



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

Controllori cognitivi Soluzioni a confronto

Antonio Francesco Gentile, Davide Macrì, Emilio Greco

RT-ICAR-CS-22-06

Maggio 2022



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)

– Sede di Cosenza, Via P. Bucci 8-9C, 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, Via Ugo La Malfa, 153, 90146 Palermo, URL: www.icar.cnr.it

Sommario

<i>Introduzione.....</i>	<i>3</i>
<i>Sistemi basati su controllori cognitivi.....</i>	<i>4</i>
<i>Capacità di acquisire conoscenza</i>	<i>6</i>
<i>Sistema di gestione del bilanciamento energetico di un edificio</i>	<i>7</i>
<i>Sistema di controllo del confort visivo</i>	<i>11</i>
<i>Sistema di controllo del confort termico.....</i>	<i>15</i>

Introduzione

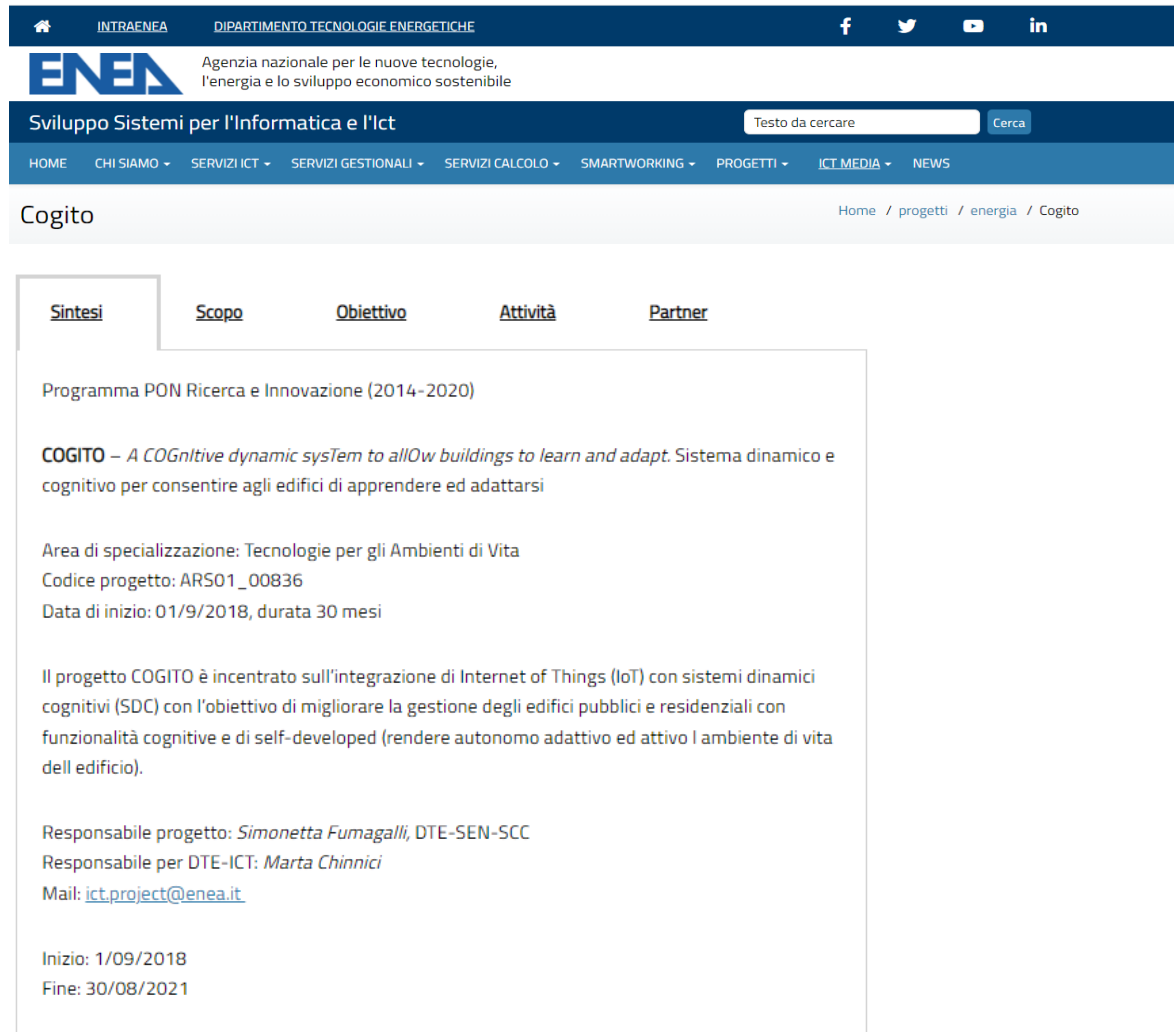
Il presente documento ha come obiettivo quello di confrontare approcci differenti per la progettazione e lo sviluppo di un sistema di controllo cognitivo nel contesto delle Smart Building. In particolare analizzeremo gli approcci e le metodologie adoperate per lo sviluppo di tre tipi di controllori basati su tecniche di Reinforcement Learning (RL) e che si differenziano per velocità di apprendimento, risorse impiegate e versatilità nell'utilizzo. L'obiettivo finale è quello di fornire una metodica ed un approccio pratico alla risoluzione di problemi di controllo per sistemi complessi ed altamente variabili.

In questo documento verranno presentati tre algoritmi di ML e i risultati ottenuti su test eseguiti in ambiente reale.

Sistemi basati su controllori cognitivi

Con tale termine vogliamo definire una classe di sistemi dinamici costituiti da dispositivi, algoritmi ed infrastrutture, con funzionalità cognitive e di self-developed. Nel contesto degli smart buildings ciò significa realizzare delle infrastrutture capaci di rendere autonomo, adattivo ed attivo l'ambiente di vita di un edificio.

