare l'allocazione intelligente dell'acqua alle varie utenze mantenendo livelli di pressione costante dell'acqua. Nel modello è utilizzato un approccio agent-based per unire il layer fisico e quello cyber, dove gli agenti ricevono informazioni dai sensori monitorando le componenti fisiche e forniscono queste informazioni alle componenti cyber che si occupano di indicare le azioni da intraprendere. Il terzo modello riguarda il monitoraggio dell'inquinamento acustico attraverso una tecnica a macro sensori virtuali basata su un approccio decentralizzato che consente ai sensori di una zona (esempio una strada) di auto-organizzarsi per definire le regioni di rumore e valutare forme di aggregazione per valutare il punto che origina il rumore e il rumore medio percepito nella regione.

I vari modelli studiati costituiscono la base per la realizzazione della DEMO per la città di Cosenza rappresentano una palestra su cui testare la piattaforma realizzata nel progetto.

7.1 Smart Street

Una Smart Street è composta da un insieme di entità fisiche, relative ad una strada urbana, instrumentate con sensori, attuatori ed altri dispositivi. Una applicazione relativa alla Smart Street definisce delle relazioni tra entità ed attori al fine di ottenere un comportamento complessivo coordinato.

Il contesto studiato si riferisce ad una strada pedonale sulla quale affacciano un certo numero di pubblici esercizi, (negozi, bar, ristoranti, poste, banche), e nella quale è presente un museo all'aperto, costituito da alcune statue situate in diversi punti della strada. Sono presenti inoltre un certo numero di cassonetti o cestini per l'immondizia. La strada è utilizzata dai passanti, che, dotati di smartpone, costituiscono gli utenti della Smart Street.

L'architettura hardware prevede la dislocazione dei dispositivi contenente la piattaforma (*raspberry*) sulla strada, presumibilmente all'interno dei piedistalli delle statue ed all'interno dei locali delle attività commerciali presenti. Le schede contenente i dispositivi reali basati su Arduino sono utilizzate per collezionare i dati da sensori locali e inoltrarli ai *raspberry*, oltre che per eseguire sugli attuatori disponibili le istruzioni ricevute dai *raspberry*.

I vari oggetti che costituiscono il sistema sono instrumentati con appositi dispositivi per il monitoraggio, l'attuazione e la presentazione delle informazioni relative alla smart street.

Ad ogni statua si associano dei sensori di prossimità, degli altoparlanti ad ultrasuoni e delle luci programmabili. Ogni pubblico esercizio è instrumentato con sensori di temperatura, dispositivi per il rilevamento del numero di persone presenti nei locali (telecamere + face-detection o fotocellule all'ingresso o tappetino segnapersone). Inoltre, i pubblici esercizi che prevedono code mettono a disposizione le informazioni della coda sul sistema. Un cestino dei rifiuti è instrumentato con sensori che ne rilevano la capacità disponibile, delle luci programmabili, un display a led ed un altoparlante. Lungo la strada sono installate altre luci programmabili, delle telecamere orientabili e dei display lcd.

Ogni persona fisica partecipa al sistema attraverso azioni naturali o attraverso l'utilizzo di un proprio smartphone, sulla quale sarà installata una applicazione in grado di interconnettersi con la rete di Raspberry.

I dispositivi descritti consentono la realizzazione di diverse applicazioni, tra le quali si sono individuate:

- Smart Basket Network. Si occupa del monitoraggio e del controllo dei cestini della spazzatura sulla smart street, coordinando il comportamento dei dispositivi e degli utilizzatori per mantenere la rete efficiente, eventualmente educando gli utilizzatori ad un buon uso della stessa.
- Gestione code, con la quale si intende diminuire, per gli utenti, i tempi morti di attesa in coda nei vari pubblici esercizi.

- Open-air Museum, per la conservazione delle opere d'arte e per garantire maggiore visibilità al museo all'aperto.
- Eco-shop, attraverso la quale si vogliono monitorare i consumi dei negozi, educare ai consumi, e ottenere un certo risparmio energetico.
- Smart Advertising System, applicazione per fornire agli utenti informazioni sui negozi o su eventi di interesse nell'area.
- Sistema di sorveglianza intelligente, con funzioni automatizzate di monitoraggio e sorveglianza dell'area.

L'architettura, inoltre, è pensata per poter sviluppare/importare e avviare nuove applicazioni dinamicamente. Senza necessità di intervento sul luogo o riavvio del sistema (o parte di esso) è possibile utilizzare le informazioni provenienti dal sensing, elaborarle nel modo desiderato e utilizzare le risorse di attuazione. Ad: esempio si potrebbe cambiare la policy delle luci sulle statue accendendole e spegnendole in modo da realizzare una coreografia, oppure integrare i dati relativi ai negozi con informazioni reperibili su internet in modo da creare nuovi servizi per il cittadino.

Di seguito sono presentate le applicazioni individuate.

7.1.1 Smart Basket Network

Obiettivo. Monitoraggio e controllo dei cestini della spazzatura sulla smart street, coordinare il comportamento dei dispositivi e degli attori umani per mantenere la rete efficiente, eventualmente tentando di educare gli utilizzatori al corretto uso della Smart Basket Network.

Strumentazione. Per ogni Smart Basket, un sensore per valutarne il carico, una luce principale, un display a led, un altoparlante.

Attori/Entità. Smart Basket, Smart Basket Network, Utilizzatore cestino, Ente Manutenzione rifiuti.

Descrizione. La luce principale cambia colore a seconda lo stato del cestino, pieno o vuoto. Se il cestino è pieno, il display indica (con una freccia) la posizione del cestino vuoto più vicino. Se il cestino è pieno, e qualcuno butta qualcosa nel cestino, l'altoparlante richiama l'utente attraverso un messaggio acustico, con lo scopo di educare l'utilizzatore al funzionamento della Smart Basket Network. Se il cestino è pieno, è inviato un messaggio all'ente preposto alla manutenzione.

7.1.2 Rete di gestione code

Obiettivo. Diminuire, per gli utenti, i tempi morti di attesa in coda nei vari uffici/negozi.

Strumentazione. Per ogni negozio, dispositivo o sistema informatico per la gestione dei ticket in una coda.

Attori/Entità. Utente, Gestore Coda Servizio, Applicazione per smartphone.

Descrizione. Un utente di un servizio fisico (ad es. Poste, Farmacia, etc) si reca sul posto e prenota una richiesta di servizio. All'utente è associato un numero, tramite ticket cartaceo, leggibile tramite QRCode dallo smartphone dell'utente. L'applicazione su smartphone associata, collegandosi al sistema gestione coda, può visualizzare il numero associato all'utente ed il numero attualmente in servizio. L'app invia una notifica se il turno dell'utente è vicino. Se l'utente si allontana dall'area della smart street, viene notificato che potrebbe essere escluso dalla coda di servizio.

7.1.3 Open-air Museum

Obiettivo. Conservazione delle opere d'arte. Visibilità del museo all'aperto.

Strumentazione. Per ogni statua/piedistallo: sensore di prossimità, altoparlanti ad ultrasuoni, sensore di luminosità, luce regolabile.

Attori/Entità. Utente, Animale, Statua, Smartphone utente.

Descrizione. Ogni qual volta i sensori di prossimità si attivano, l'altoparlante emette un segnale ad ultrasuoni, non udibile per gli utenti, ma fastidioso per gli animali (cani, uccelli, etc) con l'obiettivo di far allontanare questi ultimi dalla statua. Di notte, quando un utente si avvicina alla statua, la luce regolabile aumenta d'intensità finché l'utente rimane nei dintorni. Anche le luci sulle statue vicine aumentano parzialmente di intensità. Dopo che un utente rimane nei paraggi di una statua per un certo tempo, un notifica con la descrizione della statua è inviata al suo smartphone unitamente con altre informazioni relative al museo, oppure un messaggio registrato è attivato sull'altoparlante.

7.1.4 Eco-shop

Obiettivo. Monitoraggio consumi, educazione ai consumi, risparmio energetico.

