

Università degli studi di Padova
Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

Tesi di Laurea

OTTIMIZZAZIONE DELL'ENERGY STORAGE
PER APPLICAZIONI SMART GRID

Relatore:

Prof. Tomaso Erseghe

Correlatore:

Prof. Andrea Zanella

Laureando:

Claudio Giuseppe Codemo

Ottobre 2012

A.A. 2011/2012

*Dedico la tesi ai miei genitori:
grazie di cuore per tutto.*

Indice

Indice	iii
Sommario	iii
Introduzione	v
1 Primo modello di sistema	1
1.1 Modello delle batterie	1
1.2 Modello di costo	2
1.3 Modello di stato	3
1.4 Def. del problema di ottimizzazione	4
1.5 Modello di generazione potenza	5
1.6 Unico vincolo: E_{\max} finita	10
1.7 Aggiunta di vincoli a batteria	14
1.8 Simulazioni numeriche	15
1.9 Caso con due batterie	17
1.9.1 Risultati delle simulazioni numeriche	19
2 RL	27
2.1 Interfaccia del RL	27
2.2 MDP	29
2.2.1 <i>Value Functions</i>	30
2.2.2 <i>Optimal Value Functions</i>	32
2.3 DP	34
2.3.1 <i>Policy Evaluation</i>	35
2.3.2 <i>Policy Improvement</i>	36
2.3.3 <i>Policy Iteration</i>	38
2.3.4 <i>Value Iteration</i>	40
2.3.5 <i>Generalized Policy Iteration</i>	40
2.3.6 Efficienza della DP	42
3 Soluzione tramite DP	45
3.1 Applicazione di DP al pr. di ottimizzaz.	45
3.2 Simulazioni numeriche con DP	48
3.3 Simulazioni con 3 funzioni di costo lineari a tratti	55
3.4 Simulazioni ad hoc per confronto tra DPM e DPI	59
3.5 Doppia soglia vs strategia ottima	62

4	Secondo modello di sistema	65
4.1	Nuovo modello di sistema	65
4.1.1	Nuovo modello di batteria	65
4.1.2	Nuovo modello di costo	68
4.2	Def. del nuovo problema di ottimizzazione	70
4.3	Simulazioni con nuovo modello di sistema	72
4.3.1	Risultati relativi al caso con una sola fascia oraria	73
4.3.2	Risultati relativi al caso con due fasce orarie	76
5	Approfondimento analitico	85
5.1	Caratterizz. analitica della strategia ottima	85
5.2	Proprietà di V^*	88
5.3	Caso con costo lineare	91
5.4	Caso con costo lineare a tratti	96
5.5	Caso con ammortamento della batteria	101
5.6	Caso con perdita di carica nel tempo	103
5.7	Verifica numerica dei risultati	105
	Sviluppi futuri	107
	Bibliografia	109
	Ringraziamenti	111

Sommario

La presente tesi si pone il problema di studiare la strategia ottima di *energy storage* per minimizzare il costo energetico da pagare da parte degli utenti di una micro-grid che abbia a disposizione una batteria per immagazzinare energia. La logica alla base di questo possibile risparmio è quella di caricare la batteria in momenti giudicati *convenienti*, in base sia alla funzione di costo dell'energia richiesta alla grid che alla statistica delle richieste di potenza che giungono al suo controllore. Si definiscono i modi tramite i quali vengono caratterizzati batteria, funzione di costo e stato del sistema, che costituiscono *le diverse versioni del modello di sistema* che vengono considerate nella seguente trattazione.

Viene poi formulato il problema di ottimizzazione in modo analitico, sia nella sua versione offline (che costituisce un lower bound sul costo complessivo e suppone di conoscere l'intera sequenza delle richieste di potenza per decidere quale strategia di carica/scarica adottare nei confronti della batteria) sia online (che si traduce in una minimizzazione stocastica rispetto dell'aspettazione del costo totale). Le due versioni del problema si prestano a tecniche risolutive completamente diverse: nel primo caso si può utilizzare l'algoritmo di Viterbi, familiare al mondo delle Telecomunicazioni, nel secondo caso algoritmi di Programmazione Dinamica (che richiedono la stima della statistica della richiesta di potenza), confrontati con un semplice algoritmo a singola soglia noto in letteratura [1], che ha costituito il punto di partenza del seguente lavoro di tesi. Si sono implementati in Matlab gli algoritmi menzionati e si sono simulate e confrontate le loro prestazioni in termini di costo complessivo da pagare da parte degli utenti della grid al variare sia del modello di batteria (e dei suoi parametri) che delle funzione di costo.

Oltre a ciò, in modo simile a quanto eseguito dagli autori di [2], si conduce un approfondimento analitico riguardo alla determinazione della strategia ottima da adottare nei confronti della batteria, che si ritiene *costituisca il contributo più significativo allo stato dell'arte di questa tesi*. Partendo da un'analisi delle proprietà della *state-value function* V (definita nel contesto della Programmazione dinamica) e del suo sub-differenziale rispetto all'azione da eseguire nei confronti della batteria, si dimostra qual è la *la struttura* della miglior *policy* - dal punto di vista stocastico - da applicare per minimizzare il costo complessivo quando la funzione che lo definisce è, rispettivamente, lineare, lineare a tratti e anche quando tiene conto dell'ammortamento per la spesa iniziale della batteria e/o del fatto che essa possa perdere carica nel tempo (purchè il problema di ottimizzazione risultante rimanga *convesso*).

Le simulazioni numeriche precedentemente citate confermano quanto dimostrato dal punto di vista analitico: la strategia ottima di carica/scarica per risolvere il problema di ottimizzazione stocastica del costo ha sì bisogno di una certa potenza di calcolo per poter essere ricavata tramite algoritmi di Programmazione Dinamica, ma poi, finchè la statistica della richiesta di potenza è assunta stazionaria, si può memorizzare all'interno del controllore della grid attraverso un *numero molto limitato* di parametri, costituiti dalle soglie rispetto al livello energetico della batteria, in funzione delle quali si sceglie se intraprendere un'azione di carica/scarica (e di quale entità) o se *non* alterare la quantità di energia presente nel dispositivo di *energy storage*.

Introduzione

Stato dell'arte

L'evoluzione della rete elettrica verso le smart grid [3] è tra i processi di trasformazione più interessanti del sistema elettrico italiano, europeo e mondiale. Da un punto di vista prettamente tecnico – nel campo quindi dell'ingegneria elettronica e delle telecomunicazioni – una smart grid altro non è che una rete di informazioni che coopera con la distribuzione dell'energia elettrica, funzionale in quanto in grado di prevenire cadute di tensione, sprechi e sovraccarichi, in altri termini: una infrastruttura che facilita risparmio energetico ed efficienza.

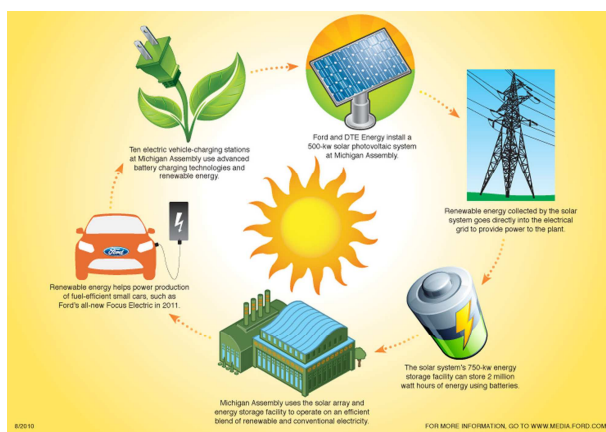


Figura 1: Esempio di smart grid.

in maniera diretta con la rete. Lo stesso consumatore – che secondo un termine coniato di recente diverrebbe *prosumer*, ovvero produttore e consumatore al tempo stesso – può decidere di utilizzare secondo le sue preferenze l'energia da lui stesso prodotta, andando anche ad alimentare in prima persona la rete elettrica. Le smart grid si impongono dunque come soluzione vincente in termini di risparmio economico (oltre che di risparmio energetico) e di sviluppo nel consumo delle fonti rinnovabili. L'utilizzo delle reti intelligenti è mediato da un apposito contatore, lo *smart meter*, con funzionalità digitali e possibilità di lettura remota in grado di garantire la massima efficienza di rete. Lo sviluppo della rete intelligente è strettamente relazionato all'esigenza di raggiungere gli obiettivi europei in termini di risparmio energetico e sfruttamento di fonti di energia rinnovabili. Entro il 2020 l'obiettivo italiano è quello di raggiungere il 17 % di energia prodotta da fonti rinnovabili. In tale circostanza, pertanto, una rete di produzione aggiuntiva, da affiancarsi a quella tradizionale, non può che facilitare la piena realizzazione dell'intento: la massima efficienza distributiva e la massima integrazione della rete con le fonti rinnovabili.

Una transizione fondamentale per il nostro Paese, che entro fine anno raggiungerà i 25000

MW di potenza elettrica installata da fonti non programmabili come fotovoltaico ed eolico [4]: occorre quindi sviluppare soluzioni per la gestione intelligente dell'elettricità in tutte le fasi, dalla produzione all'utilizzo finale, passando per la trasmissione e la distribuzione. In tale contesto, il problema di accumulare l'energia degli impianti non programmabili (ad es il fotovoltaico) è da sempre in primo piano nella ricerca di soluzioni tecnologiche che permettano un accumulo sostenibile dal punto di vista dei costi e della capacità di *storage*. Con il sistema dello scambio sul posto (l'energia consumata e prelevata dalla rete viene compensata in moneta se prodotta in momenti diversi) quello dell'accumulo in impianti domestici non è ancora un tema davvero rilevante, mentre potrà esserlo quando (secondo le prime indiscrezioni) *gli incentivi del Conto Energia saranno alternativi al meccanismo dello scambio sul posto*, oltre a risultare di vitale importanza in impianti in isola (come ad esempio le baite che non possono usufruire di una connessione alla rete). *Panasonic Corporation* ha annunciato che ha già avviato la produzione di massa di un sistema di batterie agli ioni di litio compatto, sicuro, con una lunghissima vita utile sviluppato per le abitazioni europee. Questo tipo di batterie non sono una novità, ma è la prima volta che l'azienda produce su una scala così ampia un sistema progettato per l'Europa. In Germania e nelle altre nazioni europee, i governi stanno *riducendo* il prezzo di acquisto dell'energia solare per frenare la sempre più diffusa generazione domestica di energia fotovoltaica, che si è tradotta in un aumento dei costi energetici. Il provvedimento dovrebbe spingere i consumatori ad *ottimizzare l'autoconsumo della corrente prodotta dai propri impianti fotovoltaici*. Prevedendo che in un futuro non lontano le famiglie genereranno in proprio l'energia che consumeranno, Panasonic ha deciso di rafforzare il proprio settore degli accumulatori e lo sviluppo di sistemi di batterie (agli ioni di litio) per accumulatori a larga scala, la cui richiesta è prevista in rapida crescita. Il sistema di batterie agli ioni di litio consiste in un modulo batteria Panasonic, che ha una capacità nominale di 1.35 kWh e un sistema di gestione della batteria, studiato per controllare la carica e scarica della batteria in base alle esigenze del consumatore. Il sistema accumula l'energia in eccesso, prodotta dal pannello fotovoltaico nelle ore di picco, e scarica la corrente in base alle necessità. È la risposta ideale per chi cerca un accumulatore che aiuti a ottimizzare l'autoconsumo dell'energia generata in proprio, una soluzione che facilita le famiglie nel ridurre la dipendenza dall'energia della rete e contribuisce a diffondere sempre di più l'energia pulita. Un'ottima panoramica tra *tutti* i vantaggi legati ad un uso intelligente dell'*energy storage* si può trovare in [5], tra i quali i principali si possono riassumere in ¹ *Electric Energy Time-shift, Electric Supply Capacity, Load Following, Area Regulation, Time-of-use Energy Cost Management, Electric Service Power Quality, Renewables Energy Time-shift, Renewables Capacity Firming and Wind Generation Grid Integration*.

Le smart grid possono quindi sfruttare il fatto che il costo della produzione e della distribuzione dell'energia è sottoposto a variazioni significative nel corso della giornata, a causa di variazioni nella domanda e nella capacità di generazioni di energia. Queste fluttuazioni sono generalmente nascoste all'utente finale, che paga un *fixed retail energy price*. Un prezzo dell'energia così dinamico, tuttavia, crea un'opportunità per gli utenti, come le unità familiari o i data center, di ridurre il prezzo pagato per l'acquisto di energia sfruttandone le fluttuazioni di valore. Sfortunatamente, nella pratica, *al momento* le richieste di potenza degli utenti non sono ancora così reattive da riuscire a trarre vantaggio dai cambiamenti del prezzo dell'energia, come illustrato in [6, 7, 8, 9]. Una possibile soluzione, come già accennato, è proprio quella di attrezzare i singoli utenti, o un piccolo gruppo di essi organizzato in una micro-grid, con un dispositivo di *energy storage*, e.g., una batteria di capacità abbastanza elevata, che possa immagazzinare energia quando il prezzo è basso e rilasciarla quando il prezzo è più elevato: questo permetterebbe

¹Ho preferito non tradurre questo elenco di vantaggi ottenibili tramite un *energy storage* intelligente perché sono quasi tutti termini tecnici, la cui traduzione suonerebbe ridicola.

effettivamente agli utenti di *trarre beneficio dalla variazione dei prezzi dell'energia senza dover aggiustare i propri consumi*. Esiste una quantità notevole di articoli in letteratura su tematiche di *energy storage optimization*, tra i quali i più importanti per lo sviluppo di questa tesi sono stati sicuramente [1] e [2]: prima di descrivere nel dettaglio il loro apporto nella stesura dei successivi capitoli e la descrizione dei punti in comune e delle differenze nella modellizzazione del sistema rispetto ad essi, penso però valga la pena citare anche [10], articolo che mi ha fatto entrare in confidenza col mondo del Reinforcement Learning (RL) ed in particolare dei Markov Decision Process (MDP).

In tale articolo viene riportata la prima analisi empirica dell'applicazione del RL al problema di massimizzare il guadagno totale degli utenti nel rivendere potenza alla grid. Prima vengono identificate un insieme di variabili che possono effettivamente rappresentare lo stato corrente del sistema e delle caratteristiche di immagazzinamento relativo alla generazione di una potenza unitaria in ogni slot temporale. Poi il problema viene convertito in un MDP e viene applicata la *policy iteration* per calcolare la strategia ottima su una serie fissata di prezzi. Per modellare la generazione di potenza, *l'energia presente nella batteria cresce di un'unità ad ogni slot temporale*, finché non raggiunge il suo valore massimo. Nella realtà, la quantità di energia generata dipende da molti fattori quali, ad esempio, il tempo, l'intensità del sole e/o del vento e dalle prestazioni del dispositivo di generazione della potenza. In tal caso, andrebbe adottato invece un modello più completo del rate di generazione di potenza. Il guadagno ottenuto dall'utente è definito come la quantità di soldi che il cliente riceve dall'operatore energetico per aver venduto la propria energia; il guadagno complessivo è la somma cumulativa dei guadagni ottenuti su un certo orizzonte temporale. Oltre all'approccio MDP, vengono citati altri algoritmi per ottenere la strategia ottima, come il *Q-learning*, che calcola la strategia ottima "imparando" una funzione di tipo azione-valore che fornisce il guadagno atteso relativo ad intraprendere un'azione in un certo stato seguendo una strategia fissata da lì in avanti: esso ha come punto di forza che risulta un algoritmo molto "libero", dato che non richiede un modello d'ambiente; oppure una strategia greedy, che corrisponde alla massimizzazione del guadagno istantaneo senza prendere in considerazione quelli futuri: tale approccio risolutivo consiste quindi semplicemente nel vendere tutta l'energia rimasta nella batteria in qualunque stato il sistema si trovi. Le simulazioni riportate nel paper sono state ottenute usando i dati *Location Marginal Pricing* presi da [11]: i risultati mostrano come l'approccio RL possa migliorare significativamente il guadagno complessivo, derivante dalla vendita di energia, rispetto all'approccio greedy. Viene anche studiato come differenti fattori abbiano un ruolo cruciale nell'influenzare la scelta della decisione ottima, come il fattore di sconto, la capacità di immagazzinamento e la dimensione della finestra temporale. Tali risultati suggeriscono che, con un alto valore del fattore di sconto, aumentano i guadagni complessivi ottenibili con l'approccio RL. L'influenza della capacità di immagazzinamento è del tutto analoga: con una maggior capacità di immagazzinamento, infatti, l'algoritmo RL ha più flessibilità nel prendere le decisioni e ciò può portare a guadagni complessivi più elevati. Con differenze nelle sequenze dati di *training* e di *testing*, invece, i ricavi ottenibili decrescono all'aumentare della dimensione delle finestre temporali.

Per quanto riguarda l'articolo [1], invece, viene affrontato il problema del controllo ottimo dell'immagazzinamento di energia dal punto di vista di un operatore che alimenti una grid. Il controllore ha accesso ad *un* dispositivo di *energy storage* di capacità finita. Le ipotesi e gli obiettivi posti nella risoluzione del problema sono i seguenti:

- il costo operativo della grid viene modellato come una funzione convessa della domanda di potenza istantanea che viene soddisfatta dalla grid, per catturare il guadagno via via marginale da parte dell'operatore al crescere delle richieste di energia;

- viene studiato un problema di controllo *online dynamic* dell'immagazzinamento dell'energia definendo un modello stocastico per le richieste di potenza continuamente generate ed il loro completamento, e si considera la minimizzazione del costo medio sul lungo termine; viene derivata una strategia di controllo a soglia che tenta di mantenere bilanciato il consumo di potenza all'interno della griglia tramite una gestione adattativa dei processi di carica e scarica della batteria, soddisfacendo parte delle richieste che arrivano al sistema tramite la batteria stessa;
- modello, approccio e struttura della strategia ottima vengono adattati al caso con presenza di *una* sorgente di energia rinnovabile che alimenta la batteria.

Invece di procedere risolvendo il MDP che si potrebbe facilmente formulare per il problema di minimizzazione del costo, viene derivata una semplice strategia dimostrando che essa risulta asintoticamente ottima per valori elevati della capacità della batteria; viene poi anche verificato numericamente che la strategia si comporta abbastanza bene anche per valori finiti di capacità. Viene considerata la seguente strategia di controllo dinamico. Esiste una certa soglia, P_0 . Ogni volta che arriva una nuova richiesta al sistema, il controllore esamina il valore $P(t)$ della richiesta di potenza. Se $P(t) \leq P_0$, allora tutte le richieste attive sono servite dalla grid e viene presa la decisione di *caricare* la batteria, con un rate di carica pari a $\hat{h}(t) = P_0 - P(t)$. Se $P(t) = P_0$, allora la batteria non viene caricata. Se la decisione di carica viene presa all'istante t_1 , e la carica ha luogo nell'intervallo compreso tra t_1 e t_2 (dove t_2 è determinato dall'occorrenza di un nuovo evento come un arrivo, il completamento del servizio o dal fatto che la batteria sia completamente carica), il livello energetico della batteria sarà $E(t_2) = E(t_1) + \hat{h}(t_1) \cdot (t_2 - t_1)$. D'altra parte, se $P(t) > P_0$, viene presa allora la decisione di *scaricare* la batteria, con rate $\hat{h}(t) = P(t) - P_0$, fino al prossimo evento. Di nuovo, se la scarica ha luogo tra gli istanti t_1 e t_2 e la decisione iniziale di scarica viene presa a t_1 , allora $E(t_2) = E(t_1) - \hat{h}(t_1) \cdot (t_2 - t_1)$. Ogni volta che una richiesta servita dalla grid è completata, o il livello energetico della batteria raggiunge lo zero, il controllore osserva di nuovo il valore di $P(t)$ e viene applicata la stessa strategia di controllo sulla carica o scarica della batteria. Nel caso in cui $E(t) = 0$ durante la procedura di scarica, il controllore re-inoltra alla grid le richieste che stavano venendo servite dalla batteria al tempo t . La strategia appena descritta può essere riassunta nel modo seguente:

- se $P(t) \leq P_0$, si carica la batteria con rate $P_0 - P(t)$;
- se $P(t) > P_0$, si scarica la batteria con rate $P(t) - P_0$.

Dato che $P_0 - P(t)$ può assumere sia valori positivi che negativi, la strategia appena descritta (che verrà d'ora in poi indicata come Single-Threshold (ST)) può essere espressa nel modo più riassuntivo possibile come

$$\delta(t) = \max\{-E(t), P_0 - P(t)\}. \quad (1)$$

e nel paper [1] viene dimostrato che la soglia ottima è data da $P_0 = P_{av} = \mathbb{E}[P(t)] = \frac{\lambda}{s}$, dove λ è il rate medio di arrivo delle richieste (modellizzato come un processo di Poisson) e s è il rate medio di servizio delle richieste di potenza, che viene modellizzato tramite lo stesso tipo di processo aleatorio. Il semplice algoritmo di controllo dell'*energy storage* ST, di cui verranno discusse in modo approfondito le prestazioni in termini di costo medio associato al variare di vari parametri del sistema, si dimostra essere *asintoticamente ottimo* quando la capacità della batteria tende ad infinito. Tuttavia, esso non prende in considerazione nè rate di carica/scarica della batteria diversi da infinito, nè un processo di generazione della richiesta di potenza che sia anche vagamente realistico, dal momento che esso viene modellizzato come lo stato di un sistema di coda di tipo $M/M/\infty$.

In [2], invece, si affronta il problema di organizzare gli acquisti di energia sfruttando la possibilità di *energy storage* per minimizzare il costo medio sul lungo periodo in condizioni di variabilità sia della domanda che del prezzo: tale problema prende in considerazione la scelta di soddisfare la richiesta di potenza direttamente dalla grid o dalla batteria, e a quale livello caricare o scaricare quest'ultima (si fa cioè riferimento ad uno scenario del tipo quello illustrato in Figura 2).

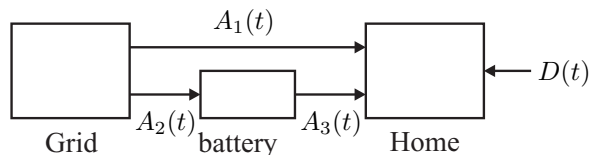


Figura 2: Rappresentazione del modello di sistema descritto in [2].

Il problema di ottimizzazione che ne risulta è complicato dalla natura stocastica del prezzo e della domanda e dal fatto che si cerca di minimizzare il *costo sul lungo termine*. Il problema viene quindi modellizzato come un MDP e *viene dimostrato che esiste una strategia stazionaria a due soglie che minimizza il costo*: quando la batteria è carica ad un livello inferiore alla soglia più bassa, viene caricata fino a tale valore; in caso contrario viene scaricata fino al livello energetico corrispondente all'altra soglia.

Confrontando i costi ottenuti applicando tale *policy* con quelli che si hanno soddisfacendo l'intera richiesta di potenza direttamente dalla grid, viene mostrato che *l'energy storage può portare ad un significativo risparmio sul costo*. Come altre considerazioni interessanti sulla base delle simulazioni numeriche condotte, verso la fine del paper si osserva che il risparmio (percentuale) sul costo rispetto al caso senza batteria assume il suo valore massimo da 16 kWh in poi: aumentare ulteriormente la capacità della batteria oltre tale valore non determina un'ulteriore diminuzione del costo e sembra che la strategia ottima non riesca ad utilizzare capacità di immagazzinamento di energia superiori da parte della batteria. Tale punto di saturazione viene giustificato tramite il fatto che il valore dell'energia immagazzinata decresce nel tempo a causa del *fattore di sconto*; inoltre, la dimensione di un tipico *hybrid vehicle battery pack* è proprio dell'ordine dei 16 kWh, e questo suggerisce che le batterie delle macchine elettriche siano del giusto ordine di grandezza per essere utili ad un *energy storage* domestico, dato che la loro dimensione corrisponde alla quantità di energia che è necessario poter immagazzinare per poter applicare la strategia ottima. Anche conclusioni/sviluppi futuri tratti dagli autori di [2] sono molto interessanti, che si è parzialmente cercato di tenere in considerazione nel modello di sistema descritto nei capitoli successivi: viene accennato al fatto che un possibile passo successivo al loro lavoro potrebbe essere quello di incorporare nel modello il tempo di vita della batteria (come verrà effettuato nel Capitolo 4 di questa tesi) e che i costi di acquisto e sostituzione della batteria vengono influenzati dalla strategia adottata. Prendere in considerazione i costi per acquistare e sostituire la batteria introduce quindi il problema del suo dimensionamento: batterie piccole sono più economiche, ma potrebbero dare minori possibilità di sfruttare le fluttuazioni del prezzo. La dimensione ottimale della batteria dipende molto verosimilmente dallo *spread* e dalla volatilità dei prezzi dell'energia, e potrebbe diversificarsi sulla base del mercato energetico preso in considerazione, come illustrato in [12]. Nella conclusione di [2] viene affermato che risulta essenziale prendere in considerazione i costi di sostituzione della batteria quando si vuole accertare l'*efficacia globale* dell'*energy storage*: sebbene si sia dimostrato che l'immagazzinaggio di energia possa portare a significativi risparmi sul costo da pagare da parte degli utenti all'interno di una micro-grid, questi corrono il rischio di essere limitati da un sistema di *energy storage* troppo costoso. Conseguentemente, sebbene le batterie abbiano visto significativi riduzioni di prezzo ed aumento dell'efficienza negli ultimi anni, gli autori di [2] sostengono che non sia ancora chiarissimo se l'*energy storage* sia o meno una strada conveniente da intraprendere dal punto di vista economico. Va osservato che i risultati di

[2], tuttavia, sono stati ottenuti sotto la (pesante) assunzione che la funzione di costo dell'energia sia sempre *lineare* rispetto alla richiesta di potenza che perviene alla grid: si tratta cioè di un modello che non riesce a catturare il più-che-lineare aumento del costo per produrre e distribuire energia quando la richiesta di potenza cresce.

Struttura della tesi

In questo lavoro di tesi, si è cercato di rendere più chiaro l'effetto che i differenti parametri del modello di sistema possono avere sulle performance delle strategie di *energy storage*. Come primo passo in tale direzione, è stato essenziale considerare sequenze di *richiesta di potenza* che fossero il più possibile realistiche. Per raggiungere tale scopo risulta davvero ottimo il tool messo a disposizione dagli autori di [13] (che viene citato in [2]), che permette di generare un segnale sintetico di richiesta di potenza con una granularità del minuto, simulando i tipici pattern di richieste di potenza delle più comuni applicazioni domestiche. Tale modello di "generazione sintetica" è stato validato tramite un confronto esauriente delle caratteristiche statistiche dei dati sintetici ed empirici (misurati da *smart meters*) dislocati in 22 diverse abitazioni negli UK. Volendo poi che ² la *richiesta netta di potenza* da parte degli utenti all'interno della grid potesse assumere anche valori minori di zero (corrispondenti ad una maggior quantità di energia prodotta tramite fonti rinnovabili rispetto a quella richiesta in un certo slot temporale), si è deciso di generare sinteticamente anche dei campioni di produzione di potenza tramite pannelli fotovoltaici, tramite un modello pseudo-aleatorio (la sequenza prodotta viene infatti moltiplicata per una finestra ripetuta ciclicamente, per rappresentare il fatto che la generazione di energia fotovoltaica *deve* essere nulla nel corso della notte) che non ha pretese di essere particolarmente realistico; la statistica della richiesta netta di potenza era già abbastanza complicata da modellizzare considerando il contributo del tool precedentemente citato: si è ritenuto che introdurre nel sistema un tool analogo ad esso (sicuramente reperibile in letteratura) rendesse davvero ardua l'interpretazione dei risultati.

Per quanto riguarda, invece, il *modello di batteria*, si è deciso di adottarne uno astratto e molto semplice, che prendesse però in considerazione le limitazioni fisiche più importanti di questo tipo di dispositivo, come la capacità di immagazzinare una quantità finita di energia, i rate di carica e scarica limitati e le perdite di energia durante le operazioni di carica/scarica, che corrispondono ad un'efficienza diversa da 1. Altri aspetti, come la perdita di carica nel tempo da parte della batteria e l'ammortamento della sua spesa iniziale, vengono presi in considerazione nella seconda parte della tesi, anche se, dopo la verifica tramite simulazioni, si è consapevoli del fatto che il loro effetto possa essere *abbastanza trascurabile* rispetto al modo in cui viene modellizzato il sistema: la ragione per la quale si è deciso valesse la pena farne vedere le conseguenze che hanno in termini costo medio complessivo è perchè si ritiene sia *abbastanza innovativo rispetto allo stato dell'arte*. Infatti, anche nell'ottima descrizione delle tecniche di modellizzazione delle batterie presente in [14], risulta troppo marcata la differenza tra modelli *stocastici* (come quello di Chiasserini e Rao, descrivibile tramite la catena di Markov a stati finiti, rappresentata in Figura 3) e modelli *analitici* (come quello denominato *Kinetic* o il *Rakhmatov*

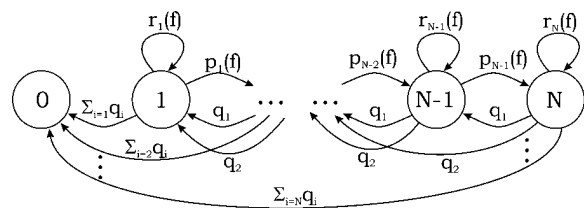


Figura 3: Modello di batteria tramite una catena di Markov di Chiasserini e Rao.

²Anche se è stato effettivamente implementato solo nella seconda parte della tesi.

and Vdrudhula's diffusion model rappresentato in Figura 4). Nel primo caso è stato provato per via simulativa in [15] che i risultati del modello stocastico hanno una massima deviazione dal modello elettro-chimico del 4 % ed una deviazione media dell'1 %: la descrizione della batteria del modello stocastico è cioè davvero buona, però, solo dal punto di vista qualitativo, dato che *non* si riesce ad ottenere neanche quale sia l'ordine di grandezza del tempo di vita della batteria corrispondente. Il secondo caso, invece, che introduce equazioni differenziali nelle densità di corrente e nella capacità della batteria, è sicuramente molto preciso nel caso si voglia un riscontro il più realistico possibile con l'effettivo comportamento della batteria, ma *non risulta trattabile nè dal punto di vista della modellizzazione matematica nè da quello delle simulazioni per via numerica* nel modello del sistema all'interno del quale si andrà a definire il problema di minimizzazione della spesa energetica con l'ausilio della batteria. Si ritiene che la scelta fatta per la descrizione dell'*energy storage*, quindi, si tratti di un buon compromesso tra l'approccio stocastico e quello analitico avente un elevato grado di precisione.

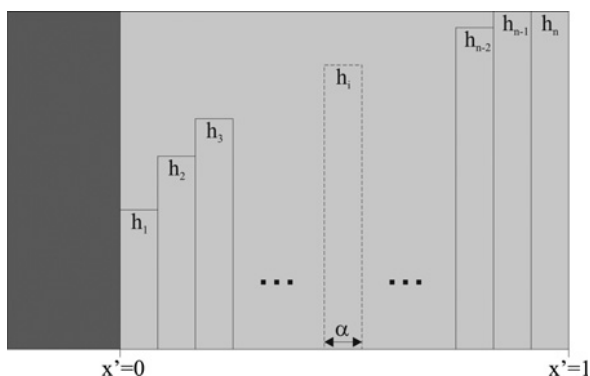


Figura 4: Discretizzazione del modello a diffusione di Rakhmatov e Vrudhula.

costo medio in un sistema senza possibilità di *energy storage* avente in ingresso una distribuzione statistica delle richieste di potenza che coincida con quella che verrà successivamente descritta. Nell'ultima parte della tesi viene infine descritta e analizzata una funzione di costo più simile all'attuale realtà energetica italiana: si tratta di una che emula la tariffa bi-oraria dell'ENEL, con dei prezzi che si differenziano sia per la fascia oraria (diurna o notturna) in cui avviene lo scambio di energia, sia in base al fatto che si tratti di un acquisto o di una vendita. Come sarà chiaro leggendo la tesi, inoltre, è proprio in tale funzione di costo, e non nel modello di batteria, che viene aggiunto un termine relativo all'ammortamento del dispositivo di *energy storage*.

Considerando poi gli *algoritmi* che vengono descritti ed implementati per la risoluzione del problema della minimizzazione del costo energetico da pagare da parte degli utenti all'interno della grid, si è deciso di considerare i quattro denominati rispettivamente come A-posteriori Optimal Strategy (AOS), Dynamic Programming with Markovian requests (DPM), Dynamic Programming with Independent request (DPI) e Single-Threshold (ST). Tali strategie sono state selezionate per il fatto che richiedono progressivamente *meno informazione a priori* sulla statistica delle richieste di potenza degli utenti, ma, d'altro canto, offrono anche una diminuzione in buona approssimazione monotona del risparmio sul costo rispetto al sistema di riferimento *senza batteria*. Confrontando le prestazioni dei quattro algoritmi variando i vincoli del modello di batteria e la funzione di costo, si possono effettuare delle interessanti considerazioni sulla strategia di carica/scarica della batteria in setting molto diversi tra loro.

Il *Capitolo 1*, quindi, descrive il (primo) modello di sistema in tutti gli elementi di cui esso è composto, definisce il problema di ottimizzazione ed espone la strategia di risoluzione AOS, che non è altro che l'algoritmo di Viterbi, ben noto nel mondo delle Telecomunicazioni, che risulta la soluzione ottima del caso offline, ovvero presupponendo di conoscere *a priori* la richiesta di potenza che giungerà alla grid. Viene poi implementato anche un confronto tra la risoluzione del problema di *energy storage* nel caso ad una e a due batterie rispettivamente, argomento non ulteriormente approfondito perchè nell'ultimo caso il problema della complessità computazionale nel calcolo della strategia ottima va al di là degli scopi che ci si è prefissati.

Il *Capitolo 2*, prendendo come testo di riferimento il Sutton-Barto [16], introduce la tecnica di risolutiva della Programmazione Dinamica a partire dal contesto del RL, che è essenziale per arrivare alla strategia ottima *dal punto di vista stocastico* del problema di minimizzazione online, supponendo di poter stimare (periodicamente) la statistica (del secondo ordine) della richiesta netta di potenza che giunge al controllore della grid. In questo capitolo vengono definiti in modo molto generale gli elementi e le equazioni di base della strategia della Programmazione Dinamica, che è uno strumento matematico molto potente, che permette di risolvere problemi con una natura aleatoria molto più marcata di quello affrontato in questa tesi.

Il *Capitolo 3* applica la tecnica della Dynamic Programming (DP) (esplicitando la notazione ad essa associata), definita nel Capitolo 2, al problema di minimizzazione del costo medio da parte degli utenti della grid definito nel Capitolo 1. Vengono riportati, poi, i risultati di alcune simulazioni numeriche eseguite per confrontare le performance degli algoritmi AOS, DPM, DPI e ST con in ingresso due diverse sequenze di richiesta di potenza $\{P_k\}$. Vengono infine eseguiti dei confronti con la strategia a doppia soglia proposta in [2], i cui autori hanno dimostrato essere *ottima* nel caso in cui la funzione di costo risulti *perfettamente lineare* (ipotesi *non* verificata con quella lineare a tratti e con coda quadratica qui considerata) rispetto della richiesta di energia alla grid ed un altro confronto tra il costo corrispondente a tre diverse funzioni di costo lineare a tratti.

Il *Capitolo 4* definisce un nuovo modello di sistema, leggermente diverso da quello definito nel Capitolo 1 per quanto riguarda il modello di batteria (più accurato), la sequenza della richiesta *netta* di potenza (che ora comprende anche un termine di generazione rinnovabile di energia tramite il fotovoltaico, che si sottrae alla domanda di potenza degli utenti della grid) e la funzione di costo (che emula la tariffa bi-oraria attualmente offerta dall'ENEL e dà la possibilità agli utenti della grid di *vendere l'energia in eccesso*, ad un prezzo diverso da quello di acquisto). Viene effettuato uno step simulativo "intermedio" con una tariffa mono-oraria che però permetta di vendere energia e viene poi riportato il confronto sulla base di simulazioni numeriche tra i quattro algoritmi risolutivi più volte citati in presenza di una tariffa bi-oraria di prezzatura dell'energia.

Il *Capitolo 5*, che si ritiene essere il *contributo più importante allo stato dell'arte di questo lavoro di tesi*, deriva *analiticamente* la strategia di ottimizzazione dell'*energy storage* sotto diverse ipotesi sulla funzione di costo dell'energia ed anche sul modello di batteria, dimostrando che finchè si ha a che fare con un problema di ottimizzazione *convesso* con una funzione di costo lineare a tratti la strategia ottima dal punto di vista stocastico può essere definita in *forma chiusa* e con un numero sorprendentemente limitato di parametri. Alla luce di tali dimostrazioni, nella sezione conclusiva vengono riportate una serie di osservazioni sui risultati ottenuti in tutti i capitoli precedenti e qualche spunto per futuri lavori di approfondimento.

Capitolo 1

Primo modello di sistema

1.1 Modello delle batterie

L'integrazione delle sorgenti di energia rinnovabile nella grid elettrica presenta delle sfide dovute all'intermittenza nella generazione di potenza. Queste difficoltà possono essere alleviate utilizzando sistemi intermedi di immagazzinamento dell'energia per migliorare la loro affidabilità nella generazione di potenza e bilanciare l'integrazione di fonti di energia rinnovabile, quali il sole e il vento, in un disegno energetico di larga scala. Una tecnologia avanzata relativa all'immagazzinamento dell'energia gioca un ruolo importante nel smorzare il divario tra la richiesta e l'effettiva erogazione di potenza. Negli ultimi anni, sono stati studiati diversi sistemi di immagazzinamento dell'energia. Questi includono batterie elettrochimiche, super-condensatori, batterie ad aria compressa e batterie a volano. Tuttavia, le batterie al litio (o, come discusso successivamente, agli ioni di litio) rimangono il tipo di batteria più comune per una finalità applicativa che sia l'impiego nelle smart grid, avendo una bassa perdita di carica quando non sono utilizzate, un effetto di memoria assente ed uno dei migliori rapporti energia su peso. Prendendo quindi in considerazione una di esse, le relative equazioni di carica e scarica sono di seguito riportate. F_t^d è la potenza scaricata dalla batteria durante l'intervallo temporale t . F_t^c è la potenza fornita dalla grid alla batteria, ovvero quando la batteria viene caricata nello slot temporale t . $E(t)$ è l'energia immagazzinata nella batteria al tempo t e Δt è la durata temporale di ciascun intervallo. Siano poi η_d e η_c rispettivamente le efficienze di carica e scarica (che tengono conto delle perdite per dissipazione di energia nel corso di tali processi). La batteria deve, inoltre, soddisfare i vincoli (1.2) e (1.3).

- **Vincoli sull'efficienza energetica**

$$\begin{cases} \text{Scarica : } E(t+1) = E(t) - \Delta t F_t^d / \eta_d \\ \text{Carica : } E(t+1) = E(t) + \Delta t F_t^c \eta_c. \end{cases} \quad (1.1)$$

- **Vincoli sul rate di carica/scarica**

$$\begin{cases} 0 \leq F_t^d \leq F_{\max}^d \\ 0 \leq F_t^c \leq F_{\max}^c. \end{cases} \quad (1.2)$$

- **Vincoli sulla massima energia immagazzinabile**

$$0 \leq E(t) \leq E_{\max} \quad (1.3)$$

dove F_{\max}^d è il rate massimo di scarica, F_{\max}^c è il rate massimo di carica e 0 e E_{\max} sono rispettivamente i livelli di energia minimo e massimo immagazzinabili nella batteria.

1.2 Modello di costo

Il modello di costo relativo alla prezzatura dell'energia da parte dell'operatore che fornisce il servizio elettrico assunto nella definizione del problema di ottimizzazione da risolvere è quello di una curva lineare a tratti, con coda quadratica e monotonicamente crescente, che dipenda soltanto dalla domanda di potenza in un certo slot temporale (si tratta quindi di una funzione *convessa*). Esso è motivato dal fatto che nelle ore di picco l'operatore ha meno margine di guadagno e cerca, tramite la tariffazione, di mantenere il consumo energetico *sempre contenuto entro una certa soglia*. Analiticamente ciò si può esprimere tramite l'equazione

$$C(X(t)) = \begin{cases} X(t), & X(t) \leq 5 \text{ kW} \\ 3X(t) - 10e3, & 5 \text{ kW} < X(t) \leq 10 \text{ kW} \\ 10X(t) - 80e3, & 10 \text{ kW} < X(t) \leq 25 \text{ kW} \\ (X(t) - 12e3)^2, & X(t) > 25 \text{ kW}. \end{cases} \quad (1.4)$$

il cui andamento è rappresentato in Figura 1.1.

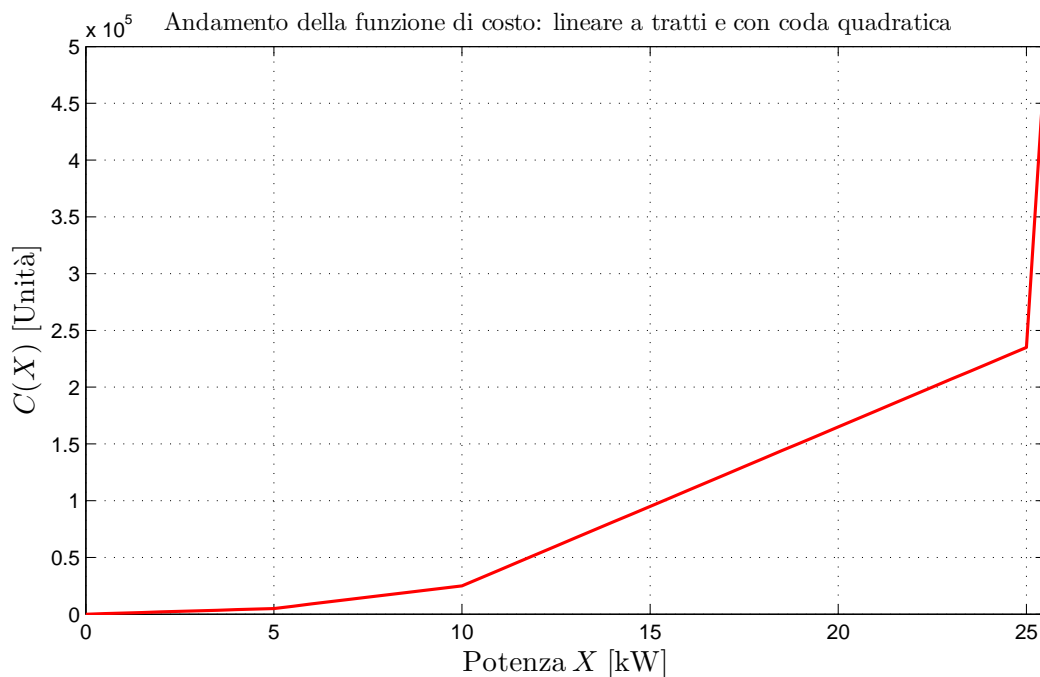


Figura 1.1: Andamento della funzione di costo $C(X)$ in funzione della potenza (compreso il contributo della batteria) richiesta alla grid, con 3 pendenze diverse per la parte lineare a tratti e con coda quadratica.