Di seguito si riporta la scheda di progetto pubblicata da ENEA che enfatizza degli elementi caratterizzanti un sistema cognitivo.



The screenshot shows the ENEA website interface. At the top, there is a navigation bar with the ENEA logo and the text 'Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile'. Below this, there is a search bar and a menu with options like 'HOME', 'CHI SIAMO', 'SERVIZI ICT', 'SERVIZI GESTIONALI', 'SERVIZI CALCOLO', 'SMARTWORKING', 'PROGETTI', 'ICT MEDIA', and 'NEWS'. The main content area is titled 'Cogito' and includes a breadcrumb trail: 'Home / progetti / energia / Cogito'. Below the title, there are five tabs: 'Sintesi', 'Scopo', 'Obiettivo', 'Attività', and 'Partner'. The 'Sintesi' tab is selected and contains the following text:

Programma PON Ricerca e Innovazione (2014-2020)

COGITO – A *COGNitive dynamic sysTem to allOw buildings to learn and adapt*. Sistema dinamico e cognitivo per consentire agli edifici di apprendere ed adattarsi

Area di specializzazione: Tecnologie per gli Ambienti di Vita
Codice progetto: ARS01_00836
Data di inizio: 01/9/2018, durata 30 mesi

Il progetto COGITO è incentrato sull'integrazione di Internet of Things (IoT) con sistemi dinamici cognitivi (SDC) con l'obiettivo di migliorare la gestione degli edifici pubblici e residenziali con funzionalità cognitive e di self-developed (rendere autonomo adattivo ed attivo l'ambiente di vita dell'edificio).

Responsabile progetto: *Simonetta Fumagalli*, DTE-SEN-SCC
Responsabile per DTE-ICT: *Marta Chinnici*
Mail: ict.project@enea.it

Inizio: 1/09/2018
Fine: 30/08/2021

In particolare un sistema cognitivo si caratterizza nell'avere le seguenti proprietà:

1. Capacità di acquisire conoscenza e quindi apprendere dall'ambiente
2. self-developed
3. autonomo
4. adattivo
5. attivo

Analizzeremo pertanto le soluzioni tecnologiche adoperate all'interno del progetto COGITO cercando di enfatizzare gli aspetti salienti e la rispondenza ai punti sopra descritti.

Di seguito diamo una breve descrizione delle problematiche che sono state affrontate nella realizzazione di un edificio cognitivo e le soluzioni proposte:

Sistema di gestione del bilanciamento energetico dell'edificio:

Schedulazione di carichi elettrici in un edificio utilizzando un sistema cognitivo.

Obiettivo:

In questo primo progetto abbiamo voluto realizzare **un servizio di supporto alle decisioni per la schedulazione di alcuni carichi domestici**. In sintesi, date le curve di andamento dei costi energetici, il profilo di produzione fotovoltaico ed il profilo di consumo di ogni elettrodomestico, sia controllabile che non, ci poniamo l'obiettivo di stabilire una sequenza di attivazione dei dispositivi, tenendo conto delle preferenze utente, tale da minimizzare il costo energetico evitando picchi o sovraccarico della rete.

Sistema di controllo del confort visivo:

Controllare l'apertura e l'inclinazione di una tapparella motorizzata in modo da ottimizzare l'energia consumata e confort visivo

Obiettivo:

In questo secondo progetto ci siamo posti l'obiettivo di realizzare un sistema di controllo ad anello chiuso basato sull'interazione con l'utente (human centric lighting control). Si tratta della realizzazione di sistema di controllo ad anello chiuso per il confort luminoso basato su reclami. Il sistema di controllo deve individuare la posizione delle tapparelle e l'intensità delle luci in modo da sfruttare l'apporto termico gratuito esterno per effetto serra col fine di ottimizzare i consumi e mantenere il confort interno. Inoltre le azioni compiute devono prevedere eventuali reclami dovute a situazioni di abbagliamento o di poca luce.

Sistema di controllo del confort termico:

Controllo del sistema di climatizzazione utilizzando un sistema cognitivo

Obiettivo:

In questo terzo approccio si è voluto realizzare un sistema di controllo capace di acquisire informazioni ambientali e comportamentali da parte degli utenti. Tali informazioni verranno quindi utilizzate dal controllore cognitivo per inferire una politica di controllo che ottimizzi il consumo energetico e che riduca condizioni di discomfort per gli utenti.

Capacità di acquisire conoscenza

Prima di addentrarci nell'analisi delle tre soluzioni proposte e quindi individuare la rispondenza a questo requisito, andiamo a meglio esplicitare il significato attribuito alla capacità di apprendimento di un sistema elettronico.

Ovvero ci muoviamo all'interno di quell'ambito di ricerca denominato Machine Learning. In quest'area troviamo diverse tecniche utilizzate al fine di consentire a macchine elettroniche di inferire conoscenza. Una macchina può inferire nuova conoscenza o se vogliamo una regola che governa un sistema, analizzando semplicemente i dati che raccoglie nel tempo. A partire dai dati, quindi è possibile estrapolare una o più correlazioni che accomuna varie grandezze, ad esempio pensiamo al livello CO² in una stanza che aumenta all'aumentare del numero di persone presenti. Per catturare quindi questa regola si adoperano tecniche di Deep Learning che fanno uso di reti neurali, le quali immagazzinano le correlazioni esistenti tra le varie grandezza di un sistema, per poi essere utilizzate come elemento di predizione del comportamento dello stesso sistema. Questa tecnica prevede pertanto l'acquisizione di un set di dati e nella maggior parte dei casi anche la loro etichettatura da parte di un operatore umano. A questa tecnica si affianca quella nota col nome di Reinforcement Learning, che si allinea molto di più a quello che è il processo di apprendimento dell'essere umano, ovvero il sistema impara dai propri errori e dai propri successi. La tecnica in questo caso prevede la realizzazione di un sistema di feedback per un decisore. Il sistema di feedback deve avere la capacità di valutare ogni azione intrapresa dal decisore e quindi elargire una ricompensa o una penalità in funzione del risultato raggiunto. Se ad esempio in una stanza fa caldo ed il decisore accende il riscaldamento, il sistema di feedback intercetta l'azione ed elargisce un feedback negativo al decisore. Naturalmente in tali tipi di sistemi è necessario, o quantomeno è auspicabile utilizzarli in contesti dove sono presenti le seguenti condizioni:

- a) difficile formalizzare esattamente il problema (es. intervento umano)
- b) presenza di rumore e/o incertezza
- c) mancanza di conoscenza o di un modello associato al problema da risolvere

Affinché si possa formulazione un problema come problema di Reinforcement Learning è necessario prendere in considerazione quando segue:

- ✓ occorre trasformare il problema dato come problema decisionale one-step
- ✓ il sistema è modellato ed evolve per stati
- ✓ il passaggio da uno stato ad un altro è compiuto attraverso un'azione

- ✓ la scelta dell'azione è eseguita da un Agente in maniera “autonoma”
- ✓ l'agente riceverà un **premio** se l'azione da lui scelta risulterà corretta.

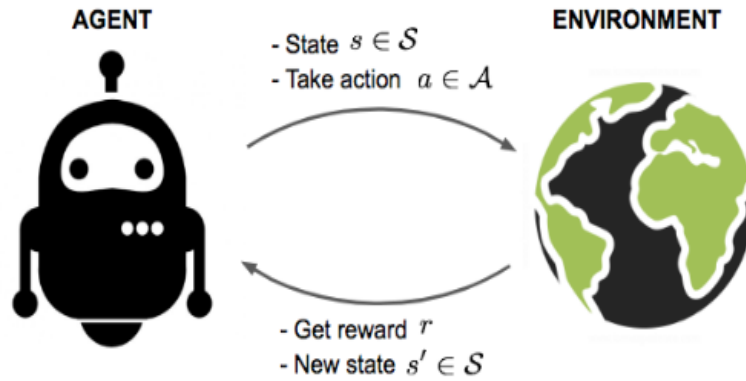


Figura 1 - Modello Reinforcement Learning

Sistema di gestione del bilanciamento energetico di un edificio

Per quando riguarda il problema di schedulazione dei carichi, possiamo sintetizzare l'approccio col seguente grafo:

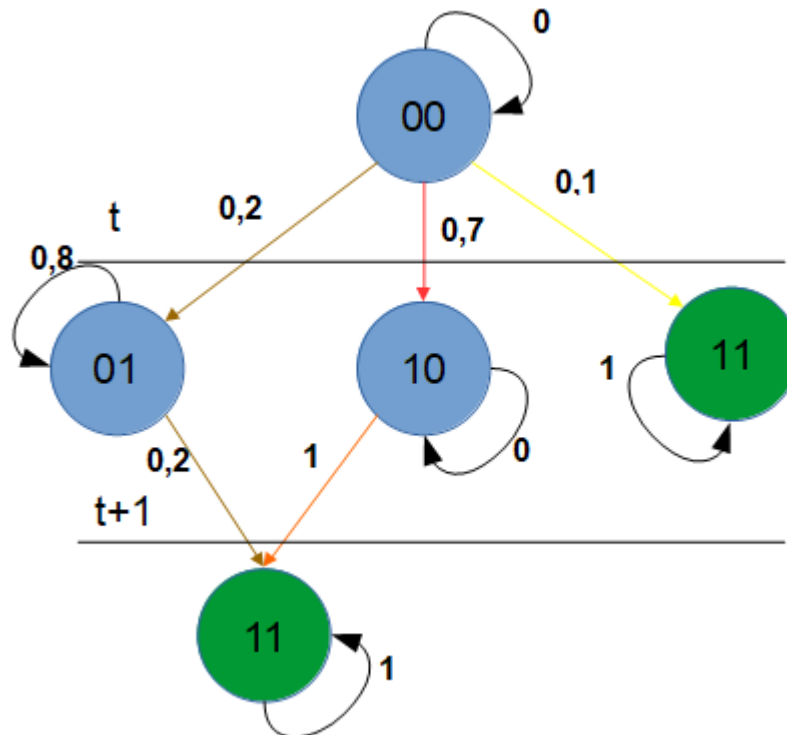


Figura 2 - MDP (markov decision process)

Partendo da uno stato iniziale definito dagli elettrodomestici su cui si richiede l'attivazione, si deve raggiungere uno stato finale, costituito da dispositivi tutti attivi, rispettando i vincoli di costo.

Nel nostro esempio, il decisore dovrà cercare il percorso tra il nodo iniziale e quello terminale massimizzando la ricompensa che possiamo attribuire al peso degli archi:

- Percorso 1 con ricomp. 0,1
- Percorso 2 con ricomp. 1,7
- Percorso 3 con ricomp. 0,4

Il sistema di acquisizione e di supporto alle decisioni può essere raffigurato dal seguente schema:

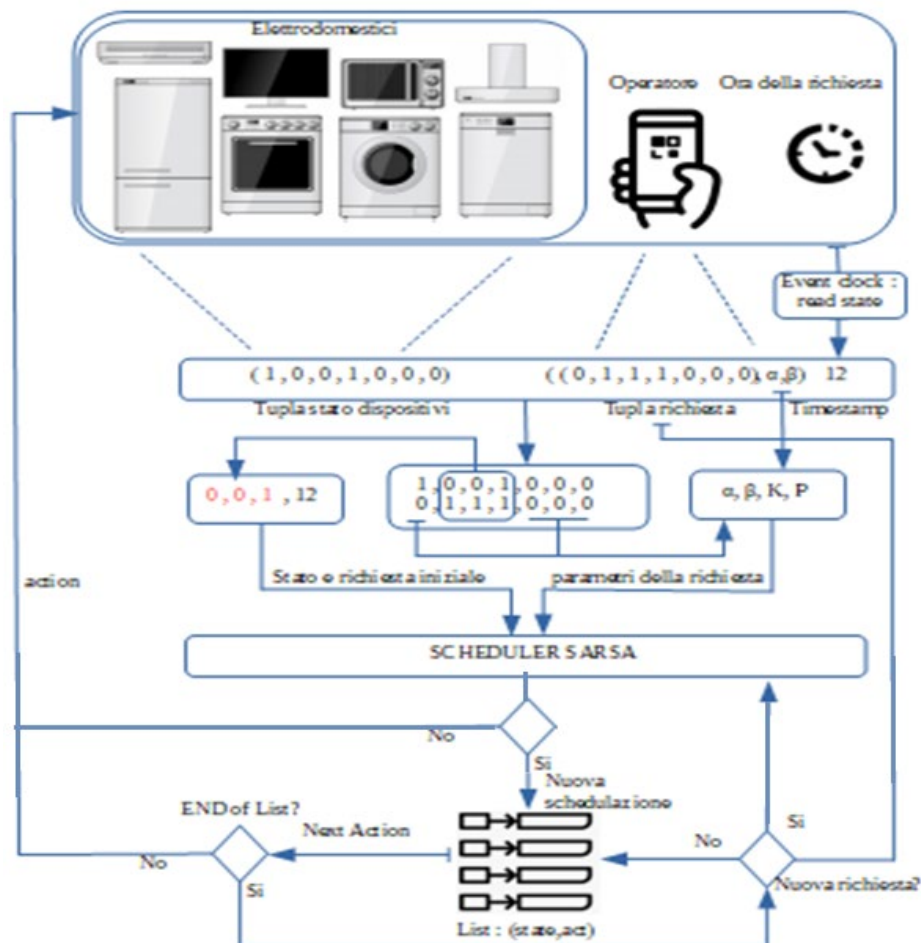


Figura 3 - Caso d'uso - schedulazione carichi

Rimanendo nella descrizione dei casi d'uso, possiamo sintetizzare il processo di utilizzo del sistema attraverso i seguenti step:

1. **Richiesta schedulazione utente**
 - scelta algoritmo Q-Learning /SARSA
 - scelta dispositivi
 - tempo di esecuzione
 - scelta tempi di differimento
2. **Elaborazione di un piano di attivazione**
 - produzione grafico di schedulazione
 - report vincoli non soddisfatti
 - report risparmio ottenibile
3. **Conferma esecuzione**
 - creazione di cronjob
 - attivazione attraverso msg mqtt

Più in dettaglio, la fase di elaborazione di un piano di attivazione segue la seguente logica operativa:

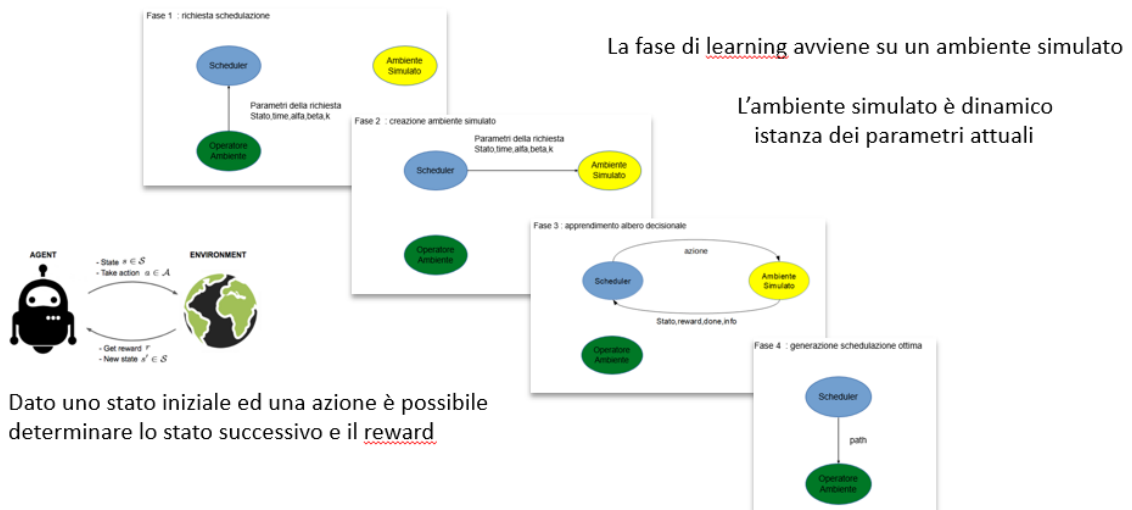


Figura 4 - Elaborazione schedulazione

Ogni istanza inoltrata da un utente produce la creazione un ambiente simulato. Il modulo, che fornisce il reward al decisore in funzione di ogni coppia stato azione, elabora le informazioni acquisite nell'arco del tempo dai sensori ambientali (Produzione fotovoltaico, consumo carichi controllabili e non, etc.).

I risultati prodotti sono i seguenti:

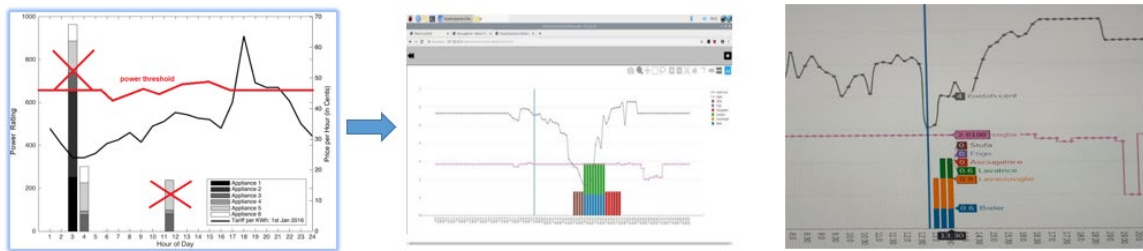


Figura 5 - Esempi di schedulazione

Vediamo quindi quali solo i vantaggi e le caratteristiche di questo approccio:

1. La cardinalità delle coppie stato-azione è tale da consentire **un approccio basato su Q-table**
2. L'addestramento non può avvenire on-line. Si esegue un **addestramento su un ambiente simulato** istanziato ad ogni richiesta di elaborazione in funzione dei dati correnti.
3. **L'ambiente simulato è istanziato on demand.** Elabora i dati raccolti (profilo consumi, produzione fotovoltaico, costi energetici) e restituisce un valore di reward per ogni richiesta espressa dal decisore.
4. Il sistema converge in poche centinaia di iterazioni (decine di secondi) ad una soluzione. **Sistema in tempo reale**
5. **Sistema è cognitivo:** apprende una politica di schedulazione attraverso una **strategia trial-and-error**
6. **Self-developed:** per ogni richiesta viene prodotta una nuova istanza del problema ed un nuovo environment
7. **Autonomo:** è possibile lanciare l'esecuzione di elaborazioni ancor prima che avvenga effettuata la richiesta in modo da proporre all'utente degli scenari ottimali
8. **Adattivo:** al variare del numero dei dispositivi, della loro potenza, dell'autoproduzione e del prezzo dell'energia alla rete, il sistema rielabora una nuova strategia di attuazione.
9. **Attivo:** per questo specifico problema non è richiesto questo requisito poiché si vuole lasciare autonomia all'utente e non al decisore.

Sistema di controllo del confort visivo

All'interno del progetto COGITO sono stati proposti e sviluppati tre diversi tipi di sistemi di controllo per il confort visivo:

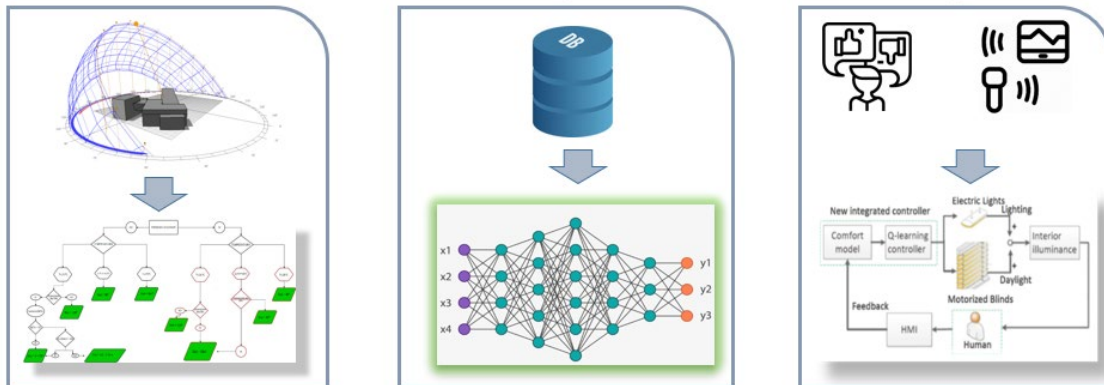


Figura 6 - Modelli di controllo confort visivo

Il primo sulla destra rappresenta un modello di gestione parametrizzato in funzione dell'ambiente in cui viene collocato. Questo modello viene generato attraverso l'uso di un software di illuminotecnica e considera come input, oltre che ai parametri fisici dell'ambiente, la misura dell'irraggiamento solare incidente sulla superficie vetrata.

Il secondo modello, basato sulla tecnica del deep learning, genera un modello simile al primo, ma questa volta viene utilizzato un dataset di informazioni prodotte in simulazioni attraverso un software di illuminotecnica.

Infine il terzo modello cerca di inglobare tutte le caratteristiche di un sistema cognitivo, puntando sull'**autoapprendimento** on-line e all'**adattabilità** ad ogni ambiente in cui viene collocato, senza conoscenza alcuna dei suoi parametri fisici.

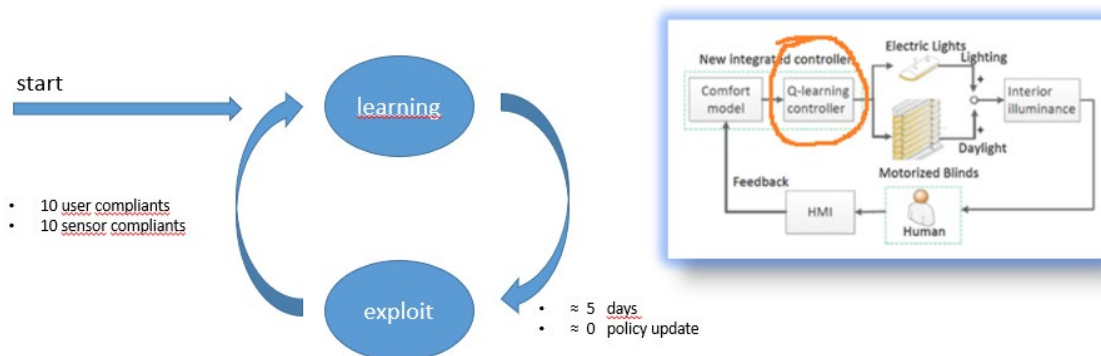


Figura 7 - Ciclo di adattamento

Il sistema diventa **autonomo ed adattativo** grazie ai meccanismi di Reinforcement learning messi in campo. Il sistema è capace di riadattarsi ai cambiamenti ambientali innescando una fase di learning ogni volta che si ritiene necessario.

Questo può avvenire tenendo il conto del numero di reclami acquisiti nell'arco di una prefissata finestra temporale.

Ad esempio se nell'arco di cinque giorni vengono recepiti 10 reclami provenienti dagli utenti o 10 reclami provenienti dai sensori (ovvero sfioramento del valore di luminosità oltre le soglie impostate), in automatico il sistema si porta in modalità di training. La fase di training, nel nostro caso studio si conclude nell'arco di 5 giorni. A seguito di ciò il controllore interagisce sul sistema prevedendo condizioni di discomfort ed agisce di conseguenza.