Strumentazione. Per ogni negozio-locale, dei sensori di temperatura all'interno ed all'esterno, sensori sul consumo elettrico istantaneo, sensori per la rilevazione del numero di persone nel negozio, uno o più display esterni.

Attori/Entità. Negozio, Acquirente, Utente.

Descrizione. Ogni negozio mette a disposizione i dati sulla temperatura interna, il numero di persone presenti all'interno, i consumi istantanei. Questi dati, aggregati o meno, vengono resi disponibili sui display esterni. L'idea è far entrare i negozi in competizione sul risparmio energetico. I display daranno più tempo pubblicitario ai negozi più virtuosi.

7.1.5 Smart Advertising System

Obiettivo. Fornire agli utenti informazioni sui negozi o su eventi di interesse nell'area.

Strumentazione. Display, sparsi sull'area.

Attori/Entità. Utente, Smartphone utente, Sistema gestione,

Descrizione. Attraverso l'apposita applicazione su smartphone, l'utente segnala i suoi interessi, spuntandoli da una lista. Ogni display pubblicitizzerà i negozi nell'area a seconda degli interessi segnalati dagli utenti del sistema.

Il sistema può essere utilizzato anche per pubblicizzare eventi singoli durante manifestazioni come notti bianche od altre feste. Se un utente è interessato, ad esempio, agli spettacoli dei giocolieri di strada, questo può essere pubblicizzato sui display, sempre aggregando gli interessi di tutti gli utenti del sistema.

7.1.6 Sistema di sorveglianza intelligente

Obiettivo. Monitoraggio e sorveglianza dell'area.

Strumentazione. Sparsi per l'area: sensori di rumore, sensori di presenza, telecamere orientabili, display.

Attori/Entità. Sistema di sorveglianza, Autorità.

Descrizione. Utilizzando i dati misurati dai sensori, è possibile identificare aree di interesse verso le quali orientare automaticamente le telecamere di sorveglianza. Le immagini delle telecamere sono disponibili sui display. In caso di attività particolarmente interessanti o fuori norma, possono essere notificate le autorità.

7.2 Water grid

In questa sezione analizzeremo un caso studio relativo all'applicazione delle tecniche e dei concetti dei *self-adaptive Smart Object* nel contesto delle reti di distribuzione dell'acqua. In particolare sarà descritto un approccio *bio-ispirato* che permette di gestire la problematica in oggetto in modo completamente distribuito. In questo modo il sistema beneficerà delle proprietà di: flessibilità, robustezza (tolleranza ai guasti), scalabilità e adattabilità. Inoltre il sistema si adatta in real time alle perturbazioni impreviste (es: fenomeni meteorologici). L'idea è stata approfondita partendo dal lavoro [Dot10].

7.2.1 Water Distribution Networks (WDN)

Una WDN serve essenzialmente per portare l'acqua in città ai cittadini, sia per quel che riguarda le residenze private che per quelle commerciali.

Una WDN può essere schematizzata come un grafo nel quale ogni arco rappresenta una condotta dell'acqua, mentre i nodi rappresentano entità di vario tipo descritte di seguito (vedi figura 1):

- Giunzioni: servono ad interconnettere le condutture (ju in figura)
- Bacini idrici: le fonti che immettono l'acqua nella rete (re in figura)
- Serbatoi (o cisterne): permettono di accumulare acqua (ta in figura)

Inoltre alcuni archi del grafo (alcune condutture) contengono anche pompe il cui scopo è quello di fornire energia meccanica in modo da modificare il flusso d'acqua nella rete (pu in figura).

Lo scopo di una WDN è quella di ottenere un'opportuna pressione dell'acqua nei vari punti della rete. Tale pressione è misurata tipicamente dal valore del "carico idraulico" (*hydraulic head*). Mantenere il carico idraulico nei vari punti della rete ad un valore opportuno equivale a garantire che il servizio di distribuzione funzioni correttamente.

A tale scopo può essere usata la forza di gravità: banalmente, se i bacini idrici (che supponiamo per semplicità avere una capacità di produzione infinita) sono posti più in alto rispetto ai nodi di interesse ciò potrebbe già garantire un sufficiente carico idraulico. Naturalmente, visto che i bacini idrici sono: laghi, fiumi etc., la loro ubicazione non può essere decisa dal progettista della rete ma rappresenta un vincolo del problema.

Proprio per questo motivo sono presenti nella rete i serbatoi e le pompe. I serbatoi

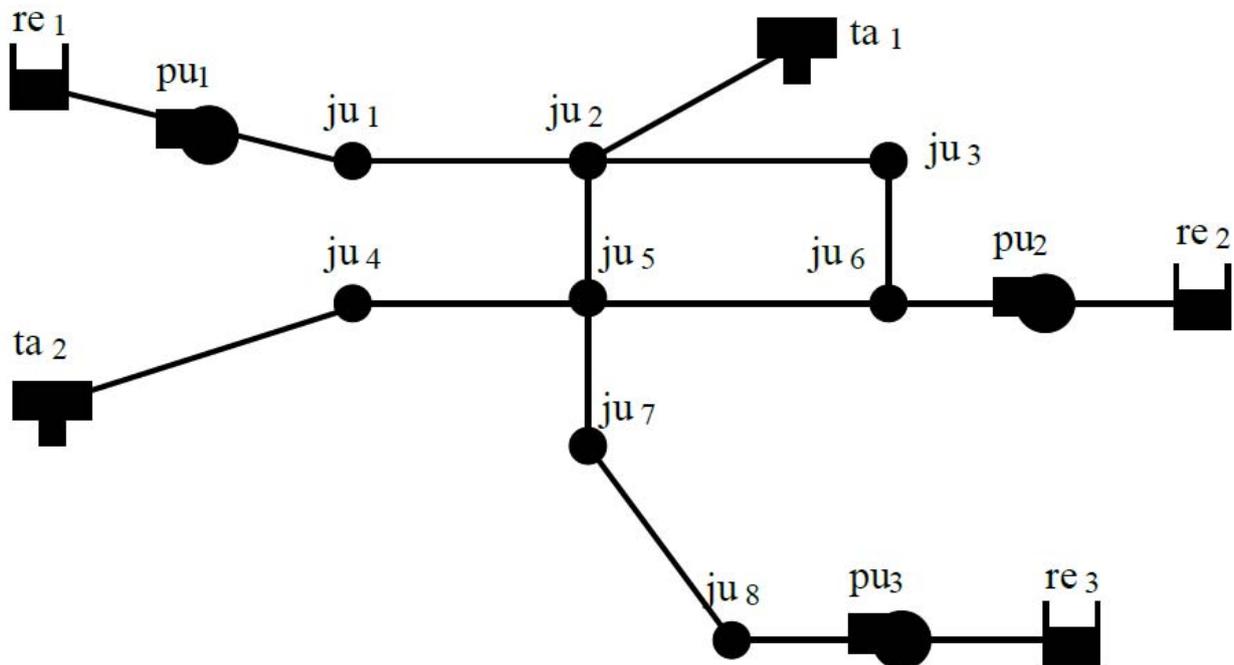


Figura 9. Esempio di WDN.

servono ad immagazzinare l'acqua in modo da far fronte ai cambiamenti repentini di richiesta dell'acqua nella rete. La capacità di immagazzinamento dei serbatoi è limitata, per cui è necessario portare l'acqua dai bacini idrici ai serbatoi, questo è il

ruolo svolto dalle pompe: ogni pompa è incaricata di portare l'acqua da un punto più basso (il bacino idrico) ad un punto più alto (il serbatoio). Il comportamento di una pompa è usualmente descritto dalla sua *pump-curve* che definisce come il carico idraulico fornito a destinazione dipenda dall'energia fornita alla pompa e dal flusso presente alla sorgente.

Nell'uso comune, le WDN utilizzano un approccio reattivo basato sul livello di riempimento dei serbatoi. Tale approccio unito con interventi umani esperti permette di solito di ottenere il soddisfacimento dei bisogni idraulici dei cittadini ad un non troppo alto consumo di energia.