Già questo modello, pur nella sua elementarietà, prende chiaramente le distanze dalla tariffa energetica bi-oraria introdotta nel 2010 dall'ENEL, la quale prevede una differenziazione del prezzo dell'energia solo in base alla fascia oraria a cui appartiene la richiesta di potenza ed, eventualmente, al fatto di trovarsi in un giorno ferialo o festivo. È chiara, tuttavia, la differenza tra l'ottica centralizzata dell'ENEL nella distribuzione e tariffazione dell'energia e quella distribuita delle smart grid, dove l'ordine di grandezza dell'area in cui risulta interessante ottimizzare il consumo energetico e la spesa da parte dei clienti che le appartengono è dell'ordine delle decine di Km².

Con una funzione di costo di questo tipo, diventa *fondamentale* il ruolo della batteria come dispositivo di immagazzinamento di energia, che può essere inserita o prelevata da esso in modo che la richiesta di potenza complessiva all'operatore energetico sia sempre la più bassa possibile, dato che per valori di X inferiori a 5 kW il costo cresce con la pendenza più ridotta.

1.3 Modello di stato

Il punto di vista che la seguente trattazione assume per quanto riguarda lo stato del sistema è di ipotizzare inizialmente un modello che risulti ragionevolmente generale, riducendo la dimensione dello spazio degli stati assumibili dal sistema e dei vincoli ad esso associati soltanto in mancanza della conoscenza in letteratura di una tecnica risolutiva efficiente per risolvere il problema di ottimizzazione che ne deriva. Si consideri quindi una generica smart grid con la presenza di un controllore centralizzato delle richieste di potenza dei *prosumers* (termine che si riferisce al fatto che il generico utente è sia un consumatore che un produttore di energia tramite sorgenti rinnovabili), le quali vengono accorpate nel segnale scalare $L(t)$ funzione del tempo, che viene campionato ad intervalli di durata Δt . Questo stesso controllore centralizzato si prende anche in carico di accorpare nel segnale $R(t)$ i contributi relativi alla generazione di potenza distribuita nella grid grazie alle sorgenti di energia rinnovabile (ad esempio, pannelli fotovoltaici e generatori eolici). Per formalizzare quanto appena scritto, sia $L(t)$ che $R(t)$ sono dei processi aleatori monodimensionali approssimabili come ciclo-stazionari (supponendo che il comportamento dei *prosumers* in termini di consumo e produzione di potenza abbia una ciclicità di 24 h, trascurando eventuali dipendenze dei processi dal trovarsi temporalmente in un giorno festivo o feriale e dalle condizioni meteorologiche) a tempo discreto che possono assumere solo valori non-negativi (per definizione di potenza elettrica) che verranno opportunamente quantizzati in sede di simulazione e di analisi dei dati. Per non complicare troppo la trattazione, *almeno in questa prima parte della tesi*, viene assunto che la potenza complessiva $R(t)$ generata dalle sorgenti rinnovabili sia *molto inferiore* a quella $L(t)$ delle richieste di potenza: in questo modo, *la richiesta netta* $P(t) = L(t) - R(t)$ di energia che deve essere presa in carico dall'operatore energetico (a meno del contributo della batteria) assume sempre valori *non-negativi*. Non è stata infatti nemmeno definita la funzione di costo per valori negativi di $P(t)$: ciò corrisponde ad assumere che l'energia possa solo essere acquistata e che le sorgenti rinnovabili producano una traslazione verso il basso dei valori della richiesta complessiva di potenza. *In questa prima parte della tesi*, senza che sia mai presente nella grid un eccesso di produzione energetica locale, il vantaggio derivante dalla disponibilità della batteria non è quello di poterla caricare in tali situazioni al posto di vendere l'energia all'operatore energetico (ad un prezzo che, nella pratica, tralasciando gli eventuali incentivi statali, è *sempre* inferiore a quello di acquisto), ma di comprarla da esso in momenti giudicati *convenienti* sulla base dell'andamento della funzione di costo. Se ad un certo istante la richiesta $P(t)$ è molto bassa, ma si prevede che essa manifesterà un drastico aumento nelle ore successive, caricare il più possibile la batteria può risultare la scelta migliore per non finire poi nel range di ascisse a massima pendenza della funzione di costo e pagare quindi, complessivamente, il minimo possibile. Il modello di sistema appena discusso è rappresentato in Figura 1.2.

Si supponga poi che nella grid siano presenti M batterie, definite dal modello della Sezione 1.1, che si generalizza nel modo seguente.

- **Equazioni di aggiornamento del livello energetico**

$$\begin{cases} \text{Scarica : } E_i(t+1) = E_i(t) - \Delta t F_{i,t}^d / \eta_{i,d}, & i = 1, 2, \dots, M \\ \text{Carica : } E_i(t+1) = E_i(t) + \Delta t F_{i,t}^c \eta_{i,c}, & i = 1, 2, \dots, M. \end{cases} \quad (1.5)$$

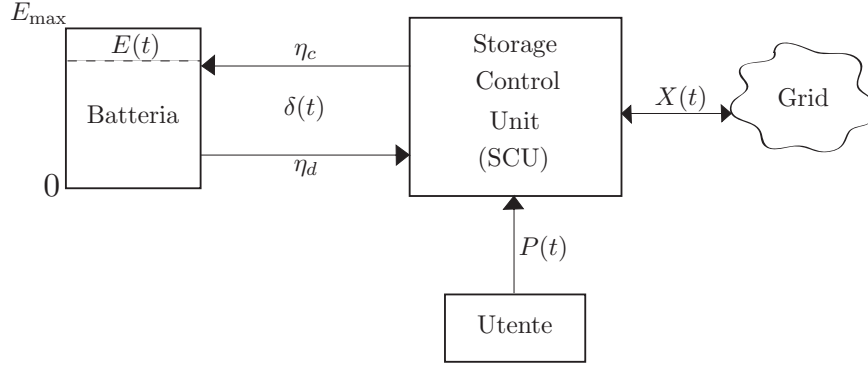


Figura 1.2: Architettura del modello di sistema.

- **Vincoli sul rate di carica/scarica**

$$\begin{cases} 0 \leq F_{i,t}^d \leq F_{i,\max}^d, & i = 1, 2, \dots, M \\ 0 \leq F_{i,t}^c \leq F_{i,\max}^c, & i = 1, 2, \dots, M. \end{cases} \quad (1.6)$$

- **Vincoli sull'immagazzinamento di energia**

$$0 \leq E_i(t) \leq E_{i,\max} \quad i = 1, 2, \dots, M. \quad (1.7)$$

Si osservi che (1.5), (1.6) e (1.7), componendo opportunamente le disuguaglianze che le costituiscono, si possono riassumere rispettivamente come

$$-\frac{\Delta t F_{i,\max}^d}{\eta_{i,d}} \leq E_i(t+1) - E_i(t) \leq F_{i,\max}^c \Delta t \eta_{i,c} \quad i = 1, 2, \dots, M. \quad (1.8)$$

e

$$0 \leq E_i(t) \leq E_{i,\max} \quad i = 1, 2, \dots, M. \quad (1.9)$$

Sia quindi $\mathbf{E}(t)$ il vettore ad M componenti che tiene conto dei livelli di energia immagazzinati nelle diverse batterie del sistema durante lo slot temporale t : esso contiene tutte le informazioni che servono a memorizzare lo stato del sistema così definito. In quest'ottica, $P(t) = L(t) - R(t)$ costituisce l'input del sistema, ed il suo andamento è assunto come *conosciuto* dal controllore della grid in ogni intervallo temporale (ed, in particolare, i loro due valori pervengono al controllore *all'inizio* di ogni intervallo di durata Δt ; poi, il controllore, a "sua" discrezione, può memorizzare un certo numero di valori passati, che formano la memoria dello stato del sistema). Nel modello con M batterie, la richiesta di potenza in ingresso alla grid al tempo t (sulla quale è effettivamente applicata la tariffazione del fornitore del servizio energetico) è data da

$$X(t) \triangleq P(t) + \delta(t) = L(t) - R(t) + \mathbf{1}^T (\mathbf{E}(t+1) - \mathbf{E}(t)) / \Delta t \quad (1.10)$$

dove $\mathbf{1}$ rappresenta un vettore colonna con tutti gli M elementi uguali ad 1.

1.4 Definizione del problema di ottimizzazione

Premesso quanto riportato nelle Sezioni 1.1, 1.2 e 1.3, i problemi di ottimizzazione che si possono definire sono molteplici. Viene di seguito assunto di voler minimizzare la somma dei costi relativi

al consumo energetico pagati al fornitore del servizio elettrico da parte degli utenti della grid sfruttando la possibilità di prelevare ed immagazzinare energia nelle M batterie in slot temporali giudicati convenienti. Ipotizzando un orizzonte di ottimizzazione della generica durata di T slot temporali, si vuole determinare quali sono i livelli energetici ottimi $\{\mathbf{E}_{opt}(t)\}_{t=1,2,\dots,T-1}$ da mantenere nelle batterie durante questo arco di tempo, supponendo che esse siano inizialmente cariche a dei livelli di energia noti memorizzati nel vettore \mathbf{E}_0 . Ciò si può formalizzare come

$$\{\mathbf{E}_{opt}(t)\}_{t=1,2,\dots,T-1} = \underset{\{\mathbf{E}(t)\}_{t=1,2,\dots,T-1}}{\arg \min} \mathbb{E} \left[\sum_{t=0}^{T-1} C(X(t)) \right]. \quad (1.11)$$

subject to : (1.8), (1.9), (1.10)

Nella formulazione (1.11) risulta abbastanza evidente che la minimizzazione del costo totale su un certo orizzonte temporale abbia con lo stato del sistema, $\mathbf{E}(t)$, un legame istantaneo anche se quest'ultimo non compare esplicitamente come argomento della funzione costo: se un "genio portatile" (neologismo coniato nella letteratura della teoria della probabilità, ad esempio in [17]) fornisse il minor costo raggiungibile fino allo slot temporale corrente e lo stato attuale del sistema ad esso associato, il processo di ottimizzazione temporalmente successivo non avrebbe bisogno di tener traccia dei valori che lo stato ha assunto precedentemente. Questa *assenza di memoria* rispetto allo stato del sistema è la peculiarità di un processo di Markov e il fatto che si debba effettuare una sequenza finita di decisioni (in ogni slot temporale, si assumono come conosciuto e fissato il valore di $P(t)$ e si possono scegliere i livelli di energia presenti nelle M batterie, compatibilmente con i vincoli citati in (1.11)) porta a far rientrare questo problema di ottimizzazione tra i MDP. A causa della natura vettoriale del segnale che rappresenta la decisione ottima da effettuare, il processo risolutivo risulta *intrinsecamente complesso e computazionalmente intrattabile* per un generico numero M di batterie: tuttavia in fase di modellizzazione si è preferito essere il più generici possibile. Una prima serie di considerazioni che portino alla definizione di una strategia per risolvere (1.11) può essere ottenuta esaminando un caso particolare rispetto a quello a M batterie nella grid: con *una sola* batteria, lo stato $\mathbf{E}(t)$ assume natura scalare, ovvero

$$\mathbf{E}(t) \stackrel{M=1}{=} E(t) \quad (1.12)$$

la quale rimane soggetta ai vincoli (1.1),(1.2) e (1.3).

1.5 Modello per la generazione della richiesta complessiva di potenza $P(t)$

Dato che ai fini del problema di ottimizzazione definito in (1.11) interessano soltanto i valori della richiesta *netta* di potenza, $P(t) = L(t) - R(t)$, che giunge al controllore della grid ad ogni intervallo temporale di durata Δt , ci si può non preoccupare per il momento di generare i due segnali distinti $L(t)$ (che costituisce la richiesta di potenza comprensiva del contributo di tutte le abitazioni presenti all'interno della grid) ed $R(t)$ (che rappresenta la produzione in termini di potenza da parte delle sorgenti di energia rinnovabili della medesima area). Sulla base soltanto di una qualche minima conoscenza di modellizzazione statistica, senza aver bisogno di alcun riscontro da parte di simulazioni numeriche, ci si rende subito conto dell'importanza di un modello quanto più realistico possibile per la generazione della sequenza $P(t)$: in particolare la sua correlazione può influenzare moltissimo sia la forma della soluzione ottima, che il costo complessivo che si cerca di minimizzare. In letteratura, per testare gli algoritmi proposti, vengono spesso usate modalità di generazione del segnale $P(t)$ completamente *inverosimili*: assumere ad

esempio i valori di $\{P(t)\}_{t=0,1,2,\dots}$ independent identically distributed (iid) e la loro distribuzione stazionaria uniforme all'interno di un certo range di potenze, sono assunzioni molto azzardate senza che ne venga garantito alcun riscontro con dati misurati da qualche *power meter*; permane inoltre la curiosità di come alcuni algoritmi euristici si comportino ricevendo in ingresso una sequenza $P(t)$ con una statistica *molto diversa da quella su cui sono stati testati*, perchè è lecito si insinuì il dubbio che le prestazioni in termini del costo complessivo che si cerca di minimizzare sarebbero molto diverse, nella direzione di un aumento di tale costo.

Non essendo la generazione di richieste di potenza con una statistica quanto più possibile simile a quella di dati reali lo scopo principale di questa tesi, si è cercato in letteratura un tool che, tarato opportunamente, mettesse a disposizione una sequenza $\{P(t)\}$ che servisse come ingresso del problema di ottimizzazione (1.11); c'era la consapevolezza che non sarebbero stati di interesse modelli eccessivamente accurati, che ne descrivessero anche il legame con la potenza reattiva e/o con le correnti e tensioni corrispondenti: tutti segnali legati alla matrice di impedenza della grid (e quindi alle perdite lungo le linee di distribuzione), che non sono stati presi in considerazione. Si reputa comunque veramente eccellente la routine messo a disposizione come *macro* di Excel dagli autori di [13], di cui riporto una breve descrizione nelle prossime righe.

- **Applicazione:** il modello usa il concetto di applicazione come il proprio *building block* basilare, dove per applicazione si intende qualsiasi carico elettrico *domestico*, come una TV, una lavastoviglie o un aspirapolvere; possiede poi l'importante caratteristica di rappresentare un utilizzo delle applicazioni *correlato nel tempo*. Le applicazioni nel modello sono configurate usando statistiche che descrivano la loro domanda media annuale di energia e le caratteristiche associate relative all'uso della potenza, ad esempio includendo il loro comportamento a regime oppure di tipo ciclico. Il passo successivo nella definizione del modello considera quando è verosimile che le specifiche applicazioni vengano usate: l'utilizzo di un'applicazione all'interno di un'abitazione è sicuramente legato al numero di persone *svegli* che ci sono all'interno. Tale elemento viene catturato dall'articolo come l'*active occupancy* e viene rappresentato per ogni abitazione all'interno del modello come un intero che varia nel corso del giorno in maniera pseudo-casuale. Tale approccio si basa sui dati derivati da *UK 2000 Time Use Survey* (TUS), un'indagine esauriente di come le persone spendono il loro tempo in Gran Bretagna, che raccoglie migliaia di diari giornalieri registrati con una risoluzione di 10 minuti: il modello dell'*active occupancy* richiede in ingresso il numero totale di residenti (da 1 a 5, per ogni abitazione che si intende simulare) e questo valore viene stimato in accordo con i dati statistici del popolo inglese reperibili in [18]. La rappresentazione dell'*active occupancy* nel modello fornisce il meccanismo primario per creare i dati relativi alla richiesta energetica con profili giornalieri aggregati appropriati; costituisce inoltre una base su cui modellizzare l'uso dell'elettricità *correlato nel tempo* sia all'interno della stessa abitazione che tra abitazioni distinte.

Al fine di raffinare la modellizzazione della temporizzazione della domanda di potenza viene introdotto anche un secondo meccanismo, che si basa sulle attività degli abitanti. I dati di TUS vengono usati stavolta per creare *profili di attività*, che mostrano, per fare un esempio ovvio, che la gente tenda ad essere impegnata con attività inerenti la cucina durante le ore dei pasti. Il passo successivo è quello di collegare queste attività alle applicazioni: assegnando un *profilo di attività* ad ogni applicazione nel modello, la mutevole verosimiglianza che un'applicazione venga usata nel corso della giornata può essere presa in considerazione all'interno di una simulazione stocastica. Tali passi assicurano che le applicazioni nel modello siano attivate ad istanti del giorno opportuni, senza la necessità di considerare una statistica delle applicazioni molto dettagliata; instaurare una

relazione tra l'uso dell'energia e l'attività di un residente all'interno di una abitazione, inoltre, è particolarmente importante nello studio del *demand side management*, che presuppone una domanda flessibile. Infine, le applicazioni possono essere usate, come è ovvio, contemporaneamente da più di un abitante. Utilizzare la *active occupancy* come base del modello permette che la condivisione delle applicazioni possa essere agevolmente presa in considerazione: la probabilità modellizzata che una certa applicazione venga utilizzata cresce non linearmente col numero di *active occupants*.

- **Struttura del modello:** la struttura del modello è illustrata in Figura 1.3: sulla sinistra del diagramma, ci sono un insieme di profili di attività giornalieri, che rappresentano la verosimiglianza che le persone eseguano determinate attività al variare dell'ora del giorno, dove questi profili sono gli stessi per tutte le abitazioni; sulla destra del diagramma, le abitazioni sono rappresentate dal blocco quadrato più esterno. Ad ogni abitazione vengono assegnati una serie di dati relativi alla *active occupancy* ed un insieme di applicazioni installate. Ogni applicazione è mappata su uno dei profili di attività giornalieri. Quando avviene l'accensione di un dispositivo, le caratteristiche sul consumo di potenza da parte dell'applicazione vengono usate per determinare la sua domanda energetica. Sommando le richieste di potenza di tutte le applicazioni di una casa fornisce la corrispondente richiesta energetica aggregata: la domanda di potenza complessiva si ottiene sommando i singoli contributi di tutte le abitazioni.

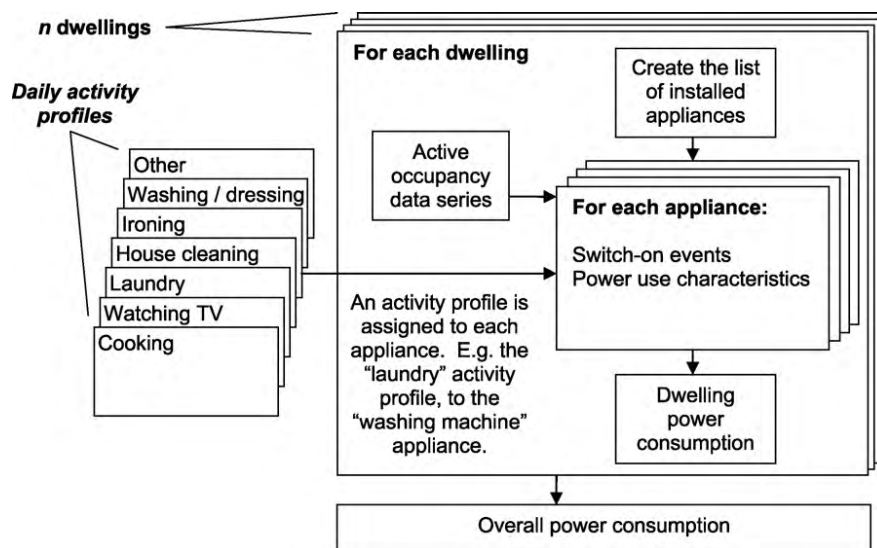


Figura 1.3: Architettura del modello relativo alla domanda energetica.

Ogni profilo di attività giornaliero, sulla sinistra di Figura 1.3, quantifica la probabilità che venga intrapresa una determinata attività in funzione dell'ora del giorno; tale insieme di profili include delle varianti per prendere in considerazione il numero corrente di *active occupants* e se si tratta di un giorno feriale o festivo. Per quanto riguarda le applicazioni della grid di cui viene generata la domanda energetica, all'inizio di ogni simulazione il modello popola ogni abitazione di applicazioni in modo random sulla base di una statistica matchata con vari database inglesi relativi al possesso di una vasta gamma di dispositivi: il modello può includere fino a 33 applicazioni, con la possibilità che ci sia più di un possessore. Ad ogni applicazione vengono inoltre assegnati una domanda energetica annuale espressa in [kWh/anno] e un fattore di potenza, che rappresenta il valore medio su un intervallo

di un minuto: un valore unitario viene impiegato per applicazioni resistive che producono calore, come il bollitore, il forno o il ferro da stiro.

- **Mappatura applicazioni-attività ed eventi di *switch-on*:** le applicazioni il cui uso è dipendente dal fatto che una particolare attività abbia luogo sono assegnate al profilo di attività corrispondente; ci possono essere più applicazioni assegnate ad una singola attività: questo *non* implica che tutte siano usate necessariamente quando essa ha luogo, ma modella soltanto la possibilità che vengano utilizzate, *anche* contemporaneamente. Le applicazioni che non sono associate ad una particolare attività hanno un profilo classificato come *other*; per i dettagli si rimanda a [13].

La procedura per determinare quando, all'inizio di ogni intervallo temporale della simulazione, avviene un evento di accensione da parte di un'applicazione è rappresentato in Figura 1.4. Vengono seguiti i seguenti passi:

- innanzitutto, viene selezionato il profilo di attività in accordo con l'attività a cui appartiene l'applicazione, il numero corrente di *active occupants* e se si tratta di un weekend o meno;
- poi, la probabilità che *ciascuno* degli *active occupants* si dedichi a tale attività nell'intervallo temporale corrente viene letta dal profilo di attività;
- tale probabilità viene quindi moltiplicata per uno *scalare di calibrazione*, la cui stima è discussa di seguito;
- infine, il risultato dello step precedente è paragonato ad un numero random tra 0 e 1: se la probabilità in questione è maggiore di tale quantità aleatoria, allora si verifica un evento di *switch-on*.

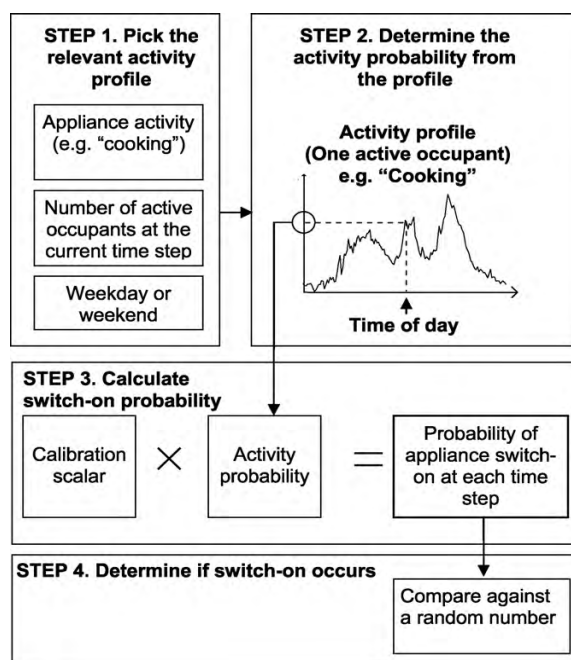


Figura 1.4: Eventi di *switch-on*.

Ogni applicazione ha il proprio scalare di calibrazione, che viene impiegato nel calcolo della probabilità di *switch-on* come mostrato in Figura 1.4, e determina quindi il numero

medio di volte che l'applicazione viene usata in un anno. Tale fattore viene tarato in modo che, su un grande numero di esecuzioni di simulazioni stocastiche, il consumo medio annuo dell'applicazione sia corretto; il calcolo è ovviamente più complesso per le applicazioni che appartengono ad un profilo di attività giornaliero, perchè si devono prendere in considerazione le distribuzioni statistiche sia della occupazione dell'abitazione che dei profili di attività. Innanzitutto va calcolato il valore medio complessivo che un'attività abbia luogo in un certo intervallo temporale. Questo può essere ottenuto usando il teorema (sella probabilità condizionata) di Bayes. Il primo input di tale calcolo è la probabilità di ciascun livello della *active occupancy* ($\in \{0, 1, \dots, 5\}$) ad ogni istante della giornata: questo può essere determinato dal modello di occupazione. Il secondo input è la probabilità condizionata che un'attività abbia luogo, dato ciascun livello di *active occupancy*: questa informazione è disponibile nel profilo di attività. La probabilità media che un'attività abbia luogo, una volta moltiplicata per lo scalare di calibrazione, dovrebbe essere uguale alla probabilità media che l'applicazione preveda un evento di *switch-on*.

- **Validazione del modello:** per tutti i dettagli sulla validazione del modello di generazione della sequenza $\{P(t)\}$ appena descritto si rimanda all'articolo più volte citato. Per brevità di seguito si riportano solo i risultati delle simulazioni relative alla *domanda media giornaliera*. Il profilo medio giornaliero sintetico e misurato sono riportati in Figura 1.5, che mostra anche l'andamento di quello tipico degli UK. In generale, tutti i profili seguono un *pattern* simile. C'è un livello ridotto di richiesta energetica durante la notte, seguito da un picco iniziale ad ora di colazione. Poi la domanda è ragionevolmente piatta durante il resto della giornata, fino alla metà del pomeriggio, in cui comincia a crescere per raggiungere il picco serale; la domanda cala poi nella parte rimanente della serata.

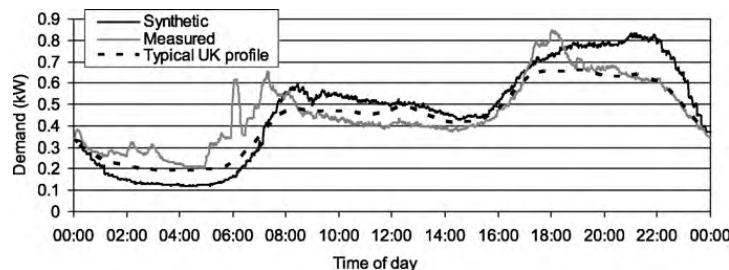


Figura 1.5: Profilo giornaliero medio della domanda energetica su base annuale.

Per un paragone con i dati nazionali, la Figura 1.5, come sopra menzionato, riporta anche il profilo della domanda media (su base annuale) per *tutti* gli utenti domestici degli UK, sulla base dei profili standard sviluppati dalla *Electricity Association*. I dati sintetici corrispondono al profilo domestico medio inglese in modo anche più accurato di quanto non facciano i dati misurati localmente. Tuttavia, dalla curva si nota che il modello predice un consumo energetico eccessivo nel tardo pomeriggio (dalle 18.00 in poi). Questo potrebbe essere causato dal fatto che le persone diventano più sedentarie (meno attive) verso la fine della giornata e meno propense ad un consumo energetico *multi-tasking*. Nonostante il modello permetta che più applicazioni siano eseguite contemporaneamente, i profili di attività sono determinati solo sulla base della attività primaria descritta nei dati TUS, che non rappresentano quindi accuratamente gli utenti *multi-tasking*.

- **Sequenza $\{P(t)\}$ generata sulla base del tool fornito in [13]:** per rendere la generazione della richiesta di potenza “stazionaria nella maggior misura possibile” si è scelto di far girare la *macro* in Excel settando il mese di riferimento sempre a gennaio, simulandone i suoi 31 giorni (con un totale di 23 giorni feriali e 8 giorni festivi) per 6 diverse realizzazioni. Nell’inizializzazione delle simulazioni, invece, sono stati fatti variare le *active occupancy* giornalmente, mentre il numero ed il tipo di applicazioni presenti nelle abitazioni in media 2 volte nell’arco di una singola realizzazione, affinché il pattern di campioni di potenza prodotti fosse il più possibile realistico. Il tool che si è utilizzato fornirebbe in output i dati relativi alla generazione di una realizzazione della richiesta $P(t)$ di un giorno del mese di gennaio con la *risoluzione temporale di un minuto e con riferimento ad un’abitazione*. Ai fini di avere un dato “trattabile” come ingresso al problema di ottimizzazione precedentemente definito, si è scelto di modificare la *macro* in modo che generasse la richiesta di potenza relativa a 20 case, andando poi a mediare i dati con risoluzione di un minuto sull’intervallo $\Delta t = 1$ h (con la consapevolezza di perdere in tal modo le variazioni di potenza ad alta frequenza dei carichi: si tratta di un tradeoff tra il volume di dati su cui effettuare il *processing* e la *demand curve smoothing*). Così facendo sono stati generati $24 \cdot 31 \cdot 6 = 4464$ campioni della sequenza $\{P(t)\}$, avente un minimo di 1.3 kW, un massimo di 61.9 kW ed un valore medio di 13.7 kW, la cui corrispondente Probability Density Function (PDF) è riportata in Figura 1.6.

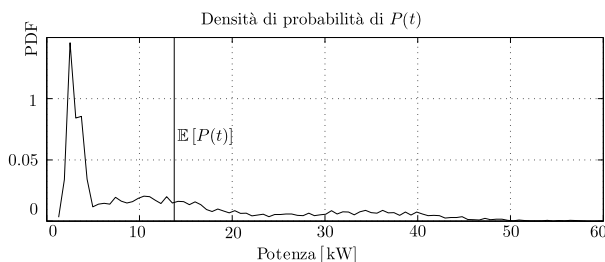


Figura 1.6: PDF della sequenza della richiesta netta di potenza $\{P(t)\}$.

Nella Figura 1.7 si riporta, invece, *l’andamento medio giornaliero* con granularità di un’ora della richiesta di potenza sulle 6 realizzazioni generate per il mese di gennaio. Dalla Figura 1.7 si nota come il segnale potenza mediato presenti un minimo di 2.9 kW ed un massimo di 24.6 kW (parecchio ridotto rispetto a quanto sopra riportato, dal momento che è stata effettuata una media tra realizzazioni puntualmente anche molto diverse tra loro). Si osserva, poi, che l’andamento della richiesta di potenza media (e le stesse considerazioni valgono per le singole realizzazioni) sia, almeno apparentemente, piuttosto correlato nel tempo: come sarà chiarito soltanto nei capitoli seguenti, tuttavia, la sua statistica (generata con un tool che offre una risoluzione temporale originaria di un minuto) vari così velocemente che risulta *ben* approssimabile da un modello iid, che si comporta altrettanto bene di uno di tipo markoviano. Questa considerazione, tuttavia, ha creato non poche difficoltà nell’interpretazione dei risultati successivamente riportati e merita un grado di approfondimento superiore: ciò verrà discusso nei Capitoli 3 e 4.

1.6 Unico vincolo: E_{\max} finita

Vale la pena iniziare la seguente trattazione da un caso estremamente semplificato rispetto a (1.11): quello con *una* sola batteria di capacità *finita* pari a E_{\max} , *senza nessun vincolo sui*

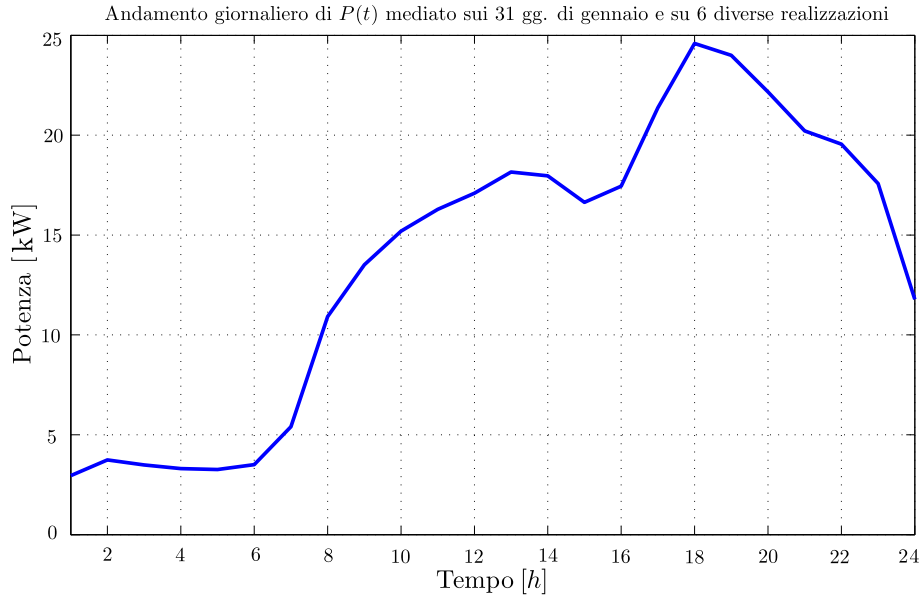


Figura 1.7: Andamento medio giornaliero della richiesta di potenza $P(t)$.

massimi rate di carica e scarica $n\grave{e}$ *sull'efficienza della batteria*. Questa scelta è motivata dal fatto che, così operando, in sede di simulazione si può avere un riscontro immediato con i risultati citati nell'introduzione: in [1], infatti, viene descritta la strategia ottima al tendere all'infinito della capacità della batteria, senza imporre alcun vincolo alle sue equazioni di aggiornamento del livello energetico. Riuscendo a trovare una soluzione *esatta* al problema con una possibilità di immagazzinamento *finita*, le prestazioni offerte dall'algoritmo ST dovrebbero risultare *inferiori* in termini di costo complessivo pagato dagli utenti della grid, soprattutto per valori di capacità "modesti" ($\ll 24$ kWh). Il problema di ottimizzazione risultante si può quindi esprimere come

$$\{E_{\text{opt}}(t)\}_{t=1,2,\dots,T-1} = \underset{\{E(t)\}_{t=1,2,\dots,T-1}}{\text{arg min}} \quad \mathbb{E} \left[\sum_{t=0}^{T-1} C(X(t)) \right] \quad (1.13)$$

subject to : $0 \leq E(t) \leq E_{\max}$, (1.14)

dove

$$X(t) = L(t) - R(t) + (E(t) - E(t-1)) / \Delta t. \quad (1.14)$$

Tale modello di sistema si differenzia da quello descritto in [1] innanzitutto per il fatto che la sorgente di energia rinnovabile *non* alimenta direttamente la batteria, ma concorre soltanto a diminuire la quantità di potenza, $P(t)$, su cui viene applicata la tariffazione. Un'osservazione molto più importante consiste, invece, nel notare che l'aver introdotto il vincolo sulla massima capacità di immagazzinamento della batteria modifichi minimamente il problema di ottimizzazione da risolvere, ma comporti un cambiamento *radicale* della strategia di risoluzione ad esso associata. Se $E_{\max} \rightarrow \infty$, infatti, non si può pi\`u dimostrare essere ottima una semplice strategia a soglia, e questo perchè la disuguaglianza di Jensen è "ben lontana" dal valere con il segno di uguaglianza.

Se l'obiettivo del controllore fosse quello di trovare la soluzione di (1.13) offline, ovvero conoscendo la sequenza delle richieste di potenza nella grid, $\{P(t)\}_{t=0,1,\dots,T-1}$, allora l'operatore di aspettazione risulterebbe "superfluo", nel senso che non ci sarebbe pi\`u niente di aleatorio

nell'espressione da minimizzare, che diventerebbe una semplice sommatoria esprimibile come

$$\{E_{\text{opt}}(t)\}_{t=1,2,\dots,T-1} = \underset{\{E(t)\}_{t=1,2,\dots,T-1}}{\arg \min} \sum_{t=0}^{T-1} C(X(t)). \quad (1.15)$$

subject to : $0 \leq E(t) \leq E_{\max}$, (1.14)

In tal caso, una volta quantizzati opportunamente i valori della sequenza $E(t)$, con passo ΔE , in modo da appartenere ad un insieme finito (definito come $\mathcal{E} = \{E_0 = 0, E_1, E_2, \dots, E_{N-1}\}$), la soluzione ottima non potrebbe che essere fornita da un approccio “alla Viterbi” (che si ricorda essere indicato come strategia AOS), in cui il vettore che memorizza i valori energetici presenti nella batteria al variare del tempo corrisponderebbe alla consueta sequenza di simboli trasmessi $\{a_k\}$, alla quale al *detector* ottimo viene richiesto di risalire in presenza di rumore ed in generale di *Inter-symbol Interference* (ISI) causata da un canale di tipo *multipath*. L'assenza di memoria del problema descritta nella Sezione 1.4 (che ha un'analogia nel mondo delle telecomunicazioni in un canale non *multipath* ma Additive White Gaussian Noise (AWGN)) *non* significa comunque che un approccio di tipo greedy, che determina il livello energetico ottimo $E_{\text{opt}}(t)$ per $t = 1, 2, \dots, T - 1$ solo sulla base del valore della funzione di costo $C(X(t))$ all'istante corrente t , risulti ottimo: se non si osserva l'intera sequenza di potenza effettivamente richiesta alla grid, definita istante per istante come $P(t) = L(t) - R(t)$, si corre il rischio di scaricare la batteria entro pochi intervalli di tempo (commettendo la scelta sbagliata); essendo la funzione di costo, fissata $P(t)$, crescente in $\delta(t) = E(t) - E(t - 1)$, l'algoritmo greedy non giudicherà mai più conveniente ricaricare la batteria, anche in presenza di picchi futuri della richiesta di potenza, che beneficerebbero della capacità di immagazzinamento per cui la stessa batteria è stata introdotta nella grid.

Per ottimizzare la complessità computazionale nel determinare la soluzione di (1.15) senza ricorre ad un algoritmo che enumeri tutte le possibili soluzioni valutandone il costo (che avrebbe complessità esponenziale), risulta quindi *davvero necessario* ricorrere all'algoritmo di Viterbi: c'è bisogno soltanto di un piccolo sforzo per adattare la “terminologia energetica” a quella delle Telecomunicazioni. Cercando di conformarsi il più possibile alla notazione dell'algoritmo di Viterbi descritto in [19], lo *stato* del sistema in questo particolare problema di ottimizzazione può essere definito semplicemente come

$$s_k = (E_k) \quad \text{con } s_{-1} = 0 \text{ per definizione.} \quad (1.16)$$

Indicando con \mathcal{S} l'insieme dei possibili stati, ovvero dei possibili valori assumibili da s_k , si ha che

$$s_k \in \mathcal{S} = \{E_0, E_1, \dots, E_{N-1}\} \quad (1.17)$$

dalla quale segue che il numero degli stati è dato da $N = \left\lfloor \frac{E_{\max}}{\Delta E} \right\rfloor + 1$, con una forte dipendenza dal *passo di quantizzazione* scelto per il livello di energia (in letteratura, considerando batterie aventi una capacità che va circa da 5 kWh a 20 Kkh, una scelta comune è quella di porre $\Delta E = 0.5$ kWh; tale scelta sarà adottata anche in tutto il resto della tesi, anche per la quantizzazione dalla sequenza delle richieste di potenza $P(t) \rightarrow \{P_k\}$ con $\Delta P = 0.5$ kW). Definendo quindi la metrica¹

$$\Gamma_k = \sum_{i=0}^k C(P_i + s_i - s_{i-1}) \quad (1.18)$$

vale la seguente equazione ricorsiva

$$\Gamma_k = \Gamma_{k-1} + C(P_k + s_k - s_{k-1}) \quad (1.19)$$

¹Assumendo da ora in avanti, Without Loss Of Generality (WLOG), $\Delta t = 1$ h.

dove vanno valutati i costi di tutte le N possibili transizioni tra uno stato e l'altro, non avendo imposto alcun vincolo sui rate di carica e scarica della batteria; in particolare esistono N possibili transizioni che lasciano lo stato s_{k-1} ed N possibili transizioni che arrivano allo stato s_k .

Ad ogni possibile valore E_j , per $j = 0, 1, \dots, N-1$, assunto dallo stato s_k all'istante temporale k -esimo vengono associate due quantità:

1. la *path metric*, o funzione di costo, definita da

$$\Gamma(s_k = E_j) = \min_{s_0, s_1, \dots, s_k = E_j} \Gamma_k \quad (1.20)$$

2. la *survivor sequence*, definita come la sequenza di livelli energetici che finisce in tale stato e che determina $\Gamma(s_k = E_j)$

$$\mathcal{L}(s_k = E_j) = (s_0, s_1, \dots, s_k = E_j) \quad (1.21)$$

dove si ricordi che $s_0 = E_0$ è il livello di energia inizialmente presente nella batteria e supposto conosciuto (lo si può assumere pari a 0 WLOG).

Queste due quantità si possono determinare in modo ricorsivo. È facile verificare, infatti, che se, all'istante temporale k -esimo, una *survivor sequence* include $s_k = E_j$, allora, all'istante temporale $k-1$ -esimo, la stessa sequenza *deve* includere $s_{k-1} = E_{i_{\text{opt}}}$, che si può esprimere nel modo seguente

$$E_{i_{\text{opt}}} = \arg \min_{s_{k-1} = E_i \in \mathcal{S} \rightarrow s_k = E_j} \Gamma(s_{k-1} = E_i) + C(P_k + E_j - E_i) \quad (1.22)$$

dove il termine $C(P_k + E_j - E_i)$ è chiamato *branch metric*. Si ottiene, quindi, che

$$\Gamma(s_k = E_j) = \Gamma(s_{k-1} = E_{i_{\text{opt}}}) + C(P_k + E_j - E_{i_{\text{opt}}}) \quad (1.23)$$

e la *survivor sequence* viene aggiornata nel modo seguente

$$\mathcal{L}(s_k = E_j) = (\mathcal{L}(s_{k-1} = E_{i_{\text{opt}}}), E_j). \quad (1.24)$$

Partendo da $k=1$, con stato iniziale $s_0 = E(0)$ assunto noto, tale procedura ricorsiva è ripetuta finché $k = T-1$. La sequenza ottima di stati è data dalla *survivor sequence* $\mathcal{L}(s_{T-1} = E_{j_{\text{opt}}})$ associata a $s_{T-1} = E_{j_{\text{opt}}}$ avente *costo minimo*. Assumendo come noto lo stato s_0 di valore E_0 , risulta conveniente associare agli altri valori assumibili da s_0 i seguenti costi

$$\Gamma(s_0 = E_i) = \begin{cases} 0, & E_i = E_0 \\ +\infty, & \text{altrimenti} \end{cases} \quad (1.25)$$

Nella pratica, se l'estremo temporale superiore $T-1$ è parecchio grande, la lunghezza delle *survivor sequences* viene limitata ad un valore T_d , chiamato profondità del traliccio: questo significa che ad ogni istante temporale k si decide il livello energetico s_{k-T_d} dell'intervallo $(k-T_d)$ -esimo, un valore che viene rimosso dal diagramma. La decisione è basata sulla *survivor sequence* alla quale è associato il minimo tra i valori di $\Gamma(s_k)$. Un altro aspetto puramente implementativo è che i valori di $\Gamma(s_k)$ possono diventare molto elevati: tale quantità può quindi essere normalizzata sottraendo lo stesso valore da tutte le metriche (senza che questo vada ad influire sul processo di ottimizzazione), per esempio la più piccola tra le quantità $\Gamma(s_k = E_i)$, $i = 0, 1, \dots, N-1$.

La memoria necessaria per conservare le metriche $\Gamma(s_k)$, $\Gamma(s_{k-1})$ e le *survivor sequences* risulta proporzionale al numero di stati N . Per quanto riguarda la *complessità computazionale* dell'algoritmo, il numero di somme e confronti è proporzionale al numero di transizioni, sempre uguale a N : quella totale dell'algoritmo, che ripete iterazione dopo iterazione sempre le stesse operazioni, risulta quindi lineare in T .