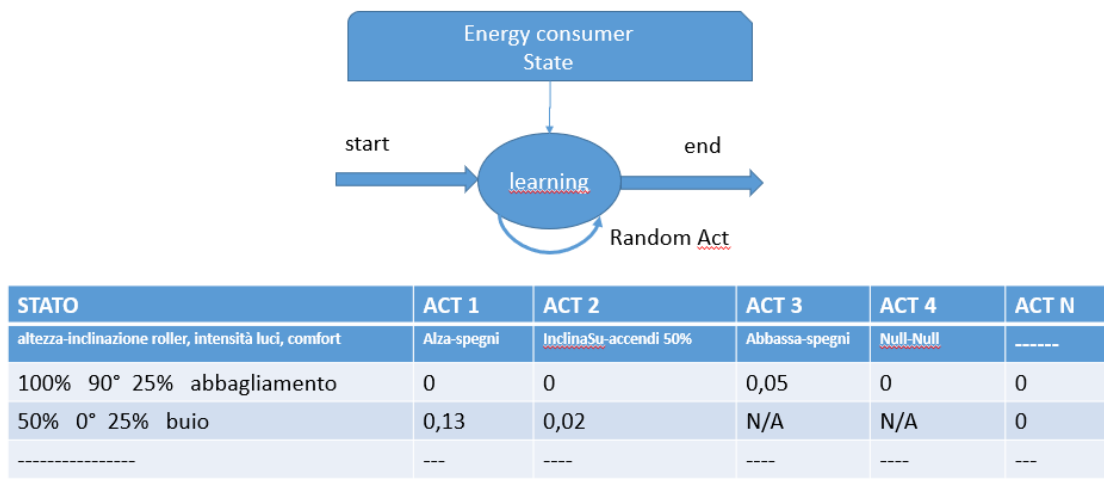


Figura 8 - Self-programming

La caratteristica di **self-programming** viene realizzata come segue:

In fase di learning il decisore compie un certo numero di scelte causali e ne valuta l'impatto sull'ambiente. La valutazione avviene attraverso le misure dirette sui consumi e sul livello di luminosità raggiunto. Ad esempio se l'azione porta in uno stato di discomfort, l'azione precedente viene valutata con reward negativo e viene scelta immediatamente una nuova azione. Se invece il sistema si mantiene in condizioni di confort luminoso, si calcola l'energia media consumata dall'esecuzione dell'azione fino al prossimo cambio di stato. Il valore letto viene normalizzato e restituito come valore di reward.

La caratteristica di **proattività** viene realizzata utilizzando il seguente approccio:

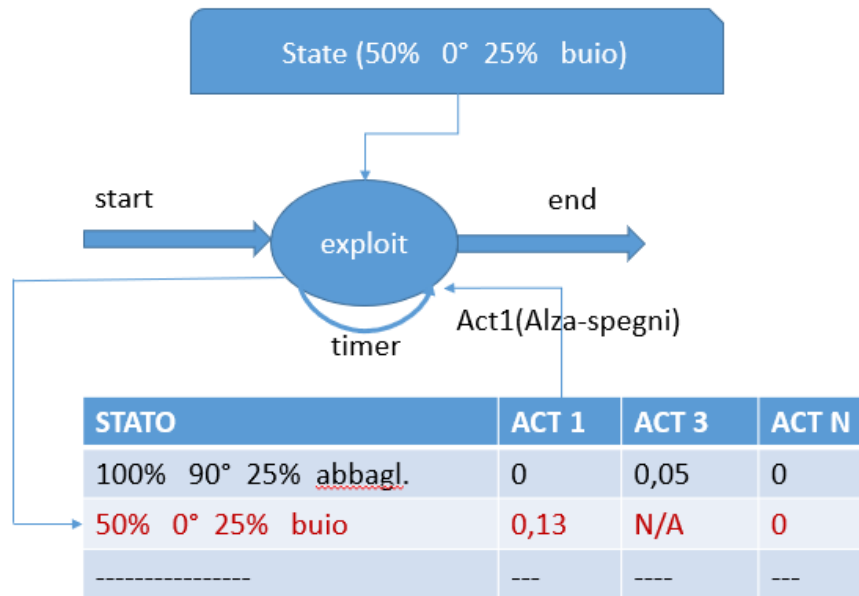


Figura 9 - Gestione della proattività

Il controllore non deve comportarsi come un sistema tradizione, ovvero interviene se riceve un feedback negativo dall'anello di retroazione, ma deve intervenire in modo proattivo per scongiurare situazioni di disconfort, in prima istanza, ma soprattutto deve intervenire in modo attivo nel cercare di ottimizzare le risorse energetiche.

Per realizzare questo meccanismo vengono usati dei timer che hanno una duplice funzione. Innanzitutto servono a cadenzare gli step decisionali, dato che operiamo con un problema di Reinforcement learning di tipo Time division, poi ci occorrono per stimolare il controllore a selezionare ripetutamente delle azioni che consentono di inseguire l'evoluzione del sole durante l'arco della giornata.