Ciononostante, considerando che la richiesta d'acqua è un parametro che cambia repentinamente in modo poco predicibile, un approccio di tipo self-adaptive può risultare vincente allo scopo di minimizzare il consumo d'energia spesa per il corretto funzionamento della rete.

7.2.2 Indirect defense strategy

Tra i possibili approcci self-adaptive alle WDN descriviamo quello ispirato dal meccanismo naturale chiamato "*indirect defense*".

In biologia, quando una pianta è attaccata da animali erbivori o funghi essa emette una particolare sostanza chimica che si propaga nell'ambiente allo scopo di difendersi. Tale sostanza chimica attira (se presenti) animali carnivori che possono essere interessati ad attaccare quelli erbivori che stanno attaccando la pianta in quel momento. Da qui il nome di "difesa indiretta". Queste sostanze chimiche sparse nell'ambiente sono classificate sotto la dicitura più generale di '*infochemicals*'. Anche i feromoni emessi dalle formiche o le sostanze emesse sempre dalle piante per attirare le api sono esempi di *infochemicals*.

La analogia nel contesto delle WDN riguarda i serbatoi e le pompe. I primi possono emettere *infochemicals* per notificare (tramite l'ambiente) che hanno necessità di acqua o che, al contrario, stanno per raggiungere la loro capacità massima. Le pompe, ricevendo *infochemicals* dall'ambiente, possono modificare il loro comportamento in modo da venir incontro alle esigenze dei serbatoi.

Dal punto di vista informatico occorre, quindi, modellare innanzitutto gli *infochemicals* e la loro propagazione nell'ambiente. In particolare un *infochemical* trasporta una serie di informazioni specifiche del contesto applicativo ed è caratterizzato da parametri fisici che ne definiscono la propagazione nell'ambiente: il coefficiente di *diffusione* e quello di *evaporazione*. In generale ne esiste un terzo: il coefficiente di *aggregazione* che riguarda il mescolamento di *infochemicals* di natura diversa ma esso non è utilizzato in questo esempio. L'ambiente di diffusione degli *infochemicals*, nel nostro caso è la WDN, ovvero essi si propagano (virtualmente) tramite le condutture.

Le pompe e i serbatoi sono realizzati da agenti che sono in grado di emettere e percepire gli *infochemicals* nonché di effettuare reasoning locali.

7.2.3 Algoritmo di indirect defense strategy applicato alle WDN

Quando un agente serbatoio rileva che sta per essere svuotato (o al contrario completamente riempito) emette un *infochemical* per notificare all'ambiente la sua richiesta di ricevere ulteriore acqua o meno. Nell'*infochemical* esso aggiunge il suo identificativo univoco e la quantità algebrica di acqua. Quando tale *infochemical* è ricevuto dall'agente pompa, esso stabilisce il comportamento da tenere per ottemperare la richiesta. In particolare, esso prende in considerazione il carico idraulico in uscita alla pompa e il flusso corrente e utilizza la propria *pump-curve* per determinare a quale velocità impostare la propria ventola in modo da fornire l'energia meccanica idonea per la richiesta. Contemporaneamente, alla ricezione dell'*infochemical* di richiesta, esso emette un altro tipo di *infochemical* per notificare le altre pompe che può prendere in carico la richiesta. Questo secondo tipo di *infochemical* trasporta le informazioni relative all'identificativo della pompa, l'identificativo del serbatoio e una terza informazione utile per stabilire quale pompa dovrà prendere in carico la richiesta del serbatoio. Vi sono 2 versioni dell'algoritmo che si differenziano esclusivamente sulla natura di questa terza informazione: la prima versione utilizza la concentrazione di *infochemical* ricevuta in modo da tenere conto della distanza tra pompa e serbatoio, la seconda versione cerca invece di stimare (sempre tramite la sua *pump-curve*) l'energia minima da spendere per ottemperare alla richiesta. In entrambe le versioni gli agenti pompe, ricevuto il secondo tipo di *infochemical*, valutano autonomamente quale pompa debba prendersi carico della richiesta. Quando un agente pompa valuta che è proprio lui il "prescelto" attua il comportamento prestabilito.

Nel lavoro preso in esame sono mostrati i risultati sperimentali dell'algoritmo nelle 2 versioni. In particolare è dimostrato che la prima versione riesce a garantire un corretto funzionamento della WDN adattandosi alle esigenze dinamiche delle utenze. La seconda versione garantisce, inoltre, una efficienza energetica migliore.

7.3 Monitoraggio dell'inquinamento acustico

Il rumore presente nell'ambiente esterno è un agente inquinante di tipo fisico che può essere originato da diverse sorgenti legate alle attività umane, al traffico veicolare, ferroviario, aereo e alle attività industriali, commerciali e artigianali. Tra queste, il traffico veicolare costituisce senza dubbio la causa prevalente ed a maggiore diffusione nell'intera area urbana. Un contributo particolare, in quanto non omogeneo, nell'area urbana, è quello dovuto a motociclette e convogli ferroviari; tali mezzi di trasporto possono rappresentare in aree limitate, la principale causa di disturbo mentre possono essere ininfluenti su altre parti del territorio. Esempi concreti sono: i centri storici o comunque le aree a traffico limitato per i motocicli, le zone prossime alle linee ferroviarie nel secondo caso. Il rumore da traffico non è però l'unica causa di produzione del rumore nelle aree urbane sebbene sia la più rilevante e la più diffusa; tra le altre sorgenti sono senza dubbio le attività produttive quelle che determinano il contributo maggiore. In termini di "disagio acustico" percepito dalla popolazione esposta non vanno inoltre dimenticate una moltitudine di sorgenti acustiche che, seppure caratterizzate da un basso valore di emissione, possono produrre incrementi "disturbanti" del rumore ambientale a breve distanza. In questa tipologia rientrano

frequentemente gli impianti per la diffusione della musica nei luoghi di trattenimento, ma anche gli stessi impianti tecnologici condominiali, in particolare i condizionatori. Ogni persona è sottoposta quotidianamente ad una quantità innumerevole di stimoli acustici; alcune esperienze condotte con campionatori personali hanno accertato che stili di vita assolutamente consueti espongono ad un Lep diurno (livello di esposizione personale) superiore ai 70dBA; tale valore è individuato come soglia di sicuro disturbo per il rumore ambientale in un area urbana. Si deve inoltre sottolineare come la reazione agli stimoli acustici in molti casi sia soggettiva in quanto le persone reagiscono in modo diverso sia a livello individuale che anche in funzione del contesto nel quale l'esposizione si verifica. In ogni caso lo stimolo sonoro può risultare gradevole o sgradevole e viene pertanto alternativamente definito suono o rumore. Una serata in discoteca, può produrre un'esposizione al rumore espressa in Lep superiore ai 90dBA; in un ambiente di lavoro, un tale livello di esposizione al rumore renderebbe obbligatorio l'uso di idonei DPI (le cuffie protettive) e la predisposizione di un piano di bonifica. È dimostrato che un'esposizione prolungata al rumore può essere causa di effetti sulla salute umana che in modo schematico possono essere distinti in uditivi o extrauditivi; questi ultimi possono essere di tipo somatico oppure psicosociale. I danni specifici all'udito sono quelli studiati da maggior tempo e non sono più oggetto di discussione; è noto che l'esposizione a livelli di rumore elevati, superiori a 75dBA, provoca l'innalzamento temporaneo della soglia uditiva. La ipoacusia temporanea può divenire permanente a seguito di esposizioni prolungate; tali condizioni si verificano, di solito, per particolari mansioni in alcuni ambienti lavorativi mentre non vengono raggiunte dal rumore in ambiente urbano. Per i lavoratori esposti a livelli di esposizione media giornaliera superiore agli 80dBA sono previsti particolari garanzie di tutela oltre alla sorveglianza sanitaria, al fine di evidenziare per tempo il verificarsi di deficit uditivo. Diversi studi hanno ormai accertato che tra i residenti in aree ad elevata rumorosità ambientale si presentano con frequenza maggiore sia malesseri di tipo psichico sia alterazioni di alcune funzioni fisiologiche soprattutto a carico del sistema cardiocircolatorio, gastrointestinale, endocrino e respiratorio. Tali disturbi possono manifestarsi anche a livelli inferiori ai 75dBA e pertanto sono tipici della esposizione al rumore urbano, che di solito è compreso tra i 50dBA e i 75dBA. La terza tipologia di disturbi, quelli prettamente psicologici, comprendono i disagi o gli svantaggi connessi ad una elevata rumorosità dell'ambiente di vita. Tra questi si possono annoverare: l'insonnia, la stanchezza, l'irritazione, le difficoltà di concentrazione, il disturbo della conversazione, etc.