1.7 Conseguenze dell'aggiunta dei vincoli alla batteria sulla tecnica risolutiva

Si consideri quindi il caso di un sistema con una sola batteria di capacità finita pari a E_{\max} , e che rispetti inoltre tutti i vincoli definiti nella Sezione 1.1. Tale situazione si dimostra parecchio diversa da quella riportata nella Sezione 1.6: il *problema di ottimizzazione* che si può definire in questo caso è dato da

$$\{E_{\text{opt}, k}\}_{k=1, 2, \dots, T-1} = \underset{\{E_k\}_{k=1, 2, \dots, T-1}}{\text{arg min}} \quad \mathbb{E} \left[\sum_{k=0}^{T-1} C(X_k) \right] \quad (1.26)$$

subject to : (1.27), (1.28), (1.14)

dove

$$-\frac{F_{\max}^d \Delta t}{\eta_d} \leq E_{k+1} - E_k \leq F_{\max}^d \Delta t \eta_c \quad (1.27)$$

con $\delta_k \triangleq E_{k+1} - E_k$, e

$$0 \leq E_k \leq E_{\max}. \quad (1.28)$$

Ponendosi l'obiettivo di minimizzare (1.26) offline, anche stavolta l'operatore di aspettazione diventa "superfluo" per l'assenza di aleatorietà di tale problema, che si può riformulare come

$$\{E_{\text{opt}, k}\}_{k=1, 2, \dots, T-1} = \underset{\{E_k\}_{k=1, 2, \dots, T-1}}{\text{arg min}} \quad \sum_{k=0}^{T-1} C(X_k) \quad (1.29)$$

subject to : (1.27), (1.28), (1.14).

In questo caso, però, ci si può rendere conto come i rate e le efficienze di carica e scarica della batteria, diversi rispettivamente da $+\infty$ e da 1, restringano l'insieme immagine dei valori assumibili dal livello energetico della batteria nello slot temporale $k+1$, dato quello al tempo k . Per questo motivo, nell'algoritmo di Viterbi descritto nella Sezione 1.6, oltre al cambiamento del numero di stati (dato ora da $N = \lfloor \frac{E_{\max}}{\Delta E} \rfloor + 1$), il numero di transizioni possibili dipende dal livello energetico dello stato corrente. Detto E_k il livello energetico (che deve ovviamente rispettare il vincolo $0 \leq E_k \leq E_{\max}$) dell'intervallo temporale k -esimo, quello dello stato successivo E_{k+1} deve rispettare la disuguaglianza

$$\max \left\{ 0, -\frac{F_{\max}^d \Delta t}{\eta_d} \right\} \leq E_{k+1} - E_k \leq \min \{ E_{\max}, F_{\max}^c \Delta t \eta_c \} \quad (1.30)$$

dove, ovviamente, i valori assumibili da s_{k+1} vanno discretizzati con passo di quantizzazione ΔE . Il numero *massimo* di transizioni possibili dal generico stato s_k allo stato successivo s_{k+1} risulta quindi definito dal seguente insieme di equazioni

$$\begin{aligned} M_{\text{su}} &= \min \left\{ N, \lfloor (F_{\max}^c \Delta t \eta_c / \Delta E) \rfloor \right\} \\ M_{\text{giu}} &= \min \left\{ N, \lfloor F_{\max}^d \Delta t / (\eta_d \Delta E) \rfloor \right\} \\ M &= \min \{ N, M_{\text{su}} + M_{\text{giu}} + 1 \}. \end{aligned} \quad (1.31)$$

Fornire ulteriori dettagli delle modifiche da apportare all'algoritmo risulterebbe straniante non essendo ancora in un'ottica implementativa. Quello che è importante notare è che la complessità computazionale del Viterbi si riduce imponendo i vincoli sui rate di carica e scarica finiti e sulle efficienze energetiche non unitarie. Per quantificare la mole di questa riduzione computazionale, tuttavia, non si può far altro che mettersi nel caso peggiore, in cui il numero *massimo* di

transizioni possibili da stato a stato ad ogni iterazione dell'algoritmo è pari al valore M definito in (1.31): tuttavia, nella pratica, le transizioni effettivamente valutate saranno molto inferiori.

Le soluzioni del problema offline con i vincoli introdotti risulterà ovviamente “sub-ottima” (nel senso che corrisponde ad un costo superiore) rispetto a quella trovata nella Sezione 1.6: in questo caso il Viterbi osserva sempre l'intera sequenza $\{P_k\}$, ma se si rende conto, ad esempio, che in qualche intervallo temporale successivo a quello corrente c'è un forte picco della richiesta di potenza, che necessiterebbe di ricaricare la batteria il più possibile prima di tale evento, può sì ricaricarla, ma compatibilmente con il vincolo di rate di carica finito ed efficienza di carica non unitaria, arrivando verosimilmente ad un livello energetico sub-ottimo, in quanto *inferiore*. Un approccio di questo tipo, tuttavia, è l'unico vagamente (dato che non modella ulteriori vincoli a livello fisico delle batterie effettivamente in commercio, analizzati, in parte, nel Capitolo 4) realistico, anche se spesso in letteratura si preferisce prendere in considerazione solo che $E_{\max} \neq \infty$ (o nemmeno questo, come, ad esempio, in [1]); si ricordi, inoltre, che la soluzione del caso offline, anche con l'introduzione di tutti i vincoli sulla batteria, presenta interesse solo come lower bound per il problema di ottimizzazione stocastico online trattato nei capitoli seguenti.

1.8 Risultati delle simulazioni numeriche

Implementando in Matlab quanto appena descritto, si è scelto di analizzare quanto il lower bound offline calcolabile tramite l'algoritmo di Viterbi sia più performante in termini di costo complessivo rispetto a quello corrispondente al semplice algoritmo a singola soglia online che, senza conoscere l'intera sequenza delle richieste di potenza $\{P_k\}$, prevede di caricare o scaricare la batteria se la richiesta di potenza istantanea è rispettivamente inferiore o superiore rispetto al suo valore medio $P_{\text{thr}} = \mathbb{E}[P]$, assunto come noto. Con la funzione di costo lineare a tratti e con coda *quadratica* precedentemente definita, vengono confermate le attese di una *evidente differenza di prestazioni* da parte dei due algoritmi (a differenza di quello che si osserverà nel Capitolo 4 con una funzione di costo lineare rispetto alla potenza ma dipendente anche dalle caratteristiche della batteria): ciò è facilmente giustificabile dal fatto che, senza conoscere l'andamento futuro della richiesta di potenza e prendendo la decisione di carica/scarica in modo greedy solo sulla base del valore istantaneo, si può venire penalizzati pagando un costo molto superiore a quello minimo. L'azione di carica/scarica della batteria si caratterizza come un possibile shift del valore istantaneo della richiesta di potenza: quando essa assume un valore elevato, se si ha a disposizione energia perchè in passato si è stata caricata la batteria in un momento in cui tale valore era molto inferiore, si può risparmiare notevolmente in termini di costo complessivo.

Come prima simulazione si è scelto di far variare la capacità E_{\max} della batteria, tenendo i rate di carica e scarica normalizzati fissati a $\left\lfloor \frac{E_{\max}}{4\Delta E} \right\rfloor^2$: in ogni intervallo temporale di durata Δt la batteria può quindi essere caricata o scaricata di circa un quarto della propria capacità massima. Già in Figura 1.8, si nota il comportamento di difficile interpretazione dell'algoritmo ST: essendo basato su un approccio euristico, quando gli vengono imposti i vincoli relativi ai rate di carica/scarica finiti della batteria, determina un costo complessivo *inferiore* al caso in cui sia sempre possibile caricare/scaricare completamente la batteria in ogni intervallo temporale di durata Δt . Ad ogni modo, in entrambi i casi appena descritti, il costo complessivo risulta

²Sono state scelte inoltre, *in questa ed in tutte le altre simulazioni*, delle efficienze di carica/scarica tali che $\eta = \eta_c = \eta_d = 0.90$, avendo verificato che l'effetto della variazione di tale parametro (che aumenta il numero di variabili indipendenti del sistema) è sempre *lineare* rispetto a quella del costo complessivo da pagare da parte degli utenti della grid, come riportato nella Sezione 3.3.

superiore *almeno* del 21 % rispetto a quello ottenibile tramite l'algoritmo AOS in presenza dei vincoli sui rate di carica/scarica.

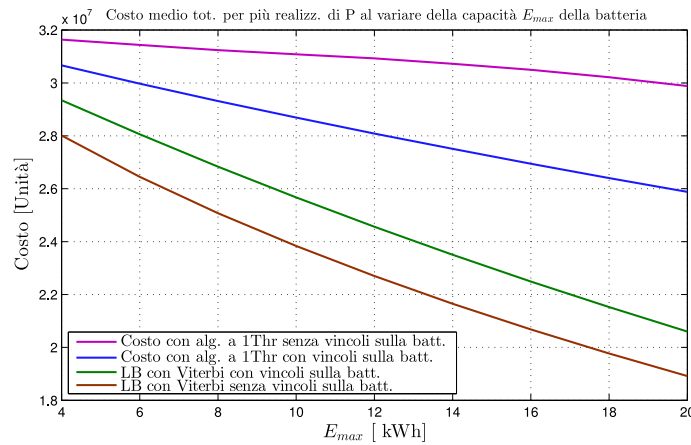


Figura 1.8: Simulazione al variare della capacità della batteria per calcolare il costo medio.

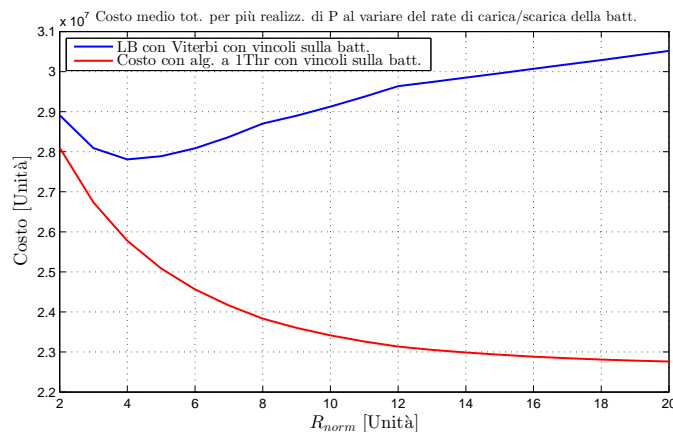


Figura 1.9: Simulazione al variare dei rate di carica/scarica normalizzato della batteria per calcolare il costo medio.

Il risultato riportato in Figura 1.9, si riferisce ad una simulazione in cui si tiene fissa la capacità della batteria a 12 kWh e si fanno variare i rate di carica/scarica normalizzati (assunti uguali in valore assoluto): nel caso del costo complessivo relativo al lower bound calcolato con Viterbi, si osserva un andamento monotono decrescente di facile interpretazione (l'assenza di vincoli sul rate corrisponderebbe ad avere $R_{norm}=24$). Per quanto riguarda l'algoritmo a soglia, la quale è data dalla media della sequenza $\{P_k\}$, invece, il costo inizialmente scende all'aumentare di R_{norm} , ma poi comincia a crescere, come se l'aver più gradi di libertà nella scelta dell'azione di carica/scarica da intraprendere nei confronti della batteria *non* giovasse alle prestazioni di tale algoritmo euristico: ciò è giustificabile dal fatto che tale algoritmo risulta ottimo *sia* quando la capacità della batteria tende all'infinito *sia* quando non sono presenti i vincoli sul rate di carica/scarica che sono stati introdotti. Risulta quindi non predicibile il suo comportamento in termini di costo complessivo risultante al variare della intensità di tali vincoli, imposti su una batteria di capacità 12 kWh (gli autori dell'algoritmo hanno trovato per via numerica che le sue

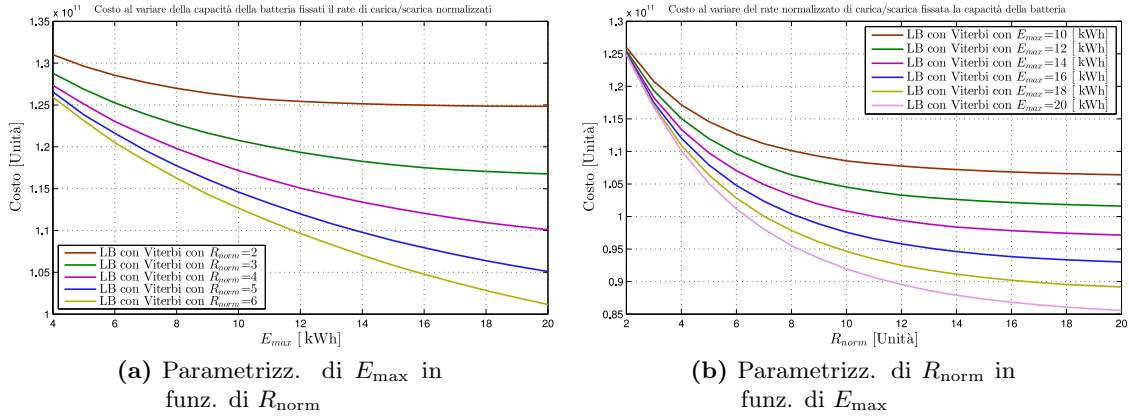


Figura 1.10: Simulazione al variare della capacità E_{\max} parametrizzata in funzione di R_{norm} (*e viceversa*) per calcolare il costo medio.

prestazioni tendono ad approssiarsi al lower bound di Viterbi quando la capacità della batteria è $\gg 24$ kWh).

In Figura 1.10a, invece, si riportano delle curve, parametrizzate in funzione dei rate di carica/scarica normalizzati al variare della capacità E_{\max} della batteria, relative al costo complessivo ottenuto soltanto con l'algoritmo di Viterbi, per non appesantire troppo la visualizzazione. Questo risultato dà un'idea di quanto i vincoli sui rate di carica/scarica finiti influenzino il costo complessivo: la distanza della curva con $R_{\text{norm}} = 2$ da quella con $R_{\text{norm}} = 3$ è visibilmente maggiore di quella dalla curva con $R_{\text{norm}} = 3$ alla curva con $R_{\text{norm}} = 4$. Lo stesso si ha per la transizione tra i due valori di R_{norm} successivi; poi, invece, la distanza tra una curva e l'altra tende a stabilizzarsi. Questo dimostra che l'influenza del vincolo si fa sentire particolarmente quando il numero di transizioni possibili tra uno stato della batteria all'altro in un singolo intervallo temporale è molto *limitato*, per poi portare a decrementi del costo complessivo all'aumentare del modulo del rate di carica/scarica *tutti con la stessa costante di proporzionalità rispetto a tale incremento*.

In Figura 1.10b, in modo complementare, si riportano gli andamenti, parametrizzati in funzione della capacità E_{\max} della batteria, al variare dei rate di carica/scarica normalizzati, relativi al lower bound della strategia AOS. A differenza del caso precedente, aumentando a passi di 2 kWh la capacità della batteria, il costo complessivo corrispondente scende in modo pressochè *lineare* per qualunque range di energia immagazzinabile nella batteria: non c'è un decremento di costo significativamente maggiore nei primi aumenti della capacità rispetto a quelli successivi. Ciò significa che la funzione di costo risulta *meno sensibile* all'aver posto il vincolo $E_{\max} \neq \infty$ rispetto a quello sulla finitezza dei rate di carica/scarica in un singolo intervallo temporale di durata Δt .

1.9 Caso con due batterie

Si consideri quindi il caso di un sistema con *due* batterie di capacità *finita* pari a $E_{\max, 1}$ ed $E_{\max, 2}$, e che rispettino inoltre tutti i vincoli definiti nella Sezione 1.1. Il problema di ottimizzazione che

si può definire in questo nuovo caso è dato da ³

$$\{\mathbf{E}_{\text{opt}, k}\}_{k=1, 2, \dots, T-1} = \underset{\{\mathbf{E}_k\}_{k=1, 2, \dots, T-1}}{\text{arg min}} \quad \mathbb{E} \left[\sum_{k=0}^{T-1} C(X_k) \right] \quad (1.32)$$

subject to : (1.19), (1.20), (1.14)

dove

$$-\frac{F_{\text{max}, i}^d \Delta t}{\eta_{d, i}} \leq E_{k+1, i} - E_{k, i} \leq F_{\text{max}, i}^d \Delta t \eta_{c, i} \quad \text{per } i = 1, 2 \quad (1.33)$$

e

$$0 \leq E_{k, i} \leq E_{\text{max}, i} \quad \text{per } i = 1, 2. \quad (1.34)$$

Ponendosi l'obiettivo di minimizzare (1.32) offline, l'operatore di aspettazione diventa come al solito "superfluo" per l'assenza di aleatorietà di tale problema, che si riformula come

$$\{\mathbf{E}_{\text{opt}, k}\}_{k=1, 2, \dots, T-1} = \underset{\{\mathbf{E}_k\}_{k=1, 2, \dots, T-1}}{\text{arg min}} \quad \sum_{k=0}^{T-1} C(X_k). \quad (1.35)$$

subject to : (1.19), (1.20), (1.14)

I rate e le efficienze di carica e scarica della batteria, come già discusso nella Sezione 1.8, restringano l'insieme immagine dei valori assumibili dai livelli energetici delle due batterie nello slot temporale $k + 1$, dato quello al tempo k . Nell'algoritmo di Viterbi il numero di stati diventa $N = N_1 \cdot N_2$ (con $N_1 = \lfloor \frac{E_{\text{max}, 1}}{\Delta E_1} \rfloor + 1$ e $N_2 = \lfloor \frac{E_{\text{max}, 2}}{\Delta E_2} \rfloor + 1$) e il numero di transizioni possibili dipende dai livelli energetici raggiungibili ad un passo dello stato corrente $\mathbf{E}_k = (E_{k, 1}, E_{k, 2})$. Detta \mathbf{E}_k la coppia dei livelli energetici nelle due batterie (che deve ovviamente rispettare il vincolo $0 \leq E_{k, i} \leq E_{\text{max}, i}$ per $i = 1, 2$) dell'intervallo temporale k -esimo, quella dello stato successivo \mathbf{E}_{k+1} deve rispettare la disuguaglianza

$$\max \left\{ 0, -\frac{F_{\text{max}, i}^d \Delta t}{\eta_{d, i}} \right\} \leq E_{k+1, i} - E_{k, i} \leq \min \{ E_{\text{max}, i}, F_{\text{max}, i}^c \Delta t \eta_{c, i} \} \quad (1.36)$$

dove, in generale, i valori assumibili dalle due componenti di \mathbf{s}_{k+1} possono essere discretizzati con passi di quantizzazione ΔE_1 e ΔE_2 diversi tra loro. Il numero *massimo* di transizioni possibili dal generico stato \mathbf{s}_k allo stato successivo \mathbf{s}_{k+1} risulta quindi definito dal seguente insieme di equazioni

$$\begin{aligned} M_{\text{su}, i} &= \min \left\{ N_i, \lfloor F_{\text{max}, i}^c \Delta t \eta_{c, i} / \Delta E_i \rfloor \right\} \quad \text{per } i = 1, 2 \\ M_{\text{giu}, i} &= \min \left\{ N_i, \lfloor F_{\text{max}, i}^d \Delta t / (\eta_{d, i} \Delta E_i) \rfloor \right\} \quad \text{per } i = 1, 2 \\ M_i &= \min \{ N_i, M_{\text{su}, i} + M_{\text{giu}, i} + 1 \} \quad \text{per } i = 1, 2. \end{aligned} \quad (1.37)$$

Fornire ulteriori dettagli delle modifiche da apportare all'algoritmo risulterebbe ancora più tedioso del caso con una sola batteria. Si osserva solo che, sebbene la equazioni del caso "mono e bi-dimensionale" siano soltanto la naturale estensione di un problema di ottimizzazione rispetto ad una sequenza di scalari ad un altro rispetto ad una sequenza di coppie di scalari, implementare l'algoritmo di Viterbi, e farlo in modo *efficiente dal punto di vista computazionale*, non è stato affatto un problema banale.

³dove $\{\mathbf{E}_{\text{opt}, k}\}$ è un vettore avente come 2 componenti i livelli energetici delle 2 batterie al passare del tempo, $\{E_{\text{opt}_1, k}\}$ e $\{E_{\text{opt}_2, k}\}$ rispettivamente.

1.9.1 Risultati delle simulazioni numeriche

Come appena menzionato, implementare l'algoritmo di Viterbi nel caso in cui il controllore della grid si trovi a dover affrontare il problema dell'*ottimizzazione congiunta*⁴ dei livelli energetici delle due batterie che ha a disposizione al fine di minimizzare il costo complessivo è un problema più complesso dal punto di vista implementativo che da quello concettuale. Per questa ragione nelle righe successive si è scelto, piuttosto che descrivere a grandi linee l'idea definitiva avuta per l'implementazione, di riportare il codice Matlab dei due punti salienti dell'algoritmo AOS, che sono, rispettivamente, il calcolo di un'opportuna "matrice di provenienti" rispetto ad ogni stato (nel Codice 1.1) e l'aggiornamento della *survivor sequence* allo scorrere del tempo (nel Codice 1.2), perchè si ritiene possano essere un punto di partenza per chiunque generalizzerà l'implementazione del problema al caso M -dimensionale.

Codice 1.1: Codice Matlab per calcolare la "matrice dei provenienti" per l'algoritmo di Viterbi applicato al problema di minimizzazione del costo complessivo con due batterie a disposizione.

```
% Creazione di un vettore (SOLO a valori numerici) in 3 dimensioni
% che rappresenti i possibili stati assumibili dalle due batterie
v_stati_n=zeros(N_tot,1,2);
i=1;
while (i<=N_tot)
    for q_2=1:N_2
        for q_1=1:N_1
            v_stati_n(i,1,1)=q_1;
            v_stati_n(i,1,2)=q_2;
            i=i+1;
        end
    end
end
end
```

Codice 1.2: Codice Matlab per aggiornare la *survivor sequence* allo scorrere del tempo all'interno dell'algoritmo di Viterbi applicato al problema di minimizzazione del costo complessivo con due batterie a disposizione.

```
for k=1:T
    for r=1:N_tot
        branch.metric=zeros(N_tot,1);
        costo_tmp=1e40*ones(N_tot,1);
        r_1=v_stati_n(r,1,1);
        r_2=v_stati_n(r,1,2);
        q_i_1=max(r_1-M.su_1,1);
        q_i_2=max(r_2-M.su_2,1);
        q_f_1=min(r_1+M.giu_1,N_1);
        q_f_2=min(r_2+M.giu_2,N_2);
        for q_1=q_i_1:q_f_1
            for q_2=q_i_2:q_f_2
                ind_stato_prov=-1;
                for kk=1:N_tot
                    if ((v_stati_n(kk,1,1)==q_1) && (v_stati_n(kk,1,2)==q_2))
                        ind_stato_prov=kk;
                    end
                end
            end
        end
    end
end
```

⁴Sono state condotte anche delle simulazioni di ottimizzazione disgiunta rispetto a $\{E_{k,1}\}$ e $\{E_{k,2}\}$, che qui non vengono riportate sia per mancanza di spazio sia perchè non hanno portato a dei risultati di particolare interesse.

```

branch_metric(ind_stato_prov)=fun_cost(max(P(k)+((r_1-1)
-(q_1-1))*Delta_E_1+((r_2-1)-(q_2-1))*Delta_E_2,0));
costo_tmp(ind_stato_prov)=Gamma(ind_stato_prov,k)
+branch_metric(ind_stato_prov);
end
end
min_cost=min(costo_tmp);
ind_opt_v=find(costo_tmp==min_cost);
ind_opt=ind_opt_v(floor(length(ind_opt_v)*rand)+1);
Gamma(r,k+1)=min_cost;
LG(r,k+1)=ind_opt;
end
end

```

Detto ciò, si crede risulti più interessante, piuttosto che riportare nel caso a due batterie simulazioni analoghe a quelle della Sezione 1.8 al variare dei loro rate, capacità, etc., effettuare un confronto tra le prestazioni in termini di costo complessivo da pagare da parte degli utenti della grid (per alcuni valori di capacità) avendo a disposizione un'unica batteria o due batterie, le quali, in generale, possono avere anche caratteristiche completamente diverse. L'aspetto non banale è rendere questo confronto *equo* dal punto di vista della *fairness*: non basta infatti che le capacità delle due batterie sommino a quelle del caso a singola batteria, ovvero $E_{\max} = E_{\max,1} + E_{\max,2}$, ma bisogna prestare attenzione anche ai rate e alle efficienze di carica/scarica, nonché ai passi di quantizzazione. Rimanendo quindi nel caso a parità di capacità totale e supponendo di aver già effettuato la quantizzazione riassunta dall'equazione (1.37), si è condotta una prima simulazione in cui $M_{\text{su}} = M_{\text{giu}} = \min \left\{ N, \left\lfloor \frac{E_{\max}}{4\Delta E} \right\rfloor \right\}$ e $M_{\text{su},i} = M_{\text{giu},i} = \min \left\{ N_i, \left\lfloor \frac{E_{\max,i}}{4\Delta E_i} \right\rfloor \right\}$ per $i = 1, 2$: nel caso a singola a batteria e in quello a due batterie, cioè, c'è a disposizione *la stessa percentuale della corrispondente capacità totale* per effettuare le azioni di carica/scarica. La simulazione ha considerato un passo di quantizzazione del livello energetico della batteria/batterie pari a $\Delta E = \Delta E_1 = \Delta E_2 = 0.5$ kWh: intuitivamente, a parità di capacità totale, di passo di quantizzazione e con rate di carica/scarica proporzionali alla capacità corrispondente *con la stessa costante di proporzionalità*, il caso a due batterie dovrebbe portare a prestazioni *migliori* di quello con una sola batteria, perchè è maggiore la libertà dell'algoritmo di Viterbi in termini di azioni nel relativo spazio degli stati. I risultati riportati nella Tabella 1.1 in termini di costo medio cumulativo e nelle Figure 1.11 e 1.12 in termini di costo medio istantaneo suggeriscono, però, che tale affermazione va ulteriormente approfondita, come discusso nella conclusione di questa sezione: si è preferito mantenere nell'esposizione anche questo step intermedio perchè aiuta a capire il procedimento logico che ha condotto alle osservazioni finali.

Ci si focalizzi sul costo complessivo riportato nella Tabella 1.1 (che è quanto pagato in totale dagli utenti di un'ipotetica grid con una sola batteria e da quelli di un'altra con due batterie): al contrario delle previsioni, i valori del caso a singola batteria rispetto a quello con due dispositivi di *energy storage* non solo sono vicinissimi, ma addirittura il costo con un'unica batteria di capacità $E_{\max} = E_{\max,1} + E_{\max,2}$ è in generale inferiore (anche se, pragmaticamente, *uguale con ottima approssimazione*) a quello con le due batterie. Questo risultato, di difficile interpretazione, si ritiene sia legato al passo di quantizzazione dei livelli energetici: ragionando a valori discreti, se $\Delta E = \Delta E_1 = \Delta E_2 = 0.5$ kWh, può capitare che per alcuni valori della capacità la somma di $M_{\text{su},1} + M_{\text{su},2} \neq M_{\text{su}}$ e $M_{\text{giu},1} + M_{\text{giu},2} \neq M_{\text{giu}}$ per *ragioni numeriche* dettate dalla loro definizione presente nell'equazione (1.37), in cui è presente la funzione $\lfloor \cdot \rfloor$. Quando si è in uno di questi ultimi casi la soluzione a doppia batteria risulta svantaggiata rispetto a quella a singola batteria, perchè all'interno dell'algoritmo di Viterbi non c'è la stessa possibilità di escursione in termini di livello energetico complessivo (dato dalla somma del modulo delle differenze di quelli

Tabella 1.1: Risultati numerici sul costo complessivo offline (calcolato con l'algoritmo AOS) relativo al confronto tra il caso a singola e a doppia batteria: $\Delta E = \Delta E_1 = \Delta E_2 = 0.5$ kWh.

Descrizione	Costo ottenuto
1 batteria da 4kWh	1.99019e+10
2 batterie da 2 kWh	1.99020e+10
1 batteria da 6 kWh	1.90584e+10
2 batterie da 3 kWh	1.93728e+10
1 batteria da 8 kWh	1.82558e+10
2 batterie da 4 kWh	1.82558e+10
1 batteria da 10 kWh	1.74914e+10
2 batterie da 5 kWh	1.77738e+10
1 batteria da 12 kWh	1.67640e+10
2 batterie da 6 kWh	1.67639e+10

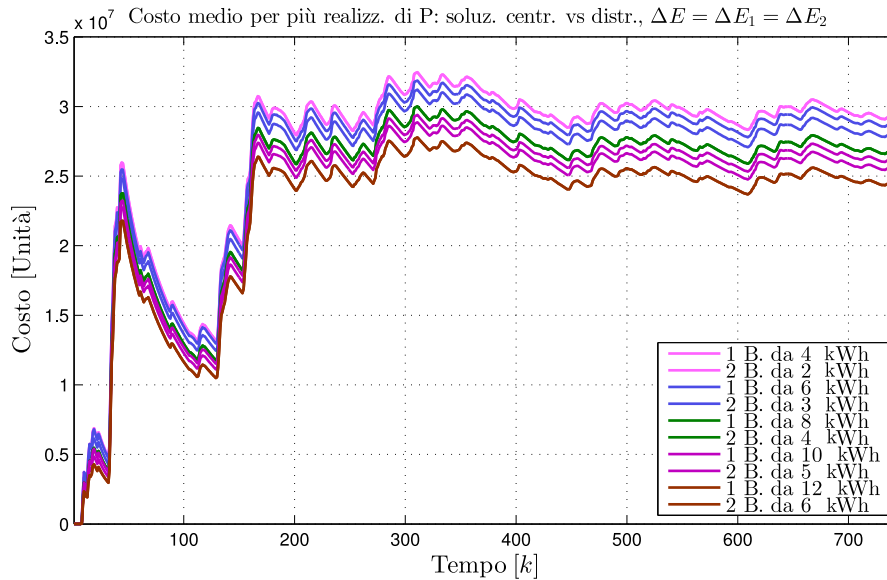


Figura 1.11: Simulazione al variare della capacità $E_{\max} = E_{\max,1} + E_{\max,2}$ per confrontare il caso a 1 batteria rispetto a quello a 2 batterie in termini di costo medio istantaneo (andamento complessivo).

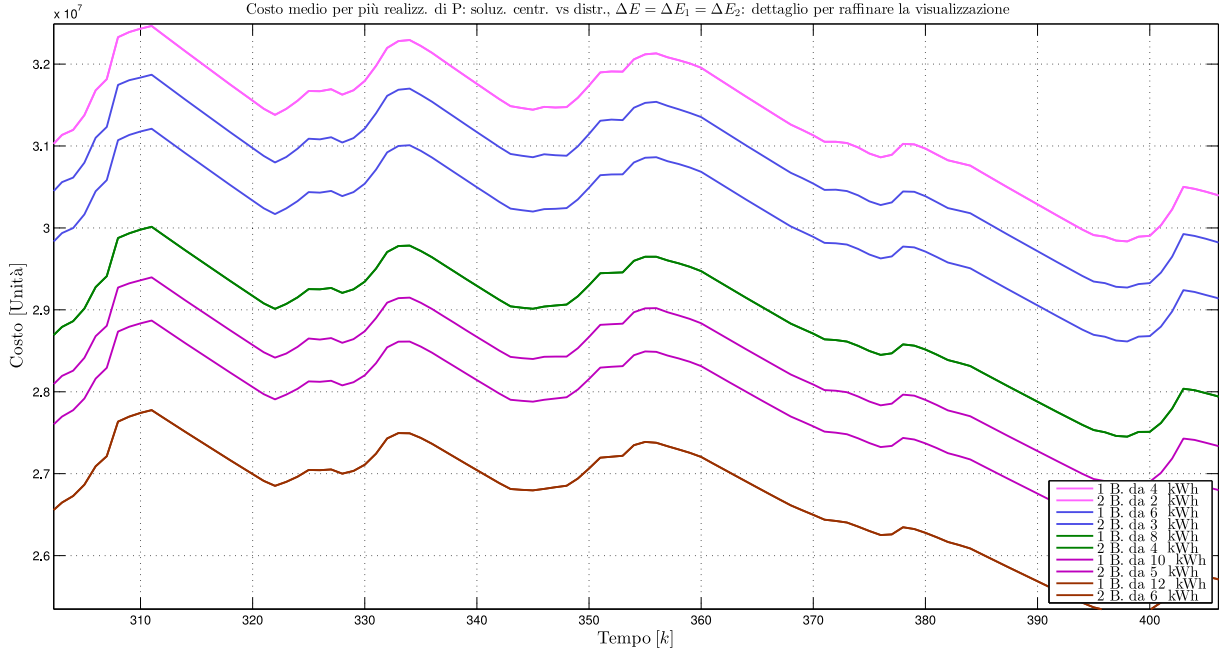


Figura 1.12: Simulazione al variare della capacità $E_{\max} = E_{\max,1} + E_{\max,2}$ per confrontare il caso a 1 batteria rispetto a quello a 2 batterie in termini di costo medio istantaneo (dettaglio).

delle due batterie) rispetto al caso con un solo dispositivo di *energy storage* di capacità E_{\max} . Per avere conferma di quanto appena affermato, si è eseguita una seconda simulazione identica alla prima a parte il fatto che $\Delta E_1 = \Delta E_2 = \frac{\Delta E}{2} = 0.25$ kWh in modo da non favorire mai la situazione a batteria centralizzata rispetto a quella con doppia batteria; il “prezzo da pagare” è un aumento notevole del tempo di calcolo del costo di quest’ultima soluzione, dal momento che la complessità computazionale dell’algoritmo di Viterbi si sa $\sim O(T \cdot N_1 \cdot N_2)$ e che N_1 ed N_2 sono legati al passo di quantizzazione impiegato dalla relazione $N_i = \left\lceil \frac{E_{\max,i}}{\Delta E_i} \right\rceil + 1$ per $i = 1, 2$. I risultati ottenuti sono riportati nella Tabella 1.2 in termini di costo medio cumulativo e nelle Figure 1.13 e 1.14 in termini di costo medio istantaneo.

I costi cumulativi riportati nella Tabella 1.1 confermano con buona approssimazione l’osservazione precedentemente riportata sul passo di quantizzazione dei livelli energetici: per valori elevati della capacità, la soluzione con due batterie offre ora prestazioni in termini di costo leggermente superiori di quella a singola batteria, ma in modo così modesto in termini percentuali che si possono considerare a tutti gli effetti *equivalenti*. Ciò è confermato anche dalla Figura 1.13: le corrispondenti curve del costo medio istantaneo risultano sovrapposte, e nemmeno nella Figura 1.14, che ne riporta un forte ingrandimento, si riesce ad osservare visivamente la differenza.

La spiegazione che si ritiene essere la più ragionevole di questo confronto tra costo della soluzione con uno o più dispositivi di *energy storage*, a parità di capacità totale, è legata a quanto riportato nella Sezione *E - Resource pooling* dell’articolo [2]: lì gli autori del paper, ragionando in modo *complementare* rispetto a quanto fatto in questa sezione, considerano come *alternativa* all’*energy storage* da parte di singole abitazioni, ciascuna con una propria batteria di capacità limitata, quella di usare un’unica grande batteria condivisa da più utenti, ovvero, al posto del caso con n utenti ciascuno con una batteria da 16 kWh quello con un’unica batteria condivisa da $16 \cdot n$ kWh di capacità. Viene impiegato il tool impiegato nella Sezione 1.5 per generare richieste di potenza il più possibile *indipendenti e incorrelate* per ciascuna abitazione. Nel caso con più

Tabella 1.2: Risultati numerici sul costo complessivo offline (calcolato con l'algoritmo AOS) relativo al confronto tra il caso a singola e a doppia batteria: $\Delta E = 2 \Delta E_1 = 2 \Delta E_2 = 0.5$ kWh.

Descrizione	Costo ottenuto
1 batteria da 4kWh	1.99019e+10
2 batterie da 2 kWh	1.99020e+10
1 batteria da 6 kWh	1.90584e+10
2 batterie da 3 kWh	1.90728e+10
1 batteria da 8 kWh	1.82558e+10
2 batterie da 4 kWh	1.82558e+10
1 batteria da 10 kWh	1.74914e+10
2 batterie da 5 kWh	1.74909e+10
1 batteria da 12 kWh	1.67640e+10
2 batterie da 6 kWh	1.67639e+10

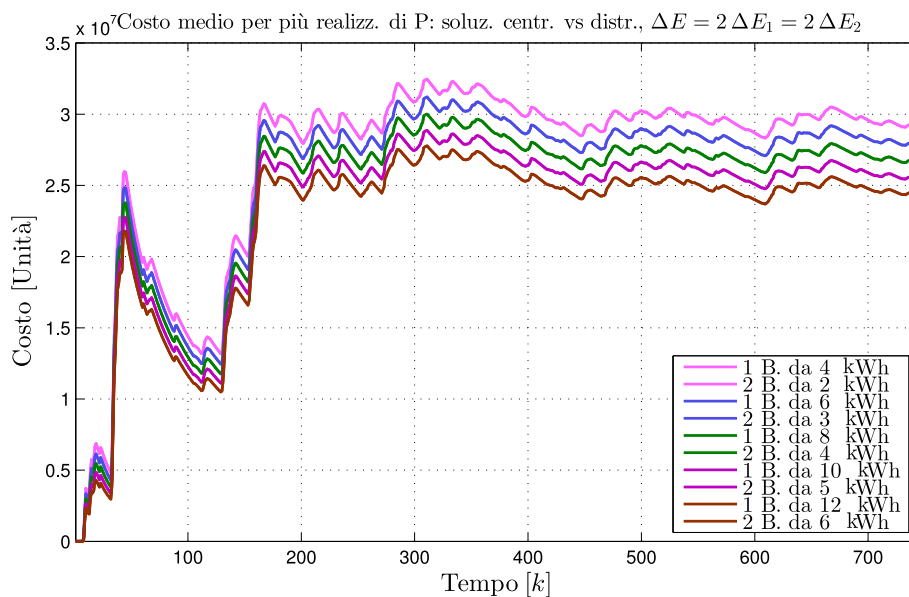


Figura 1.13: Simulazione al variare della capacità $E_{\max} = E_{\max,1} + E_{\max,2}$ con $\Delta E = 2 \Delta E_1 = 2 \Delta E_2 = 0.5$ kWh per confrontare il caso a 1 batteria rispetto a quello a 2 batterie in termini di costo medio istantaneo (andamento complessivo).

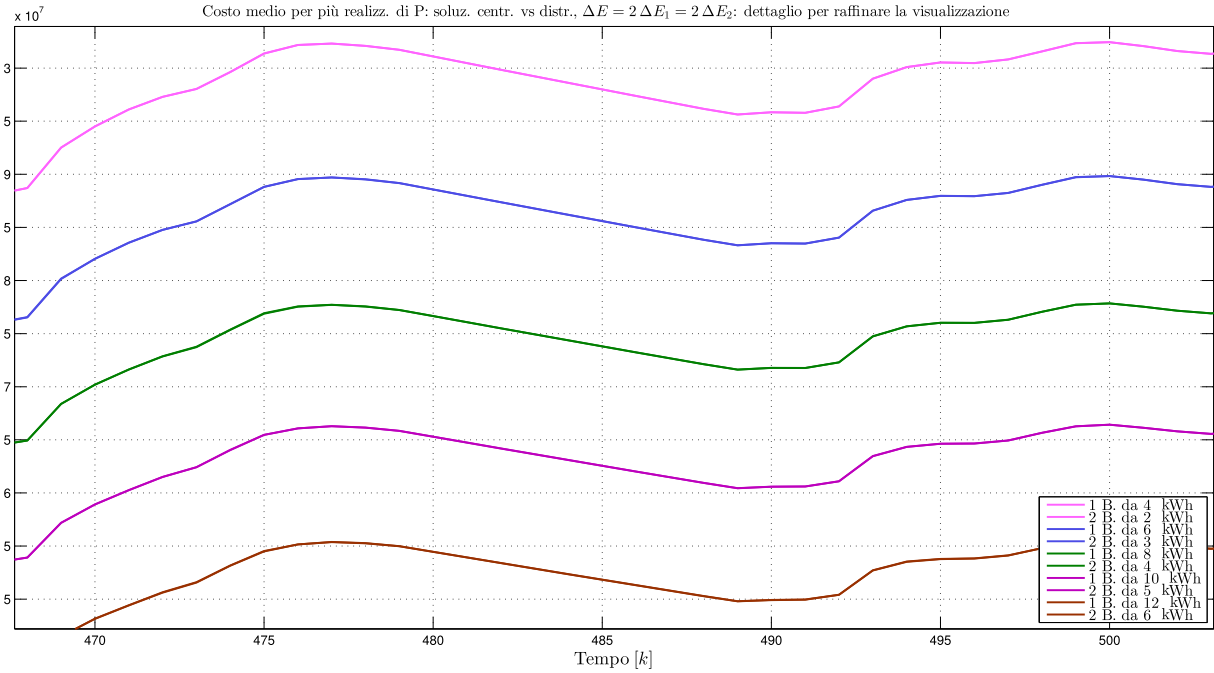


Figura 1.14: Simulazione al variare della capacità $E_{\max} = E_{\max,1} + E_{\max,2}$ con $\Delta E = 2 \Delta E_1 = 2 \Delta E_2 = 0.5$ kWh per confrontare il caso a 1 batteria rispetto a quello a 2 batterie in termini di costo medio istantaneo (dettaglio).

dispositivi di *energy storage*, ogni sequenza $\{P_{k,i}\}$ per $i = 1, 2, \dots, n$ viene data in ingresso al controllore della batteria corrispondente. Quando il dispositivo di *energy storage* è condiviso, invece, la *aggregate demand data* di tutte le abitazioni viene considerata come un'unica sequenza di richieste di potenza dal controllore dell'unica batteria del sistema. Nel paper si osserva per via numerica (a dire il vero senza specificare nessun parametro usato per confrontare le prestazioni in sede simulativa) che la *condivisione della batterie*, equivalente ad un unico dispositivo di *energy storage* con capacità n volte maggiore, porta a *pochissimi vantaggi* rispetto ad utilizzare più batterie con minor possibilità di immagazzinamento. Ciò viene giustificato dal fatto che si può verificare che le sequenze delle richieste di potenza siano *fortemente correlate*⁵: dato che tutti gli utenti hanno il consumo di picco approssimativamente alla stessa ora, c'è una modesta possibilità di poter sfruttare la batteria condivisa che, per poter portare ad un guadagno sul costo davvero significativo, dovrebbe avere una capacità $\gg 16 \cdot n$, in modo da poter essere utile anche nelle ore più "critiche" all'interno della giornata che, per la correlazione presente nelle richieste di potenza, sono comuni a tutte le abitazioni.

Rovesciando l'ottica dell'argomentazione, va ricordato che nello scenario considerato, quando si usano più batterie con capacità che sommano allo stesso valore di un'unica batteria più grande, queste ricevono in ingresso *non* richieste di potenza fortemente correlate, ma *esattamente la stessa sequenza* $\{P_k\}$: questo perchè si è sempre supposto che il controllo della grid sia *centralizzato* e quindi l'aggregazione delle domande di energia delle diverse abitazioni avvenga *a monte* dell'ingresso al controllore della batteria o delle batterie. È quest'ultima osservazione che si ritiene possa giustificare i risultati relativi al confronto tra il costo della soluzione ottenuta

⁵Ciò non risulta un limite del tool proposto dagli autori di [13], ma una peculiarità endemicamente diffusa del consumo energetico contemporaneo.

con uno o due dispositivi di *energy storage*: semplicemente il setting della grid del modello di sistema considerato, ed il problema di ottimizzazione che ne deriva, *non* è predisposto in maniera intelligente all'uso di più batterie distribuite al posto di un'unica centralizzata. Perché questo risulti una scelta che porti ad un guadagno sul costo davvero significativo, infatti, il problema di minimizzazione avrebbe dovuto essere definito ⁶ avendo in ingresso le sequenze *disgiunte* della richieste nette di potenza $\{P_{k,i}\}$ per $i = 1, 2, \dots, M$, ipotizzando che ciascuna abitazione sia dotata di una propria batteria e di un relativo controllore e sperando, come osservato in [2], che tali richieste energetiche risultino il più possibile *scorrelate*. Solo in tal caso il nuovo (e più complesso) problema di minimizzazione vettoriale congiunta rispetto alla somma dei costi da pagare da parte di ciascun utente potrebbe trarre dei vantaggi rilevanti dal *resource pooling*, mentre quello definito dall'equazione (1.11) non "guadagna in generalità" se ottimizzato rispetto alle M sequenze $\{E_{k,i}\}$ per $i = 1, 2, \dots, M$ dei livelli energetici di altrettante batterie.