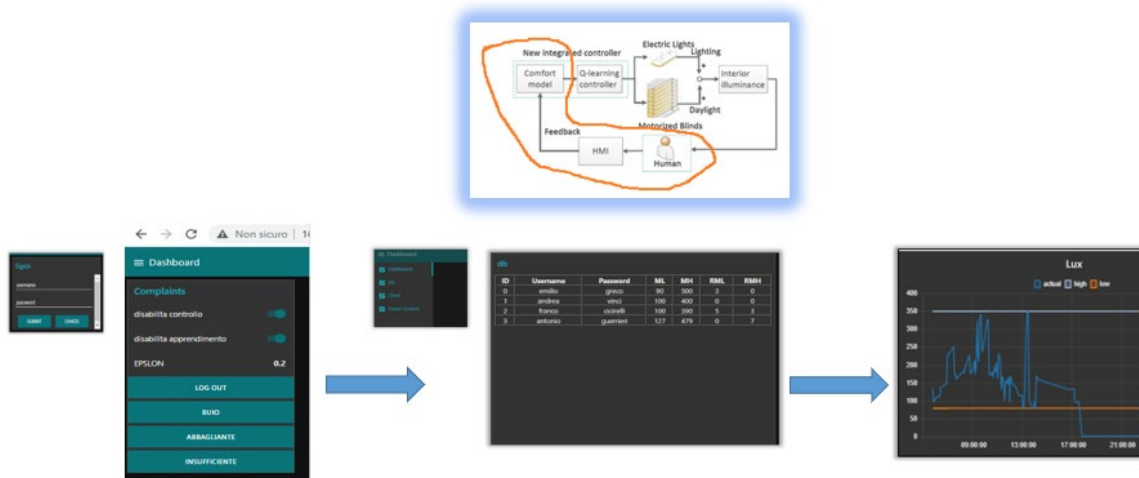


Figura 10 - human-centric approach

Infine descriviamo come è stata realizzata la caratteristica human-centric del sistema. Abbiamo già accennato al fatto che il sistema lavora sui reclami. Reclami che possono essere di due tipi: reclami utente e reclami provenienti dai sensori. Abbiamo già visto come al raggiungimento di un certo numero il controllore passa in modalità di training. Occorre aggiungere a tutto questo l'idea di abbinare ad ogni reclamo le informazioni ambientali presenti al momento della sua sottomissione. Se ad esempio un utente produce un reclamo di abbagliamento, andremo a salvare il valore medio di luminosità misurato in quel momento. Questo valore servirà in un secondo momento per valutare un aggiornamento della soglia massima consentita. Per mediare le preferenze di tutti gli occupanti della stanza viene usato un approccio che prende il nome dal suo ideatore, ovvero il miglioramento paretiano.

Possiamo ora sintetizzare le differenze sostanziali col primo sistema descritto all'inizio di questo lavoro:

1. In questo approccio si esegue un **addestramento on-line** su misure reali
2. Il reward valuta il consumo generato da tutti i dispositivi presenti (HVAC, Luci, Ventilazione) per inferire le azioni energivore che fanno transitare il sistema tra due stati di confort. **Discriminante delle azioni energy save**.
L'intervento manuale dell'operatore durante la fase di training viene gestita come apprendimento supervisionato.
3. Sistema cognitivo, apprende:
 - le soglie sul confort percepito attraverso l'analisi dei reclami
 - una politica di azionamento proattivo attraverso la strategia trial-and-error su ambiente reale
 - gli schemi comportamentali basandosi sulla sistematicità ed attraverso l'apprendimento supervisionato
4. Self-developed: il sistema opera in assenza di dati e di un modello che acquisisce con osservazioni dirette
5. Autonomo: se il modello pre-acquisito non risponde più alle aspettative il sistema si riaddestra
6. Adattivo: Il sistema ha informazioni sui consumi prodotti dai dispositivi presenti, sul livello di luminosità e nient'altro. Può essere inserito in qualsiasi ambiente senza pre-addestramento.
7. Attivo: il sistema interviene prima che si possa verificare una situazione di discomfort muovendosi tra stati di confort contigui: approccio incrementale

Sistema di controllo del confort termico

Nel paragrafo precedente abbiamo descritto in che modo possono essere usate le reti neurali per produrre un modello di funzionamento di un ambiente per poi essere utilizzato per eseguire previsioni o effettuare delle azioni. Il modello che avevamo proposto si basava su un procedimento di raccolta dati ed addestramento off-line, dopo di che poteva essere utilizzato allo stesso modo di un sistema a logica cablata.

Il passo successivo a questo approccio è fare in modo che il modello non rimanga invariato nel corso del tempo ma che si adatti di volta in volta alle nuove condizioni che vi si presentano. In questo caso si fa ricorso al seguente schema di controllo:

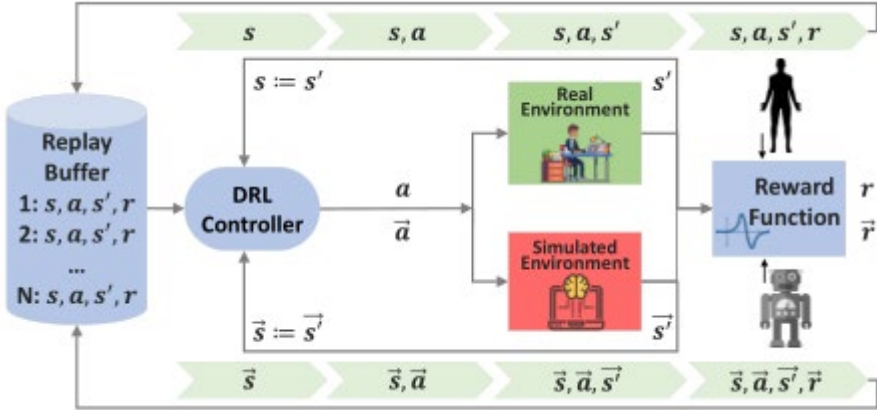
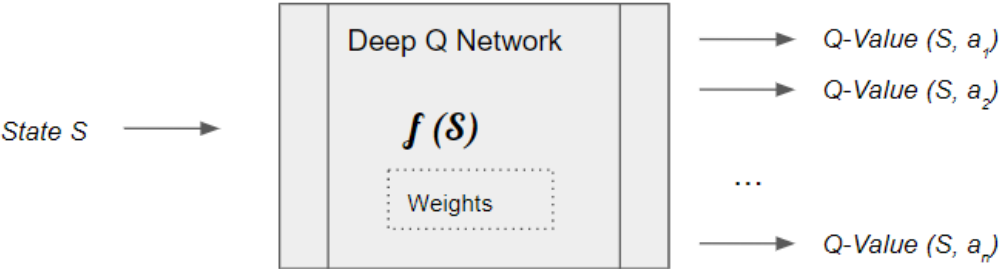