Monitorare l'inquinamento acustico è un'attività estremamente importante che può essere effettuata attraverso rete di sensori wireless distribuiti tridimensionalmente all'interno di un'area urbana. Molti problemi si generano quando una rete di sensori è utilizzata. Primo, la rete di sensori, deve essere in grado di controllare il suo consumo di energia. In altre parole, i costi energetici dovranno essere bilanciati dalla rete, in modo da assicurare una durata minima garantita o per dispositivi autoricaricabili un consumo medio predefinito. Secondo, la rete dovrà essere in grado di fornire agli utenti (sia remoti che locali) informazioni significative e compatte legate ai fenomeni che sono rilevati invece di dati grezzi. E' necessario quindi, in reti di sensori di grandi dimensioni, tenere conto che i trasferimenti di dati e l'analisi ex-post potrebbe

diventare ingestibile. Infine, la rete dovrebbe rispondere rapidamente alle richieste degli utenti. Dal momento che gli utenti possono essere altamente mobili, una risposta in ritardo ad una query può non riuscire a raggiungere l'utente o a raggiungerlo in un luogo in cui le informazioni della risposta sono inutili.

Per superare queste problematiche ci siamo ispirati ad un approccio innovativo capace di monitorare l'inquinamento acustico attraverso una rete di sensori gestita in maniera decentralizzata e capace di aggregare in maniera distribuita i dati dei sensori attraverso la creazione dinamica di regioni. Il risultato di questo processo è che una rete di sensori può essere vista come costituita da macro sensori virtuali, ciascuno associato ad una ben caratterizzata regione dell'ambiente fisico (una regione che presenta un profilo uniforme per qualche proprietà specifica come *rumore*, *luminosità*, *temperatura*, etc.). All'interno di ogni regione, ogni sensore fisico ha la disponibilità locale di dati aggregati della sua regione e può agire come un *access point* per tali dati. Questo approccio consente ad utenti multipli e mobili di accedere tempestivamente a informazioni globali sull'ambiente circostante, semplicemente interrogando il sensore più vicino, senza alcun costo aggiuntivo per la rete di sensori. Inoltre, permette di trasferire efficacemente dati aggregati verso un punto di raccolta centralizzato in modo più efficiente, tuttavia evitando la perdita di informazioni tipica di algoritmi di aggregazione globali.

7.3.1 L'approccio a "Macro sensori virtuali"

L'approccio Virtual Macro Sensors (ViMS) si basa su lavori di ricerca di [Bic10]. In sintesi, l'approccio ViMS comprende (vedi Figura 2):

- *un algoritmo di formazione delle regioni self-organized*, per dividere logicamente una rete di sensori (Figura 10 (a)) in una serie di regioni spaziali, ognuna caratterizzata da specifici patterns ambientali nei dati rilevati (Figura 10 (b));
- *algoritmi di aggregazione in-network* localizzati (regionali) per fornire ad ogni sensore in una regione informazioni aggregate sullo stato globale della regione (Figura 10 (c));
- *soluzioni auto-adattative* innovative per gestire le situazioni transitorie e dinamiche. Tali soluzioni assicurano che, sia le informazioni di regione che quelle aggregate all'interno delle regioni, riflettano sempre la situazione corrente della rete e dell'ambiente (Figura 10 (d)).

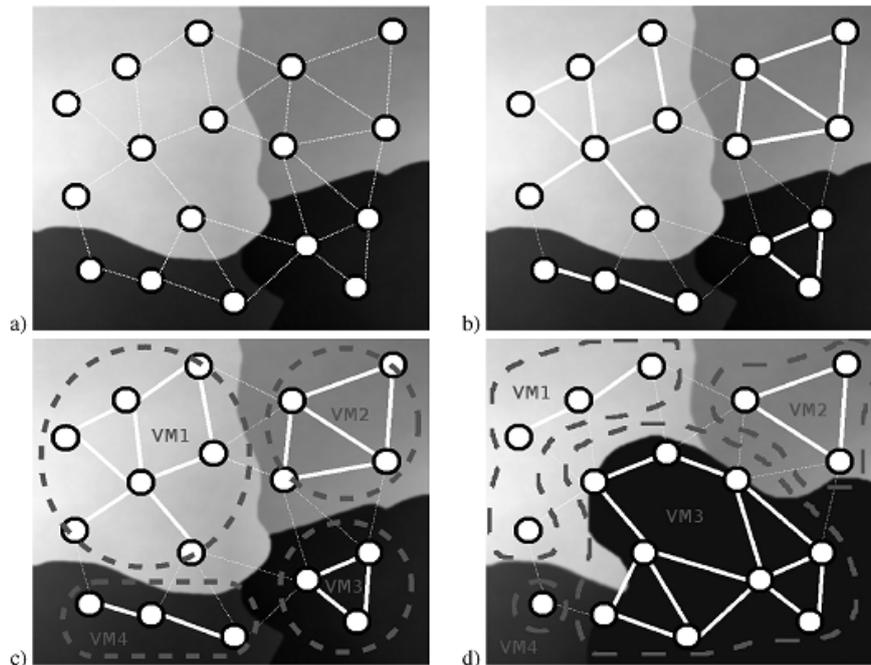


Figura 10 Approccio VIMS: (a) una rete di sensori caratterizzata da differenti patterns di qualche valore rilevato; (b) Una rete overlay di connessioni virtuali in base a tali patterns; (c) i clusters risultanti di sensori associati ad ogni regione possono agire come una sorta di macro sensore virtuale per fornire un'informazione aggregata dei dati rilevati in quella regione; (d) le regioni, e quindi, i virtual macro sensor, si rimodellano in risposta ad una modifica delle condizioni.

Questo processo, che induce costi energetici prevedibili e controllabili sulla rete, permette ad una rete di sensori di essere percepita e sfruttata come se fosse costituita da macro sensori virtuali, ciascuno associato ad una regione ben caratterizzata dell'ambiente fisico. All'interno di ogni regione, ogni sensore fisico ha la disponibilità locale dei dati aggregati relativi alla sua regione e può agire come un punto di accesso a tali dati. Nella parte restante di questa sezione si vedranno in dettaglio: l'algoritmo di formazione della regione, gli algoritmi di aggregazione, e la gestione delle situazioni transitorie e dinamiche.

7.3.2 Algoritmo per la formazione delle regioni

Consideriamo una rete di sensori distribuiti in un ambiente in cui il valore, v , di qualche proprietà specifica viene rilevato localmente. Il valore v potrebbe rappresentare il valore di rumore, un livello di luce, o qualunque proprietà che un sensore è in grado di rilevare. L'algoritmo proposto per la formazione della regione ha l'obiettivo di far sì che i sensori si auto-organizzino in insiemi disgiunti di regioni spaziali ognuna caratterizzata da misure simili della proprietà v . Chiaramente, l'ipotesi sottostante è che l'ambiente monitorato presenti qualche discontinuità spaziale della proprietà, riflettendo le diverse regioni ambientali. Tuttavia, sia paesaggi naturali che ambienti modellati dall'uomo presentano, nella maggior parte dei casi, tali discontinuità. L'organizzazione nelle regioni avviene attraverso un processo di

costruzione di una sovrapposizione di collegamenti virtuali pesati tra nodi vicini (tra i nodi che si trovano entro il range radio). In particolare, i nodi appartenenti alla stessa regione avranno collegamenti virtuali forti, mentre i nodi vicini appartenenti a diverse regioni avranno link deboli (o nulli). Per esempio: la misura del livello di luce può essere utilizzata per una rete di sensori in un edificio auto-partizionato su una base per-camera (camere diverse saranno caratterizzate da diversi livelli di luce, mentre il livello di luce all'interno di una stanza può essere abbastanza omogeneo); misurare il livello di vibrazione su un pendio di montagna potrebbe portare ad auto-organizzazione di una rete di sensori in regioni associate a superfici con differenti caratteristiche geologiche. In generale, l'organizzazione della regione riflette proprietà dello spazio fisico e porta ad una organizzazione logica della rete, così una regione può essere vista come estensione spaziale di un macro sensore virtuale.