⁶Ciò si potrebbe ben prestare ad altri lavori di studio e approfondimento da parte di altri tesisti.

Capitolo 2

Introduzione alla tecnica ottima di soluzione online: il RL

2.1 L'interfaccia agente-ambiente del RL

Il problema del Reinforcement Learning (RL), che viene trattato in modo chiaro e analiticamente preciso in [16], è nato per essere la *cornice* completa del problema di imparare da un'interazione per raggiungere un obiettivo. Il *learner* e *decision-maker* sono chiamati l'*agente*. La cosa con cui si interagisce, ovvero tutto quello al di fuori dell'agente, viene chiamato l'*ambiente*. Questi interagiscono continuamente: l'agente seleziona le azioni e l'ambiente risponde a tali azioni e presenta nuove situazioni all'agente. L'ambiente, inoltre, dà luogo a dei guadagni, valori numerici speciali che l'agente tenta di massimizzare nel corso del tempo. Una completa specificazione di un ambiente definisce un *task*, ovvero un'istanza del problema del RL.

Più in dettaglio, l'agente e l'ambiente interagiscono ad ogni passo di una sequenza di istanti temporali discreti, $t = 0, 1, 2, 3, \dots$. Ad ogni istante temporale t , l'agente riceve una qualche rappresentazione dello *stato* dell'ambiente, $s_t \in \mathcal{S}$, dove \mathcal{S} è l'insieme degli stati possibili, e su tale base seleziona un'*azione*, $a_t \in \mathcal{A}(s_t)$, dove $\mathcal{A}(s_t)$ è l'insieme di azioni disponibili nello stato s_t . Un istante di tempo dopo, in parte come conseguenza dell'azione scelta, l'agente riceve un guadagno di tipo numerico, $r_{t+1} \in \mathcal{R}$, e si ritrova in un nuovo stato, s_{t+1} ¹. La Figura 2.1 rappresenta schematicamente l'interazione tra agente ed ambiente.

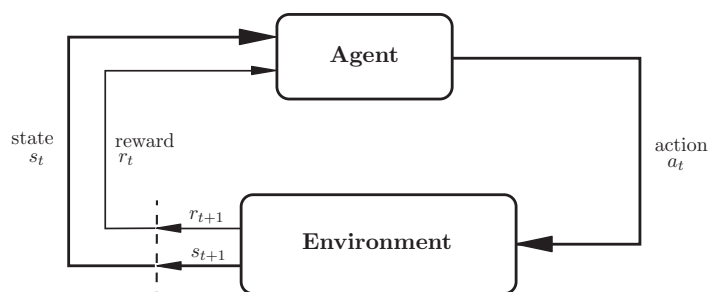


Figura 2.1: L'interazione agente-ambiente nel RL.

¹Viene usato r_{t+1} , al posto di r_t , per indicare il guadagno immediato dovuto all'azione intrapresa al tempo t , perchè ciò enfatizza il fatto che il guadagno successivo e lo stato successivo, s_{t+1} , sono determinati *congiuntamente*.

Ad ogni istante temporale, l'agente implementa una mappa dagli stati alle probabilità di selezionare ogni possibile azione. Questa mappa è definita come la *strategia* dell'agente e viene indicata con π_t , dove $\pi_t(s, a)$ è la probabilità che $a_t = a$ se $s_t = s$. I metodi di RL specificano come l'agente modifichi la propria strategia come conseguenza delle azioni intraprese. Lo scopo/obiettivo dell'agente è formalizzato in termini di un segnale speciale di guadagno che passa dall'ambiente all'agente. Ad ogni istante temporale, il guadagno è un semplice numero, $r_t \in \mathcal{R}$. In modo informale, l'obiettivo dell'agente è quello di massimizzare la somma totale dei guadagni che riceve: questo significa massimizzare non il guadagno immediato, ma quello cumulativo nel lungo periodo. Se la sequenza dei guadagni ricevuti dopo l'istante temporale t è indicata come $r_{t+1}, r_{t+2}, r_{t+3}, \dots$, è lecito chiedersi quale specifico aspetto di questa sequenza si tenti di massimizzare. In generale si cerca di massimizzare il *ritorno atteso*, dove il *ritorno*, R_t , è definito come una qualche specifica funzione della sequenza dei guadagni. Nel caso più semplice (che è anche quello considerato nel resto della trattazione) il ritorno è dato dalla *somma dei guadagni*

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_T \quad (2.1)$$

dove T è un istante di tempo finale. Questo approccio ha senso in applicazioni in cui ci sia una nozione naturale di istante temporale finale, ovvero, quando l'interazione agente-ambiente si può spezzare naturalmente in sotto-sequenze, che vengono chiamate *episodi*. Ogni episodio termina in uno stato speciale chiamato stato terminale, seguito da un reset ad uno stato di partenza o ad un campione da una certa distribuzione di stati di partenza. Nei task ad episodio ci può essere la necessità di distinguere l'insieme di tutti gli stati non-terminali, indicato come \mathcal{S} , da quello di tutti gli stati e lo stato terminale, indicato come \mathcal{S}^+ .

D'altro canto, in molti casi l'interazione agente-ambiente non si può spezzare naturalmente in episodi ben identificabili, ma va avanti continuamente senza limite: si tratta cioè di *continuing tasks*. La formulazione (2.1) risulta problematica in questi casi perchè l'istante di tempo finale sarebbe $T = \infty$, ed il ritorno, che si sta cercando di massimizzare, potrebbe anch'esso essere infinito. Risulta necessario *in questo caso*, quindi, introdurre il concetto di *sconto*. In accordo con tale nuovo approccio, l'agente prova a selezionare azioni in modo che la somma di tutti i guadagni scontati che riceve in futuro venga massimizzata. In particolare, l'agente sceglie l'azione a_t in modo da *massimizzare il ritorno scontato atteso*

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \quad (2.2)$$

dove γ è un parametro, $0 \leq \gamma \leq 1$, chiamato *fattore di sconto*. Il fattore di sconto determina il valore presente dei guadagni futuri: un guadagno ricevuto k passi temporali successivi viene pesato soltanto γ^{k-1} volte di quanto sarebbe pesato se fosse ricevuto immediatamente. Se $\gamma < 1$, la somma infinita ha un valore finito se la sequenza dei guadagni $\{r_k\}$ risulta limitata. Se, invece, $\gamma = 0$, l'agente è "miope" essendo concentrato nel massimizzare soltanto i guadagni immediati: il suo obiettivo in questo caso è quello di imparare come scegliere l'azione a_t in modo da massimizzare r_{t+1} . Se ciascuna delle azioni dell'agente influisce soltanto sul guadagno immediato e in nessuna misura sui guadagni futuri, allora un agente "miope" potrebbe massimizzare (2.2) massimizzando separatamente ogni guadagno immediato. In generale, tuttavia, comportarsi in modo da massimizzare solo il guadagno immediato può ridurre l'accesso a guadagni futuri e in tal modo il ritorno scontato atteso potrebbe risultarne penalizzato. Quando γ si avvicina ad 1, invece, l'obiettivo prende in considerazione i guadagni futuri in modo evidente: l'agente diventa di conseguenza più prudente.

Essere precisi dal punto di vista matematico riguardo ai task episodici richiede qualche raffinamento della notazione usata. Invece che una lunga sequenza di istanti temporali, bisogna

considerare una serie di episodi, ciascuno dei quali consiste in una sequenza finita di unità temporali. Gli istanti temporali di ogni episodio vengono numerati partendo di nuovo da 0. Quindi, c'è bisogno di riferirsi non solo a s_t (lo stato al tempo t) ma a $s_{t,i}$, lo stato al tempo t dell'episodio i (ed, in modo del tutto simile, ad $a_{t,i}$, $r_{t,i}$, $\pi_{t,i}$, T_i , etc.). Comunque, trattando task episodici non verrà praticamente mai fatta distinzione tra diversi episodi. Si considererà quasi sempre un singolo episodio in particolare, o verrà affermato qualcosa che risulta valido per tutti gli episodi: per tali ragioni, si adotterà come notazione s_t per riferirsi a $s_{t,i}$ e così via. Inoltre, i due casi del ritorno atteso definito come una sommatoria su un numero di termini finito (2.1) ed infinito (2.2) possono essere espressi tramite una notazione unificata considerando come termine di un episodio l'entrata in uno stato speciale *assorbente* che permetta transizioni solo verso se stesso e che generi solo guadagni nulli. Si può quindi definire in generale il ritorno, in accordo con (2.2), usando la convenzione di omettere il numero degli episodi quando questo non è esplicitamente richiesto, ed includendo la possibilità che $\gamma = 1$ se la somma rimane finita (ad esempio perchè tutti gli episodi giungono a termine: come nel problema di ottimizzazione affrontato da questa tesi nei Capitoli 3 e 4). Alternativamente, si può esprimere il ricavo nel modo seguente

$$R_t = \sum_{k=0}^T \gamma^k r_{t+1+k} \quad (2.3)$$

includendo la possibilità che $T = \infty$ o che $\gamma = 1$ (ma *non* entrambi contemporaneamente).

2.2 MDP

La proprietà di Markov: nel contesto del RL, l'agente prende le proprie decisioni in funzione di un segnale proveniente dall'ambiente chiamato *stato*, dove con stato si intende qualsiasi informazione che sia disponibile all'agente; si assume che lo stato venga fornito da un qualche sistema con funzione di *pre-processing* che sia nominalmente parte dell'ambiente. Idealmente, quello che si vorrebbe è un segnale di stato che riassume tutti gli avvenimenti del passato in modo compatto, ma in maniera tale che tutta l'informazione rilevante sia preservata. Questo normalmente richiede più che le "sensazioni" immediate, ma mai di più di tutta la storia completa delle "sensazioni" passate. Un segnale di stato che riesca a mantenere tutta l'informazione rilevante si definisce essere di Markov, o in possesso della proprietà di Markov. Ciò può essere equivalentemente visto come una sorta di proprietà di *indipendenza dal percorso*, perchè tutto quello che conta si trova nel segnale di stato corrente: il suo significato è indipendente dal percorso, dalla storia, o dai segnali che hanno condotto ad esso. Per definire formalmente la proprietà di Markov per il problema del RL, si assuma che ci sia un numero finito di stati e di valori di guadagno. Si consideri quindi come un ambiente generico possa rispondere al tempo $t+1$ ad un'azione intrapresa al tempo t . Nel caso (causale) più generale tale risposta può dipendere da qualunque cosa sia accaduta precedentemente: in tale situazione la dinamica del sistema non può che essere definite specificando la distribuzione di probabilità completa

$$\Pr \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} \quad (2.4)$$

per $\forall s', r$, e per tutti i possibili valori degli eventi passati: $s_t, a_t, r_t, \dots, r_1, s_0, a_0$. Se il segnale di stato possiede la proprietà di Markov, tuttavia, la risposta dell'ambiente al tempo $t+1$ dipende solo dalle rappresentazioni dello stato e dell'azione al tempo t , ed in tal caso la dinamica dell'ambiente si può definire specificando soltanto

$$\Pr \{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} \quad (2.5)$$

per $\forall s', r, s_t, a_t$. In altre parole, un segnale di stato possiede la proprietà di Markov, ed è *uno stato di Markov*, se e solo se (2.5) è equivalente a (2.4) per $\forall s', r$ e “storie”, $s_t, a_t, r_t, \dots, r_1, s_0, a_0$. In questo caso, l’ambiente ed il *task* nel loro complesso sono detti avere la *proprietà di Markov*.

Se un ambiente possiede la proprietà di Markov, allora la sua dinamica ad un passo (2.5) consente di predire lo stato ed il guadagno totale atteso complessivi dati lo stato e l’azione corrente. Si può dimostrare che, iterando questa equazione, si possono predire tutti gli stati e guadagni attesi futuri dalla conoscenza dello stato corrente soltanto, allo stesso modo in cui sarebbe possibile farlo conoscendo la *storia completa* del sistema fino al tempo corrente. In questo modo la strategia migliore per scegliere le azioni in funzione dello stato di Markov è altrettanto valida della strategia migliore per selezionare le azioni in funzione delle storie complete. Anche quando il segnale di stato *non* è di Markov, risulta ancora appropriato pensare allo stato nel RL come un’approssimazione di uno stato di Markov. In particolare, si vuole sempre che lo stato sia una buona base per poter predire i guadagni futuri e selezionare le azioni. Può quindi essere utile pensare allo stato ad ogni istante temporale come un’approssimazione di uno stato di Markov, pur ricordando che esso potrebbe non soddisfare la proprietà di Markov. La proprietà di Markov è molto importante nel RL perchè si assume che decisioni e valori siano *funzioni soltanto dello stato corrente*: affinché questi possano essere efficaci nel trasportare la corretta informazione, la rappresentazione dello stato deve essere sufficientemente accurata. La teoria sviluppata per il caso di Markov, tuttavia, si rende utile nella comprensione degli algoritmi che verranno proposti, e tali algoritmi possono essere anche applicati con successo a molti task con stati che non siano strettamente di Markov.

Un task di RL che soddisfi la proprietà di Markov viene chiamato Markov Decision Process (MDP). Se poi lo spazio degli stati e quello delle azioni sono finiti, viene allora chiamato *finite MDP*. I MDP finiti sono particolarmente importanti per la teoria del RL. Un particolare MDP finito viene definito *completamente* dai suoi insiemi degli stati e delle azioni e dalla dinamica dell’ambiente ad un passo. Dati qualunque stato ed azione, s ed a , la probabilità di ogni possibile stato successivo, s' , è data da

$$\mathcal{P}_{s s'}^a = \Pr \{s_{t+1} = s' \mid s_t = s, a_t = a\}. \quad (2.6)$$

Tali quantità sono chiamate *probabilità di transizione*. Allo stesso modo, da qualsiasi stato corrente ed azione, s ed a , insieme a qualsiasi stato successivo, s' , l’aspettazione del valore del guadagno atteso si può esprimere come

$$\mathcal{R}_{s s'}^a = \mathbb{E} \{r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s'\}. \quad (2.7)$$

Le quantità $\mathcal{P}_{s s'}^a$ e $\mathcal{R}_{s s'}^a$ specificano completamente gli aspetti più importanti della dinamica di un MDP finito (viene persa soltanto l’informazione sulla distribuzione dei guadagni attorno al loro valore medio).

2.2.1 Value Functions

Praticamente tutti gli algoritmi di RL sono basati sulla stima delle *value functions* - funzioni dello stato (o della coppia azione-stato) - che cercano di rappresentare “quanto positivo” è per l’agente trovarsi in un certo stato (od eseguire una certa azione a partire da un certo stato). La nozione di “quanto positivo” è qui definita in termini di guadagni futuri che possono essere attesi, o, per essere precisi, in termini di ritorno medio. Ovviamente i guadagni che l’agente si può aspettare di ricevere in futuro dipendono da quali azioni adotterà: conformemente, le *value functions* sono definite rispetto a particolari strategie. Si ricordi che una strategia, π , è una mappa da ogni stato, $s \in \mathcal{S}$, ed azione, $a \in \mathcal{A}(s)$, alla probabilità $\pi(s, a)$ di intraprendere

l'azione a quando ci si trova nello stato s . In modo non formale, il *valore* di uno stato s seguendo una certa strategia π , indicato come $V^\pi(s)$, è il *ritorno medio* partendo dallo stato s e seguendo la strategia π da lì in avanti. Per i MDP, $V^\pi(s)$ può essere definita formalmente come

$$V^\pi(s) = \mathbb{E}_\pi \{R_t | s_t = s\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} | s_t = s \right\} \quad (2.8)$$

dove $\mathbb{E}_\pi \{\cdot\}$ indica il valore atteso dato che l'agente segue la strategia π . Si noti che il valore dello stato terminale, se presente, è sempre zero. La funzione V^π viene chiamata *state-value function* per la strategia π .

In modo simile, si definisce il valore derivante dall'intraprendere l'azione a nello stato s seguendo la strategia π , rappresentato da $Q^\pi(s, a)$, come il *ritorno medio* partendo dallo stato s , intraprendendo l'azione a , e seguendo la strategia π da lì in avanti

$$Q^\pi(s, a) = \mathbb{E}_\pi \{R_t | s_t = s, a_t = a\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} | s_t = s, a_t = a \right\}. \quad (2.9)$$

La funzione Q^π viene chiamata *action-value function* per la strategia π .

Le *value functions* V^π e Q^π possono essere stimate dall'esperienza. Per esempio, se un agente segue una strategia π e mantiene una media, per ogni stato incontrato, dei ritorni effettivi che hanno seguito tale stato, allora la media convergerà al valore dello stato, $V^\pi(s)$, non appena il numero di volte che lo stato è incontrato tende ad infinito. Se medie *separate* sono memorizzate per ogni azione intraprendibile a partire da un certo stato, queste convergeranno, in modo simile, agli *action values* $Q^\pi(s, a)$. I metodi di stima di questo tipo vengono classificati come *metodi di Monte Carlo*, dal momento che coinvolgono un'operazione di media su campioni random di ritorni effettivi. Ovviamente, se ci sono davvero molti stati, allora potrebbe non essere molto pratico mantenere medie separate per ogni stato individualmente. L'agente dovrebbe, invece, mantenere traccia di V^π e Q^π come funzioni parametrizzate e aggiustare i parametri per matchare nel modo più accurato possibile i ritorni osservati.

Una proprietà fondamentale delle *value functions* usata in ogni parte del RL e della DP è che esse soddisfano particolari relazioni ricorsive. Per qualsiasi strategia π e per qualsiasi stato s , vale la seguente *condizione di consistenza* tra il valore di s e quello di tutti i possibili stati successivi

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \{R_t | s_t = s\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} | s_t = s \right\} = E_\pi \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+2+k} | s_t = s \right\} \\ &= \sum_a \pi(s, a) \sum_{s'} \mathcal{P}_{s s'}^a \left\{ \mathcal{R}_{s s'}^a + \gamma \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+2+k} | s_{t+1} = s' \right\} \right\} \\ &= \sum_a \pi(s, a) \sum_{s'} \mathcal{P}_{s s'}^a [\mathcal{R}_{s s'}^a + \gamma V^\pi(s')] \end{aligned} \quad (2.10)$$

dove è implicito che le azioni, a , siano scelte dall'insieme $\mathcal{A}(s)$, e che gli stati successivi, s' , siano presi dall'insieme \mathcal{S} , o da \mathcal{S}' nel caso di un problema episodico. L'equazione (2.10) è chiamata *equazione di Bellmann* per V^π : essa esprime la relazione tra il valore di uno stato e i valori dei suoi stati successivi. Si pensi di guardare in avanti da uno stato verso i possibili successori, come suggerito in Figura 2.2(a). Ogni cerchio vuoto rappresenta uno stato e ogni circoletto colorato di nero rappresenta una possibile coppia azione-stato. Partendo dallo stato s , la radice dell'albero

in alto, l'agente può intraprendere qualsiasi azione dall'insieme di quelle ammissibili per tale stato (in figura ne sono rappresentate tre). Da ciascuna di esse, l'ambiente può rispondere con uno tra diversi stati, s' , insieme ad un guadagno r . L'equazione di Bellman (2.10) media tra tutte le possibilità, pesando ciascuna per la propria probabilità di occorrenza: afferma cioè che *il valore dello stato di partenza deve essere uguale al valore medio (scontato) dello stato successivo, più il guadagno atteso lungo la via*.

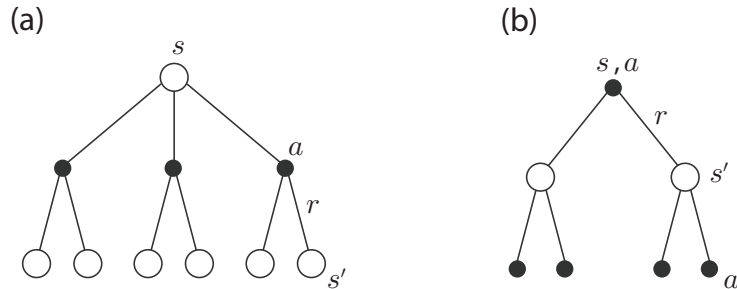


Figura 2.2: Diagrammi di backup per (a) V^π e (b) Q^π .

La *value function* V^π risulta l'unica soluzione dell'equazione di Bellman, che si può dimostrare essere anche alla base di un numero di modi di calcolare, approssimare ed apprendere V^π . I diagrammi come quelli mostrati in Figura 2.2 sono chiamati diagrammi di backup perchè rappresentano le relazioni che sono alla base delle operazioni di aggiornamento o backup che sono il cuore dei metodi di RL. Queste operazioni trasferiscono l'informazione sul valore *indietro* verso un certo stato (o coppia stato-azione) a partire dai suoi stati successivi (o coppie stato-azione): forniscono quindi una rappresentazione grafica immediata dei vari algoritmi usati per sfruttare le equazioni di Bellman (si noti che, a differenza dei grafi di transizione, i nodi di stato dei diagrammi di backup non rappresentano necessariamente stati distinti; per esempio, uno stato può benissimo essere un suo successore; vengono, inoltre, omesse le punte delle frecce perchè il tempo in un diagramma di backup scorre sempre verso il basso).

2.2.2 Optimal Value Functions

Risolvere un problema di RL significa, in parole povere, trovare una strategia che garantisca il maggior guadagno possibile sul lungo periodo. Per MDP finiti, si possono definire strategie ottime in modo analiticamente preciso nella maniera seguente. Le *value functions* stabiliscono un ordine parziale tra le strategie. Una strategia π si definisce migliore od uguale ad un'altra strategia π' se il suo ritorno medio risulta maggiore od uguale a quello di π' per tutti gli stati: ovvero, $\pi \geq \pi'$ se e solo se $V^\pi(s) \geq V^{\pi'}(s) \forall s \in \mathcal{S}$. C'è sempre *almeno una* strategia che è migliore od uguale di tutte le altre: essa costituisce *una strategia ottima*. Sebbene ce ne possano essere più di una, tutte le strategie ottime saranno indicate con π^* : esse condividono la stessa *state-value function*, chiamata *optimal state-value function*, indicata con V^* e definita come

$$V^*(s) = \max_{\pi} V^\pi(s) \quad \forall s \in \mathcal{S}. \quad (2.11)$$

Le strategie ottime condividono anche la stessa *optimal action-value function*, indicata con Q^* e definita come

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s). \quad (2.12)$$

Per la coppia stato-azione (s, a) , questa funzione dà il ritorno medio intraprendendo l'azione a a partire dallo stato s e seguendo da lì in avanti la strategia ottima. Quindi Q^* può essere espressa

in termini di V^* nel modo seguente

$$Q^*(s, a) = \mathbb{E} \{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a \}. \quad (2.13)$$

Inoltre, dato che V^* è la *value function* per una certa strategia, essa deve soddisfare la condizione di auto-consistenza che è data dall'equazione di Bellman (2.10) per i valori degli stati. Dato che si tratta della *optimal value function*, tuttavia, la condizione di consistenza per V^* può essere scritta in una forma particolare, senza alcun riferimento ad una specifica strategia: si tratta dell'equazione di Bellman per V^* , o della *equazione di ottimalità di Bellman*. Intuitivamente, l'equazione di ottimalità di Bellman esprime il fatto che *il valore di uno stato seguendo una strategia ottima deve essere uguale al ritorno medio per l'azione migliore intraprendibile dallo stato in questione*, ovvero

$$\begin{aligned} V^*(s) &= \max_{a \in \mathcal{A}(s)} Q^{\pi^*}(s, a) = \max_a \mathbb{E}_{\pi^*} \{ R_t \mid s_t = s, a_t = a \} \\ &= \max_a \mathbb{E}_{\pi^*} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \mid s_t = s, a_t = a \right\} \\ &= \max_a \mathbb{E}_{\pi^*} \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+2+k} \mid s_t = s, a_t = a \right\} \\ &= \max_a \mathbb{E} \{ r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a \} \\ &= \max_a \sum_{s'} \mathcal{P}_{s s'}^a [\mathcal{R}_{s s'}^a + \gamma V^*(s')]. \end{aligned} \quad (2.14)$$

Le ultime due equazioni sono due forme della equazione di ottimalità di Bellman per V^* . L'equazione di ottimalità di Bellman per Q^* è data invece da

$$\begin{aligned} Q^*(s, a) &= \mathbb{E} \left\{ r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right\} \\ &= \sum_{s'} \mathcal{P}_{s s'}^a \left[\mathcal{R}_{s s'}^a + \gamma \max_{a'} Q^*(s', a') \right]. \end{aligned} \quad (2.15)$$

I diagrammi di backup rappresentati in Figura 2.3 mostrano graficamente il range di stati ed azioni futuri considerati nelle equazioni di ottimalità di Bellman per V^* e Q^* . Essi sono uguali ai diagrammi di backup di V^π e Q^π ad eccezione del fatto che sono stati aggiunti degli archi nei punti di decisione dell'agente, per rappresentare che viene eseguita un'operazione di massimo rispetto a tale scelta *invece* che un valore atteso data una certa strategia. La Figura 2.3(a), in particolare, rappresenta graficamente l'equazione di ottimalità di Bellman (2.14).

Per MDP finiti, l'equazione di ottimalità di Bellman (2.14) ha una *soluzione unica indipendentemente dalla strategia*. L'equazione di ottimalità di Bellman è, a dire il vero, un sistema di equazioni, una per ciascuno stato: se ci sono N stati, ci saranno di conseguenza N equazioni in N incognite. Se la dinamica del sistema è conosciuta (in termini di $\mathcal{R}_{s s'}^a$ e $\mathcal{P}_{s s'}^a$), allora, almeno in linea di principio, questo sistema di equazioni si può esplicitare in funzione di V^* usando uno dei metodi tra la varietà di quelli a disposizione per la risoluzione dei sistemi di equazioni non lineari; si può anche risolvere il corrispondente sistema di equazioni per Q^* . Una volta che si ha a disposizione V^* , è relativamente semplice determinare *una* strategia ottima. Per ogni stato s , ci saranno una o più azioni in corrispondenza delle quali viene raggiunto il massimo nelle equazioni di ottimalità di Bellman: qualsiasi strategia che assegni probabilità non-nulla solo a questo insieme di azioni risulta una strategia ottima. Si può pensare a quanto appena descritto

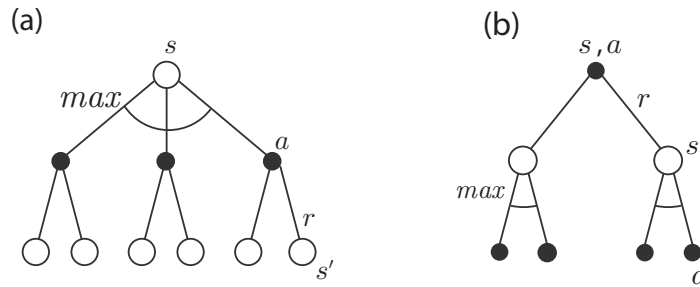


Figura 2.3: Diagrammi di backup per (a) V^* e (b) Q^* .

come ad una ricerca *a singolo step*. Avendo quindi a disposizione la *optimal value function*, V^* , allora le azioni che appaiono migliori dopo una ricerca ad un singolo passo risulteranno le azioni ottime. Un'altra maniera di ribadire lo stesso concetto è affermare che qualsiasi strategia che sia greedy rispetto alla *optimal value function* V^* è una strategia ottima. Il termine greedy viene usato nel mondo della *Computer Science* per descrivere qualsiasi ricerca o procedura decisionale che selezioni alternative basandosi soltanto su considerazioni locali o comunque immediate, senza considerare la possibilità che tale selezione possa compromettere l'accesso futuro ad alternative migliori. Conseguentemente, una procedura greedy corrisponde a strategie che selezionino azioni basandosi solo sulle loro conseguenze a breve termine. La "bellezza" di V^* è che se usata per valutare le conseguenze a breve termine delle azioni - in particolare, le conseguenze ad un solo passo - allora essa risulta *ottima sul lungo periodo* (scopo a cui si è in realtà interessati), e questo perchè V^* tiene già in considerazione le conseguenze in termini di guadagno di tutti i possibili comportamenti futuri. Attraverso la conoscenza di V^* , cioè, il ritorno medio sul lungo termine è ricondotto ad una quantità che è localmente ed immediatamente disponibile per ciascuno stato: una procedura di ricerca a singolo passo corrisponde quindi alle azioni ottime sul lungo periodo. Avere a disposizione Q^* rende la scelta delle azioni ottime ancora più semplice: all'agente non serve nemmeno effettuare una ricerca ad un passo in avanti. Per qualunque stato s , esso può semplicemente trovare qualsiasi azione che massimizzi la quantità $Q(s, a)$. La *action-value function* cattura in effetti i risultati di tutte le procedure di ricerca ad un solo step e fornisce il ritorno medio ottimo (nel senso di maggiore possibile) sul lungo periodo come un valore che è localmente ed immediatamente disponibile per ciascuna coppia stato-azione. Quindi, al costo di rappresentare una funzione di coppie stato-azione, invece che soltanto di stati, la *optimal action-value function* consente che le azioni ottime vengano selezionate senza sapere niente dei possibili stati successivi nè dei loro valori, ovvero, senza conoscere nulla riguardo alla dinamica dell'ambiente.

2.3 DP

Il termine Dynamic Programming (DP) si riferisce ad una collezione di algoritmi che possono essere usati per calcolare strategie ottime *dato un modello perfetto dell'ambiente*, come un MDP. Si assume quindi che gli insiemi degli stati e delle azioni, \mathcal{S} e $\mathcal{A}(s)$, siano finiti e che la dinamica del sistema sia espressa tramite un insieme di probabilità di transizione, $\mathcal{P}_{s s'}^a = \Pr \{s_{t+1} = s' \mid s_t = s, a_t = a\}$, e di guadagni medi immediati, $\mathcal{R}_{s s'}^a = \mathbb{E} \{r_{t+1} \mid a_t = a, s_t = s, s_{t+1} = s'\} \forall s \in \mathcal{S} \forall a \in \mathcal{A}(s) \text{ e } s' \in \mathcal{S}^+$ (dove si ricordi che \mathcal{S}^+ è costituito da \mathcal{S} più uno stato terminale nel caso che il problema sia episodico). L'idea chiave della DP, e del RL in generale, è l'uso delle *value functions* per organizzare e strutturare la ricerca di buone strategie: di seguito verrà illustrato come utilizzare la

DP per *calcolare* le *value functions* definite nella Sezione 2.2.2. Come discusso precedentemente, si possono ottenere facilmente le strategie ottime una volta che si hanno a disposizione le *optimal value functions* V^* o $Q^* \forall s \in \mathcal{S} \forall a \in \mathcal{A}(s)$ e $s' \in \mathcal{S}^+$: gli algoritmi di DP si ottengono trasformando le equazioni di Bellman in assegnamenti, ovvero, in *regole di aggiornamento* per migliorare l'approssimazione delle *value functions* desiderate.

2.3.1 Policy Evaluation

Si consideri prima come calcolare la *state-value function* V^π per un'arbitraria strategia π : ciò viene indicato come *policy evaluation* nella letteratura della DP; ci si può riferire ad esso anche come *problema della predizione*. Ricordando l'equazione (2.14) della Sezione 2.2.2, si ha che per $\forall s \in \mathcal{S}$

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \{ r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid s_t = s \} \\ &= \mathbb{E}_\pi \{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s \} \\ &= \sum_a \pi(s, a) \sum_{s'} \mathcal{P}_{s, s'}^a [\mathcal{R}_{s, s'}^a + \gamma V^\pi(s')] \end{aligned} \quad (2.16)$$

dove $\pi(s, a)$ è la probabilità di intraprendere l'azione a a partire dallo stato s seguendo la strategia π , e le aspettative hanno come pedice π per indicare che sono condizionate al fatto che venga seguita la strategia π . L'esistenza e l'unicità di V^π sono garantite non appena o $\gamma \leq 1$ o la terminazione finale è assicurata a partire da tutti gli stati seguendo la strategia π .

Se la dinamica del sistema è completamente conosciuta, allora (2.16) è un sistema di $|\mathcal{S}|$ equazioni lineari simultanee in $|\mathcal{S}|$ incognite (le $V^\pi(s)$, $s \in \mathcal{S}$): in linea di principio la soluzione è un calcolo abbastanza semplice, anche se tedioso. Per gli scopi della seguente trattazione, sono più adatti *metodi di soluzione iterativi*. Si consideri una sequenza di *value functions* approssimate $\{V_0, V_1, V_2, \dots\}$, ciascuna che mappi \mathcal{S}^+ in \mathcal{R} . L'approssimazione iniziale, V_0 , è scelta arbitrariamente (eccetto che lo stato finale, se presente, al quale deve essere assegnato valore 0) ed ogni approssimazione successiva è ottenuta usando l'equazione di Bellman (2.16) per V^π come regola di aggiornamento

$$V_{k+1}(s) = \mathbb{E}_\pi \{ r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s \} = \sum_a \pi(s, a) \sum_{s'} \mathcal{P}_{s, s'}^a [\mathcal{R}_{s, s'}^a + \gamma V_k(s')] \quad (2.17)$$

per $\forall s \in \mathcal{S}$. Ovviamente, $V_k = V^\pi$ è *un punto fisso* per questa regola di aggiornamento perchè l'equazione di Bellman per V^π assicura in questo caso l'uguaglianza. Per essere più precisi, in generale si può dimostrare che la sequenza $\{V_k\}$ converge a V^π per $k \rightarrow \infty$ sotto le stesse condizioni che garantiscono l'esistenza di V^π : questo algoritmo viene chiamato *iterative policy evaluation*. Per ricavare l'approssimazione successiva, V_{k+1} , a partire da V_k , l'*iterative policy evaluation* applica la stessa operazione a ciascun stato s : sostituisce il vecchio valore di s con un nuovo valore ottenuto dai vecchi valori degli stati successivi di s , ed i guadagni immediati attesi, seguendo tutte le possibili transizioni ad un passo rispetto alla strategia che sta venendo valutata; questo tipo di operazione sarà chiamato *full backup*. Ogni iterazione della *iterative policy evaluation* salva subito il valore di ogni stato per produrre la nuova approssimazione della *value function* V_{k+1} .

Per scrivere un programma sequenziale al computer che implementi la *iterative policy evaluation*, come espresso da (2.17), c'è la necessità di usare due array, uno per i valori vecchi, $V_k(s)$, ed uno per i valori nuovi, $V_{k+1}(s)$: in questo modo, i valori nuovi possono essere calcolati ad uno ad uno a partire da quelli vecchi senza che questi ultimi vengano sovrascritti. È chiaro che sarebbe più comodo usare un solo array ed aggiornare i valori *in-place*, ovvero, con ogni

nuovo valore salvato che va a sovrascrivere immediatamente quello precedente. In tal modo, a seconda dell'ordine con cui i valori degli stati vengono salvati, a destra del primo segno di uguale dell'equazione (2.17) a volte verrebbero usati i nuovi valori invece che quelli vecchi. Tale algoritmo leggermente modificato converge anch'esso a V^π : a dire il vero, generalmente *converge più velocemente della versione con due array*, dato che usa i nuovi dati non appena diventano disponibili. Per l'algoritmo *in-place*, inoltre, l'ordine con cui i valori degli stati vengono salvati ha una significativa influenza sul rate di convergenza: in genere è ad esso che ci si riferisce quando si citano algoritmi di DP.

Tabella 2.1: *Iterative policy evaluation.*

<ul style="list-style-type: none"> • Ingresso: π, la strategia da valutare • Inizializzazione: $V(s) = 0$, per $\forall s \in \mathcal{S}$ • Si ripeta <ul style="list-style-type: none"> – $\Delta \leftarrow 0$ – Per $\forall s \in \mathcal{S}$ <ul style="list-style-type: none"> (a) $v \leftarrow V(s)$ (b) $V(s) \leftarrow \sum_a \pi(s, a) \sum_{s'} \mathcal{P}_{s, s'}^a [\mathcal{R}_{s, s'}^a + \gamma V(s')]$ (c) $\Delta \leftarrow \max(\Delta, v - V(s))$ – Finchè $\Delta < \theta$ (un numero positivo piccolo) • Uscita: $V \approx V^\pi$
--

Un altro punto critico dell'implementazione riguarda la fase di terminazione dell'algoritmo. Formalmente, la *iterative policy evaluation* converge solo al limite, ma in pratica deve essere fermata prima di questo. Una tipica condizione di stop per la *iterative policy evaluation* è calcolare la quantità $\max_{s \in \mathcal{S}} |V_{k+1}(s) - V_k(s)|$ dopo ogni operazione di salvataggio di tutti i valori degli stati e fermarsi quando tale quantità è sufficientemente piccola: la Tabella 2.1 descrive l'algoritmo completo per la *policy evaluation* con questo criterio di stop.

2.3.2 Policy Improvement

La ragione per cui si calcola la *value function* di una strategia è per cercare di trovarne di migliori. Si supponga di aver determinato la *value function* V^π per un'arbitraria strategia deterministica π . Per qualche stato s si vorrebbe sapere se sia o meno opportuno cambiare strategia per scegliere deterministicamente un'azione $a \neq \pi(s)$. Si conosce “quanto va bene” seguire la strategia corrente a partire dallo stato s - cioè $V^\pi(s)$ - ma non se si andrebbe meglio o peggio adottando la nuova strategia. Una maniera di rispondere alla domanda è di considerare la scelta dell'azione a in s e da lì in avanti seguire la strategia π esistente. Il valore di questo comportamento è dato da

$$Q^\pi(s, a) = \mathbb{E}_\pi \{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a\} = \sum_{s'} \mathcal{P}_{s, s'}^a [\mathcal{R}_{s, s'}^a + \gamma V^\pi(s')]. \quad (2.18)$$

Il *criterio chiave* è se la quantità appena definita è più piccola o più grande di $V^\pi(s)$. Se è più grande - ovvero, se è meglio selezionare l'azione a una volta che ci si trova nello stato s e da lì in poi seguire π piuttosto che seguire *sempre* π - ci si aspetta che sia ancora meglio selezionare a *ogni volta* che si incontra lo stato s e che la nuova strategia così definita sia globalmente migliore. Che questo sia vero è un caso speciale di un risultato generale chiamato *policy improvement theorem*. Siano π e π' una qualsiasi coppia di strategie deterministiche tali per cui $\forall s \in \mathcal{S}$ si abbia che

$$Q^\pi(s, \pi'(s)) \geq V^\pi(s). \quad (2.19)$$

Allora la strategia π' deve essere migliore od uguale a π : ovvero, deve portare ad un ritorno medio maggiore od uguale a partire da qualunque stato $\forall s \in \mathcal{S}$

$$V^{\pi'}(s) \geq V^\pi(s). \quad (2.20)$$

Inoltre, se per qualche stato sussiste una stretta disuguaglianza nella (2.19), allora ci deve essere una stretta disuguaglianza anche nella (2.20) per uno o più stati. Questo risultato si applica in particolare alle due strategie appena prese in considerazione: una strategia originale deterministica, π , ed una modificata, π' , identica a π eccetto che $\pi'(s) = a \neq \pi(s)$. Ovviamente, (2.19) vale anche per tutti gli altri stati $\neq s$. Quindi, se $Q^\pi(s, a) > V^\pi(s)$, allora la strategia modificata risulta in realtà *migliore* di π .

L'*idea* che sta alla base della *prova* del *policy improvement theorem* è facile da capire. A partire da (2.19), si continua ad espandere il "lato" Q^π usando (2.18) e si riapplica (2.19) finchè non si ottiene $V^{\pi'}(s)$

$$\begin{aligned} V^\pi(s) &\leq Q^\pi(s, \pi'(s)) \\ &= \mathbb{E}_{\pi'} \{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\} \\ &\leq \mathbb{E}_{\pi'} \{r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi'(s_{t+1})) \mid s_t = s\} \\ &= \mathbb{E}_{\pi'} \{r_{t+1} + \gamma \mathbb{E}_{\pi'} \{r_{t+2} + \gamma V^\pi(s_{t+2})\} \mid s_t = s\} \\ &= \mathbb{E}_{\pi'} \{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) \mid s_t = s\} \\ &\leq \mathbb{E}_{\pi'} \{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V^\pi(s_{t+3}) \mid s_t = s\} \\ &\vdots \\ &\leq \mathbb{E}_{\pi'} \{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots \mid s_t = s\} \\ &= V^{\pi'}(s). \end{aligned} \quad (2.21)$$

Fino a questo momento si è visto come, data una strategia e la sua *value function*, si possa facilmente valutare un cambiamento della strategia in un singolo stato rispetto ad una particolare azione. È un'estensione naturale considerare cambiamenti in tutti gli stati rispetto a tutte le azioni possibili, selezionando in ogni stato l'azione che risulta essere la migliore rispetto a $Q^\pi(s, a)$. in altre parole, considerare la nuova strategia greedy, π' , data da

$$\begin{aligned} \pi'(s) &= \arg \max_a Q^\pi(s, a) \\ &= \arg \max_a \mathbb{E} \{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a\} \\ &= \arg \max_a \sum_{s'} \mathcal{P}_{s s'}^a [\mathcal{R}_{s, s'}^a + \gamma V^\pi(s')]. \end{aligned} \quad (2.22)$$

La strategia greedy seleziona l'azione che risulta la migliore sul breve periodo - dopo uno step in avanti - in accordo con V^π ; per costruzione, inoltre, soddisfa le condizioni del *policy improvement*

theorem (2.19), quindi si sa che essa è migliore o uguale alla strategia originale. Il processo di costruzione di una nuova strategia che migliora una certa strategia iniziale, rendendola greedy o “quasi” greedy rispetto alla *value function* della strategia originale, è chiamato *policy improvement*.

Si supponga che la nuova strategia greedy, π' , sia altrettanto valida, ma non migliore, della vecchia strategia π : ma allora da (2.22) $V^\pi = V^{\pi'}$ e segue che $\forall s \in \mathcal{S}$

$$V^{\pi'}(s) = \max_a \mathbb{E} \left\{ r_{t+1} + \gamma V^{\pi'}(s_{t+1}) \mid s_t = s, a_t = a \right\} = \max_a \sum_{s'} \mathcal{P}_{s,s'}^a [\mathcal{R}_{s,s'}^a + \gamma V^\pi(s')]. \quad (2.23)$$

D'altronde tale equazione è la stessa di quella di ottimalità di Bellman e, quindi, $V^{\pi'}$ deve coincidere con V^* e sia π che π' devono costituire *strategie ottime*. Il *policy improvement*, in sostanza, fornisce sempre una strategia strettamente migliore *eccetto* quando la strategia originale è già ottima.

Fino ad ora è stato considerato solo il caso speciale di strategie deterministiche. Nel caso generale, una strategia stocastica π specifica delle probabilità, $\pi(s, a)$, di intraprendere una certa azione a al variare dello stato s ; senza andare nel dettaglio, tutte le idee appena descritte si estendono perfettamente al caso aleatorio. In particolare, continua a valere il *policy improvement theorem*, che per il caso stocastico si esprime nel modo seguente

$$Q^\pi(s, \pi'(s)) = \sum_a \pi'(s, a) Q^\pi(s, a). \quad (2.24)$$

Infine, se nei passi di *policy improvement* ci sono più azioni in corrispondenza delle quali viene raggiunto il massimo, nel caso stocastico non serve banalmente selezionare una di esse. Ad ogni azione che porta al massimo, invece, può essere assegnata *una porzione di probabilità* di essere selezionata nella nuova strategia greedy: uno schema di assegnazione rimane ammissibile finché assegna probabilità nulla a tutte le azioni non-massimali.