Figura 11- Deep-Q Learning

Presente in letteratura sotto il nome di Deep-Q Learning. Quando il problema di Reinforcement learning da modellare è costituito da un numero piccolo di stati ed di azioni è possibile immagazzinare i valori della funzione utilità utilizzata dall' algoritmo all'interno di una matrice di dimensione (s,a), dove s rappresenta il numero di stati ed a il numero delle azioni. Tuttavia, in uno scenario reale, il numero di stati potrebbe essere enorme, rendendo impossibile dal punto di vista computazionale costruire una tabella. Per affrontare questa limitazione usiamo una funzione Q piuttosto che una tabella Q.



Il modo migliore per approssimare una funzione è per l'appunto usare una rete neurale. Nello schema raffigurato nella figura 11, notiamo un elemento che in uno schema di circuito Q_learning non troviamo, il replay buffer. Ma cos'è e a cosa serve questo buffer, perché non inviamo i dati provenienti dal sistema direttamente alla rete neurale?

I motivi per cui non è possibile fare ciò sono le seguenti:

- 1) Sappiamo che le reti neurali in genere richiedono un batch di dati. Se lo addestrassimo con campioni singoli, ogni campione e i gradienti corrispondenti avrebbero troppa varianza e i pesi della rete non convergerebbero mai.
- 2) una best practice consiste nel selezionare un batch di campioni dopo aver mescolato i dati di addestramento in modo casuale. Le azioni sequenziali sono altamente correlate tra loro il che produce il problema chiamato Catastrophic Forgetting

Lo schema raffigurato non è del tutto completo, in realtà è necessario quindi un modulo aggiuntivo che gestisce le informazioni presenti all'interno del replay buffer. Questo elemento prende il nome di Experience replay.

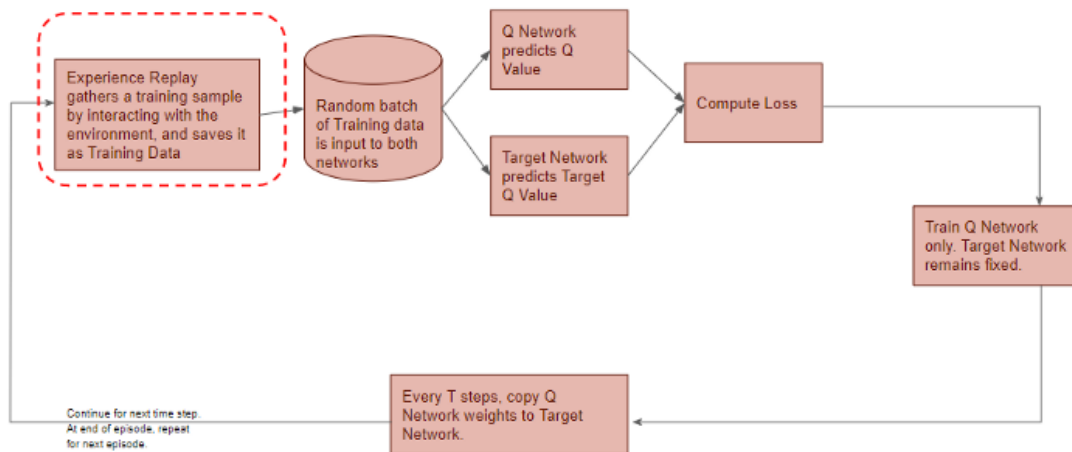


Figura 12 - Deep-Q Learning schema

L'experience replay pertanto è una sorta di coordinatore dell'intero sistema, in quanto avvia la fase di generazione dei dati di addestramento e utilizza la rete Q per selezionare un'azione ϵ -greedy da utilizzare sull'ambiente per migliorare la politica di controllo. Infine preleva l'informazione dello stato e del reward raggiunto dal sistema che la salva come campione di dati per il processo di addestramento del modello.

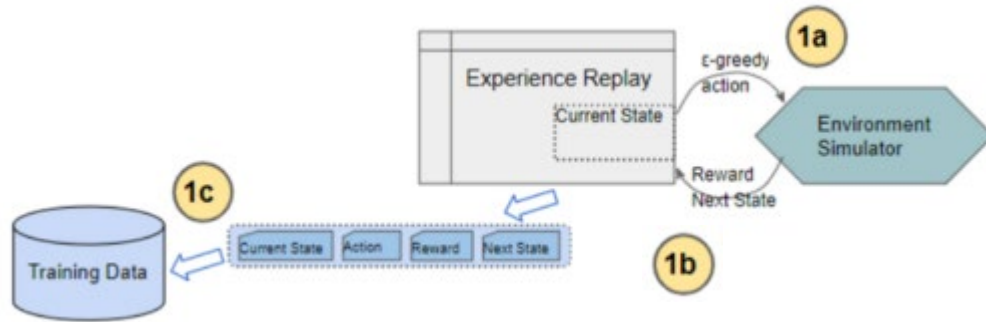


Figura 13 - Experience replay

Nello schema di funzionamento del Deep-Q Learning notiamo come anche in questo caso adoperiamo un ambiente simulato per addestrare il modello.

Possiamo ora sintetizzare le differenze sostanziali col gli altri approcci:

1. esegue un **addestramento off-line** ma su misure reali
2. il sistema è sicuramente di tipo cognitivo
3. Self-developed: il sistema opera in assenza di dati e di un modello che acquisisce con osservazioni dirette
4. Autonomo: l'experience replay mantiene sempre attiva la fase di training rendendo il sistema autonomo ed adattativo.