- Logical Links. Siano p e q due sensori vicini, cioè due sensori la cui distanza è minore del loro raggio di copertura wireless r . Siano v_p e v_q i valori di una proprietà v rilevata da p e q , rispettivamente. Supponiamo che una funzione distanza D può essere definita per le coppie di valori di v . La formazione della regione si basa quindi sul calcolo iterativo del valore di un link logico $l(p, q) \in [0,1]$ per ogni nodo del sistema, come riportato nella seguente procedura UpdateLink.

UpdateLink

if ($D(v_p, v_q) < T$) {

$l(p, q) = \min(l(p, q) + \Delta, 1);$

else {

$l(p, q) = \max(l(p, q) - \Delta, 0);$

}

T è una soglia che determina se i valori misurati sono abbastanza vicini per cui $l(p, q)$ deve essere rafforzato o indebolito, ed è un valore che incide sulla reattività dell'algoritmo nell'aggiornare un collegamento. Sulla base di questo algoritmo, è piuttosto chiaro che se $D(v_p - v_q)$ è inferiore alla soglia T , $l(p, q)$ alla fine converge a 1 (indipendentemente dal suo valore iniziale). In caso contrario, va a 0. In generale, due nodi, p e q si possono considerare essere nella stessa regione (o collegati) quando $l(p, q)$ è uguale o maggiore di una soglia di connessione T_{CONN} , cioè $l(p, q) \geq T_{CONN}$. Qui, senza perdita di generalità, si assume $T_{CONN} = 1$.

- L'Algoritmo Distribuito. Per l'effettiva esecuzione distribuita dell'algoritmo di formazione della regione, ogni nodo memorizza un vettore che descrive, per ciascuno dei suoi vicini, il valore corrente del link l verso di esso e un flag di segnalazione dello stato del link (collegato o meno). I valori iniziali di l possono

essere arbitrari. L'esecuzione distribuita dell'algoritmo si basa su uno schema periodico, che agisce come una sorta di attività continua di fondo nella rete di sensori, e non richiede alcuna forma di sincronizzazione tra i sensori. Lo schema di base è il seguente. Periodicamente, ogni nodo trasmette in broadcast a tutti i suoi vicini (tutti i sensori nel suo raggio radio wireless) un *Data Exchange Request* per richiedere a loro una sessione di scambio dati. Poi, con tutti i vicini attivi (coloro che hanno risposto alla richiesta ping), scambia i dati necessari per calcolare i valori I (ad esempio, il valore della proprietà v su cui si basa la formazione di regione), ed esegue la già descritta procedura di UpdateLink per ciascuno di essi.

7.3.3 Aggregazione

L'approccio ViMS assume che ogni sensore all'interno di una regione detiene lo stato complessivo della regione. A tal fine, è possibile integrare le forme di aggregazione diffusiva basata su gossip nello schema generale descritto. In poche parole, l'aggregazione diffusiva basata su gossip funziona con i nodi che periodicamente scambiano con i loro vicini, le informazioni di un valore locale, e localmente aggregano i valori in base ad una funzione di aggregazione (massimo, minimo, media, etc.) e scambiano ulteriormente il valore aggregato. Alla fine, i valori aggregati propagati si diffonderanno su tutta la rete e rappresenteranno la funzione di aggregazione essendo stati calcolati ad ogni nodo della rete. Così, ogni nodo della rete avrà la disponibilità locale del valore aggregato globalmente.

Minimum and Maximum. La conoscenza dei valori massimo e minimo di alcune proprietà su una regione monitorata da una rete di sensori può essere utile in molte applicazioni. Per esempio, un sensore è in grado di valutare se una lettura è un valore anomalo confrontandolo con il massimo ed il minimo, o un esperto può valutare la presenza di anomalie nella regione. Per acquisire localmente il $\max(w)$ e $\min(w)$ di qualche valore rilevato w , ogni nodo può semplicemente scambiare con i suoi vicini i dati del massimo ed del minimo ($\max(w)_p$ $\min(w)_p$, rispettivamente) conosciuti localmente fino a quel momento. Cioè, un nodo p , dopo aver scambiato tali dati con il nodo q , esegue la seguente procedura di aggregazione:

```
Aggregation
if ( $\min(w)_p > \min(w)_q$ ) {
 $\min(w)_p = \min(w)_q$ ;
}
if ( $\max(w)_p < \max(w)_q$ ) {
 $\max(w)_p = \max(w)_q$ ;
}
```

Con $\min(w)_p$ e $\max(w)_p$ inizializzate a w_p . Alla fine, la conoscenza effettiva di $\min(w)$ e $\max(w)$ raggiungerà ogni nodo della regione (o, nel caso di aggregazione globale, l'intera rete).

Average. Un'altra funzione di aggregazione molto rilevante è la media, in quanto può fornire informazioni molto compatte sullo stato generale di una regione. Per calcolare la media, la funzione di aggregazione può essere la media delle medie locali aggregate. Cioè, un nodo p , dopo aver scambiato i dati con un nodo q collegato, può eseguire semplicemente la seguente funzione aggregazione:

$$avg(w)_p = (avg(w)_p + avg(w)_q) / 2;$$

con $avg(w)_p$ inizializzato al valore locale w_p . Alla fine, ogni nodo avrà il suo valore locale di $avg(w)_p$ che convergerà alla media effettiva.

7.3.4 Situazioni dinamiche e transitorie

In questa sezione si analizza il comportamento di ViMS con ambienti dinamici e durante stati transitori.

Formazione dinamica di regioni. In generale, il valore iniziale dei collegamenti virtuali tra i nodi l , è irrilevante per la formazione della regione durante la fase transitoria dopo la distribuzione dei sensori. Per esempio, in una situazione iniziale in cui tutti i nodi sono scollegati (con i valori di l tutti inferiori alla soglia di T_{CONN}), ogni nodo rappresenta una regione in sé. All'inizio dell'esecuzione dell'algoritmo, i nodi con valori simili di v si collegheranno l'uno con l'altro, e un insieme di regioni con dimensioni crescenti inizierà a formarsi ed eventualmente si fonderanno l'un l'altra fino al raggiungimento dell'equilibrio. Ogni volta che cambia v , le regioni esistenti si potrebbero rimodellare. Per esempio, un nodo p originariamente appartenente ad una regione R_x può vedere i links che lo collegano a R_x indebolirsi finché non è più collegato a nessun nodo in questa regione. Allo stesso tempo, è possibile che questo nodo veda i suoi links con i nodi in un'altra regione R_y rafforzarsi, fino a diventare esso stesso parte di R_y . Durante tale processo, è possibile che p è temporaneamente rimosso da qualsiasi regione o che sia contemporaneamente collegato ad R_x e R_y , rendendo R_x e R_y una singola regione. Tuttavia, tali situazioni non causano alcun danno al sistema, poiché sono temporanee (e se non lo sono, è semplicemente perché non dovrebbero esserlo, ad esempio, il nodo p è una regione in sé o le regioni R_x e R_y devono effettivamente unirsi in un'unica regione).