2.3.3 Policy Iteration

Una volta che una strategia, π , è stata migliorata usando V^π portando ad una strategia migliore, π' , si può allora calcolare $V^{\pi'}$ e migliorarla di nuovo per ottenere una strategia π'' che risulti ancora più performante. Si può ottenere così una sequenza di strategie e *value functions* monotonicamente crescente

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi^* \xrightarrow{E} V^* \quad (2.25)$$

dove \xrightarrow{E} indica una *evaluation* e \xrightarrow{I} un *improvement*. Ogni strategia è garantito che costituisca uno stretto miglioramento rispetto a quella precedente (a meno che non sia già ottima). Dato che un MDP finito ha soltanto un numero finito di strategie, questo processo deve convergere in un numero finito di iterazioni ad una strategia ottima, la *optimal value function*.

Il modo di raggiungere una strategia ottima è chiamato *policy iteration*: l'algoritmo completo è descritto in Tabella 2.2. Si noti che ogni *policy evaluation*, essa stessa una computazione iterativa, comincia dalla *value function* della strategia precedente. Questo tipicamente comporta un significativo incremento della velocità di convergenza della *policy evaluation* (presumibilmente perchè la *value function* cambia di poco da una strategia a quella successiva). La *policy iteration* spesso converge in sorprendentemente poche iterazioni e, nel caso episodico, si procede verso gli stati terminali in un numero di passi molto limitato.

Tabella 2.2: *Policy iteration* (usando la *iterative policy evaluation* per il calcolo di V^* . Nell'“*argmax*” step in 3, si assume che i nodi siano risolti con un ordine consistente.

1. Inizializzazione: $V(s) \in \mathcal{R}$ e $\pi(s) \in \mathcal{A}(s)$ a valori arbitrari per $\forall s \in \mathcal{S}$
2. *Policy evaluation*
 Si ripeta
 - $\Delta \rightarrow 0$
 - Si ripeta per $\forall s \in \mathcal{S}$
 - (a) $v \rightarrow V(s)$
 - (b) $V(s) \rightarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} [\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s')]$
 - (c) $\Delta \rightarrow \max(\Delta, |v - V(s)|)$
3. Finchè $\Delta < \theta$ (un numero positivo piccolo)
4. *Policy Improvement*
 - $policy\text{-}stable \leftarrow true$
 - Per $\forall s \in \mathcal{S}$
 - (a) $b \leftarrow \pi(s)$
 - (b) $\pi(s) \leftarrow \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + \gamma V(s')]$
 - (c) Se $b \neq \pi(s)$, allora $policy\text{-}stable \leftarrow false$
 - Se $policy\text{-}stable$, allora fermati; altrimenti vai al punto 2.

2.3.4 Value Iteration

Uno svantaggio della *policy iteration* è che ciascuna delle sue iterazioni comprende la *policy evaluation*, che potrebbe essere essa stessa una computazione iterativa prolungata che richiede salvataggi multipli nello spazio degli stati. Se la *policy evaluation* viene eseguita iterativamente, allora la convergenza a V^π si ha soltanto al limite: è lecito *chiedersi se serva aspettare l'esatta convergenza*, o ci si possa fermare prima. Lo step di *policy evaluation* della *policy iteration*, infatti, può essere troncato in diverse maniere senza perdere la garanzia di convergenza di quest'ultima. Un caso speciale importante da considerare è quando la *policy evaluation* viene arrestata dopo un solo passo (che corrisponde ad un salvataggio di ogni valore degli stati): questo algoritmo viene chiamato *value iteration*. Può essere espresso come una semplice operazione di backup che combina gli step di *policy improvement* e di *policy evaluation* troncata, ovvero

$$V_{k+1}(s) = \max_a \mathbb{E} \{r_{t+1} + \gamma V_k(s_{t+1} | s_t, a_t = a)\} = \max_a \sum_{s'} \mathcal{P}_{s s'}^a [\mathcal{R}_{s s'}^a + \gamma V_k(s')] \quad (2.26)$$

per $\forall s \in \mathcal{S}$. A partire da una V_0 arbitraria, la sequenza $\{V_k\}$ si può dimostrare convergere a V^* sotto le stesse condizioni che garantiscono l'esistenza di V^* . Un altro modo di comprendere la *value iteration* è facendo riferimento all'equazione di ottimalità di Bellman (2.14): si noti infatti che la *value iteration* è ottenuta semplicemente trasformando l'equazione di ottimalità di Bellman in una regola di aggiornamento. Si noti anche che il backup della *value iteration* è identico a quello della *policy evaluation* (2.17) ad eccezione che richiede che il massimo venga calcolato rispetto a tutte le azioni ammissibili.

Considerando infine come termina la *value iteration*, si può osservare che, come la *policy evaluation*, richiede formalmente un numero infinito di iterazioni per convergere esattamente a V^* : nella pratica, tuttavia, ci si può fermare una volta che la *value function* cambia soltanto di una piccola quantità ad ogni salvataggio. La tabella 2.3 dà una completa descrizione dell'algoritmo *value iteration* implementato con tale condizione di terminazione. La *value iteration* combina effettivamente, ad ogni salvataggio, un passo della procedura di *policy evaluation* ed uno di quella di *policy improvement*: si può raggiungere spesso una più rapida convergenza interponendo più passi di *policy evaluation* tra ciascun operazione di *policy improvement*.

2.3.5 Generalized Policy Iteration

La *policy iteration* consiste di due processi simultanei e che interagiscono tra loro, uno (la *policy evaluation*) che rende la *value function* consistente con la strategia corrente e l'altro (la *policy improvement*) che rende la strategia greedy rispetto alla *value function* corrente. Nella *policy iteration* queste due procedure si alternano, e ciascuna si completa prima che l'altra inizi, ma questo non è veramente necessario. Nella *value iteration*, per esempio, viene eseguita soltanto una singola iterazione della *policy evaluation* tra ciascun *policy improvement*. Si può infatti dimostrare (ma non è affatto banale) che finchè entrambi i processi continuano ad aggiornare tutti gli stati, il risultato finale è sempre lo stesso - convergenza a *value function* e a strategia ottime. Viene usato il termine *generalized policy iteration* (GPI) per riferirsi all'idea generale di processi di *policy evaluation* e *policy improvement* che interagiscono tra loro, indipendentemente dalla granularità e da altri dettagli di tali processi. Praticamente tutti i metodi di RL sono ben descritti come una GPI: ovvero, tutti hanno strategie e *value functions* ben identificabili, con la strategia che viene sempre migliorata rispetto alla *value function* e con quest'ultima che viene sempre guidata nel suo aggiornamento dalla strategia corrente. Questo schema generale è illustrato in Figura 2.4.

Tabella 2.3: Value iteration.

- Inizializzazione di V : arbitraria, ad esempio $V(s) = 0$ per $\forall s \in \mathcal{S}^+$
- Si ripeta
 - $\Delta \leftarrow 0$
 - Per $\forall s \in \mathcal{S}$
 - (a) $v \leftarrow V(s)$
 - (b) $V(s) \leftarrow \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + \gamma V(s')]$
 - (c) $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
 - finchè $\Delta < \theta$ (un numero positivo piccolo)
- Uscita: una strategia deterministica, π , tale che

$$\pi(s) = \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + \gamma V(s')]$$

È facile rendersi conto che se sia i processi di valutazione che di miglioramento si stabilizzano, ovvero, non vengono più prodotti cambiamenti, allora sia la *value function* che la *strategia corrente* devono essere ottime. La *value function* si stabilizza soltanto quando risulta consistente con la strategia corrente, e la strategia si stabilizza soltanto quando è già greedy rispetto alla *value function* corrente. Entrambi i processi, quindi, giungono all'equilibrio soltanto quando è stata trovata una strategia che risulti greedy rispetto alla propria funzione da valutare: questo significa infatti che vale l'equazione di ottimalità di Bellman (2.14) e che quindi la strategia e la *value function* risultano ottime. Le procedure di valutazione e miglioramento della GPI possono essere viste sia in *concorrenza* che in *cooperazione* l'una rispetto all'altra. Sono in competizione nel senso che spingono in direzioni diverse tra loro: rendere la strategia greedy rispetto alla *value function* tipicamente rende quest'ultima non corretta per la strategia modificata, e rendere la *value function* consistente con la strategia fa sì che tipicamente quest'ultima non sia più greedy. Nel lungo periodo, tuttavia, questi due processi interagiscono fino a trovare una singola soluzione congiunta: *value function* e strategia ottime. Si potrebbe pensare anche all'interazione tra i processi di valutazione e miglioramento della GPI in termini di due vincoli od obiettivi - ad esempio, come a due linee in uno spazio bidimensionale, come rappresentato nella Figura 2.5.

Sebbene la reale geometria sia molto più complicata di quella di Figura 2.5, il diagramma fornisce comunque una buona idea di ciò che accade nel caso reale: ogni processo guida la *value function* o la strategia verso una delle linee che rappresentano una soluzione di *uno* dei due obiettivi, i quali interagiscono continuamente tra loro perchè le due linee non sono ortogonali. Andare direttamente verso un obiettivo causerebbe un allontanamento dall'altro obiettivo. Inevitabilmente, comunque, il processo congiunto è portato più vicino all'obiettivo globale dell'ottimalità. Le frecce di tale diagramma corrispondono al comportamento della *policy iteration* nel senso che conducono il sistema lungo tutto il percorso composto del raggiungimento parziale di uno dei due obiettivi. Nella GPI si potrebbe anche compiere passi più piccoli ed incompleti verso ciascun obiettivo: in ogni caso, i due processi insieme raggiungono l'obiettivo globale dell'ottimalità anche se nessuno dei due sta spingendo verso di essa direttamente.

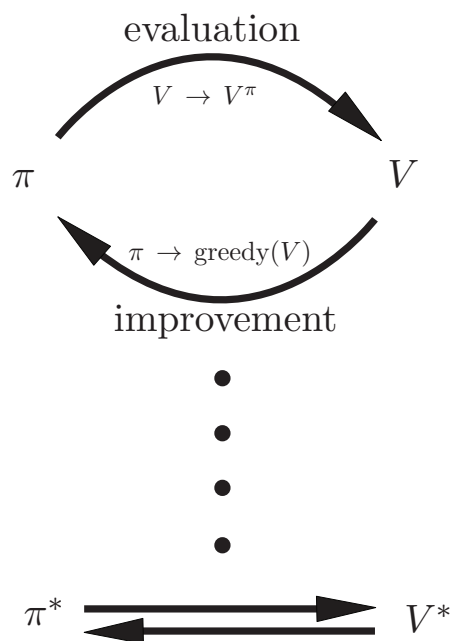


Figura 2.4: *Generalized policy iteration: value function e strategia corrente interagiscono finchè non sono ottime e quindi consistenti l'una rispetto all'altra.*

2.3.6 Efficienza della DP

La DP può non risultare implementabile per problemi di dimensioni molto grandi, ma, rispetto ad altri metodi per risolvere i MDP, la procedura di DP risulta abbastanza efficiente. Ignorando qualche dettaglio tecnico, il tempo che i metodi di DP richiedono per trovare *una* strategia ottima risulta, nel caso peggiore, polinomiale nel numero di stati e di azioni. Se n ed m indicano rispettivamente il numero di stati e di azioni, questo significa che un metodo di DP richiede un numero di operazioni computazionali che è inferiore ad una qualche funzione polinomiale di n ed m . Si può dimostrare essere garantito che un metodo di DP trovi una strategia ottima in tempo polinomiale anche se il numero totale di strategie (deterministiche) è dato da m^n . In questo senso, la DP è esponenzialmente più veloce di qualsiasi ricerca diretta nello spazio degli stati, perchè quest'ultima dovrebbe esaminare in modo *esaustivo* ogni strategia per avere la stessa garanzia di ottimalità. Anche i metodi di programmazione lineare possono essere usati per risolvere i MDP, ed in alcuni casi le loro garanzie di convergenza nel caso peggiore risultano migliori di quelle dei metodi di DP. I metodi di programmazione lineare, tuttavia, diventano impraticabili quando il numero di stati è molto inferiore di quanto accada con la DP (in [16] si parla di un fattore attorno a 100): per i problemi di dimensioni più elevate, quindi, solo le procedure di DP costituiscono una via praticabile.

La DP viene talvolta pensata di modesta applicabilità a causa del *curse of dimensionality* (Bellman, 1957), ovvero il fatto che il numero di stati spesso cresce esponenzialmente con il numero di variabili di stato. Spazi degli stati di dimensioni molto grandi creano sì difficoltà, ma sono difficoltà *intrinseche al problema*, non alla DP come metodo di risoluzione: essa risulta infatti comparativamente più adatta rispetto ai metodi concorrenti (come la ricerca diretta e la programmazione lineare) a gestire grossi spazi degli stati. Nella pratica, i metodi di DP possono

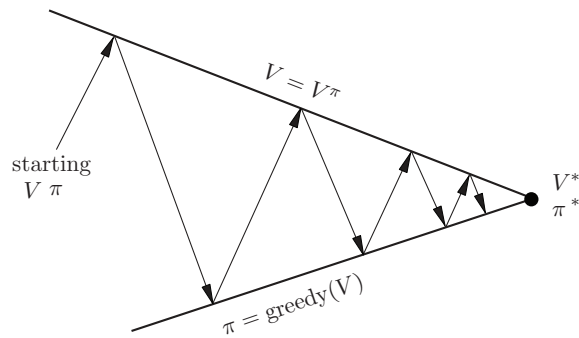


Figura 2.5: Caso bidimensionale dell'interazione continua tra i due processi che costituiscono la GPI.

essere usati con i computer moderni per risolvere MDP con milioni di stati ². Vengono largamente utilizzate sia la *policy iteration* che la *value iteration*, e non è ben chiaro, sempre che sia possibile stabilirlo in modo rigoroso, quale tra le due vada meglio nel caso generale. Dal punto di vista computazionale, inoltre, entrambi i metodi convergono in genere molto più rapidamente del loro tempo di esecuzione nel caso peggiore, in modo particolare se sono inizializzati con buone *value functions* e strategie iniziali.

²Si noti che questa affermazione è riportata, senza essere stata verificata, da un libro del 1998: nel 2012 ne andrebbe sicuramente rivisto l'ordine di grandezza.

Capitolo 3

Soluzione online ottima con una batteria tramite DP

3.1 Definizione del problema di ottimizzazione con un'unica batteria con la notazione del RL

Il modello generale di sistema descritto nel Capitolo 2 è uno strumento molto potente, le cui potenzialità sono sfruttate soltanto in minima parte nella sua applicazione alla risoluzione del problema di ottimizzazione online con una sola batteria definito da (1.13). Per quanto riguarda la definizione s di stato, esso risulta costituito semplicemente dalla coppia

$$\mathbf{s} = (E_j, P_i) = \left(s^{(1)}, s^{(2)} \right) \in \mathcal{S} \quad \text{con } \mathcal{S} = \mathcal{E} \times \mathcal{P}. \quad (3.1)$$

Vale la pena ricordare che $\{E_k\} \in \mathcal{E}^{T-1}$ costituisce la sequenza dei livelli energetici della batteria che si vuole ricavare per minimizzare il costo energetico complessivo che la grid deve pagare. $\{P_k\} \in \mathcal{P}^T$ rappresenta invece la richiesta netta di potenza che giunge al controllore della grid all'inizio di ogni intervallo temporale: il fatto di farla entrare nello stato del sistema come seconda componente costituisce la differenza *fondamentale* rispetto al modello del problema di ottimizzazione offline. Nell'equazione (1.15), infatti, si supponeva di conoscere tale sequenza: con l'aggettivo offline si intendeva la possibilità di aver memorizzato tutti i valori assunti dalla richiesta netta di potenza in un certo intervallo temporale della durata di T campioni e di voler determinare, *a posteriori*, la sequenza dei livelli energetici che la batteria *avrebbe dovuto avere* in corrispondenza ad ogni valore di P_k per poter minimizzare il costo complessivo.

In questo punto della trattazione, invece, l'ottica è completamente *diversa*: si ipotizza di avere periodicamente la possibilità di stimare la statistica di P_k da fornire in input all'agente (usando la notazione del RL) in modo che questo abbia la capacità di calcolare una *strategia ottima dal punto di vista stocastico*, ovvero che se applicata vada a minimizzare *mediamente* il costo complessivo, supponendo ovviamente che la statistica della richiesta netta di potenza sia stazionaria (quando si suppone che essa sia cambiata significativamente è necessario stimarla di nuovo). L'azione a da intraprendere, invece, rappresenta la variazione della quantità di energia immagazzinata nella batteria in ogni intervallo di durata Δt : è definita come $a_k = E_{k+1} - E_k$, per $k = 0, 1, \dots, T-1$ (con $E_T = E_{T-1}$ per convenzione) e deve soddisfare il vincolo definito in (1.30). Una volta definite le possibili azioni intraprendibili a partire da un qualunque stato, la legge di aggiornamento di stato non ha più carattere aleatorio ma deterministico, quindi può essere completamente specificata non dalle probabilità di transizione markoviane (2.6) ma dalla

seguinte equazione di aggiornamento

$$E_{k+1} = E_k + a_k. \quad (3.2)$$

Diventa poi naturale definire non più un guadagno di tipo stocastico come quello riportato in (3.3) ma un costo c *deterministico* (sempre di tipo *istantaneo*) dipendente soltanto dalla seconda componente dello stato di partenza e dall'azione ivi intrapresa

$$c_{k+1} = C(s_k^{(2)} + a_k) = C(P_k + a_k). \quad (3.3)$$

Ci si rende conto della difficoltà di uniformare questo problema di ottimizzazione alla notazione dello schema a blocchi di Figura 2.1: lì infatti l'azione scelta dall'agente, a_t , dipendeva dallo stato s_t e dal guadagno r_t riferiti allo stesso intervallo temporale, ed era poi l'ambiente a fornire in uscita guadagno e stato successivi (r_{t+1} e s_{t+1} , rispettivamente) in base ai tre valori appena menzionati. In questo caso, tuttavia, dato che la funzione del prezzo $C(\cdot)$ dipende non da un certo livello energetico ma dalla differenza tra due livelli di energia consecutivi (e ovviamente dall'intervallo temporale Δt in cui avviene questa variazione, mantenuto sempre fisso ad un'ora) e si suppone che, fissato lo stato s_k , l'istante temporale a cui è associato il "pagamento" sia all'inizio dell'intervallo temporale $k+1$ successivo, è sembrata la scelta migliore definire l'insieme di azioni intraprendibili da un certo stato come le *variazioni di energia tra il livello energetico verso cui si decide di spostarsi e quello di partenza* (con un valore nullo se si reputa la scelta migliore quella di mantenerlo invariato). Per quanto riguarda la strategia ottima che si vuole ricavare per minimizzare la somma dei costi, essa viene ancora indicata con π e risulta costituita dalla sequenza $\{E_k\}$ dei livelli energetici ottimi da mantenere nella batteria: tale sequenza, fissato il livello energetico iniziale E_0 , risulta univocamente legata tramite l'equazione (3.2) a quella delle azioni ottime $\{a_k\}$ che verranno calcolate tramite l'implementazione della procedura di DP.

Con la cornice di cui si è appena discusso, la *state-value function*, fissato lo stato di partenza \mathbf{s}_0 , risulta definita da

$$V^\pi(\mathbf{s}) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{T-1} c_{k+1} \mid \mathbf{s}_0 = \mathbf{s} \right\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{T-1} C(P_k + a_k) \mid \mathbf{s}_0 = \mathbf{s} \right\}. \quad (3.4)$$

Delle due componenti dello stato iniziale, conoscere E_0 significa che, per $\mathbf{s} = \mathbf{s}_0$, soltanto la seconda componente $s_0^{(2)} = P_0 \in \mathcal{P}$ può variare liberamente e non è nota a priori; per tutti gli altri istanti temporali diversi da 0, invece, la prima componente E_k dello stato \mathbf{s}_k verrà fissata dalla DP intraprendendo l'azione $a_{k-1} = E_k - E_{k-1}$, mentre della seconda componente P_k si suppone di avere a disposizione soltanto una stima della statistica. Per quanto riguarda invece la definizione della *action-value function*, sempre per la strategia π , essa si può esprimere come

$$Q^\pi(\mathbf{s}, a) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{T-1} c_{k+1} \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a \right\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{T-1} C(P_k + a_k) \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a \right\} \quad (3.5)$$

al variare delle azioni $a \in \mathcal{A}(\mathbf{s}_0)$ di carica/scarica o di mantenimento del livello energetico a partire da quello iniziale E_0 (si coglie qui l'occasione per osservare che l'insieme delle azioni intraprendibili a partire da un certo stato dipende soltanto dalla sua prima componente, $s^{(1)}$, ovvero dal livello energetico E_j : fino ad ora ciò era stato assunto implicitamente).

Continuando a seguire la linea espositiva del Capitolo 2, per quanto riguarda la *optimal state-value function*, indicata sempre come V^* , essa si può esprimere come

$$V^*(\mathbf{s}) = \min_{\pi} V^\pi(\mathbf{s}) \quad \text{per } \forall \mathbf{s} \in \mathcal{S}. \quad (3.6)$$

L'*optimal action-value function*, indicata sempre come Q^* , è definita e si può esprimere in funzione di V^* rispettivamente come

$$Q^*(\mathbf{s}, a) = \min_{\pi} Q^{\pi}(\mathbf{s}, a) = \mathbb{E} \left\{ c_1 + V^*(\mathbf{s}_1) \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a \right\} \quad \text{per } \forall \mathbf{s} \in \mathcal{S}, \forall a \in \mathcal{A}(\mathbf{s}) \quad (3.7)$$

dove $c_1 = C \left(s_0^{(2)} + a_0 \right) = C \left(P_0 + E_1 - E_0 \right)$. Riscrivendo le due equazioni di ottimalità di Bellman per V^* e per Q^* in una maniera che sia più congeniale all'applicazione della DP per la determinazione della strategia ottima si ha quindi che

$$V^*(\mathbf{s}) = \min_a \mathbb{E} \left\{ c_1 + V^*(\mathbf{s}_1) \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a \right\} = \min_a \sum_{s'} \mathcal{P}_{s^{(2)} s'^{(2)}}^a \left[C \left(s^{(2)} + a \right) + V^*(\mathbf{s}') \right] \quad (3.8)$$

dove $\mathcal{P}_{s^{(2)} s'^{(2)}}^a = \Pr \{ P = s'^{(2)} \mid P = s^{(2)} \}$ sono le probabilità di transizione che risultano indipendenti dall'azione a , per cui si ha che $\mathcal{P}_{s^{(2)} s'^{(2)}}^a = \mathcal{P}_{s^{(2)} s'^{(2)}}$. Per quanto riguarda l'equazione di ottimalità di Bellman per Q^* , si ha analogamente che

$$\begin{aligned} Q^*(\mathbf{s}, a) &= \mathbb{E} \left\{ c_1 + \min_{a'} Q^*(\mathbf{s}_1, a') \mid \mathbf{s}_0 = \mathbf{s}, a_0 = a \right\} \\ &= \sum_{s'} \mathcal{P}_{s^{(2)} s'^{(2)}} \left[C \left(s^{(2)} + a \right) + \min_{a'} Q^*(\mathbf{s}', a') \right]. \end{aligned} \quad (3.9)$$

Equazioni simili alla (3.8) e (3.9) si potrebbero scrivere anche per tutte quelle della DP (*policy evaluation, policy improvement, value iteration* e GPI) applicabili al problema di ottimizzazione con una sola batteria: farlo però sarebbe estremamente tedioso ed inutile. Risulta più interessante, invece, riportare l'idea dell'effettiva implementazione in Matlab che si è usata per risolvere le equazioni di Bellman ricavando quindi la strategia ottima di gestione della batteria. Come si evince dai passaggi nelle espressioni (3.8) e (3.9), le probabilità di transizione tra un generico stato e un altro non dipendono statisticamente dalla prima componente del medesimo ma solo dalla seconda: la richiesta netta di potenza $\{P_k\}$. I dati a disposizione per stimarne la statistica sono rappresentati dal segnale generato col tool descritto nella Sezione 1.5 e riportato nella Figura 1.7. Dai valori che esso assume nei vari istanti temporali (supposti sempre a distanza di un'ora) è stato possibile stimare la distribuzione di probabilità del primo ordine

$$\Pr \{ P = P_i \} \quad \text{per } P_i = P_0, P_1, \dots, P_{M-1} \quad (3.10)$$

sempre con $\Delta P = P_{i+1} - P_i = 0.5$ kW. Per la richiesta netta di potenza generata, risultano essere una scelta accettabile (sulla base del codominio del segnale) come estremi di una quantizzazione uniforme (che è ben distante dall'essere la quantizzazione di $\{P_k\}$ a cui corrisponda il più basso errore di quantizzazione possibile) $P_0 = 0$ kW e $P_{M-1} = P_{124} = 62$ kW. Oltre a quella del primo ordine, è stata stimata la statistica congiunta del secondo ordine del segnale $\{P_k\}$, che si è scelto di memorizzare (in modo insolito, ma conveniente per l'applicazione della DP) in un array bidimensionale con valori corrispondenti a quelli dell'espressione

$$\Pr \{ P_{k+1} = P_i, P_k = P_j \}. \quad (3.11)$$

Dall'equazione (3.11), tramite le regole della probabilità congiunta, è stato possibile ricavare le probabilità di transizione ad un passo - che compaiono in tutte le espressioni della programmazione dinamica

$$\Pr \{ P_{k+1} = P_i \mid P_k = P_j \} = \Pr \{ P_{k+1} = P_i, P_k = P_j \} / \Pr \{ P_k = P_j \} \longleftrightarrow \mathcal{P}_{P_i P_j} \quad (3.12)$$

le quali sono state memorizzate nella matrice \mathbf{P}_{tr} . Avendo a disposizione tali probabilità e usando dei vettori di supporto per memorizzare i valori quantizzati dei livelli energetici E_j (con passo $\Delta E = 0.5$ kWh, come già discusso nel Capitolo 1) della batteria e della richiesta netta di potenza P_i (appena menzionati), si è determinata la *strategia ottima di gestione della batteria* usando la procedura di DP indicata nel Capitolo 2 come *value iteration*, descritta nella Tabella 2.3, e la cui equazioni di aggiornamento fondamentali in questo caso diventano

$$V_{k+1}^{\text{tmp}}(\mathbf{s}) = V_{k+1}^{\text{tmp}}(E_j, P_i) = \min_{a \in \mathcal{A}(E_j)} V_k(E_j + a, P_i) \quad \forall E_j \in \mathcal{E}, \forall P_i \in \mathcal{P} \quad (3.13)$$

dove $V_{k+1}^{\text{tmp}}(\cdot)$ sono degli scalari (che non tengono ancora conto delle probabilità di transizione ad un passo della richiesta di potenza) che necessitano dell'inizializzazione

$$V_0^{\text{tmp}}(E_j, P_i) = 0 \quad \forall E_j \in \mathcal{E}, \forall P_i \in \mathcal{P}. \quad (3.14)$$

L'operazione di pesatura per le probabilità condizionate, può così poi essere implementata in modo efficiente tramite la moltiplicazione matriciale (espressa in forma *Matlab-like*)

$$V_{k+1}(:, :) = V_{k+1}^{\text{tmp}}(:, :) P_{\text{tr}}(:, :). \quad (3.15)$$

La strategia ottima π viene aggiornata (e ne vengono risparmiati i dettagli implementativi) ad ogni iterazione in modo da risultare, ovviamente, greedy rispetto a V_{k+1} . I passi appena descritti vengono ripetuti fino ad arrivare a convergenza: come criterio di stop si è scelto di adottare quello che la norma quadratica sia inferiore ad una certa costante θ (ad esempio, $\theta = 1e - 4$), ovvero

$$\left\| \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} V_{k+1}(E_j, P_i) - V_k(E_j, P_i) \right\|^2 < \theta. \quad (3.16)$$

Nella maggior parte delle simulazioni eseguite sono risultate necessarie soltanto dalle 50 alle 100 iterazioni (circa) per arrivare a convergenza.

3.2 Risultati delle simulazioni numeriche relative alla determinazione di π tramite DP

Si sono condotte quindi una serie di simulazioni, tramite l'ausilio di Matlab, sia per analizzare la *forma* della strategia ottima al variare delle due componenti dello stato (livello di energia immagazzinata nella batteria e richiesta netta di potenza) sia per misurare quantitativamente quanto si distanzia il costo della soluzione ottenuta tramite l'applicazione della strategia ottima fornita dalla DP da quello *minimo possibile* in presenza della batteria nella grid ed in assenza della medesima. Il primo di essi è ovviamente calcolabile tramite l'algoritmo AOS descritto nel Capitolo 1 (applicato alla stessa sequenza $\{P_k\}$), il quale può però solo essere applicato *a posteriori*, dopo cioè che il controllore della grid ha memorizzato l'intera sequenza della richiesta netta di potenza; il secondo, invece, si trova banalmente sommando i valori assunti dalla funzione di costo in funzione della potenza avendo come ingresso la sequenza $\{P_k\}$. Per quanto riguarda invece il costo della soluzione fornita dalla DP, è doverosa una precisazione: avendo a disposizione un numero abbastanza limitato di campioni del segnale P_k , si è deciso di utilizzare *gli stessi campioni* sia per effettuare la stima delle probabilità di transizione e quindi il calcolo della strategia ottima π (fase di *learning*), sia per applicare tale politica di decisione del livello di carica quando al controllore della grid giunge la medesima sequenza $\{P_k\}$. Supponendo che la sorgente delle richieste di potenza abbia una statistica stazionaria durante la generazione dei

campioni prodotti ed analizzati, il modo di procedere appena descritto *non* mette in discussione i risultati dell'applicazione della DP: anche se la sequenza di training per il calcolo della strategia ottima è la stessa su cui poi questa viene mediata, se si va ad osservare il costo *medio* su più realizzazioni delle richieste di potenza relative al mese di gennaio (con granularità dell'ora) non si perde di generalità e non si ottengono *in media* prestazioni migliori rispetto all'applicazione della stessa strategia ad un diverso insieme di realizzazioni.

Chiarito ciò, il setting su cui si sono svolte le simulazioni prevede una batteria di capacità $E_{max} = 12$ kWh con livelli energetici quantizzati a passi di 0.5 kWh (valori fissati in base a quelli scelti nella maggior parte dei paper relativi a problemi analoghi (ad esempio in [1]), rate di carica e scarica normalizzati fissati a $\left\lfloor \frac{E_{max}}{4\Delta E} \right\rfloor$, la sequenza $\{P_k\}$ di richiesta netta di potenza più volte menzionata e una durata degli intervalli temporali di $\Delta t = 1$ h. Come sarà chiarito solo più avanti nella trattazione (in particolare, dopo aver letto la Sezione 3.4), si è deciso di confrontare le prestazioni della politica ottima ottenuta tramite l'applicazione della DP ad un effettivo MDP finito, indicato come Dynamic Programming with Markovian requests (DPM) (completamente descritto dagli insiemi degli stati e delle azioni, dalla funzione di costo $C(\cdot)$ e dalla matrice \mathbf{P}_{tr} che memorizza le probabilità di transizione $\mathcal{P}_{P_i P_j}$) e ad un processo di decisione finito che non sia supposto di Markov ma iid, indicato come Dynamic Programming with Independent request (DPI). Quest'ultimo non è altro che un *caso particolare* di un MDP, definito dagli stessi elementi, ma avente una matrice \mathbf{P}_{tr} con *tutte le colonne uguali*: la probabilità di passare da una certa seconda componente P_i dello stato di partenza ad un'altra P_j dello stato di arrivo è cioè *sempre la stessa indipendentemente dallo stato di partenza* (analiticamente, $\Pr\{P_{k+1} = P_i | P_k = P_j\} = \Pr\{P_{k+1} = P_i\}$). Tale assunzione sembra di primo acchito una forte approssimazione se applicata alla sequenza $\{P_k\}$, ma le simulazioni faranno nascere diversi interrogativi a riguardo: in primo luogo perchè la risoluzione temporale di un minuto con cui vengono effettivamente generati i valori dei campioni della richiesta di potenza è in corrispondenza con una certa descrizione statistica, che potrebbe sì essere meglio catturata da una distribuzione markoviana che da quella iid. Andando poi a mediare, tuttavia, su intervalli temporali della durata 60 volte più elevata (infatti $\Delta t = 1$ h) per non avere problemi di capacità di memorizzazione solo per i campioni delle 6 realizzazioni del mese di gennaio, la descrizione statistica del segnale P_k *non* riesce più ad entrare in corrispondenza con la funzione di correlazione che legava i campioni che, appena generati, distavano temporalmente solo un minuto, e risulta, in alcuni casi, *matchare* meglio quella prima del campionamento se viene supposta come di Markov ed in altri se viene supposta indipendente ed identicamente distribuita.

Nelle Figure 3.1 e 3.2 si è scelto di riportare quindi l'andamento della strategia ottima π , ricavata tramite DP, supponendo prima una distribuzione iid delle probabilità di transizione di $\{P_k\}$, al variare sia del valore della richiesta netta di potenza che del livello di carica presente nella batteria, e poi al variare soltanto di quest'ultimo parametrizzando la curva per alcuni valori di P_i valutati come significativi, scelti cioè col criterio di poter osservare la massima variabilità della *policy*, che per la maggior parte dei valori non riportatati si comporta in modo molto più stabile.

Nelle Figure 3.3 e 3.4 vengono riportati degli andamenti del tutto analoghi, con l'unica variazione che la strategia ottima π è stata calcolata supponendo una distribuzione markoviana delle probabilità di transizione della sequenza $\{P_k\}$.

A questo punto, avendo a disposizione le due strategie ottime, mi è sembrato opportuno confrontarle in termini di *costo medio istantaneo* (nel senso di somma dei costi al variare del tempo delle 6 realizzazioni divisa per il numero di intervalli temporali trascorsi fino all'istante corrispondente) ad esse associato se applicate alla sequenza di richieste di potenza $\{P_k\}$. Si è esteso il paragone anche ai costi associati alle soluzioni fornite dall' algoritmo di Viterbi,

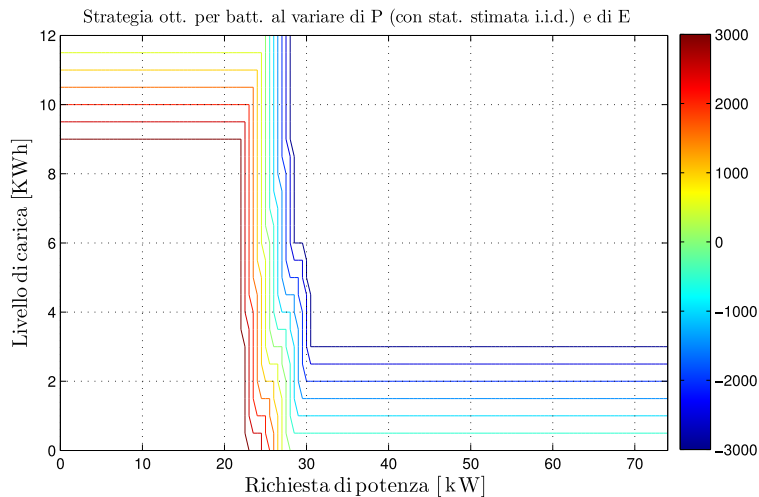


Figura 3.1: Strategia ottima π ricavata tramite DPI al variare sia del segnale P_i che del livello di energia E_j presente nella batteria.

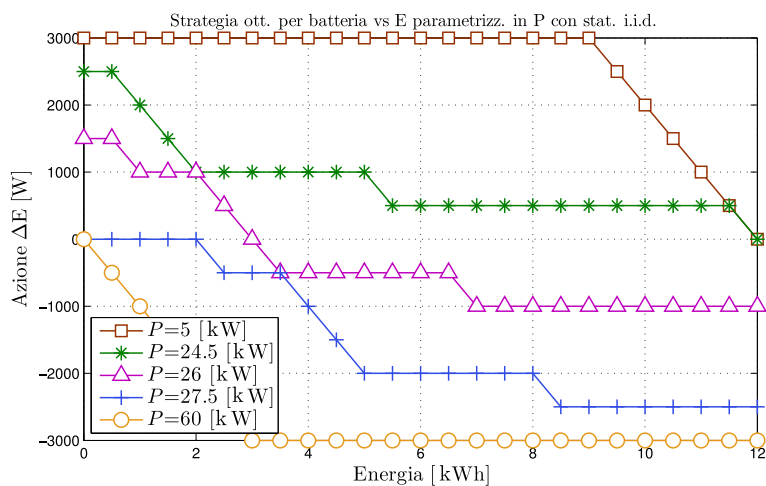


Figura 3.2: Strategia ottima π ricavata tramite DPI al variare del livello di energia E_j presente nella batteria, parametrizzata per alcuni valori di P_i .

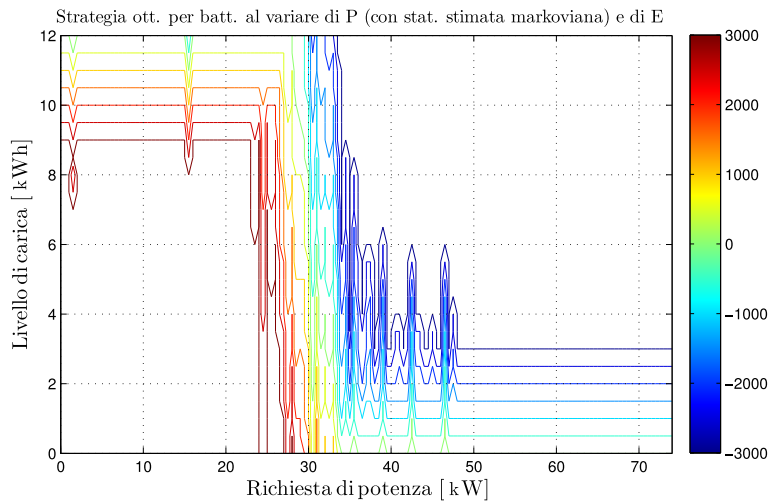


Figura 3.3: Strategia ottima π ricavata tramite DPM al variare sia del segnale P_i che del livello di energia E_j presente nella batteria.

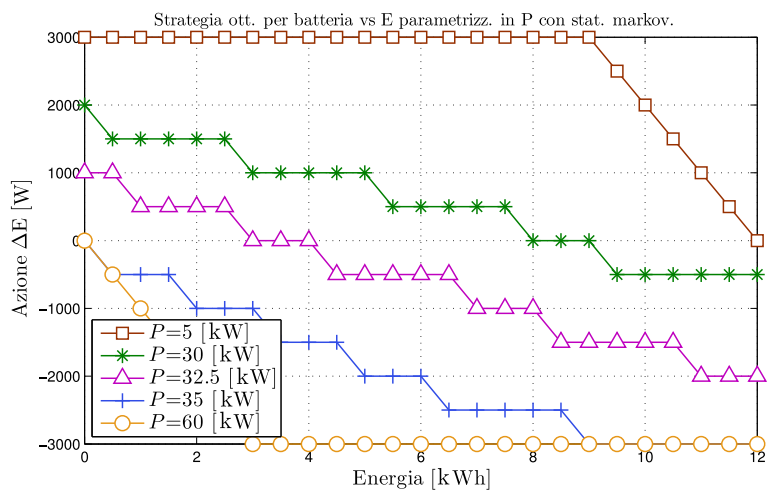


Figura 3.4: Strategia ottima π ricavata tramite DPM al variare del livello di energia E_j presente nella batteria, parametrizzata per alcuni valori di P_i .

dall'algoritmo ST descritto nella Sezione 1.8, dove tale soglia è data da $P_{\text{thr}} = \mathbb{E}[\{P\}]$ e al caso senza batteria. I risultati ottenuti in termini puramente numerici riguardo al *costo complessivo* (inteso come somma di quello relativo alle 6 realizzazioni) sono riportati nella Tabella 3.1. Gli andamenti dei costi medi istantanei, invece, sono rappresentati in Figura 3.5 al variare dell'ora dei giorni del mese di gennaio (per un totale di $744 = 24 \cdot 31$ valori).

Tabella 3.1: Risultati numerici sul costo complessivo con granularità $\Delta t = 1$ h.

Descrizione	Costo ottenuto (o % di aumento)
lower bound sul costo calcolato con alg. AOS	1.05496e+11
aumento relativo del costo con alg. DPM	5.77918 %
aumento relativo del costo con alg. DPI	7.13372 %
aumento relativo del costo con alg. ST	12.8402 %
aumento relativo del costo nel caso senza batteria	24.4147 %

Da questa tabella si evince come considerare anche come approssimazione della distribuzione di probabilità condizionata (delle richieste di potenza) quella markoviana porti ad un guadagno sul costo complessivo da pagare da parte degli utenti della grid, rispetto ad una distribuzione di probabilità assunta come iid, dell'1.34 % (però quella iid ha il vantaggio di avere la necessità di stimare soltanto la statistica del primo ordine della sequenza $\{P_k\}$ e non la distribuzione congiunta). Il gap del 12.8 % del costo delle soluzioni calcolate con la DP rispetto a quello dell'algoritmo a singola soglia si presta, invece, ad una duplice interpretazione: da un lato la differenza in termini di costo complessivo sul conto dell'energia che devono pagare gli utenti all'interno della grid comincia a diventare parecchio significativa, ma dall'altro che un algoritmo che è stato dimostrato essere ottimo in una situazione molto diversa - quella con una capacità della batteria che tende all'infinito e senza nessun vincolo sul suo rate di carica/scarica nè sulla sua efficienza - si comporti in modo discreto necessitando soltanto del valore medio della richiesta netta di potenza (che potrebbe variare lentamente nel tempo) è un risultato interessante. Va notato, infine, che l'algoritmo a singola soglia potrebbe offrire - con i vincoli aggiunti - prestazioni *decisamente migliori cambiando il valore di tale soglia*: in questo caso è molto inverosimile che la scelta ottima sia ancora $\mathbb{E}[\{P_k\}]$. Determinare la nuova soglia in modo analitico, però, va al di là degli obiettivi di questa tesi. Si osservi, infine, che il gap tra la soluzione offline con batteria e quella senza batteria assume un valore intorno al 24 %: questo giustifica lo scopo principale che si pone tutta questa trattazione, che è quello di illustrare come con una strategia di *energy storage* opportuna gli abitanti all'interno di una grid possano risparmiare sul loro conto energetico, sfruttando la possibilità di immagazzinamento all'interno della batteria nei momenti in cui il costo $C(\cdot)$ in funzione della potenza ha pendenza inferiore.

A conclusione di questa sezione, viene riportato un confronto di prestazioni (sempre in termini di costo) tra la programmazione dinamica e l'algoritmo a singola soglia, per un setting dei parametri della batteria nel quale gli autori dell'algoritmo ST hanno dimostrato in [2] che la strategia da esso calcolata tende a quella ottima. Si è fissata quindi la capacità della batteria a 30 kWh e si è supposto che essa abbia efficienze di carica e scarica unitarie e rate di carica e scarica che $\rightarrow \infty$. Nella simulazione, i cui risultati sono riportati nella Figura 3.6, viene confrontato l'algoritmo ST con la DP (con una stima della statistica della richiesta netta di potenza sia markoviana che iid) in termini di costo medio istantaneo da pagare nel mese di gennaio da parte degli utenti della grid.