Aggregazione della regione. Per quanto riguarda l'aggregazione della regione, la gestione di situazioni transitorie è più intricata. Inizialmente, quando il sistema si avvia, il processo di aggregazione della regione può iniziare in concomitanza con il processo di formazione della regione. Cominciamo col supporre che i nodi non siano inizialmente collegati. La procedura di aggregazione locale inizia effettivamente appena due nodi sono collegati, e procede gradualmente, coinvolgendo sempre più

sensori, eventualmente convergendo verso una situazione stabile della regione. Si può dimostrare (ed è abbastanza intuitivo infatti, a causa della natura cumulativa dell'aggregazione) che negli algoritmi di aggregazione proposti non si verificano problemi se eseguiti su un numero crescente di nodi, come avviene quando si formano regioni da un insieme di nodi indipendenti. Semplicemente, come nuovi valori da aggregare entrano in gioco, la stessa natura diffusiva gossip-based dell'aggregazione assicura che i nuovi valori finiranno per essere conteggiati nel processo di aggregazione generale. Considerazioni analoghe si applicano nel caso in cui nuovi sensori sono dinamicamente distribuiti in una regione del sistema o quando una regione esistente si ingrandisce o si fonde con un'altra regione. La situazione è un po' più complessa in presenza delle seguenti situazioni.

- *Restringimento delle regioni*, come accade durante un processo di scissione delle regioni, o semplicemente quando alcuni sensori "muoiono";
- *Variazioni dinamiche* nei valori da aggregare, che è in effetti una situazione normale per una rete di sensori distribuiti in qualche ambiente da monitorare.

In entrambi questi casi, i valori calcolati dalle funzioni di aggregazione locali possono non essere più validi, perché potrebbero rappresentare valori che non appartengono alla regione. Nel caso di restringimento delle regioni per esempio, il massimo precedente può aver lasciato la regione e/o la media potrebbe essere cambiata significativamente. Analogamente, nel caso di cambiamenti dinamici dei valori da aggregare, il massimo precedente può essere diminuito e/o la media potrebbe essere cambiata. In ogni caso, la natura intrinsecamente cumulativa dell'aggregazione non consente ai valori aggregati di riflettere correttamente la nuova situazione, che potrebbe indurre ad errori significativi dei valori aggregati risultanti. Una soluzione generale ed ampiamente adottata per affrontare il problema di aggregazione si basa sul riavvio periodico del processo di aggregazione. Ciclicamente, i nodi di una rete (o di una regione, nel nostro caso) iniziano una *nuova epoca* di aggregazione, azzerando i valori aggregati in precedenza e riavviando il processo di aggregazione. In questo modo, gli errori accumulati fino a quel momento sono semplicemente resettati e il processo di aggregazione appena avviato può rappresentare la situazione aggiornata. Tale approccio basato sull'epoca potrebbe essere facilmente integrato in ViMS per affrontare le situazioni dinamiche.

7.3.5 Simulazione del monitoraggio dell'inquinamento acustico

Per realizzare il monitoraggio dell'inquinamento in una zona urbana, l'architettura proposta per il monitoraggio dell'inquinamento acustico prevede l'utilizzo di sensori wireless situati a varie quote degli edifici in un ambiente urbano e conseguentemente l'impiego di una mappa 3D (vedi Fig.13). Sebbene sia possibile simulare ambienti 3D per semplicità di visualizzazione si è deciso di utilizzare una rappresentazione 2,5D

dove ogni sensore è presente in un vicinato 2D ma è dotato di un'ulteriore variabile che definisce l'altezza.

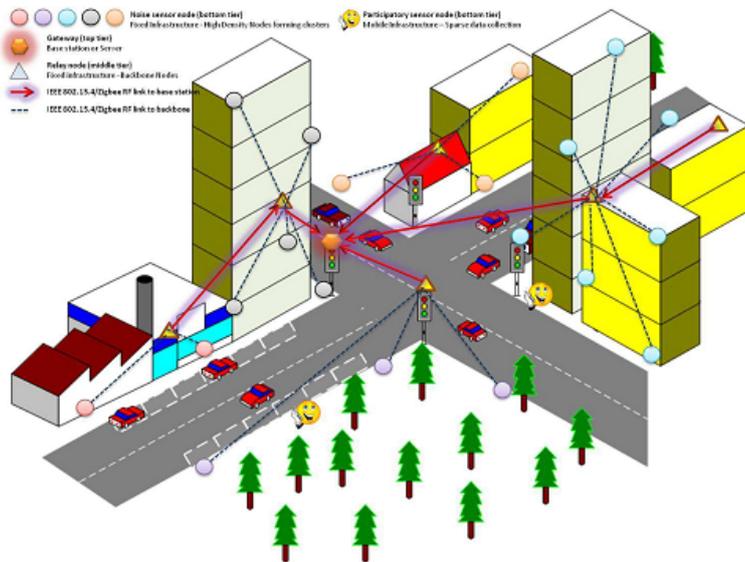


Figura 11. Esempio di utilizzo di una mappa 3D per l'architettura del monitoraggio acustico in un ambiente urbano.

La simulazione è stata realizzata utilizzando l'ambiente multi-agente Netlogo. Nella Fig. si può osservare lo scenario proposto per la simulazione in NetLogo del monitoraggio dell'inquinamento acustico in un ambiente urbano mediante l'utilizzo dei Virtual Macro Sensors. Si adopera la mappa di una città che è stata suddivisa in quattro regioni distinte, ognuna caratterizzata da un range di valori del rumore espresso in dBA: nella Fig.7 in alto a destra range 50-55 dBA; in alto a sinistra range 60-65 dBA; in basso a destra range 70-75 dBA; in basso a sinistra range 80-85 dBA. Una volta che i sensori sono stati posizionati sulla mappa in maniera del tutto casuale assegnando un'altezza anch'essa casuale vengono a determinarsi i link(blu) che denotano i collegamenti wireless dei sensori che si trovano nel range di copertura, calcolato in questo caso considerando uno scenario tridimensionale con la distanza euclidea anziché quella lineare. Riportando la porzione di codice in cui si calcola tale distanza si rammentano le formule geometriche utilizzate.

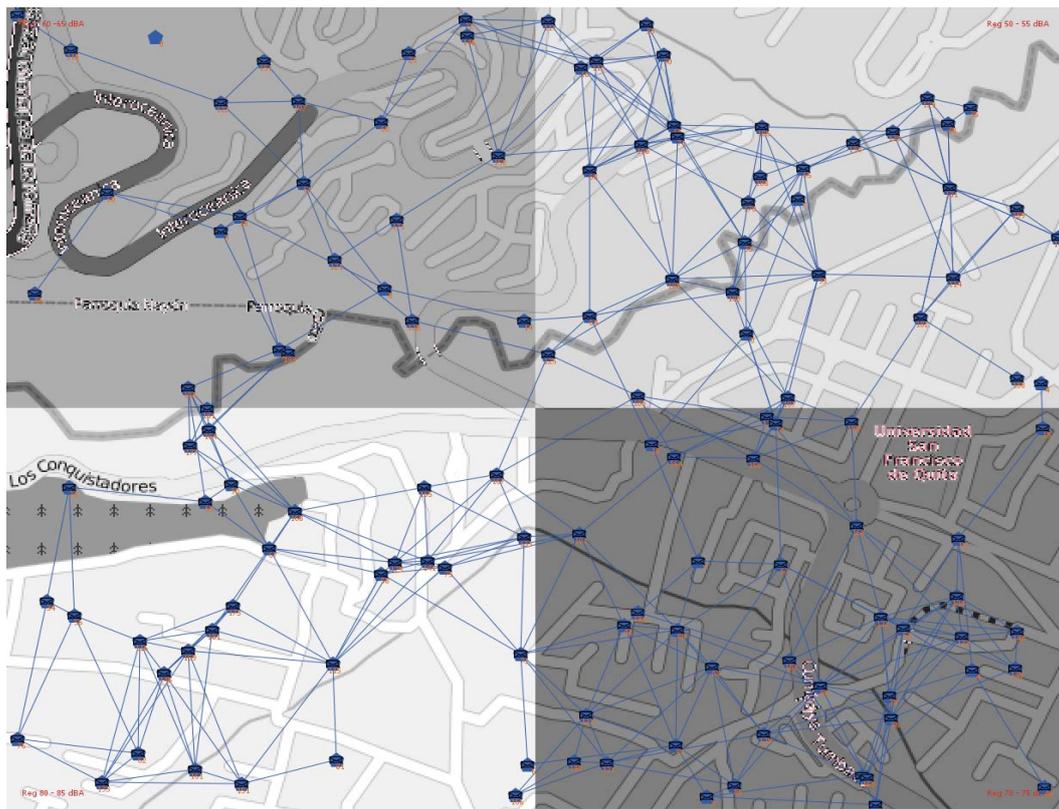


Figura 12. Mappa in NetLogo del monitoraggio del rumore mediante ViMS: situazione iniziale.

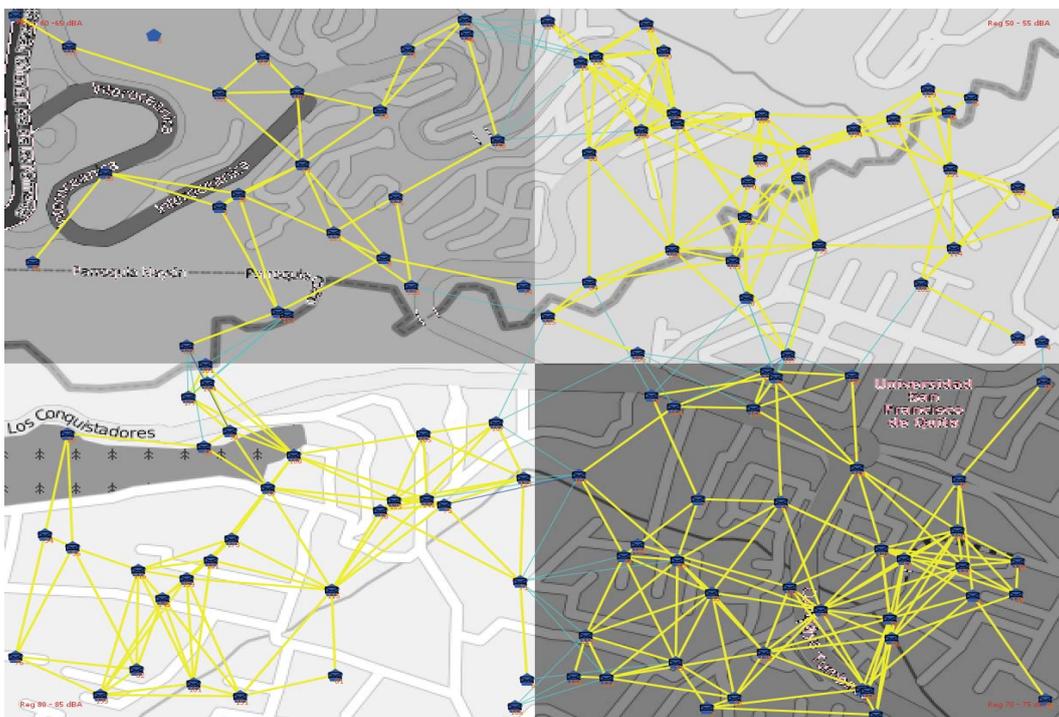


Figura 13. Mappa in NetLogo del monitoraggio del rumore mediante ViMS: formazione dei Macro Sensori.

Alcuni scenari simulati sono presentati nelle figure seguenti. Si può notare la formazione dei macro sensori nelle quattro regioni dopo lo scambio delle informazioni tra i sensori vicini, tramite l'invio dei messaggi, proprio come accadeva nell'esempio visto nel Cap.3. I sensori che hanno una differenza di rumore rilevato entro la soglia regionale (15 dBA in questo caso) vedranno i loro collegamenti ricolorarsi di giallo ed effettueranno le procedure di aggregazione sui valori medi, massimi e minimi del rumore. Nello stesso scenario sono state inserite possibili situazioni di superamento della soglia (*SogliaRum*) di rumore sopportabile dall'orecchio umano, prendendo spunto da situazioni effettivamente realizzabili in un ambiente urbano. Una situazione possibile riguarda il passaggio di un aereo di linea che provoca l'aumento del rumore sopra i 120 dBA. I ViMS risponderanno, come si può notare in Fig.16, a questo improvviso innalzamento del rumore auto-organizzando un ulteriore macro sensore nella zona interessata dal disturbo sonoro e segnaleranno opportunamente questa anomalia. Un altro esempio simile, mostrato in Fig.17, è la simulazione del passaggio di un treno.

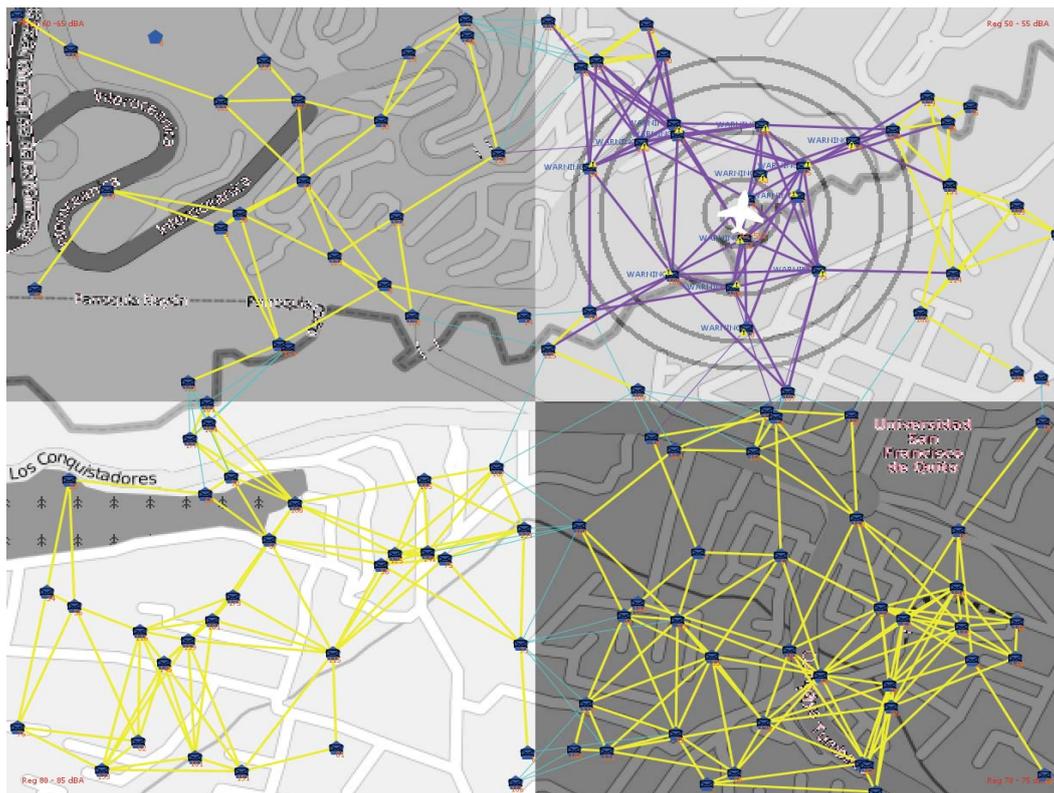


Figura 14. Mappa in NetLogo del monitoraggio dell'inquinamento acustico mediante ViMS: simulazione di una fonte improvvisa di rumore (passaggio di un Aeroplano).