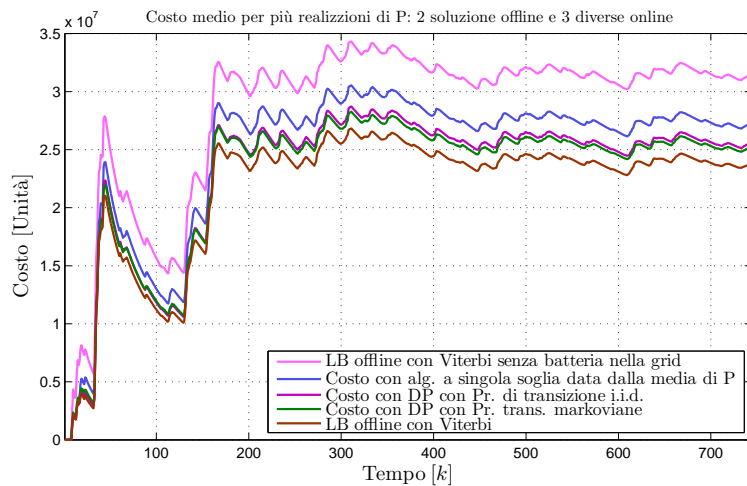


Figura 3.5: Andamento del costo medio istantaneo in funzione dell'ora del giorno dei mesi di gennaio per i 4 algoritmi analizzati.

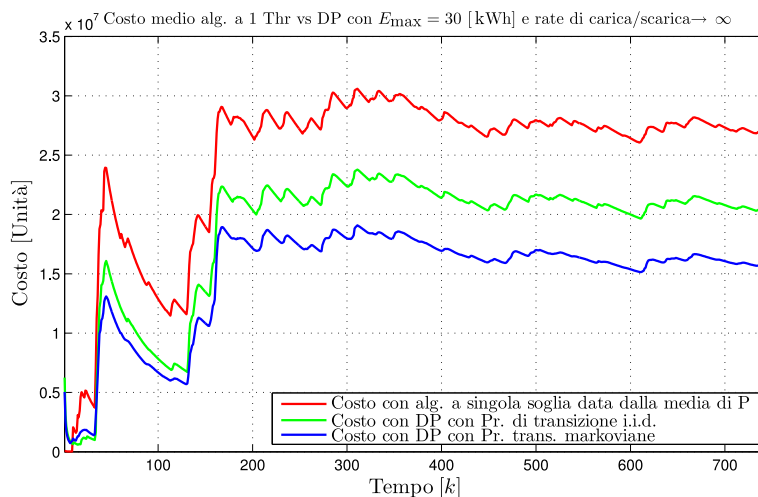


Figura 3.6: Andamento del costo medio istantaneo in funzione dell'ora del giorno dei mesi di gennaio per la DP (sia nel caso DPI che DPM) e per l'algoritmo ST.

I risultati riportati in Figura 3.6 possono sembrare di primo acchito sorprendenti (l’algoritmo a singola soglia, che dovrebbe calcolare una strategia prossima a quella ottima con capacità della batteria e rate di carica/scarica $\rightarrow \infty$, si distanzia, in termini di costo complessivo, rispettivamente del 41.42 % e del 35.53 % dal costo corrispondente alla strategia ottenuta tramite DPM e DPI), ma poi, se si rilegge con l’attenzione l’articolo [1], non è difficile trovare la giustificazione di questa stranezza. Molto semplicemente, la dimostrazione del Teorema 1 in [1] risulta *errata*, perchè non è vero che “*For increasing values of energy storage, E_m , $h(t)$ takes values in the interior of the set $\{-E(t) \leq h(t) \leq E_m - E(t)\}$ with probability that approaches 1. This means that, while discharging the battery, there is always sufficient energy such that the battery is almost never empty*”, perchè il modello di sistema equivalente a quello descritto nell’articolo equivale ad una *random walk* bidirezionale con la stessa probabilità di transizione verso il generico stato successivo e quello precedente, che si sa essere *ai limiti della stabilità*. Non è vero che la batteria, durante l’evoluzione stocastica del sistema, non risulterà quasi mai vuota, nemmeno se la capacità della batteria è molto grande, e di conseguenza quella che viene indicata come *Policy (P)* non risulta ottima nemmeno asintoticamente. Un’altra osservazione interessante che si può dedurre dalla Figura 3.6 è la netta differenza di prestazioni, in termini di costo, tra una DP eseguita con stima iid delle statistica della richiesta netta di potenza rispetto a quella con stima markoviana (circa un 6 % di gap in termini di costo complessivo): la ragione si ritiene stia nel fatto che, con rate di carica e scarica infiniti e con una capacità delle batteria di 30 kWh, ogni decisione sbagliata impatti drasticamente sul costo da pagare parte degli utenti della grid, perchè può portare, con alta probabilità, a finire nella zona della curva $C(P)$ con coda quadratica, quando ciò si può evitare con la stima più accurata delle probabilità di transizione offerta dal modello markoviano.

Si è eseguita anche un’ultima simulazione, tenendo fissata E_{\max} a 30 kWh, ma riducendo i rate di carica/scarica ad un quarto della capacità massima della batteria: ciò corrisponde al setting usato in questa tesi nella maggior parte dei casi, con una possibilità di *energy storage*, però, più che raddoppiata; i risultati corrispondenti in termini di costo medio istantaneo sono riportati in Figura 3.7.

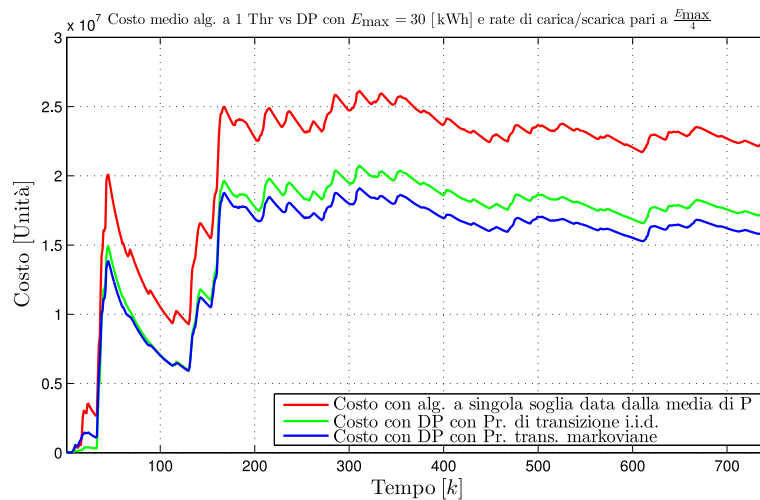


Figura 3.7: Altro esempio di andamento del costo medio istantaneo in funzione dell’ora del giorno dei mesi di gennaio per la DPM, per la DPI e per l’algoritmo ST.

In questo caso, la distanza dell’algoritmo a singola soglia in termini di costo complessivo

da quello corrispondente ad una strategia calcolata con DP e stima della statistica di $\{P_k\}$ markoviana oppure iid risulta, rispettivamente, del 28.77 % e del 22.25 % (con un gap, dovuto ad una diversa approssimazione della statistica della richiesta netta di potenza, ancora del 6 %). In una situazione maggiormente sfavorevole all’algoritmo ST (i rate di carica e scarica ora sono *finiti*) esso si avvicina cioè maggiormente, in termini di costo, alle prestazioni della strategia ottima ottenuta tramite DP. La ragione, tuttavia, è molto semplice: a rate finiti l’insieme delle azioni ammissibili da intraprendere in ogni intervallo temporale si è molto ridotto, e può accadere che l’azione che sarebbe ottima, che viene trovata dalla strategia ricavata con DP, non possa venire effettuata perché è necessiterebbe di una batteria con rate di carica/scarica maggiori. Quella che, invece, viene effettivamente intrapresa, risulta ottima nel setting che è stato simulato, ma ciò ha l’effetto di ridurre parecchio il gap prestazionale tra l’algoritmo a ST e la programmazione dinamica, sebbene gli autori di [1] ne abbiano dimostrato (in modo errato) l’ottimalità solo se non si pongono vincoli sul rate di carica/scarica, supponendo che la batteria possa essere completamente caricata o scaricata della sua intera capacità all’interno di un intervallo temporale, che in questo caso è $\Delta t = 1$ h.

3.3 Risultati delle simulazioni numeriche ottenute con 3 funzioni di costo lineari

Si ritiene sia interessante riportare anche un approfondimento su come una parte dei parametri che definiscono il modello del sistema (presentati nel Capitolo 1) influenzino le prestazioni in termini di costo della soluzione associata agli algoritmi AOS, DPM, DPI e ST. Per quanto riguarda la funzione di costo, si è scelto in questa sezione di considerarne tre, sempre *lineari a tratti senza coda quadratica*¹, adottando il seguente criterio nella loro scelta: una volta che si è fissata la prima di esse in modo che sia convessa e non-decrescente (dove la seconda proprietà è di banale interpretazione e la prima riflette il fatto che il guadagno marginale da parte dell’operatore energetico debba aumentare al crescere della richiesta di potenza, per compensare i costi progressivamente sempre più alti per produrre e trasportare ogni unità addizionale di energia agli utenti finali), le altre due sono state selezionate per garantire *lo stesso costo medio*, in un sistema senza possibilità di *energy storage*, con una distribuzione di probabilità della richiesta netta di potenza riportata in Figura 1.6. L’implementazione di quest’ultima affermazione non è stata banale, ma per non dilungarsi eccessivamente viene riportato, in Figura 3.8, solo l’andamento delle tre funzioni di costo considerate in questa sezione: la prima ha due pendenze (con punto di rottura in $\mathbb{E}[\{P_k\}]/2$) paragonabili; la seconda ha lo stesso punto di rottura della prima ma una differenza tra i 2 *slopes* molto più marcata; la terza, infine, ha l’ultima pendenza uguale alla più ripida di $C_2(\cdot)$ e i suoi punti di rottura sono $\frac{1}{3}\mathbb{E}[\{P_k\}]$ e $\frac{2}{3}\mathbb{E}[\{P_k\}]$.

In questa sezione vengono discusse le performance raggiunte dai quattro algoritmi sopra menzionati in termini di *fattore di miglioramento sul costo* rispetto al caso di riferimento di un sistema senza batteria, definito formalmente da

$$\gamma_{\text{alg}} = 1 - \frac{C_{\text{alg}}}{C_{\text{no}}} \quad (3.17)$$

dove C_{alg} è il costo corrispondente al generico algoritmo “alg” e C_{no} è il costo corrispondente al caso senza batteria. Si noti che le tre funzioni di costo riportate in Figura 3.8 sono state progettate con lo scopo di ottenere lo stesso costo C_{no} , quindi il confronto tra le prestazioni è *equo*

¹In questo modo si potranno applicare senza riserve gli approfondimenti teorici del Capitolo 5, che non si estendono al caso con coda quadratica.

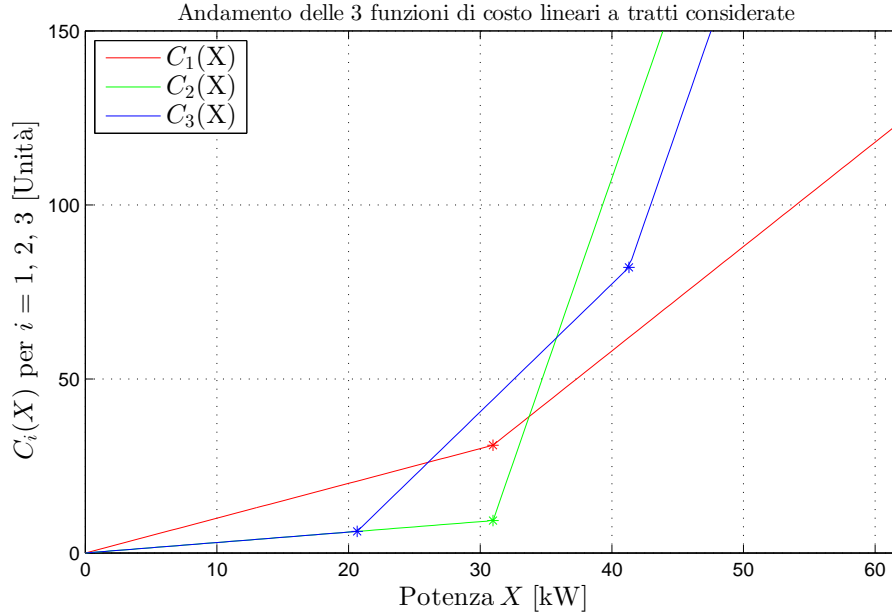


Figura 3.8: Andamento delle tre funzioni di costo lineari a tratti che garantiscono lo stesso costo medio avendo in ingresso una sequenza con la distribuzione di probabilità di $\{P_k\}$.

anche al variare dello scenario di costo. L'unica grossa differenza implementativa rispetto alle simulazioni riportate nel resto di questa tesi è che, avendo fatto variare anche la capacità della batteria ed i rate e le efficienze di carica/scarica, il passo di quantizzazione del livello energetico della batteria, ΔE , non si è tenuto fissato, ma è stato fatto variare tra 0.125 kWh e 0.5 kWh, per evitare che prestazioni in termini di costo differenti andassero a coincidere a causa di un passo di quantizzazione troppo elevato per il setting in considerazione. In Figura 3.9 vengono paragonate le prestazioni dei quattro algoritmi in funzione di E_{\max} , fissato $F_{\max} = F_{\max}^c = F_{\min}^d = \frac{1}{4} \frac{E_{\max}}{\Delta t}$, per ciascuna delle tre funzioni di costo prese in considerazione. Si nota di primo acchito che le performance crescono linearmente in funzione della capacità massima E_{\max} ; la figura fornisce inoltre un'evidente conferma che il guadagno effettivo di tutti gli algoritmi è *strettamente legato alla funzione di costo*. Si può notare, inoltre, che tutti gli algoritmi raggiungano prestazioni migliori (ovvero, maggiori risparmi sul costo) se viene usata per la prezzatura dell'energia la funzione di costo denominata $C_2(\cdot)$, che porta anche al maggior gap prestazionale tra l'algoritmo ST e gli algoritmi di Programmazione Dinamica. La ragione di questo comportamento è che più frammentaria e più ripida è la funzione di costo, maggiore è il guadagno sul costo energetico che può portare una strategia intelligente di *energy storage*; viceversa, più lineare e più *smooth* risulta la funzione di costo, meno c'è spazio per l'ottimizzazione del costo. Si osservino inoltre le ottime prestazioni degli algoritmi DPI e DPM con le funzioni di costo $C_1(\cdot)$ e $C_2(\cdot)$, che sfiorano in questi casi il bound della strategia AOS.

Come si è accennato nel Capitolo 1, si sono condotte poi delle simulazioni per verificare l'influenza delle efficienze di carica/scarica (rispettivamente η_c e η_d) sulle prestazioni in termini di costo degli algoritmi analizzati. I risultati, al variare della funzione di costo, con $E_{\max} = 12$ kWh e con $F_{\max} = \frac{1}{4} \frac{E_{\max}}{\Delta t}$, sono riportati in Figura 3.10: risulta evidente che in tutti i casi le performance *scalano linearmente con l'efficienza η* . Tenerla fissata a $\eta = 0.9$, come in tutte le altre simulazioni di questa tesi, appare quindi una scelta ragionevole, dato che l'effetto della sua variazione non è altro che traslare rigidamente in alto o in basso il costo complessivo da pagare

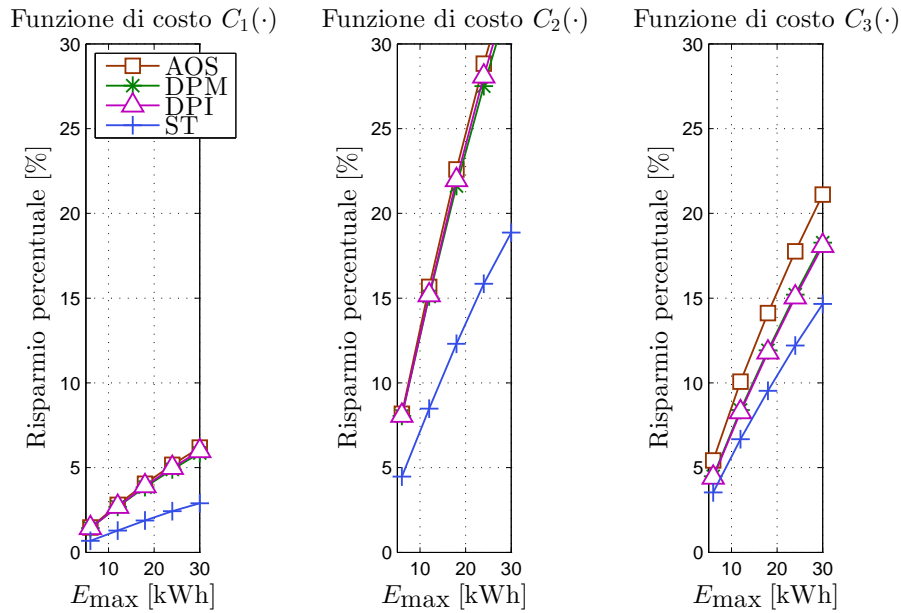


Figura 3.9: Miglioramento sul costo (in percentuale) per $F_{\max} = \frac{1}{4} \frac{E_{\max}}{\Delta t}$ e le funzioni di costo $C_1(\cdot)$, $C_2(\cdot)$ e $C_3(\cdot)$ al variare della capacità E_{\max} della batteria.

da parte degli utenti della grid.

Infine, nell'ultima simulazione viene mostrato il risparmio sul costo energetico in funzione di $\sigma \triangleq \frac{F_{\max} \Delta t}{E_{\max}}$ (che corrisponde al range normalizzato di valori delle azioni di carica/scarica della batteria prese in considerazione in ogni intervallo di tempo dai vari algoritmi implementati), con $E_{\max} = 6, 12$ e 24 kWh rispettivamente sulla sinistra, al centro e sulla destra di Figura 3.11, avendo scelto come funzione di costo $C_3(\cdot)$. Come ci si aspettava, l'algoritmo AOS offre sempre le migliori prestazioni, seguito da DPM e da DPI (a parte per piccoli valori del rapporto σ , come successivamente illustrato). DPI si distanzia di poco dalle prestazioni della strategia DPM, mentre l'algoritmo ST porta ai risultati peggiori, risentendo dei vincoli sulla capacità limitata della batteria e dei rate di carica/scarica $\neq \infty$. Le curve sono visibilmente separate per alti valori di σ , mentre per valori più limitati di σ la limitata cardinalità dell'insieme delle azioni ammissibili fa sì che i quattro algoritmi si comportino in modo molto simile. Come già accennato nella Sezione 1.8, si osserva che, inaspettatamente, *le prestazioni dell'algoritmo ST peggiorano per alti valori di σ* . Riflettendo su questo effetto, si è giunti alla conclusione che esso dipenda in buona parte dalla forte correlazione presente nella sequenza $\{P_k\}$ delle richieste di potenza, per la quale si ha che è molto probabile che i valori elevati vengano assunti a burst: in tal caso la strategia ST scaricherà la batteria in modo da soddisfare completamente la prima richiesta, lasciando in essa una carica residua che non è sufficiente a soddisfare le richieste successive. Il costo complessivo risultante sarà quindi molto maggiore di una strategia che soddisfa parzialmente ogni richiesta all'interno del burst². Per confermare questa ipotesi sono state condotte anche delle simulazioni (di seguito non riportate per mancanza di spazio) con una sequenza di richieste di potenza non correlata: si potrebbe notare come in tal caso il degrado di prestazioni per elevati valori del rate di carica/scarica svanisca completamente.

²Come viene tipicamente eseguito da algoritmi, ad esempio, DP, che sfruttino una maggior conoscenza statistica della richiesta di potenza.

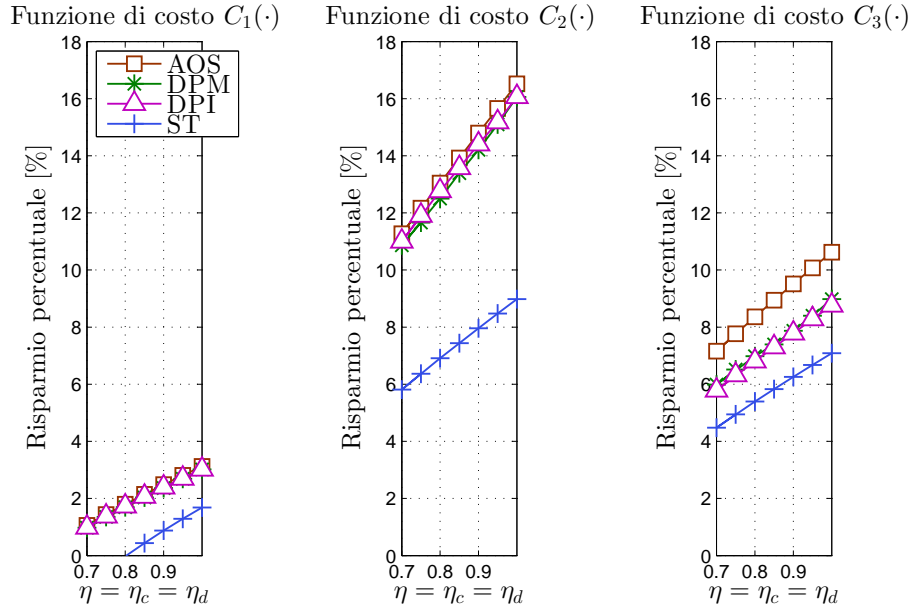


Figura 3.10: Miglioramento sul costo (in percentuale) per $F_{\max} = \frac{1}{4} \frac{E_{\max}}{\Delta t}$ e le funzioni di costo $C_1(\cdot)$, $C_2(\cdot)$ e $C_3(\cdot)$ al variare dell'efficienza di carica/scarica $\eta = \eta_c = \eta_d$ della batteria.

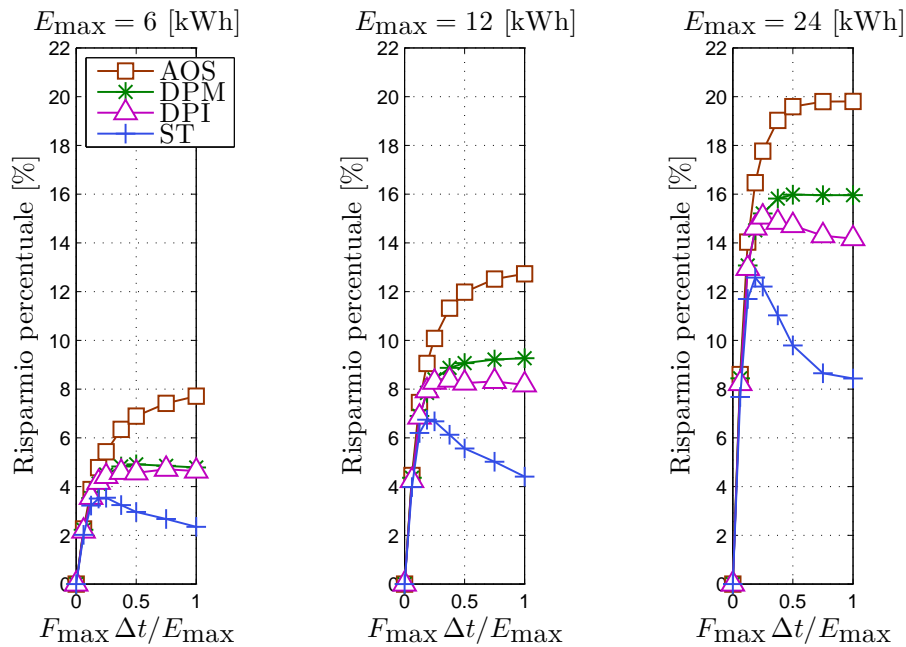


Figura 3.11: Miglioramento sul costo (in percentuale) in funzione di $\frac{F_{\max} \Delta t}{E_{\max}}$, considerando la funzione di costo $C_3(\cdot)$ e 3 diverse dimensioni della capacità della batteria.

Come conclusione della sezione, si è scelto di riportare un ingrandimento (per bassi valori di σ) della Figura 3.11, che rende visibile un effetto che si manifesterà ripetutamente anche nel Capitolo 4: il fatto che le prestazioni dell'algoritmo DPM, che sfrutta la stima della matrice delle probabilità di transizione della sequenza $\{P_k\}$, *non risultino sempre superiori* a quelle della strategia DPI, come riportato in modo qualitativo in Figura 3.12 per via grafica e nelle sezioni successive anche in modo quantitativo per via numerica. Va comunque osservato che il gap prestazionale tra i due algoritmi è modesto (si tratta sempre di decimi di punti percentuali), però non di immediata interpretazione. Proprio per questa ragione si è deciso di approfondire questo effetto nella Sezione 3.4.

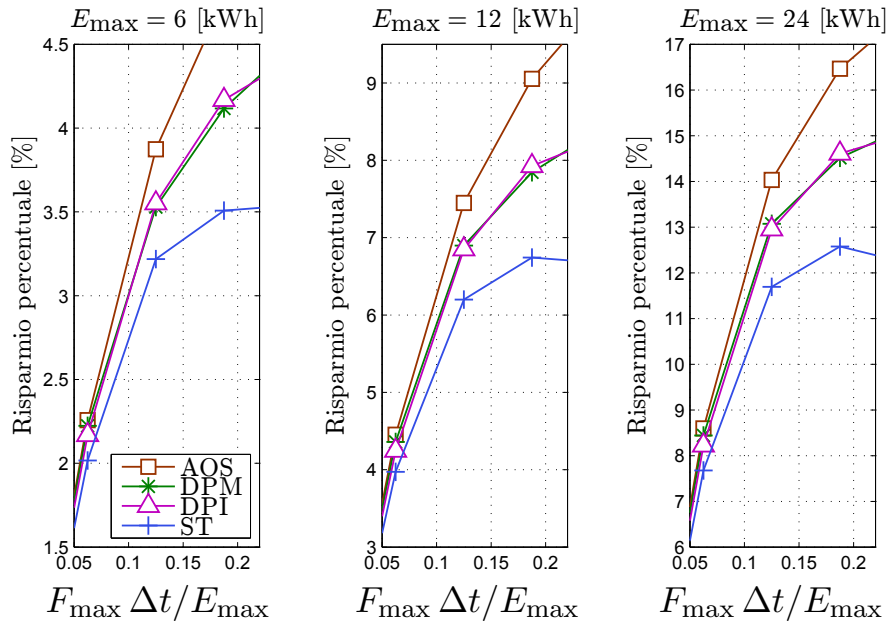


Figura 3.12: Miglioramento sul costo (in percentuale) in funzione di $\frac{F_{\max} \Delta t}{E_{\max}}$, considerando la funzione di costo $C_3(\cdot)$ e 3 diverse dimensioni della capacità della batteria: dettaglio per osservare meglio il confronto di prestazioni tra gli algoritmi DPM e DPI.

3.4 Risultati delle simulazioni numeriche condotte ad hoc per approfondire il confronto tra gli algoritmi DPM e DPI

Nel corso del lavoro di tesi il fatto che il costo ottenuto applicando la strategia fornita dalla DP con una statistica delle richieste di potenza supposta iid *non sia sempre maggiore* rispetto a quello corrispondente ad una statistica supposta markoviana (come appena menzionato nella Sezione 3.3 e come è riportato nel Capitolo 4) non è stato facile da interpretare. La spiegazione di tale comportamento si ritiene sia l'aver mediato una richiesta di potenza generata con granularità del minuto su un'intera ora perdendo così tutta la correlazione inizialmente catturabile tra campioni temporalmente vicini: si è voluto riportare di seguito anche i risultati di due prove ad hoc eseguite per validare questa affermazione. La prima di tali prove è stata quella di, usando la matrice \mathbf{P}_{tr} stimata sulla richiesta di potenza $\{P_k\}$, generare una nuova ed artificiale richiesta di potenza $\{P_k^{\text{mark}}\}$ in modo nativamente markoviano, ottenendo in tal modo dei dati la cui matrice di transizione originaria fosse proprio \mathbb{P}_{tr} . Stimando, invece, soltanto la statistica del primo

ordine di questa nuova sequenza, supponendone una statistica iid con delle colonne della matrice di transizione tutte uguali, per poi usarla nell'algoritmo di DP per ottenere la strategia ottima π , non si sarebbero potuti che ottenere dei risultati in termini di costo superiori rispetto alla soluzione associata all'effettiva statistica markoviana. Le simulazioni in Matlab hanno confermato quanto appena descritto, ma riportarne i risultati non si ritiene sia molto significativo perchè si riferiscono ad un segnale ($\{P_k^{\text{mark}}\}$) generato ad hoc per ottenere la conferma desiderata. Può essere più interessante, invece, descrivere la procedura con cui generare tali dati artificiali che si può riassumere perfettamente dal codice Matlab riportato nel Codice 3.1, di immediata comprensibilità anche senza la presenza di commenti.

Codice 3.1: Codice Matlab per generare una processo aleatorio markoviano data la sua matrice delle probabilità di transizione ad un passo.

```
N_P=125;
P_min=0;
P_max=62e3;
[P_q,P1,PT]=P_quantize_and_Pr_estimate(P,P_min,P_max,N_P);
CDFP=cumsum(PT,1);
ip=floor(length(P_q)/2)+1;
for k=1:11
    z=rand;
    iip=1+sum(z>CDFP(:,ip));
    P_synth_mark(k)=P_q(iip);
    ip=iip;
end
```

La seconda prova condotta è stata quella di non mediare sull'ora, ma di mantenere i campioni forniti dal tool per generare la richiesta di potenza della grid con la loro granularità originaria di un minuto. Per non far esplodere la mole di dati da elaborare, ci si è limitati a generare una nuova sequenza di richieste di potenza, $\{P_k^{\text{new}}\}$, per i 1440 minuti di ogni giorno di un mese di gennaio, ottenendo un segnale con $60 \cdot 24 \cdot 31 = 44640$ campioni. Su questa nuova sequenza di richieste nette di potenza sono stati applicati i medesimi algoritmi descritti nella Sezione 3.2, arrivando ai risultati (in termini di strategia ottima π con una stima della statistica di $\{P_k^{\text{new}}\}$ supposta rispettivamente iid e markoviana) riportati in Figura 3.13, in una veste grafica compatta.

Dal punto di vista qualitativo si ritiene sia interessante notare come, in queste nuove simulazioni, l'andamento della strategia ottima π con una statistica della richiesta di potenza supposta iid mantenga una struttura ben approssimabile da una *policy* a doppia soglia, come verrà discusso nella Sezione 3.5; quello con una statistica supposta markoviana, invece, risulta ancora più variabile e nervoso di quello riportato in Figura 3.3, perchè in questo caso insegue in modo molto più preciso la funzione di correlazione calcolata a distanza di un passo della sequenza $\{P_k^{\text{new}}\}$. Per quanto riguarda i risultati ottenuti in termini di costo complessivo (in questo caso riferito ad un'unica realizzazione del mese di gennaio), questi sono riportati in forma numerica nella Tabella 3.2, confermando la maggior bontà in termini di costo corrispondente alla soluzione ottenuta con DP con una stima della statistica della richiesta di potenza supposta markoviana: il suo costo dista dal lower bound di Viterbi circa il 2.71 % - quasi il doppio dell'1.34 % (relativo alla Tabella 3.1) in meno rispetto a quello ottenuto con una statistica supposta iid.

Una volta eseguite le due prove appena riportate, si è deciso di mantenere la granularità dell'ora nelle rimanenti simulazioni successivamente illustrate per la ragione seguente. Che una matrice di transizione ad un passo iid sia un caso particolare (che necessita soltanto della stima della statistica del primo ordine) di una di tipo markoviano e che non riesca quindi a catturare

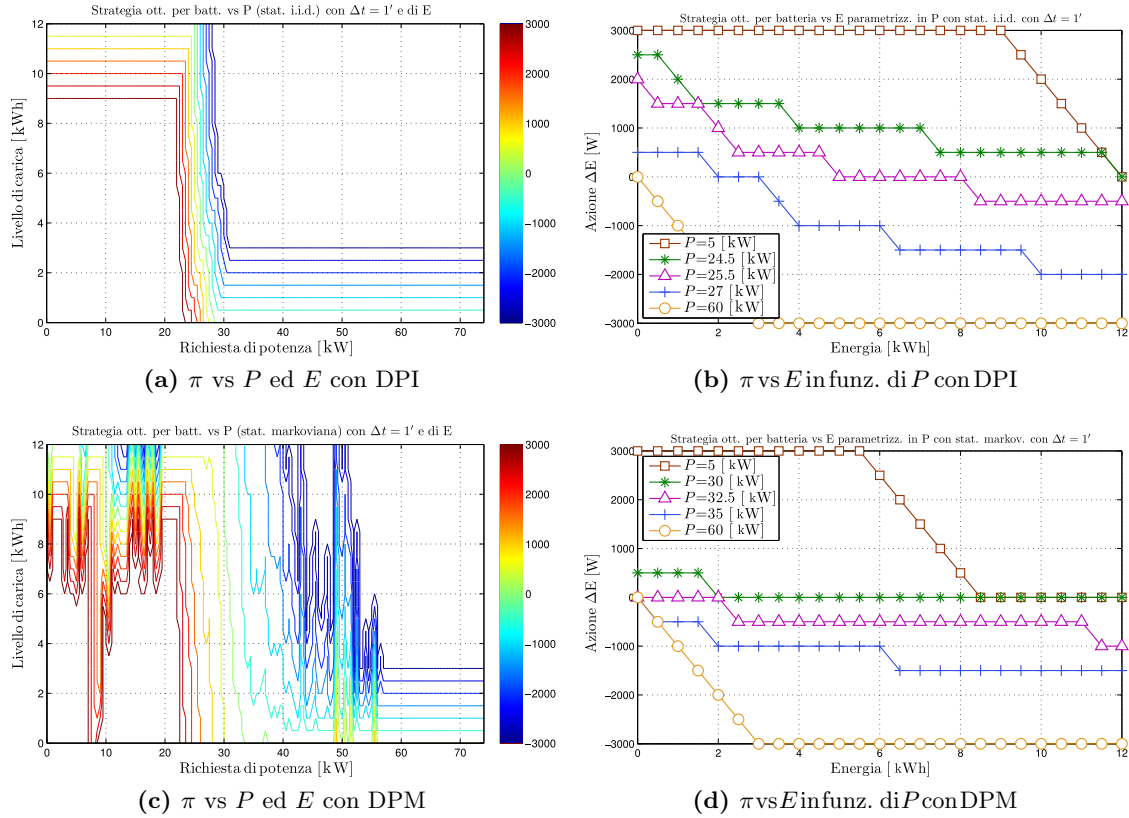


Figura 3.13: Strategie ottime π di *energy storage* fornite dalla DP considerando la sequenza $\{P_k^{\text{new}}\}$ (avente una granularità temporale $\Delta t = 1'$).

Tabella 3.2: Risultati numerici sul costo complessivo calcolato sulla sequenza $\{P_k^{\text{new}}\}$ (avente una granularità temporale $\Delta t = 1'$).

Descrizione	Costo ottenuto (o % di aumento)
lower bound sul costo calcolato con alg. AOS	8.79588e+11
aumento relativo del costo con alg. DPM	1.83841 %
aumento relativo del costo con alg. DPI	4.54393 %
aumento relativo del costo con alg. ST	5.40536 %
aumento relativo del costo nel caso senza batteria	6.5481 %

in modo altrettanto accurato la correlazione (eventualmente) presente tra campioni adiacenti di un processo aleatorio è un risultato più che banale. Nei casi in cui il costo associato alla prima delle due statistiche sia *lievemente inferiore* (come nelle Sezioni 4.3.1 e 4.3.2) è solo un problema numerico di mancanza di un numero sufficiente di campioni per una buona stima di \mathbf{P}^{tr} con una statistica della richiesta di potenza supposta markoviana. Quello che è lecito concludere è che, in assenza di una correlazione ben osservabile numericamente tra campioni di $\{P_k\}$ a distanza di un'ora, l'approccio DPI e quello DPM non presentino elevate differenze di prestazioni in termini di costo medio della corrispondente soluzione.

3.5 Approssimazione della strategia ottima tramite una *policy* a doppia soglia

Come è già stato fatto notare nell'introduzione, il paper [2] ha avuto un contributo molto importante, soprattutto nella fase di impostazione di questo lavoro di tesi, e quindi, seppur avendo definito un modello di sistema parzialmente diverso, si ritiene sia una buona idea fare un confronto sensato tra i risultati ottenuti. Il contenuto dell'articolo è già stato riassunto nell'Introduzione. Qui basta ricordare che i suoi risultati valgono con un modello di sistema, rispetto a quasi tutti i punti di vista, molto più generico di quello descritto nel Capitolo 1: il prezzo dell'energia, ad esempio, viene assunto come tempo-variante e lo stato del sistema non consiste soltanto della coppia livello di energia nella batteria-richiesta di potenza, ma comprende anche un generico *modulating state*, che può tenere conto ad esempio dell'ora del giorno o della stagione corrente. Anche se non viene esplicitamente affermato dagli autori del paper, tuttavia, l'equazione $\gamma_{\mathbf{x}} = \left[d(\mathbf{x}) + (\delta)^+ \frac{1}{\eta_c} + (\delta)^- \eta_d \right] p(\mathbf{x})$, dove $\gamma_{\mathbf{x}}$ rappresenta il costo istantaneo, assume che quest'ultimo sia lineare rispetto alla richiesta di potenza $d(\mathbf{x})$. La funzione di costo $C(\cdot)$ definita nella Sezione 1.2, invece, viola decisamente questa ipotesi e quindi non è più detto a priori che una strategia ottima per la minimizzazione del costo complessivo possa essere descritta perfettamente da una a doppia soglia (si rimanda al Capitolo 5 per degli approfondimenti analitici sulla struttura della *policy* ottima al variare della funzione di costo).

A questo punto, assunto come note le strategie ottime ricavate tramite DP (supponendo una statistica della richiesta netta di potenza iid e markoviana) riportate rispettivamente in Figura 3.1 e in Figura 3.2, risulta interessante trovare la generica strategia a doppia soglia (che rispetti i vincoli sul rate di carica e scarica finiti della batteria e sul suo livello energetico compreso tra 0 ed E_{max}) che abbia fissate le due soglie $\beta_{\mathbf{x}}^-$ e $\beta_{\mathbf{x}}^+$ in modo da risultare la più vicina possibile in norma quadratica (criteri altrettanto validi avrebbero potuto essere quelli della minimizzazione della norma \mathcal{L}^1 o \mathcal{L}^∞) alla strategia ottima π . Risparmiandone i dettagli implementativi per non dilungarsi eccessivamente nella loro descrizione (per ricavare la strategia a doppia soglia più vicina possibile in norma quadratica a π si è usato un criterio enumerativo, che ha richiesto un tempo molto limitato per determinare la soluzione a causa del numero contenuto di alternative da valutare), si riportano i risultati relativi al caso con una statistica della richiesta di potenza supposta iid nelle Figure 3.14 e 3.15, rispettivamente.

Le simulazioni condotte per via numerica hanno fornito come risultato che *l'errore relativo sul costo medio* (rispetto alle 6 realizzazioni del mese di gennaio, memorizzate nella sequenza $\{P_k\}$) commesso dalla strategia a doppia soglia con le due *thresholds* ottimizzate in norma quadratica rispetto alla *policy* ottima ottenuta tramite DPI, è dato da 2.11285 %. Questo risultato si presta a molteplici osservazioni: innanzitutto conferma che con una funzione di costo $C(\cdot)$ non lineare rispetto alla richiesta netta di potenza (che, nel caso del sistema considerato, è il suo unico argomento) anche la strategia a doppia soglia che più si avvicina, in norma quadratica, alla *policy*

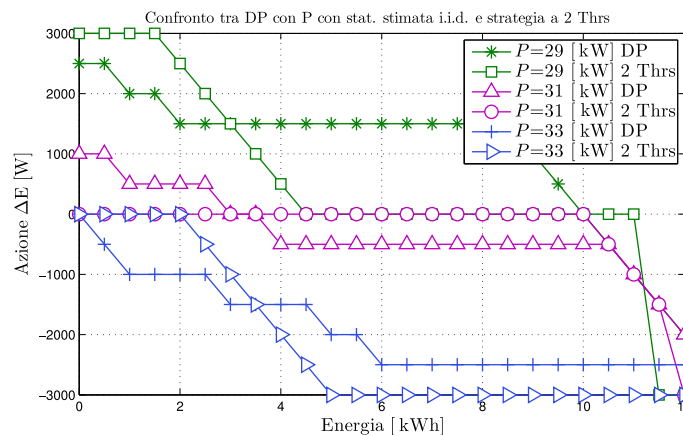


Figura 3.14: Confronto tra la strategia ottima π al variare di E_j fissati alcuni valori di P_i ottenuta tramite DPI e quella a doppia soglia più vicina ad essa in norma quadratica.

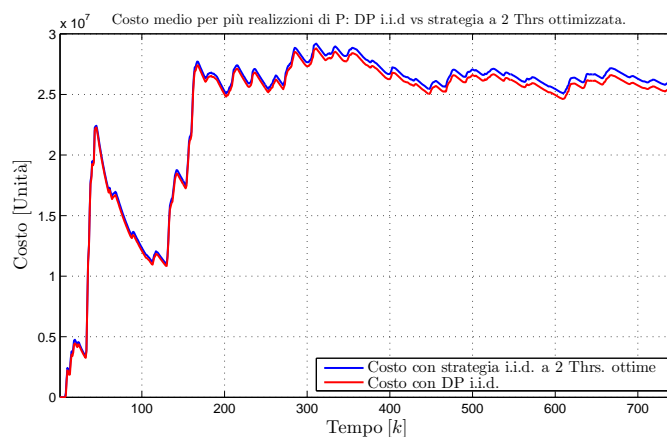
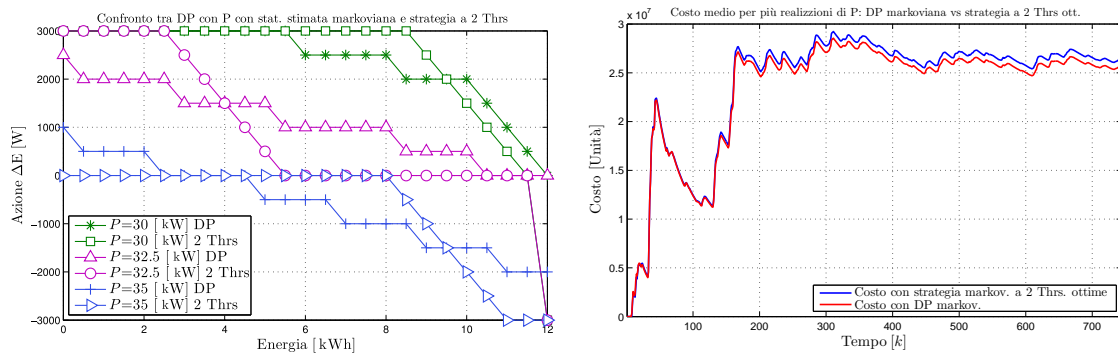


Figura 3.15: Confronto tra l'andamento del costo medio istantaneo corrispondente alla strategia ottima π ottenuta tramite DPI e a quella a doppia soglia più vicina ad essa in norma quadratica.

ottima ricavabile tramite DP *non* risulta minimizzare il costo complessivo. Se un circa 2 % di differenza sul costo relativo sia “tanto o poco” dipende da molteplici fattori quali l’effettivo prezzo al kWh richiesto dall’operatore energetico, etc., ma la soluzione proposta in [2], come fanno notare i suoi stessi autori, manifesta il problema, tranne in casi estremamente semplificati, del calcolo delle due soglie β_x^- e β_x^+ : in questo caso esso è stato possibile a posteriori minimizzando la norma quadratica rispetto a π una volta che questa era nota; senza la sua conoscenza, una tecnica di DP applicabile per il calcolo delle *thresholds* è la *value iteration*.

Considerazioni del tutto analoghe valgono quando si suppone una statistica della richiesta netta di potenza di tipo markoviano: i corrispondenti risultati sono riportati, in modo compatto, in Figura 3.16, e l’errore relativo sul costo complessivo risulta in questo caso del 2.73944 %.



(a) Confronto tra strategia a doppia soglia e π .

(b) Confronto tra l’andamento del costo medio istantaneo corrispondente

Figura 3.16: Confronto tra la strategia ottima π ottenuta tramite DPM e quella a doppia soglia più vicina ad essa in norma quadratica.

Capitolo 4

Secondo modello di sistema

4.1 Definizione del nuovo modello di sistema sulla base dei “limiti” di quello precedente

4.1.1 Nuovo modello di batteria

Il modello di batteria definito in 1.1 è più accurato di molti usati nella letteratura di *energy management* delle smart grid, ma non abbastanza da essere neanche in minima parte realistico rispetto alle effettive caratteristiche elettro-chimiche di una batteria in commercio, ad esempio quelle di tipo Li-ion (cioè agli ioni di litio) descritte in [20]. Una prima complicazione riguarda l'efficienza della batteria, precedentemente definita come frazione dell'energia riottenibile da essa dopo averne immagazzinato una certa quantità. Nel resto della trattazione seguente si sceglie di introdurre il concetto di capacità di conservazione della carica elettrica nel tempo, indipendentemente dalle azioni di carica o scarica che vengono eseguite nel corrispondente intervallo temporale: a causa di una specifica resistenza interna, il livello energetico della batteria diminuisce di una certa percentuale (nella pratica infinitesima) ad ogni Δt , in funzione del livello di carica in essa presente e della temperatura dell'ambiente, come rappresentato nella Figura 4.1.

Per modellizzare tale comportamento, supponendo di non prendere in considerazione anche la variabile temperatura per non complicare eccessivamente il problema, si introduce una legge di aggiornamento del livello energetico passando da un certo intervallo temporale a quello successivo che modella in qualche modo la (minima) perdita di carica avvenuta nella batteria indipendentemente dall'azione intrapresa, che si può specificare analiticamente, per quel che riguarda il termine legato alla perdita di energia, come

$$E_{k+1} = E_k - \theta (1 - \vartheta \cdot \exp(-E_k)) \quad (4.1)$$

dove θ e ϑ sono due parametri che andrebbero oculatamente tarati per rappresentare il corretto ordine di grandezza di questo fenomeno legato ai processi elettro-chimici della batteria. Dalla Figura 4.1, si può notare come l'ordine di grandezza con cui questa diminuzione del livello di carica nel tempo si manifesta in modo significativo dal punto di vista misuristico è dell'ordine della *decina di giorni*, un tempo enorme rispetto a Δt che rimane fissato ad un'ora. Una soluzione banale al problema sembrerebbe quella di aumentare proporzionalmente anche la costante θ : in questo modo, tuttavia, se si vuole applicare iterativamente, passo dopo passo, l'equazione (4.1) per non stravolgere il modello del sistema, le variazioni di energia nella batteria risulterebbero trascurabili rispetto al passo di quantizzazione del livello energetico $\Delta E = 0.5$ kWh che, se aumentato, farebbe esplodere la dimensione dello spazio degli stati (in ottica DP). La scelta adottata è stata quindi quella di fissare (ad arbitrio) delle costanti θ e ϑ che portino ad una

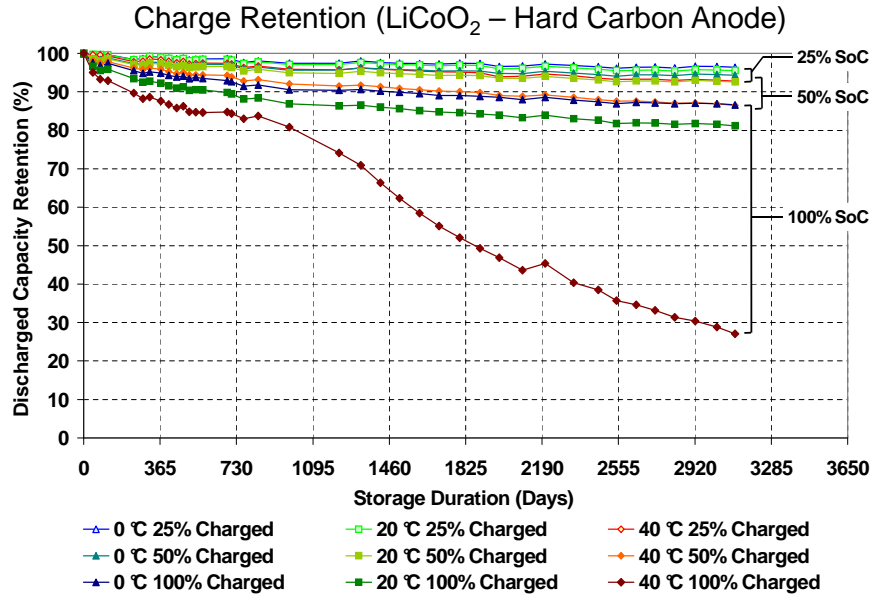


Figura 4.1: Perdita della carica da parte di una batteria Li-ion in funzione del tempo temporale e della temperatura.

perdita di energia passando dal tempo k a quello $k + 1$ irrealisticamente elevata, in modo che l'introduzione nel modello della perdita di carica nel tempo avesse delle ripercussioni evidenti sulla struttura della strategia ottima π in funzione del livello di energia E_j e della richiesta netta di potenza P_i . Sebbene questo sia un compromesso tra riscontro fisico e trattabilità del problema, si ritiene che i risultati che ne derivano siano significativi. Nella Figura 4.2 è illustrato un esempio di variazione del livello di carica presente nella batteria ($E_{k+1} - E_k$) in funzione di E_k , indipendentemente dall'azione intrapresa, per i valori delle costanti $\theta = 260$ [Wh] e $\vartheta = 0.0005$ impiegate nel modello di sistema considerato.

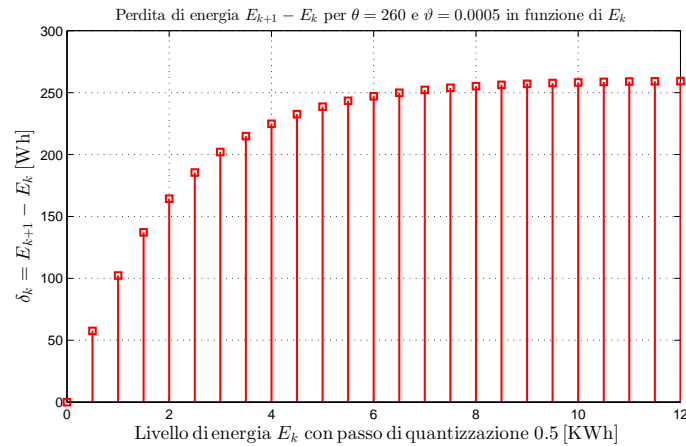


Figura 4.2: Modello di perdita nella carica nel tempo da parte della batteria in $\Delta t = 1$ h al variare della quantità di carica E_k presente in essa all'inizio dell'intervallo temporale k -esimo.

Un ulteriore fenomeno che vale la pena prendere in considerazione nella modellizzazione della batteria è il fatto che il suo tempo di vita cambi a seconda dell'utilizzo che viene adottato:

azioni di carica e/o scarica frequenti e profonde vanno a ridurre il numero medio di cicli (del corrispondente processo elettro-chimico), come è rappresentato nella Figura 4.3.

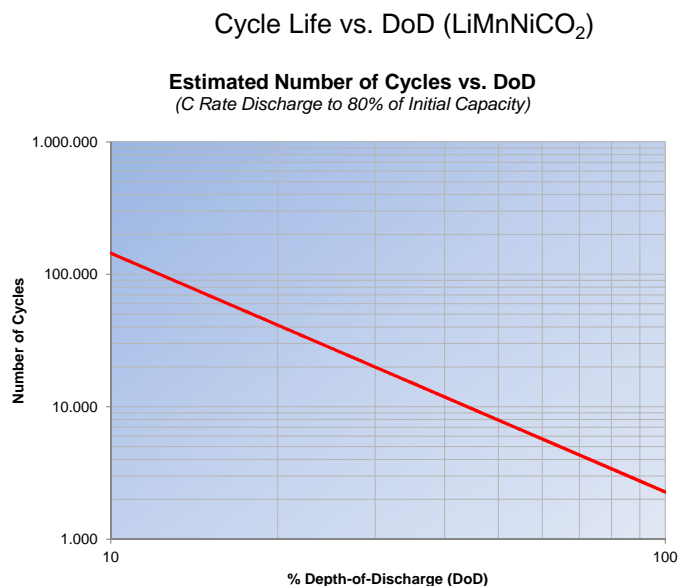


Figura 4.3: Ciclo di vita di una batteria Li-ion in funzione della percentuale di scarica.

Quanto appena descritto impatta sull'ammortamento della spesa iniziale della batteria: se ne viene fatto un cattivo uso, può venire messa in discussione la convenienza del suo acquisto nel momento in cui la spesa iniziale risulti superiore ai soldi risparmiati possedendola, legati alla possibilità di immagazzinare energia in essa quando se ne ha a disposizione in eccesso grazie alla produzione da parte delle sorgenti di energia rinnovabile o quando il prezzo dell'energia da parte dell'operatore è particolarmente conveniente e di prelevarne quando non se ne ha a sufficienza o il prezzo è nelle fasce di picco. Ciò verrà preso in considerazione nella definizione della funzione costo della Sezione 4.1.2: è chiaro che, vista la non chiara corrispondenza tra la Figura 4.3 e dei valori numerici precisi, non si può che procedere in modo euristico. Quando la batteria viene acquistata, si suppone che essa abbia un tempo di vita medio di K cicli: ogni volta che essa viene ad esempio scaricata, il contatore K viene decrementato in modo proporzionale alla profondità di scarica

$$K = K - \gamma (E_{k-1} - E_k) \quad (4.2)$$

dove γ è un parametro che va tarato - in ottica simulativa: esso nella pratica risulta fissato una volta scelto il modello di batteria - opportunamente rispetto alla funzione di costo, in modo che eseguire un *energy storage* ottimo porti comunque ad una *riduzione* del conto da pagare all'operatore energetico, anche considerato il degrado della batteria nel tempo e il fatto che quando il contatore K raggiunge lo zero è necessaria la sua sostituzione. Tale effetto verrà inserito nella funzione di costo, che avrà un andamento fortemente non lineare in corrispondenza agli eventi di sostituzione della batteria (a causa di un suo totale degrado prestazionale) e del suo successivo riacquisto. Oltre alle due novità appena menzionate, riassunte dalle equazioni (4.1) e (4.2), è ovvio che rimangono validi i vincoli definiti nella Sezione 1.1: il livello di energia non può superare un certo valore massimo, i rate di carica e scarica sono supposti finiti e le efficienze di carica/scarica sono diverse da 1, come rimane espresso da (1.1), (1.2) e (1.3).

4.1.2 Nuovo modello di costo

In precedenza la funzione di costo era stata semplicemente definita come una curva crescente lineare a tratti e con coda quadratica in funzione della richiesta istantanea (netta) di potenza all'operatore energetico. Ora, per quanto riguarda il termine dipendente dalla richiesta di potenza complessiva da parte della grid, si sceglie di considerare semplicemente una tariffazione bi-oraria molto simile a quella offerta attualmente dall'ENEL, secondo la quale, indicata come l l'ora del giorno in cui sia presente certa richiesta di potenza X_k , si ha che

$$C_{X_k, l} = \begin{cases} c_0 X_k, & l \in \{8.00, 9.00, \dots, 21.00\} \\ c_1 X_k, & l \in \{22.00, 23.00, \dots, 7.00\} \end{cases} \quad (4.3)$$

dove $c_0 > c_1$ e nella richiesta di potenza X_k è *eventualmente* (dato che il suo livello energetico può restare invariato) compreso anche il contributo necessario a caricare la batteria o a scaricarla (assumendo che le due azioni non siano possibili contemporaneamente), che si può esprimere come $X_k \triangleq L_k - R_k + F_k = L_k - R_k + \frac{E_k - E_{k-1}}{\Delta t}$.

Nella definizione della nuova funzione di costo, si è però voluto effettuare anche un raffinamento ulteriore: con quella definita da (1.4), infatti, si era assunto che la richiesta di potenza L_k da parte dei residenti all'interno della grid fosse sempre superiore alla produzione R_k da parte delle sorgenti rinnovabili. È verosimile invece che, quando molte persone sono impegnate al lavoro e la quantità di potenza prodotta dai pannelli fotovoltaici assume i suoi valori massimi (ad esempio all'ora di pranzo), si possa verificare che $P_k = L_k - R_k < 0$. L'ENEL, anche per questa ragione, definisce una diversa tariffazione energetica quando viene venduta energia e non comprata, che si può esprimere dal punto di vista analitico come

$$C_{X_k, l} = \begin{cases} c_0 c_a X_k, & l \in fo^0, X_k \geq 0 \\ c_0 c_v X_k, & l \in fo^0, X_k < 0 \\ c_1 c_a X_k, & l \in fo^1, X_k \geq 0 \\ c_1 c_v X_k, & l \in fo^1, X_k < 0 \end{cases} \quad (4.4)$$

dove con fo^0 e fo^1 si intendono, rispettivamente, la prima e la seconda fascia oraria relative al prezzo energetico. Dal punto di vista matematico, con $C(\cdot)$ che è ora lineare in funzione della potenza, è *condizione necessaria ma non sufficiente* che il prezzo per unità di energia acquistata sia superiore a quello per unità di energia venduta affinché il problema di ottimizzazione dell'*energy storage* che ci si appresta a definire abbia un senso: se i valori delle due costanti fossero invertiti, infatti, considerando la perdita di energia nel tempo della batteria e la spesa iniziale per acquistarla, che va ammortizzata nel tempo, sarebbe lecito porsi delle domande sulla sua effettiva utilità. Detto ciò, nei primi 3 capitoli l'“unico” ruolo della batteria è stato quello di immagazzinare energia in intervalli temporali giudicati convenienti dai vari algoritmi implementati al fine di ricadere il meno possibile, sul lungo periodo, nel range di valori di $C(\cdot)$ in funzione di X con pendenza elevata. Ora, invece, la sua utilità è duplice: quanto appena affermato risulta ancora applicabile rispetto alla tariffa bi-oraria, per la quale è più conveniente acquistare energia nella fascia oraria notturna, eventualmente immagazzinandola nella batteria per poi usarla di giorno se il tradeoff con la perdita di carica nel tempo e l'uso intelligente che deve venirne fatto è giudicato conveniente; oltre a ciò, la batteria diventa fondamentale anche negli intervalli di tempo in cui la produzione di potenza da parte delle sorgenti rinnovabili supera la richiesta all'interno della grid. In queste ultime situazioni, infatti, se non si avesse a disposizione la batteria, il controllore della grid non potrebbe far altro che vendere l'energia in

eccesso ad ENEL che, come sopra menzionato, paga quella venduta dagli utenti meno di quella comprata dai medesimi.

Per quanto riguarda poi il costo legato all'ammortamento della spesa iniziale per l'acquisto della batteria, si può scegliere che esso coincida o meno con il valore che la funzione di costo assume all'istante temporale 0-esimo: di seguito si suppone che

$$C_{X_0, l} = 0 \quad \forall X_0 \in \mathcal{X}, \forall l \in \{0.00, 1.00, \dots, 23.00\}. \quad (4.5)$$

Ogni volta poi che il contatore K definito nella sezione precedente tocca lo zero, la funzione di costo subisce un incremento fortemente non lineare legato alla spesa per l'acquisto di una nuova batteria

$$C_{X_k, l} \stackrel{K=0}{=} C_{X_k, l} + C_{\text{batt}}. \quad (4.6)$$

D'ora in avanti, assumendo come impliciti la dipendenza della funzione di costo dalla fascia oraria l in cui si colloca un determinato intervallo temporale k ed il fatto che nel corso di esso ci sia dipendenza dalla corrispondente richiesta di potenza X_k , essa verrà indicata semplicemente come C_k e l'obiettivo che ci si pone è quello di minimizzarne la somma su un certo orizzonte temporale (finito).

La funzione di costo non lineare definita da (4.6), tuttavia, risulta di non immediata implementazione nel modello di sistema che si sta andando a definire. Si è perciò preferito andare a spalmare il costo C_{batt} legato all'ammortamento della spesa iniziale della batteria su ogni intervallo temporale, legandolo al suo tempo di vita medio K , espresso in numero di cicli, nel modo seguente

$$C_k = C_{k-1} + C_{\text{batt}, k} = C_{k-1} + \frac{C_{\text{batt}}}{K} \frac{|E_k - E_{k-1}|}{E_{\text{max}}} \quad (4.7)$$

dove $\frac{C_{\text{batt}}}{K}$ rappresenta il costo di *un ciclo* (che si suppone essere *equivalente a caricare/scaricare la batteria di E_{max} in valore assoluto*) e $\frac{|E_k - E_{k-1}|}{E_{\text{max}}}$ corrisponde alla percentuale di carica/scarica rispetto alla capacità della batteria.

Cercando di riassumere il tutto nel modo più sintetico possibile, si definisce la funzione di costo complessiva del sistema, che tenga conto sia di (4.4) che di (4.7), la quale, con l'ausilio della variabile $X_k = L_k - R_k + \frac{E_k - E_{k-1}}{\Delta t}$, diventa esprimibile nel modo seguente

$$C_{k, l}^{\text{new}} = \begin{cases} \mathcal{X}_{\{X_k \geq 0\}} c_0 c_a X_k + \mathcal{X}_{\{X_k < 0\}} c_1 c_v X_k + C_{\text{batt}, k}, & l \in f^0 \\ \mathcal{X}_{\{X_k \geq 0\}} c_1, c_a X_k + \mathcal{X}_{\{X_k < 0\}} c_2 c_v X_k + C_{\text{batt}, k}, & l \in f^1. \end{cases} \quad (4.8)$$

dove $\mathcal{X}_{\{\mathcal{A}\}}(x)$ è definita come la funzione indicatrice di un certo insieme \mathcal{A} , ovvero

$$\mathcal{X}_{\{\mathcal{A}\}}(x) = \begin{cases} 1, & x \in \mathcal{A} \\ 0, & \text{altrimenti.} \end{cases} \quad (4.9)$$

Infine, nella definizione del nuovo problema di ottimizzazione nella Sezione 4.2 non verrà esplicitata la dipendenza della funzione di costo $C^{\text{new}}(\cdot)$ dal secondo pedice l corrispondente ad una certa fascia oraria di appartenenza: viene implicitamente assunto di trovarsi sempre o in f^0 o in f^1 , anche se ciò non è ben osservabile dal punto di vista analitico per non appesantire eccessivamente la notazione. Ciò viene invece preso in considerazione in fase di implementazione, dove, dopo che è finito ogni intervallo temporale di durata Δt , viene aumentato un contatore, per poter sempre calcolare il costo dell'energia da comprare/vendere in funzione dell'ora della giornata in cui ci si trova.

4.2 Definizione del nuovo problema di ottimizzazione

Prima di entrare nel merito della definizione del nuovo problema di minimizzazione, vale la pena inserire qui una piccola osservazione sulla generazione della sequenza $\{P_k\}$ della richiesta *netta* di potenza. Nella Sezione 4.1.2 si è osservato come ora si assuma che il segnale P_k possa assumere anche *valori negativi*, ad indicare che in tale intervallo temporale $R_k > L_k$. Nella Sezione 1.5 è stato descritto un modello accurato per generare una richiesta di potenza relativa ad una ventina di abitazioni all'interno della grid che sia la più verosimile possibile. Per non avere troppe nuove variabili da prendere in considerazione nell'analisi dei risultati di questo nuovo problema di ottimizzazione (ci sono già il nuovo modello di batteria ed una funzione di costo completamente diversa), si è preferito *non* cercare in letteratura ed utilizzare un modello di equivalente accuratezza per la generazione di energia da parte delle sorgenti rinnovabili. Per raggiungere l'obiettivo che il segnale P_k assumesse anche valori negativi, è stata creata con l'ausilio di Matlab una sequenza $\{R_k\}$ di campioni della produzione di energia rinnovabile con valori uniformemente distribuiti tra 0 e 30 kW con passo $\Delta R = 0.5$ kW durante la fascia oraria f_0^0 ed uguali a zero nel corso della fascia oraria f_0^1 . Un piccolo dettaglio tecnico, aggiunto per rendere questo segnale più verosimile, è di più facile comprensione leggendo 4.1 con il corrispondente codice Matlab che cercando di descriverlo a parole: in pratica si è fatto in modo che l'inizio della produzione di energia alla mattina e la fine al tramonto abbiano un andamento più *smooth* di quello ottenibile tramite il passaggio da un campione nullo ad una distribuzione $\sim \mathcal{U}(0, 0.5, \dots, 30)$ kW.

Codice 4.1: Codice Matlab per generare la sequenza $\{R_k\}$ che memorizza i valori relativi alla produzione di potenza tramite pannelli fotovoltaici.

```
Delta_R=0.5e3;
R_n=0;
R_g=[1e3:Delta_R:30e3];
ll=31*24*60;
R_GA_pre_f=zeros(ll,1);
for k=1:ll
    R_GA_pre_f(k)=R_g(1+floor(length(R_g)*rand));
end
R_GA=R_n*ones(ll,1);
for k=0:24:(ll-1)
    R_GA(8+k+1)=round(R_GA_pre_f(8+k+1)/8/Delta_R)*Delta_R;
    R_GA(8+k+2)=round(R_GA_pre_f(8+k+2)/2/Delta_R)*Delta_R;
    R_GA((8+k+3):(20+k-2))=R_GA_pre_f((8+k+3):(20+k-2));
    R_GA(20+k-1)=round(R_GA_pre_f(20+k-1)/2/Delta_R)*Delta_R;
    R_GA(20+k)=round(R_GA_pre_f(20+k)/8/Delta_R)*Delta_R;
end
```

Detto ciò, la finalità che ci si pone a livello di ottimizzazione stocastica rimane quella di minimizzare il costo complessivo da pagare all'operatore energetico, sfruttando la capacità offerta dalla batteria di immagazzinamento di energia in istanti temporali giudicati convenienti (ricordando poi che, quando $P_k < 0$, l'energia, se non potesse essere caricata nella batteria, verrebbe per forza venduta all'operatore energetico ad un prezzo di vendita inferiore a quello di acquisto), tenendo però in considerazione anche il costo legato all'acquisto della batteria (e alle sue successive sostituzioni per il suo inevitabile degrado), il nuovo vincolo legato alla sua perdita di energia nel tempo e una funzione di costo completamente diversa da quella precedente,

che prende in considerazione anche la fascia oraria a cui appartiene un determinato intervallo temporale. Supponendo di partire all'istante $k = 0$ con la batteria scarica (ovvero $E_0 = 0$), si ha che

$$\{E_{\text{opt}_k}\}_{k=1,2,\dots,T-1} = \underset{\{E_k\}_{k=1,2,\dots,T-1}}{\text{arg min}} \quad \mathbb{E} \left[\sum_{k=0}^{T-1} C_k^{\text{new}} \right] \quad (4.10)$$

subject to : (4.11), (4.1)

dove $C^{\text{new}}(\cdot)$ è la funzione di costo definita da (4.8) e

$$-\frac{\Delta t F_{\text{max}}^d}{\eta_d} \leq E_{k+1} - E_k \leq F_{\text{max}}^c \Delta t \eta_c. \quad (4.11)$$

è ancora il vincolo definito nel Capitolo 1 sulla capacità limitata a E_{max} , i rate di carica e scarica finiti e le efficienze di carica e scarica $\neq 1$ della batteria.

Assodato che il problema di ottimizzazione offline associato a (4.10) si può ancora risolvere applicando l'algoritmo di Viterbi (di cui per brevità vengono omessi i nuovi dettagli implementativi), tutta la teoria sulla DP definita nel Capitolo 2 rimane perfettamente valida anche per la risoluzione con il nuovo modello di sistema, prestando però particolare attenzione al seguente aspetto: anche se, come già accennato, la dipendenza dalla fascia oraria ($l \in \{fo^1, fo^2\}$) non viene esplicitata nella funzione di costo in (4.10), essa influisce in modo drastico sulla determinazione della strategia ottima π . Ci sono più modi per risolvere questa complicazione analitica: uno sarebbe quello di inserire come terza componente del vettore di stato l'intervallo temporale esatto (tra i 24 possibili con granularità $\Delta t = 1$ h) relativa ad un generico giorno al quale sono associati un certo livello energetico E_j ed una certa richiesta di potenza P_i . Questo approccio, tuttavia, presenta il difetto di aumentare in modo significativo la cardinalità dello spazio degli stati e soprattutto di avere a disposizione un numero molto limitato di valori di training su cui stimare le corrispondenti probabilità di transizione. Una soluzione alternativa è quella di considerare come terza componente dello stato la fascia oraria fo^l per $l = 0, 1$, che diventa in tal modo esprimibile come

$$\mathbf{s} = (E_j, P_i, fo^l) = (s^{(1)}, s^{(2)}, s^{(3)}) \in \mathcal{S} \quad \text{con } \mathcal{S} = \mathcal{E} \times \mathcal{P} \times \mathcal{FO}. \quad (4.12)$$

Così facendo, quando si va a stimare la statistica condizionata dello stato del sistema, le transizioni dipendono sia dalla seconda che dalla terza componente dello stato. In particolare, nella stima della distribuzione di probabilità, alla fine degli intervalli con $l \in \{8.00, 9.00, \dots, 20.00\}$ le corrispondenti transizioni vanno conteggiate tra i possibili stati con $fo^l = fo^0$; alla fine, invece, degli intervalli con $l \in \{22.00, 23.00, \dots, 6.00\}$ le corrispondenti transizioni vanno contate tra gli stati con terza componente $fo^l = fo^1$; in maniera analoga, alla fine dell'intervallo con $l = 21.00$ si verificano le transizioni tra gli stati con $fo^l = fo^0$ a quelli con $fo^l = fo^1$ ed alla fine dell'intervallo con $l = 7.00$ si verificano le transizioni tra gli stati con $fo^l = fo^1$ a quelli con $fo^l = fo^0$. L'idea alla base di quanto appena spiegato è ben rappresentata in Figura 4.4.

Ad essere precisi, la Figura 4.4 rappresenta solo in modo compatto (per non rendere difficoltosa la visualizzazione) la *catena di Markov a stati finiti associata alla stima delle nuove probabilità di transizione*. Nelle catena di Markov "effettiva", infatti, il rettangolo rosso sulla sinistra dovrebbe avere sette repliche affiancate degli M circoletti (che rappresentano i possibili livelli assumibili dalla richiesta netta di potenza quantizzata durante la fascia oraria notturna): per le prime sei repliche è possibile andare da ciascuno degli M circoletti a ciascuno degli M successivi, e così anche per la settima replica, solo che in questo caso le transizioni terminano in uno degli stati del rettangolo evidenziato in azzurro. Il ragionamento si estende in modo naturale a quest'ultimo,

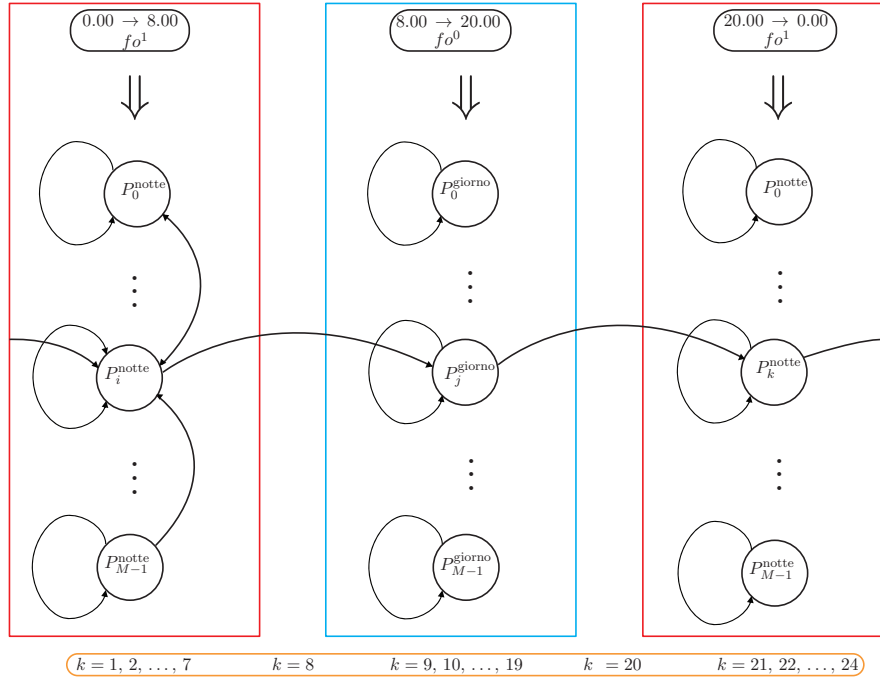


Figura 4.4: Schema che descrive la parte dello stato del sistema rispetto alla coppia richiesta di potenza-fascia oraria al variare del tempo in una generica giornata di 24 h.

con gli M circoletti (che rappresentano i possibili livelli assumibili dalla richiesta netta di potenza quantizzata durante la fascia oraria diurna) che dovrebbero essere presenti in undici repliche affiancate, con le transizioni che partono dagli stati appartenenti all'ultima di esse che entrano di nuovo negli M circoletti (da espandere in quattro repliche affiancate) relativi alla fascia oraria notturna e così via, fino a tornare nella M -upla più a sinistra della catena di Markov che emula il processo di stima delle probabilità di transizione.

L'applicazione della DP è possibile stimando le appena citate probabilità di transizione con la nuova definizione di stato, in maniera analoga a quanto descritto nella Sezione 3.1: la differenza fondamentale sta nel fatto che ora la matrice \mathbf{P}_{tr} , che memorizza i valori delle probabilità condizionate, ha dimensioni $2M \times 2M$ (ovvero il quadruplo della precedente matrice) ed è associata alle probabilità di transizione esprimibili come

$$\begin{aligned}
 & \Pr \left\{ P_{k+1} = P_i, fo = fo^l \mid P_k = P_j, fo = fo^m \right\} = \\
 & = \Pr \left\{ P_{k+1} = P_i, fo = fo^l, P_k = P_j, fo = fo^m \right\} / \Pr \left\{ P_k = P_j, fo = fo^m \right\} \quad (4.13) \\
 & \longleftrightarrow \mathcal{P}_{\{P_i, fo=fo^l, P_j, fo=fo^m\}} \quad \text{per } P_i, P_j \in \mathcal{P} \text{ e } fo^l, fo^m \in \{fo^0, fo^1\}.
 \end{aligned}$$

4.3 Risultati delle simulazioni numeriche relative alla determinazione della strategia ottima tramite DP col nuovo modello di sistema

Il setting con cui si sono svolte le simulazioni è del tutto analogo a quello della Sezione 3.2. Per quanto riguarda la definizione dei parametri della funzione di costo, essi sono riassunti dalla

Tabella 4.1.

Tabella 4.1: Coefficienti con cui è definita $C^{\text{new}}(\cdot)$ in funzione della potenza X (che comprende il contributo della batteria) e della fascia oraria.

Caso	Formula parametrica	Formula con valori effettivi: $C_{k,l}^{\text{new}}$
$X_k \geq 0$ e $l = 0$	$c_0 c_a X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_{k,l}^{\text{new}} = 10 \cdot 5 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$
$X_k < 0$ e $l = 0$	$c_0 c_v X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_{k,l}^{\text{new}} = 10 \cdot 1 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$
$X_k \geq 0$ e $l = 1$	$c_1 c_a X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_{k,l}^{\text{new}} = 9 \cdot 5 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$
$X_k < 0$ e $l = 1$	$c_1 c_v X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_{k,l}^{\text{new}} = 9 \cdot 1 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$

Anche per illustrare i risultati delle simulazioni effettuate, si segue la stessa linea guida usata nel Capitolo 3: prima quelli relativi ad una statistica della distribuzione di transizione dello stato (da intendersi, in riferimento a (4.13), come coppia richiesta netta di potenza-fascia oraria di appartenenza) supposta iid e poi quelli ad una distribuzione supposta markoviana.

4.3.1 Risultati relativi al caso con una sola fascia oraria

Per catturare le differenze nei risultati numerici date dalla scelta della nuova funzione di costo rispetto alla Sezione 3.2, è stato implementato, inizialmente, il nuovo modello di sistema con un'unica fascia oraria, la cui funzione di costo può essere descritta dalla Tabella 4.2.

Tabella 4.2: Coefficienti con cui è definita $C^{\text{new}}(\cdot)$ con un'unica fascia oraria in funzione della potenza X (che comprende il contributo della batteria).

Caso	Formula parametrica	Formula con valori effettivi: C_k^{new}
$X_k \geq 0$	$c_a X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_k^{\text{new}} = 5 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$
$X_k < 0$	$c_v X_k + \frac{C_{batt}}{K} \frac{ E_k - E_{k-1} }{E_{max}} \implies$	$C_k^{\text{new}} = 1 X_k + \frac{10e3}{5e3} \frac{ E_k - E_{k-1} }{12e3}$

Analogamente al Capitolo 3, nelle Figure 4.5 e 4.6 si è scelto di riportare l'andamento della strategia ottima π , ricavata tramite DP, supponendo prima una distribuzione iid delle probabilità di transizione relative alla sequenza $\{P_k\}$ (che ora però può assumere anche valori negativi), al variare sia del valore della richiesta netta di potenza che del livello di carica presente nella batteria e poi al variare soltanto di quest'ultimo, parametrizzando la curva per alcuni valori di P_i valutati come significativi, scelti cioè col criterio di poter osservare la massima variabilità della *policy*, che per la maggior parte dei valori non riportati si comporta in modo molto più stabile.

Anche se soltanto nel Capitolo 5 si cercherà di dare una spiegazione analiticamente rigorosa dei risultati trovati, confrontando, ad esempio, la Figura 3.4 con la Figura 4.6 salta subito all'occhio una forte analogia: le curve che descrivono la strategia ottima π al variare del livello di carica E_j continuano ad avere un andamento monotono non-crescente. Che tale caratteristica sia preservata (la strategia ottima, parametrizzata per un certo valore di P_i , ha ancora un andamento lineare a tratti non-crescente all'aumentare della quantità di energia presente nella batteria) risulta molto interessante per due motivi: innanzitutto il nuovo modello di batteria tiene in conto il fatto che il livello di energia immagazzinato diminuisca nel tempo, e ciò complica la decisione ottima riguardo all'azione da intraprendere in un certo intervallo temporale; la nuova funzione di costo, inoltre, prende anche in considerazione l'ammortamento della spesa

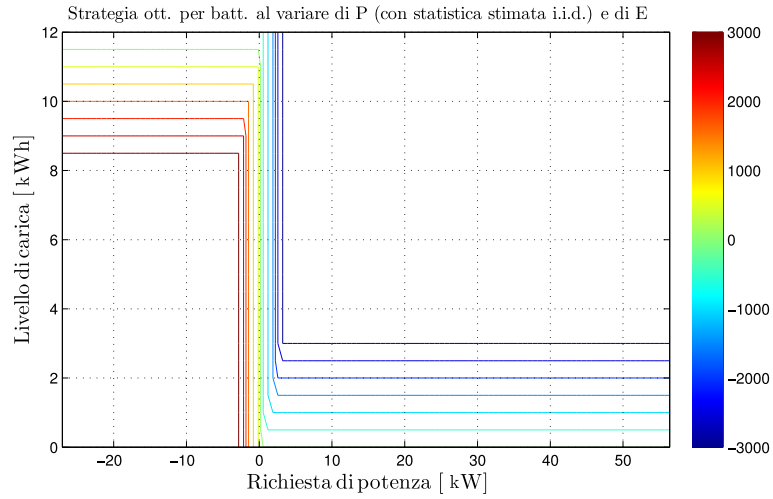


Figura 4.5: Strategia ottima π ricavata tramite DPI applicata al modello di costo con un'unica fascia oraria al variare sia del segnale P_i che del livello di energia E_j presente nella batteria.

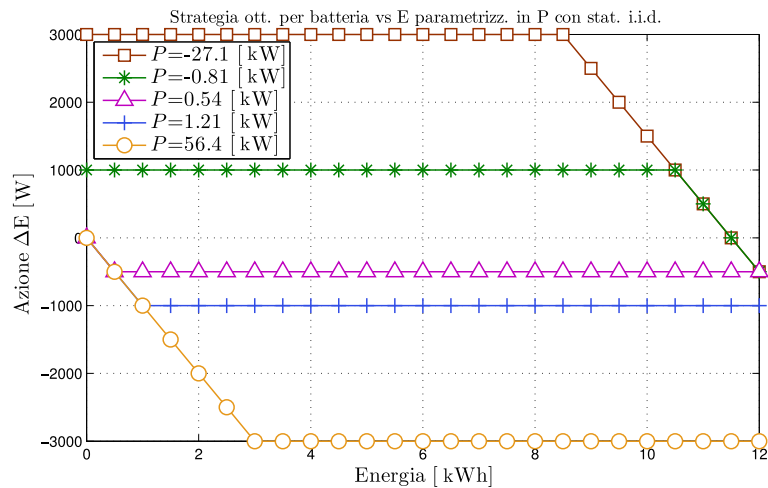


Figura 4.6: Strategia ottima π ricavata tramite DPI applicata al modello di costo con un'unica fascia oraria al variare del livello di energia E_j presente nella batteria, parametrizzata per alcuni valori di P_i .

iniziale della batteria, il cui tempo di vita dipende dalla profondità delle azioni di carica/scarica intraprese: anche questo fattore potrebbe stravolgere la forma della strategia ottima π (se ciò non si nota osservando la Figura 4.5 è solo una conseguenza del settaggio dei parametri scelto, che non enfatizza l'effetto dell'ammortamento affinché l'acquisto della batteria risulti comunque conveniente). Nelle Figura 4.7 vengono riportati, in forma compatta, degli andamenti del tutto analoghi, con l'unica variazione che la strategia ottima π è stata calcolata supponendo una distribuzione markoviana delle probabilità di transizione della sequenza $\{P_k\}$.

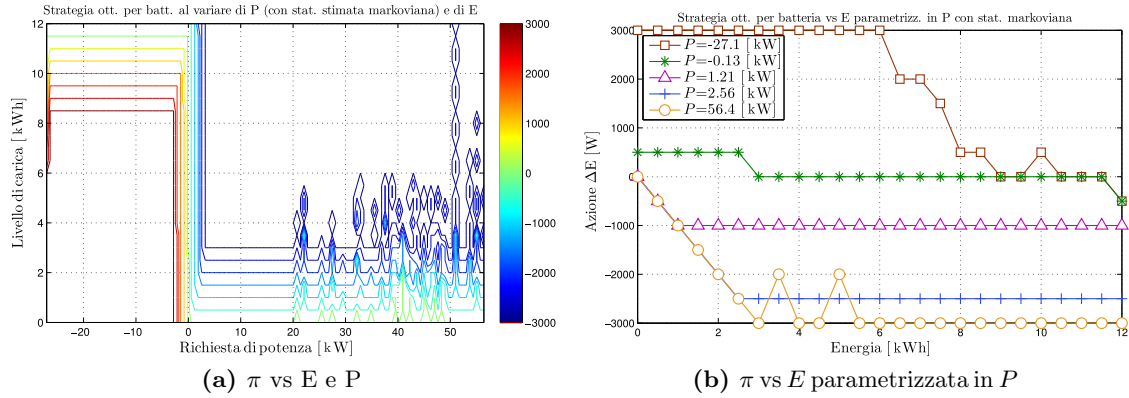


Figura 4.7: Strategia ottima π ricavata tramite DPM applicata al modello di costo con un'unica fascia oraria.

Nel caso con stima della statistica supposta markoviana, invece, si osserva già dalla Figura 4.7a che l'andamento della *policy* ottima risulti molto complesso da descrivere per valori elevati della richiesta di potenza. Come si vede chiaramente dalla Figura 4.7b, che rappresenta la strategia ottima, parametrizzata in P_i , al variare del livello di energia della batteria, essa non risulta propriamente monotona non-crescente, presentando diversi micro-picchi, e ciò, a colpo d'occhio, non si sposa bene con le dimostrazioni analitiche che verranno riportate nel Capitolo 5, secondo le quali sia l'aver introdotto l'ammortamento della batteria nella funzione costo che il prendere in considerazione la perdita di carica del tempo (quando questa risulta una funzione *convessa* del livello di energia E_j) non dovrebbero alterare la forma della strategia ottima, la quale dovrebbe preservarsi lineare a tratti e monotona non-crescente. Simulazioni numeriche ad hoc, qui non riportate, con uno solo di questi due fattori attivo, hanno verificato che è soltanto la perdita di carica nel tempo a determinare la non completa monotonicità di π di cui si sta discutendo, e non l'ammortamento del costo iniziale della batteria. La funzione di perdita di carica nel tempo, tuttavia, è sì convessa rispetto al livello di energia presente nella batteria (ed il problema complessivo rimane così un problema di ottimizzazione convessa), ma ciò vale senza considerare gli *effetti della quantizzazione* a $\Delta E = 0.5$ kWh, che fa sì che la funzione di perdita quantizzata abbia un andamento analogo al gradino, 1_k , a tempi discreti, e quindi *non convessa*: questa spiega il disaccordo con le dimostrazioni matematiche riportate nel Capitolo 5. Un'altra spiegazione che si concilia con il rigore della matematica è che, per un numero di campioni a disposizione non sufficientemente elevato, la stima delle probabilità di transizione supponendo una statistica markoviana potrebbe non essere molto accurata, e generi tali micro-picchi come artefatto.

Seguendo sempre la stessa linea guida del Capitolo 3, disponendo delle due strategie ottime, si è scelto confrontarle in termini di costo medio istantaneo, estendendo il paragone anche ad i costi associati alle soluzioni fornite dall'algorithm AOS, dall'algorithm ST e al caso senza batteria. I risultati ottenuti in termini puramente numerici riguardo al costo complessivo sono riportati nella

Tabella 4.3; gli andamenti dei corrispondenti costi medi istantanei, invece, sono rappresentati in Figura 4.8 e 4.9 al variare dell'ora dei giorni del mese di gennaio.

Tabella 4.3: Risultati numerici sul costo complessivo con un'unica fascia oraria con granularità $\Delta t = 1$ h.

Descrizione	Costo ottenuto (o % di aumento)
lower bound sul costo calcolato con alg. AOS	1.97775e+08
aumento relativo del costo con alg. DPI	0.0977069 %
aumento relativo del costo con alg. DPM	0.107773 %
aumento relativo del costo con alg. ST	2.14064 %
aumento relativo del costo nel caso senza batteria	3.51835 %

Più che l'andamento grafico successivamente riportato, è questa tabella (come anticipato nel Capitolo 3) a confermare come considerare anche come approssimazione della distribuzione di probabilità condizionata delle richieste di potenza una statistica iid risulti, per alcune funzioni di costo, una scelta più che valida: il costo complessivo, rispetto al caso markoviano, risulta in questo caso *leggermente inferiore*.