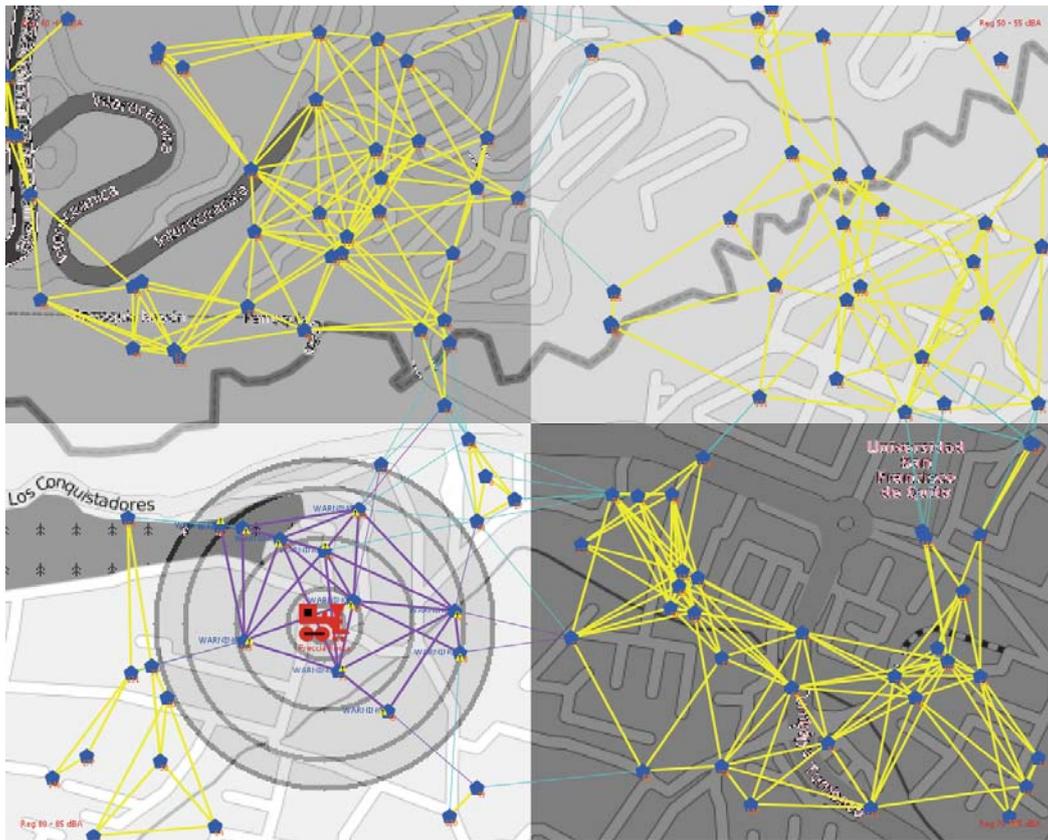


Figura 5. Mappa in NetLogo del monitoraggio dell'inquinamento acustico mediante ViMS: simulazione di una fonte improvvisa di rumore (passaggio di un Treno).

8 Bibliografia

[aisweb] AISWeb The Online Home of Artificial Immune Systems, <http://www.artificial-immune-systems.org/>

[bhu13] B. Bhushan, S.S. Pillai, "Particle Swarm Optimization and Firefly Algorithm: Performance analysis" in Advance Computing Conference (IACC), 2013 IEEE 3rd International.

[bon99] E. Bonabeau, M. Dorigo, G. Theraulaz , "Swarm Intelligence: From Natural to Artificial Systems", New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity, Paper: ISBN 0-19-513159-2,1999 .

[car98] G.D. Caro and M. Dorigo "AntNet: a mobile agents approach to adaptive routing," Proceedings of the Thirty-First Hawaii International Conference on System Science, vol.7, pp.74-83, 1998.

[car98b]G. D. Caro and M. Dorigo, "Two ant colony algorithms for best-effort routing in datagram networks," Proceedings of the Tenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98), pp.541-546, 1998.

[dec02] L.N. de Castro, F.J. Von Zuben, "Learning and Optimization Using the Clonal Selection Principle" (PDF). IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE) 6 (3): 239–251. 2002.

[for94] S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri, "Self-nonsel self discrimination in a computer", in Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on. 1994.

[fos08] K.R. Foster, "Behavioral ecology: Altruism", In: Encyclopedia of Ecology, pp 154-159. Elsevier, 2008

[kar08] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", Applied Soft Computing 8 (2008), pp. 687-697, 2008.

[kar12] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications " Springer Science+Business Media B.V. 2012.

[kav10] A. Kaveh, S. Talatahari, "A novel heuristic optimization method: charged system search", Acta Mechanica 213, 267-289, 2010.

[ken01] J. Kennedy, R.C. Eberhart, "Swarm Intelligence", Morgan Kaufmann publishers, 2001.

[mar07] Swarm Intelligence , School of Computer and Communication Sciences EPFL, WS 2007-08, http://disalw3.epfl.ch/Teaching/Swarm_Intelligence/AY_2007-08/Lecture/SI_07-08_W01_lecture.pdf

[min11] Minh Tri Tran, Vilmos Simon, "Can altruism spare energy in ad hoc networking?", in MoMM '11 Proceedings of the 9th International Conference on

Advances in Mobile Computing and Multimedia, Pages 214-217, ACM New York, NY, USA ©2011

[mir90] R. Mirollo, S. Strogatz, "Synchronization of pulse-coupled biological oscillators", SIAM, 50(6):1645–1662, 1990

[mis13] A.K. Mishra, M.N.Das, T.C. Panda, "Swarm Intelligence Optimization: Editorial Survey", International Journal of Emerging Technology and Advanced Engineering, Volume 3, issue 1, January 2013.

[pay05] Swarm Intelligence Resources
<http://faculty.washington.edu/paymana/swarm/>.

[rab07] P. Rabanal, I. Rodríguez, and F. Rubio, "Using River Formation Dynamics to Design Heuristic Algorithms", ;in Proc. UC, 2007, pp.163-177.

[ras09] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "GSA: A Gravitational Search Algorithm", Information Sciences 179 (2009) 2232–2248

[sha09] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm". International Journal of Bio-inspired Computation (IJBIC) 1 (1/2): 71–79, 2009

[wer05] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects",in SenSys '05 Proceedings of the 3rd international conference on Embedded networked sensor systems, pages 142 - 153, ACM New York, NY, USA ©2005.

[fol02] G. Folino, G. Spezzano, "An Adaptive Flocking Algorithm for Spatial Clustering", in PPSN VII Proceedings of the 7th International Conference on Parallel Problem Solving from Nature, Pages 924-933, Springer-Verlag London, UK ©2002

[fol09] G. Folino, A. Forestiero, G. Spezzano, "An adaptive flocking algorithm for performing approximate clustering", in Information Sciences: an International Journal archive, Volume 179 Issue 18, August, 2009, Pages 3059-3078, Elsevier Science Inc. New York, NY, USA.

[for13] A. Forestiero, C. Pizzuti, G. Spezzano, "A single pass algorithm for clustering evolving data streams based on swarm intelligence", in Data Mining and Knowledge Discovery archive, Volume 26 Issue 1, January 2013, Pages 1-26, Kluwer Academic Publishers Hingham, MA, USA.

[rey87] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model". Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'87) (ACM). doi:10.1145/37401.37406. (1987).

[mar07] Swarm Intelligence , School of Computer and Communication Sciences EPFL, WS2007-08,http://disalw3.epfl.ch/Teaching/Swarm_Intelligence/AY_2007-08/Lecture/SI_07-08_W01_lecture.pdf

[bon99] E. Bonabeau, M. Dorigo, G. Theraulaz , "Swarm Intelligence: From Natural to Artificial Systems", New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity, Paper: ISBN 0-19-513159-2,1999 .

[jel05] M. Jelasity, A. Montresor, O. Babaoglu.. "Gossip-based aggregation in large dynamic networks." *ACM Trans. Comput. Syst.* 23, 3 (August 2005), 219-252. 2005.

[Dot10] Dötsch, F.; Denzinger, J.; Kasinger, H. & Bauer, B. (2010), Decentralized Real-Time Control of Water Distribution Networks Using Self-Organizing Multi-agent Systems., in 4th IEEE International Conference on [Self-Adaptive and Self-Organizing Systems \(SASO\)](#), IEEE Computer Society, pp. 223-232, 2010.

[Bic10] N. Biccocchi, M. Mamei, F. Zambonelli, "Self-organizing Virtual Macro Sensors", *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol 7, Issue 1, 2012.