Osservando poi le percentuali di aumento relativo del costo rispetto al lower bound di Viterbi riportate in Tabella 4.3 e confrontandole con quelle della Tabella 3.1 risulta evidente la minor variabilità delle %: il costo corrispondente alle strategie ottenute tramite DP si distanzia soltanto per qualche centesimo di punto dal minimo possibile e anche quello relativo al caso senza batteria è diverso solo di pochi punti percentuali. La giustificazione è molto semplice ed è dovuta all'assenza dell'andamento lineare a tratti e con coda quadratica: l'andamento puramente lineare, infatti, appiattisce tutti i costi cumulativi; la batteria, inoltre, risulta uno strumento meno potente per immagazzinare energia in intervalli temporali giudicati convenienti, perchè il confine tra conveniente e non conveniente non è più un andamento lineare con pendenza modesta, andamenti lineari con pendenza elevata o addirittura un andamento quadratico, ma si rimane sempre su andamenti lineari con pendenze paragonabili. Chiarito ciò, la fortissima vicinanza della π ricavata con la programmazione dinamica alla strategia offline ottima non può che essere una buona notizia, nel senso che garantisce che, implementandola nel controllore della grid, essa porti a prestazioni in termini di costo da pagare da parte degli utenti vicinissime al minimo possibile.

Il costo relativo al caso iid, infine, risulta lievemente inferiore a quello del caso con una stima della statistica di $\{P_k\}$ di tipo markoviano, per i motivi già citati nella Sezione 3.2: in questo caso, tuttavia, non si hanno evidenti benefici nel considerare la generazione di campioni relativi alla richiesta di potenza L_k con granularità di 1 minuto, come riportato nella Tabella 4.4.

4.3.2 Risultati relativi al caso con due fasce orarie

Avendo già riportato il setting dei parametri della funzione di costo nella Tabella 4.1 ed avendo eseguito uno step simulativo intermedio nella Sezione 4.3.1, non resta che continuare a seguire la linea guida della Sezione 3.2 per illustrare i nuovi risultati numerici. Va notato che, con la doppia fascia oraria, sono stati definiti due nuovi coefficienti nella funzione di costo, che rappresentano il costo per unità di energia scambiata (da intendersi come acquistata o venduta) quando ci si trova in un intervallo temporale appartenente rispettivamente a fo^0 o fo^1 . Anche in questo caso dei valori esemplificativi dei coefficienti sono stati scelti in maniera euristica, tenendo però presente la seguente importante considerazione: *fissata la funzione di costo e le caratteristiche della batteria, bisogna stabilire quando conviene acquistarla*. Premesso che ciò verrà approfondito

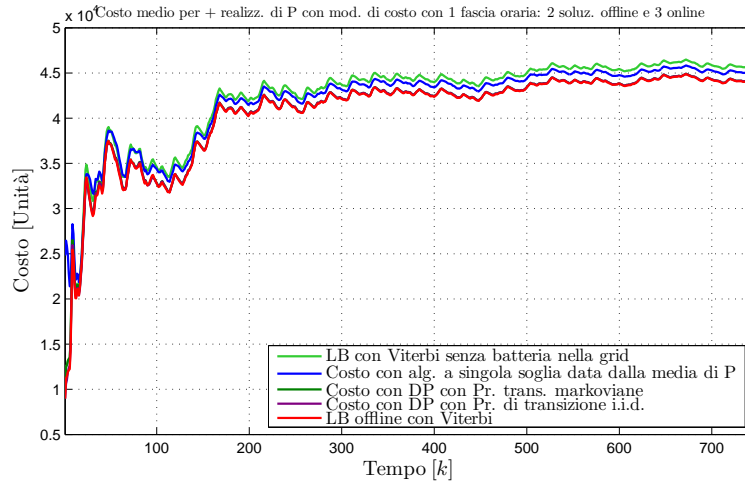


Figura 4.8: Andamento (complessivo) del costo medio istantaneo con un'unica fascia oraria in funzione dell'ora del giorno dei mesi di gennaio per i 4 algoritmi analizzati.

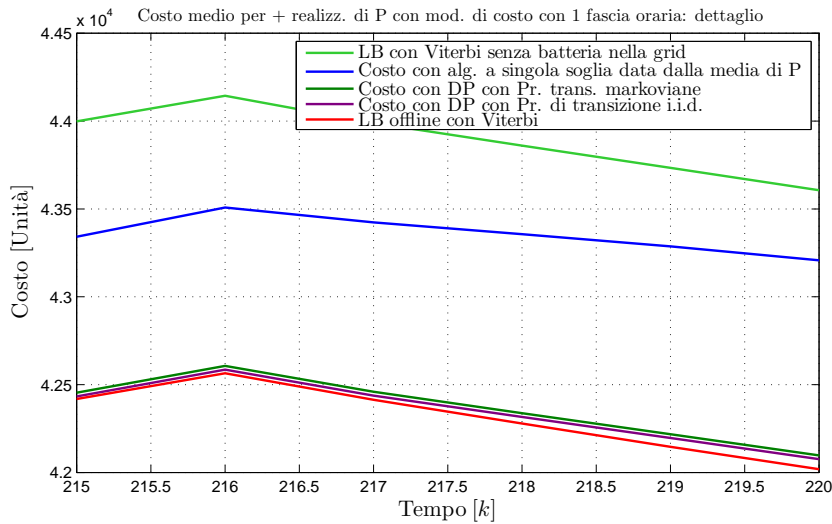


Figura 4.9: Dettaglio dell'andamento del costo medio istantaneo con un'unica fascia oraria in funzione dell'ora del giorno dei mesi di gennaio per i 4 algoritmi analizzati.

Tabella 4.4: Risultati numerici sul costo complessivo calcolato sulla sequenza $\{P_k^{new}\}$ (avente una granularità temporale $\Delta t = 1'$) e con una funzione di costo con un'unica fascia oraria.

Descrizione	Costo ottenuto (o % di aumento)
lower bound sul costo calcolato con alg. AOS	2.22871e+09
aumento relativo del costo con alg. DPM	0.162545 %
aumento relativo del costo con alg. DPI	0.177809 %
aumento relativo del costo con alg. ST	2.18522 %
aumento relativo del costo nel caso senza batteria	2.57575 %

analiticamente nel Capitolo 5, ci si può intanto accontentare di una considerazione più intuitiva: rispetto alla variazione del livello energetico δ della batteria, la funzione di costo $C(\cdot)^{\text{new}}$ risulta lineare con due costanti di proporzionalità che possono valere, rispettivamente, $a_1 = c_0 \cdot c_a$, $a_2 = c_0 \cdot c_v$, $a_3 = c_1 \cdot c_a$ e $a_4 = c_1 \cdot c_v$ per quanto riguarda il termine che rappresenta l'azione di carica/scarica e $b = \frac{C_{\text{batt}}}{K} \frac{1}{E_{\text{max}}}$ per quanto riguarda il termine che tiene conto dell'ammortamento della spesa iniziale per l'acquisto della batteria. Considerando la derivata di questi due termini rispetto a δ , risulta immediato convincersi che possa essere conveniente procedere all'acquisto della batteria (in quanto considerazione necessaria ma non sufficiente: se ne deve anche fare un uso intelligente) nel caso in cui $a_i \geq b \forall i = 2, 4$ (la batteria risulta infatti un'alternativa al vendere l'energia prodotta all'operatore energetico). Dai dati riportati in Tabella 4.2, basta una semplice divisione tra coefficienti per convincersi di essere in quest'ultimo caso (e di svariati ordini di grandezza).

Detto ciò, nelle Figure 4.10, 4.11, 4.12 e 4.13 si riportano i risultati relativi alle strategie ottime ottenute tramite DP sia nella fascia oraria diurna che in quella notturna.

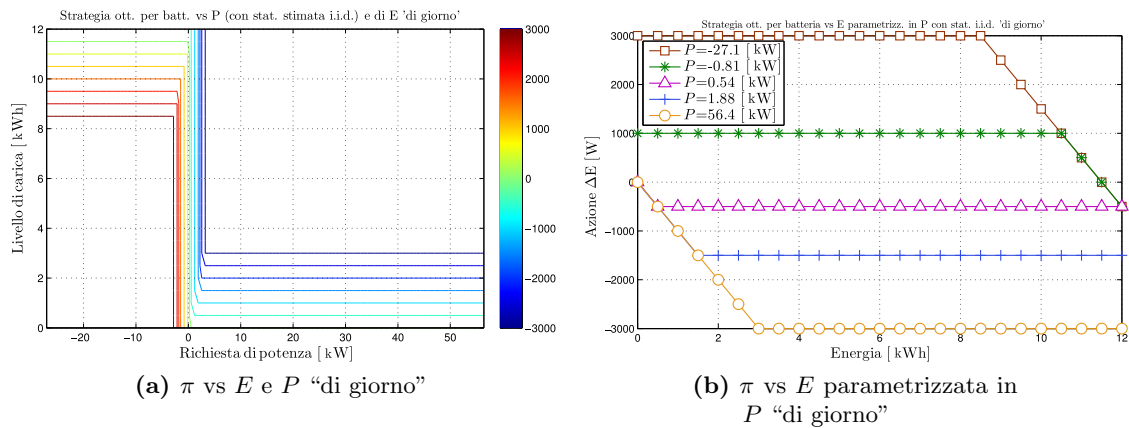


Figura 4.10: Strategia ottima π “di giorno” ricavata tramite DPI applicata al modello di costo con 2 fasce orarie.

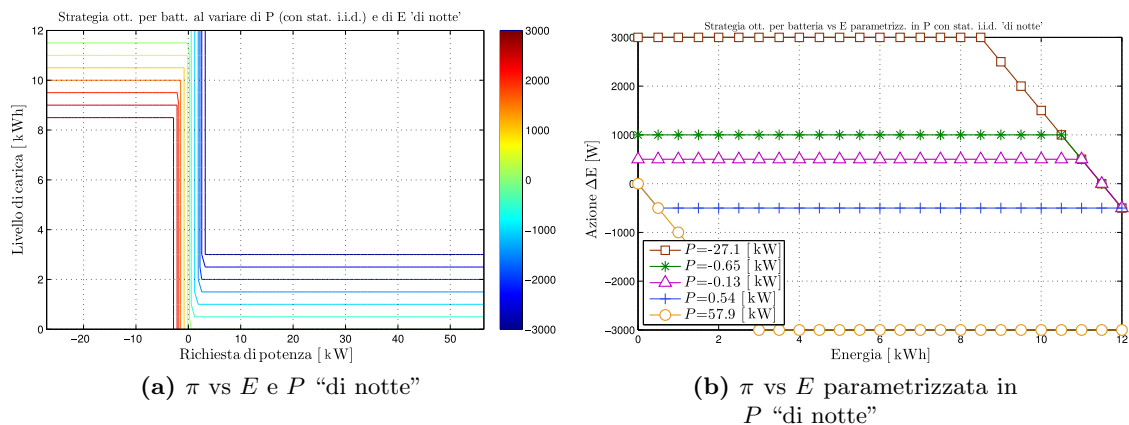


Figura 4.11: Strategia ottima π “di notte” ricavata tramite DPI applicata al modello di costo con 2 fasce orarie.

Non ci sono novità sostanziali in tali figure rispetto a quelle precedentemente riportate per quanto riguarda il caso iid, illustrato nelle Figure 1.18b e 1.20b: l'andamento della strategia

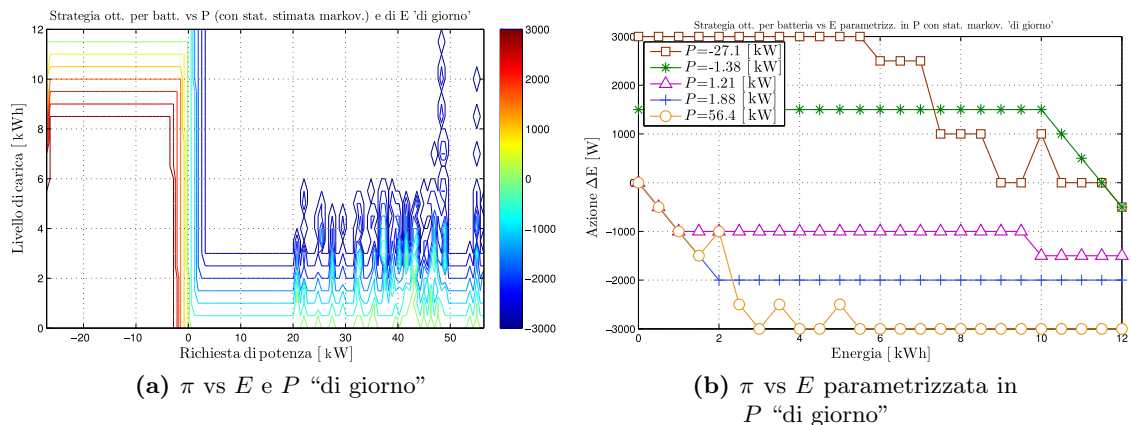


Figura 4.12: Strategia ottima π "di giorno" ricavata tramite DPM applicata al modello di costo con 2 fasce orarie.

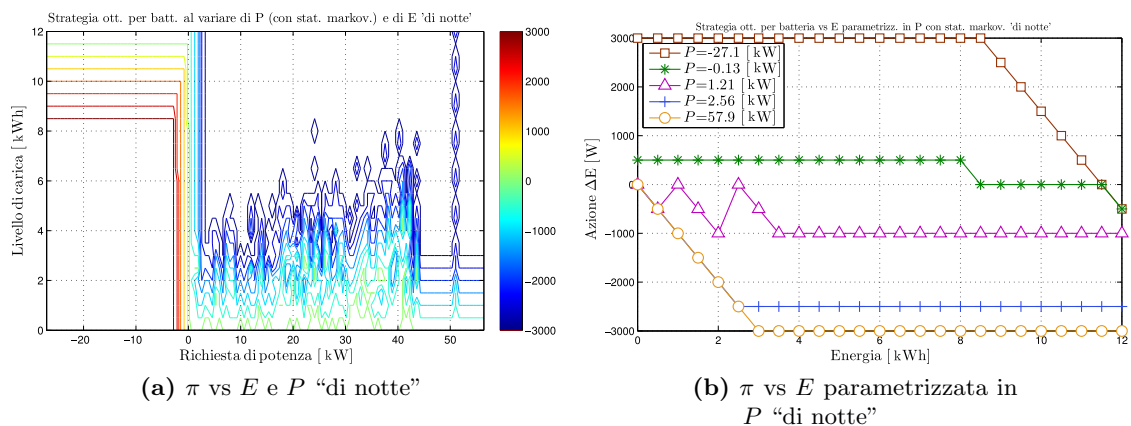


Figura 4.13: Strategia ottima π "di notte" ricavata tramite DPM applicata al modello di costo con 2 fasce orarie.

ottima risulta infatti sempre lineare a tratti e monotono non-crescente. Per quanto riguarda, invece, il caso con stima della statistica supposta markoviana, valgono le stesse considerazioni discusse nella Sezione 4.3.1 relativamente alla stima della statistica della richiesta netta di potenza quando viene supposta markoviana: come si evince chiaramente dalle Figure 4.12b e 4.13b, le strategie ottime al variare del livello di energia della batteria e parametrizzate in P_i non risultano propriamente monotone non-crescenti, presentando diversi micro-picchi, in corrispondenza dei quali la *policy* ottima π^* diventa localmente crescente, sempre a causa della *non-convessità della funzione di perdita quantizzata*.

Per quanto riguarda i risultati riguardo al costo numerico complessivo e al costo medio istantaneo, corrispondenti alle strategie di DP e agli altri algoritmi con cui le si confronta, essi sono riportati, rispettivamente, nella Tabella 4.5 e nella Figure 4.14 e 4.15.

Tabella 4.5: Risultati numerici sul costo complessivo con 2 fasce orarie con granularità $\Delta t = 1$ h.

Descrizione (o % di aumento)	Costo ottenuto
lower bound sul costo calcolato con alg. AOS	1.87428e+09
aumento relativo del costo con alg. DPI	0.484936 %
aumento relativo del costo con alg. DPM	0.512283 %
aumento relativo del costo con alg. ST	2.26700 %
aumento relativo del costo nel caso senza batteria	3.96025 %

In merito alla Tabella 4.5 ci sono un paio di osservazioni interessanti da fare. Innanzitutto l'incremento sul costo cumulativo corrispondente alla strategia ottenuta tramite DP non è più di qualche centesimo di punto percentuale, come nel caso della Sezione 4.3.1, ma arriva fino ad (un sempre modesto) 0.51 % nel caso di una stima markoviana della statistica della richiesta di potenza: la ragione di ciò è, presumibilmente, imputabile al limitato numero di campioni relativi alle transizioni di fascia oraria notte-giorno e giorno-notte su cui sono state stimate le probabilità di transizione di stato, ricordando che ora esse dipendono sia dal valore di P_i che dalla fascia oraria di appartenenza. Con una stima delle probabilità condizionate non molto precisa, è ovvio che anche la *state-value function* e quindi la strategia ottima π ne risentano, e ciò corrisponde ad un costo da pagare da parte degli utenti della grid leggermente più alto; l'aumento relativo del costo, tuttavia, è modesto, e il paradigma della programmazione dinamica rimane molto performante.

Infine, è stato verificato quanto il processo delle richieste di potenza, di cui è stata stimata la statistica del secondo ordine condizionata con $\Delta t = 1$ h supponendo che sia markoviano, *si possa approssimare come iid*. La metrica usata per quantificare questa approssimazione è la divergenza di Kullback-Leibler, che in statistica è usata come misura della distanza tra due PDF ed è in generale definita da

$$D(p \parallel q) = \sum_{a \in \mathcal{A}} p(a) \log_e \left(\frac{p(a)}{q(a)} \right). \quad (4.14)$$

In questo caso è stata usata la divergenza di Kullback-Leibler della distanza tra la distribuzione di massa degli stati con probabilità diversa da zero nella matrice della probabilità di transizione e quella della corrispondente distribuzione iid (relativa sempre ai possibili valori della richiesta netta di potenza con $\Pr \{ \cdot \} \neq 0$). Il frammento di codice Matlab che implementa il calcolo della divergenza di Kullback-Leibler è riportato nel Codice 4.2.

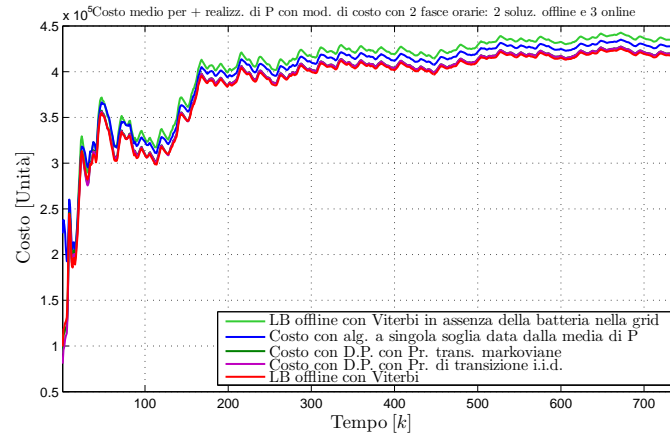


Figura 4.14: Andamento (complessivo) del costo medio istantaneo con 2 fasce orarie in funzione dell'ora del giorno dei mesi di gennaio per i 4 algoritmi analizzati.

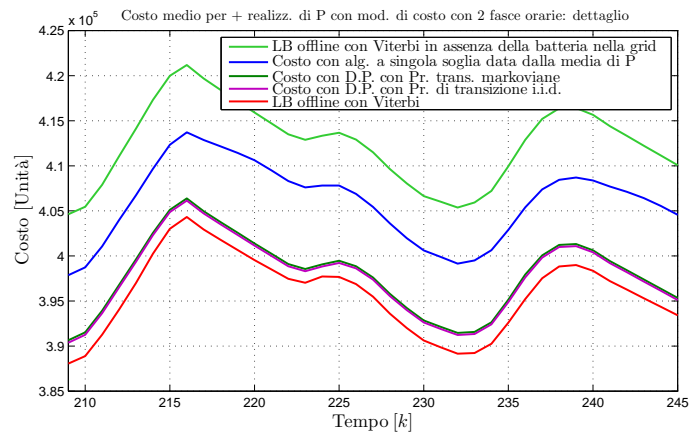


Figura 4.15: Dettaglio dell'andamento del costo medio istantaneo con 2 fasce orarie in funzione dell'ora del giorno dei mesi di gennaio per i 4 algoritmi analizzati.

Codice 4.2: Codice Matlab per il calcolo di un'opportuna divergenza di Kullback-Leibler per quantificare quanto la statistica del secondo ordine di $\{P_k\}$, supposta markoviana, si possa approssimare come iid.

```

KL=zeros(N_P_m,1);
for r=1:N_P_m
    for s=1:N_P
        if (PT_m(s,r)≠0 && P1(s)≠0)
            KL(r)=KL(r)+PT_m(s,r)*log(PT_m(s,r)/P1(s));
        end
    end
end
end

```

Nel Codice 4.2, “KL” rappresenta la divergenza di Kullback-Leibler che si sta cercando di stimare, “N_P_m” il numero di valori assumibili dalla richiesta netta di potenza la cui corrispondente probabilità è diversa da zero, “N_P” il numero totale di tali valori, “PT_m” la matrice che memorizza le stime delle probabilità di transizione ad un passo diverse da zero di $\{P_k\}$ e “P_1” la stima della distribuzione di massa del primo ordine di $\{P_k\}$. Tramite tale routine è stato possibile stimare la divergenza di Kullback-Leibler relativa alla sequenza delle richieste nette di potenza, andando a considerare separatamente i valori che cadono della fascia oraria diurna ed in quella notturna: i risultati ottenuti sono riportati, rispettivamente, nelle Figure 4.15 e 4.16.

Quello che si nota in modo evidente dalla Figura 4.17 è che la stima della divergenza di Kullback-Leibler durante le ore della fascia oraria fo^1 risulta prossima allo zero rispetto ad una buona parte dei valori assumibili da P_i . Quando invece la divergenza è $\neq 0$, avrebbe poco senso affermare che, da un punto di vista qualitativo, i suoi valori sono piccoli o grandi. È più interessante, invece, notare che, in corrispondenza alla parte centrale (relativa ad un insieme di valori assumibili dalla richiesta netta di potenza la cui somma delle rispettive probabilità risulta prossima ad 1: la zona più a sinistra e quella più a destra, quindi, sono di scarso significato rispetto a questa metrica stocastica) l'escursione della funzione è molto limitata o, in altre parole, la varianza di tali campioni è modesta. Ricordando che, se la statistica della sequenza $\{P_k\}$ fosse perfettamente approssimabile come iid la divergenza di Kullback-Leibler considerata sarebbe nulla, si può quindi concludere che, andando a mediare sull'ora i campioni di una sequenza generati con una granularità temporale di $\Delta t = 1$ ', il processo aleatorio che ne deriva ha una statistica che risulta descrivibile come iid con buona approssimazione, e che ciò si nota ancora di più durante la fascia oraria notturna. Per concludere, avendo a disposizione parecchia potenza di calcolo (da intendersi sia come memoria che come capacità computazionale), stimare la statistica con un modello markoviano - che ha memoria unitaria, e deve quindi essere definito attraverso la matrice delle probabilità di transizione, legata alla statistica del secondo ordine della richiesta netta di potenza - può portare ad un guadagno sul costo complessivo (derivante dell'applicazione della strategia ottima π) abbastanza significativo per determinate funzioni di costo. In caso contrario, mediando i valori della sequenza $\{P_k\}$ per lavorare con $\Delta t = 1$ h, approssimarne la statistica come iid - definita soltanto tramite la distribuzione di massa del primo ordine - risulta la scelta migliore, portando in genere ad un costo complessivo non molto superiore superiore

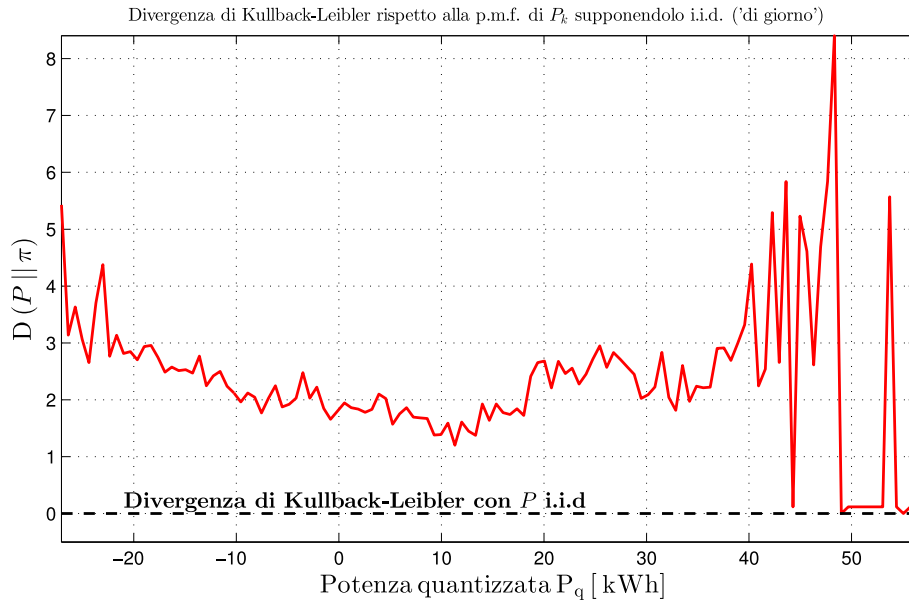


Figura 4.16: Stima della divergenza di Kullback-Liebler della distanza tra la matrice di transizione \mathbf{P}^{tr} corrispondente ad una stima della statistica di $\{P_k\}$ approssimata a markoviana e a quella con statistica approssimata ad iid (e quindi con tutte le colonne uguali alla distribuzione di massa del primo ordine) durante la fascia oraria diurna.

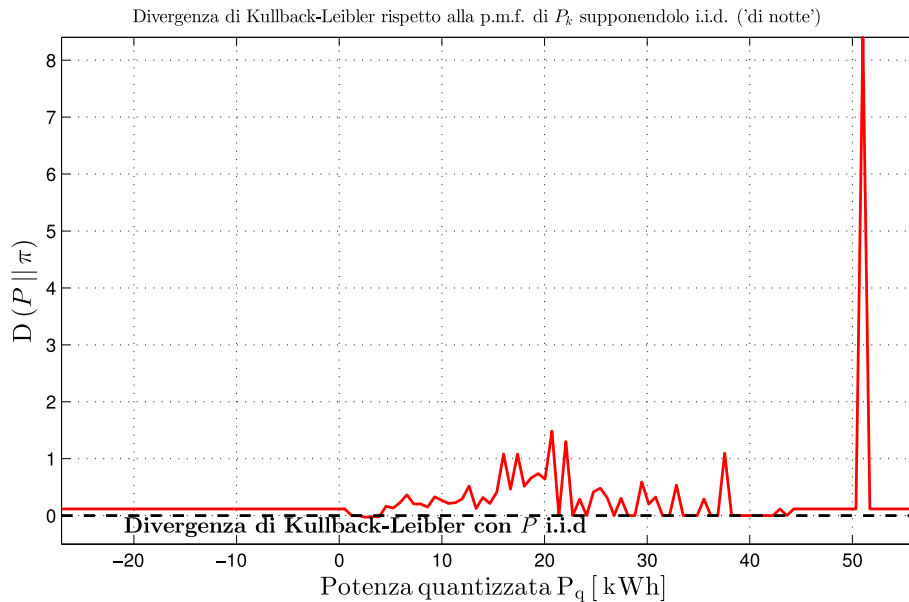


Figura 4.17: Stima della divergenza di Kullback-Liebler della distanza tra la matrice di transizione \mathbf{P}^{tr} corrispondente ad una stima della statistica di $\{P_k\}$ approssimata a markoviana e a quella con statistica approssimata ad iid (e quindi con tutte le colonne uguali alla distribuzione di massa del primo ordine) durante la fascia oraria notturna.

(rispetto a quello corrispondente ad un'approssimazione markoviana della statistica della richiesta netta di potenza).

Capitolo 5

Approfondimento analitico

5.1 Ridefinizione del problema di ottimizzazione dell'*energy storage* in modo più generale e caratterizzazione analitica della strategia ottima sotto opportune ipotesi

L'obiettivo di questo capitolo è quello di dimostrare analiticamente alcuni risultati sulla strategia di *energy storage* ottima π^* , con una funzione di costo assunta inizialmente lineare nei confronti dell'azione δ intrapresa nei confronti del livello energetico della batteria, per poi cercare di adattare la tecnica di dimostrazione descritta ad un caso più generale (ad esempio, la funzioni di costo $C(\cdot)$ definita nel Capitolo 4). La seguente trattazione, tuttavia, assume un modello di stato diverso da quello definito dalle equazioni (3.1) e (4.12), che ne risultano dei casi particolari: vale quindi la pena ridefinire brevemente il modello di sistema che viene considerato.

Sia E_k il livello di energia presente nella batteria al tempo k , essa può essere caricata/scaricata di una certa quantità δ_k in modo da avere, all'inizio dell'intervallo temporale ad esso successivo

$$E_{k+1} = E_k + \delta_k \quad (5.1)$$

ricordando che devono essere soddisfatti i seguenti requisiti: 1) il livello di energia nella batteria deve stare sempre nell'intervallo $0 \leq E_k \leq E_{\max}$; 2) l'azione di carica/scarica è vincolata a rispettare $\delta_{\min} \leq \delta_k \leq \delta_{\max}$, dove $\delta_{\min} < 0$ è il massimo livello di scarica e $\delta_{\max} > 0$ è il massimo livello di carica. Mettendo insieme questi due vincoli si ha quindi che

$$\max(\delta_{\min}, -E_k) \leq \delta_k \leq \min(\delta_{\max}, E_{\max} - E_k) \quad (5.2)$$

ovvero

$$E_{k+1} = E_k + \delta_k \quad \text{con } \delta_k \in \mathcal{U}(E_k) \quad (5.3)$$

dove

$$\mathcal{U}(E) = [\mathcal{U}^-(E), \mathcal{U}^+(E)] = [\max(\delta_{\min}, -E), \min(\delta_{\max}, E_{\max} - E)] \quad (5.4)$$

come illustrato in Figura 5.1a. Dal momento che il suo utilizzo tornerà presto comodo, si introduce di seguito la definizione di proiezione ortogonale sull'insieme $\mathcal{U}(E)$, che dal punto di vista analitico si esprime come

$$\perp_{\mathcal{U}(E)}(\delta) = \min(\max(\delta, \mathcal{U}^-(E)), \mathcal{U}^+(E)) = \min(\max(\delta, \delta_{\min}, -E), \delta_{\max}, E_{\max} - E). \quad (5.5)$$

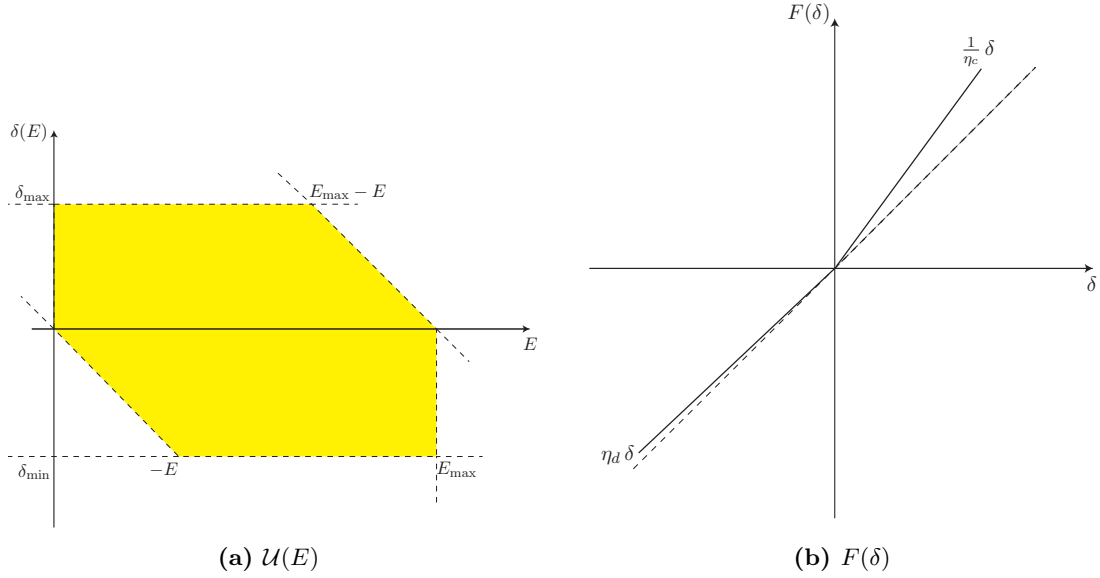


Figura 5.1: Esempi di rappresentazione dell'insieme $\mathcal{U}(E)$ e della funzione $F(\delta)$.

Assumendo inoltre le consuete inefficienze di carica e scarica, espresse rispettivamente come $\eta_c \leq 1$ ed $\eta_d \leq 1$, la quantità di potenza richiesta dalla grid o ceduta alla medesima assume la forma

$$F(\delta) = \frac{1}{\eta_c} (\delta)^+ + \eta_d (\delta)^- \quad (5.6)$$

che, come illustrato in Figura 5.1b, risulta un operatore *convesso*. Si ricordi poi che, in un certo intervallo temporale k , L_k indica la richiesta di potenza da parte degli utenti all'interno della grid e R_k la produzione di energia tramite sorgenti rinnovabili. Sia poi M_k un *modulating state* (come quello che viene considerato nell'articolo [2]) che influenza la produzione, la richiesta ed i costi (ad esempio, l'ora all'interno di una certa giornata, il mese dell'anno, le condizioni meteorologiche, etc.). Lo stato del sistema durante tale intervallo temporale si può esprimere allora come

$$\mathbf{s}_k = (E_k, L_k, R_k, M_k) \quad \text{con } \mathbf{s}_k \in \mathcal{S} = \{\mathcal{E} \times \mathcal{L} \times \mathcal{R} \times \mathcal{M}\}. \quad (5.7)$$

Si definisca poi anche, sempre per uso futuro, lo stato parziale (privato cioè della sua prima componente) del sistema

$$\mathbf{x}_k = (L_k, R_k, M_k) \in \mathcal{X} \quad (5.8)$$

in modo che $\mathbf{s}_k = (E_k, \mathbf{x}_k)$. Sia poi $\delta = a(\mathbf{s})$ l'azione che si è interessati a valutare in accordo con qualche criterio di ottimalità e sia \mathcal{A} l'insieme delle azioni ammissibili (ad esempio, quelle che non implicano vendere energia all'operatore se ciò non è consentito). Il costo risulta, come il solito, una funzione dello stato e dell'azione, ovvero $\gamma = \gamma(\mathbf{s}, \delta)$. In uno scenario tipico delle smart grid il costo può essere una funzione, f , della potenza effettivamente richiesta all'operatore energetico, definita da u , e si ha quindi che, all'istante k

$$\gamma(\mathbf{s}_k, \delta_k) = f(u(L_k, R_k, F(\delta_k)), M_k) \quad (5.9)$$

dove u prende in considerazione le perdite della linea. Proprietà di interesse per la funzione costo γ sono:

1. $\gamma(\mathbf{s}, \delta)$ è non-decrescente in δ – Questa proprietà è sempre verificata dal momento che u risulta non-decrescente in δ (perchè un aumento della richiesta di potenza degli utenti all'interno della grid implica un aumento della potenza che quest'ultima deve provvedere a rendere disponibile), ed il costo f è non-decrescente in u (altrimenti consumare di più costerebbe meno, che è irragionevole).
2. $\gamma(\mathbf{s}, \delta)$ è convessa in δ – Questa è un requisito più forte dato che u è in generale una funzione non-convessa dei suoi argomenti. La convessità, tuttavia, è assicurata quando le perdite delle linee di trasmissione risultano trascurabili o quando la batteria si trovi nel punto di *common coupling*. In tale situazione si ha che

$$\gamma(\mathbf{s}, \delta) = f(L - R + F(\delta), M) \quad (5.10)$$

e così il costo γ sarà convesso se la funzione f è convessa nel suo primo argomento.

3. $\gamma(\mathbf{s}, \delta)$ è lineare in δ – In assenza di perdite lungo la linea di trasmissione e per una funzione di costo f che sia lineare nel suo primo argomento, sia la non-decrescenza che la convessità sono assicurate, e si ha che

$$\gamma(\mathbf{s}, \delta) = f(\mathbf{x}) \cdot (L - R + F(\delta)). \quad (5.11)$$

Siano $p(\mathbf{t} | \mathbf{s}, \delta(\mathbf{s}))$ le probabilità di transizione di stato, che supponiamo definiscano completamente il sistema stazionario considerato. Sia $\mu(\mathbf{s}, a)$ la PDF in condizioni stazionarie. Il criterio che si adotta rispetto alla funzione obiettivo rimane quello della minimizzazione del costo medio (su un orizzonte potenzialmente infinito), espresso tramite la sua formulazione *DP-like*

$$a^* = \arg \min_{a \in \mathcal{A}} \mathbb{E} \left\{ \sum_{k=0}^{\infty} \alpha^k \gamma(\mathbf{s}_k, a(\mathbf{s}_k)) \right\}. \quad (5.12)$$

Mettendosi nella prospettiva della programmazione dinamica, si definisca il costo per $\mathbf{s}_0 = \mathbf{s}$ e la strategia π (da intendersi come una funzione: $\mathcal{S} \rightarrow \mathcal{A}$)

$$V_\pi(\mathbf{s}) = \mathbb{E} \left\{ \sum_{k=0}^{\infty} \alpha^k \gamma(\mathbf{s}_k, \pi(\mathbf{s}_k)) \mid \mathbf{s}_0 = \mathbf{s} \right\} \quad (5.13)$$

ed il valore ottimo sia dato da

$$V^*(\mathbf{s}) = \min_{\pi} V_\pi(\mathbf{s}) \quad \forall \mathbf{s} \in \mathcal{S}. \quad (5.14)$$

$V^*(\mathbf{s})$ risulta (per quanto già descritto seguendo un percorso leggermente diverso dal Capitolo 2) l'unica soluzione di

$$V^*(\mathbf{s}) = \min_{\delta \in \mathcal{U}(E)} Q(\mathbf{s}, \delta) \quad (5.15)$$

con

$$Q(\mathbf{s}, \delta) = \gamma(\mathbf{s}, \delta) + \alpha G(\mathbf{s}, \delta) \quad (5.16)$$

e

$$G(\mathbf{s}, \delta) = \int_{\mathcal{S}} p(\mathbf{t} | \mathbf{s}, \delta) V^*(\mathbf{t}) d\mathbf{t} \quad (5.17)$$

e la strategia ottima sarà data da

$$\pi^*(\mathbf{s}) = \arg \min_{\delta \in \mathcal{U}(E)} Q(\mathbf{s}, \delta). \quad (5.18)$$

Nel problema di ottimizzazione che si sta analizzando, fortunatamente, è possibile semplificare le equazioni (5.17) e (5.18). Infatti, come già osservato, dal momento che $\mathbf{s} = (E, \mathbf{x})$ e $E_{k+1} = E_k + \delta_k$ si ha che

$$\begin{aligned} p(\mathbf{t} | \mathbf{s}, \delta(\mathbf{s})) &= p(\varepsilon, \mathbf{y}, | E, \mathbf{x}, \delta(\mathbf{s})) = p(\mathbf{y} | \varepsilon, E, \mathbf{x}, \delta(\mathbf{s})) p(\varepsilon | E, \mathbf{x}, \delta(\mathbf{s})) \\ &= p(\mathbf{y} | \varepsilon, E, \mathbf{x}, \delta(\mathbf{s})) \delta_{\varepsilon, E + \delta(\mathbf{s})} \end{aligned} \quad (5.19)$$

dove $\delta_{(\cdot, \cdot)}$ è la funzione impulso di Dirac. Grazie a (5.19), l'integrale presente nell'equazione (5.17) si può semplificare come espresso da

$$\begin{aligned} G((E, \mathbf{x}), \delta) &= \int_{\mathcal{S}} p(\mathbf{t} | \mathbf{s}, \delta) V^*(\mathbf{t}) d\mathbf{t} \\ &= \int_{\mathcal{X}} \int_{\mathcal{E}} p(\mathbf{y} | \varepsilon, E, \mathbf{x}, \delta(\mathbf{s})) \delta_{\varepsilon, E + \delta(\mathbf{s})} V^*(\varepsilon, \mathbf{y}) d\mathbf{y} d\varepsilon \\ &= \int_{\mathcal{X}} p(\mathbf{y} | \mathbf{x}) V^*(E + \delta, \mathbf{y}) d\mathbf{y}. \end{aligned} \quad (5.20)$$

In maniera analoga si dimostra che (5.16) si può esprimere come ¹

$$Q(\mathbf{s}, \delta) = Q((E, \mathbf{x}), \delta) = \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}(\mathbf{x}, E + \delta) \quad (5.21)$$

e si ha inoltre che

$$\begin{aligned} V^*(\mathbf{s}) &= \min_{\varepsilon \in E + \mathcal{U}(E)} \bar{\gamma}(\mathbf{x}, \varepsilon - E) + \alpha \bar{G}(\mathbf{x}, \varepsilon) \\ &= \min_{\delta \in \mathcal{U}(E)} \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}(\mathbf{x}, E + \delta). \end{aligned} \quad (5.22)$$

5.2 Proprietà di V^*

Il seguente risultato è un'estensione di un risultato tratto da [2]: esso ha carattere molto generale ed è una conseguenza diretta del fatto che la funzione di costo $\gamma(\mathbf{s}, \delta)$ è non-decrescente in δ , e della peculiare definizione dell'intervallo di azioni ammissibili data da (5.4).

Lemma 1. $V^*(\mathbf{s}) = V^*(E, \mathbf{x})$ è una funzione non-crescente di E .

Dimostrazione. Si definisca la *state-value function* ottima ad n passi come

$$V_n^*(\mathbf{s}) = \min_{\delta \in \mathcal{U}(\mathbf{s})} \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}_{n-1}(\mathbf{x}, E + \delta) \quad (5.23)$$

dove

$$\bar{G}_{n-1}(\mathbf{x}, \varepsilon) = \int_{\mathcal{X}} p(\mathbf{y} | \mathbf{x}) V_{n-1}^*(\varepsilon, \mathbf{y}) d\mathbf{y}. \quad (5.24)$$

La (5.23), grazie all'equazione di ottimalità di Bellman, soddisfa la seguente relazione

$$\lim_{n \rightarrow \infty} V_n^*(\mathbf{s}) = V^*(\mathbf{s}). \quad (5.25)$$

Si prova quindi per induzione che $V_n^*(\mathbf{s})$ è non-crescente in E . La base dell'induzione è che $V_0(\mathbf{s}) = 0$. Il passo induttivo consiste nel provare che, assumendola verificata per $V_{n-1}^*(\mathbf{s})$, essa vale anche per $V_n^*(\mathbf{s})$: si vuole cioè dimostrare che, se $E_1 \leq E_2$, risulta che $V_n^*(E_2, \mathbf{x}) \leq V_n^*(E_1, \mathbf{x})$. A

¹Si noti che viene aggiunta la $\bar{(\cdot)}$ sopra una funzione quando se ne considera la versione che dipende solo dalla seconda, \mathbf{x} , componente dello stato $\mathbf{s} = (E, \mathbf{x})$.

tale scopo si definiscano come $\delta_1 = a^*(E_1, \mathbf{x})$ e $\delta_2 = a^*(E_2, \mathbf{x})$ le azioni ottime da intraprendere, rispettivamente, in E_1 ed E_2 . Si noti ovviamente che $\delta_1 \in \mathcal{U}(E_1)$ e che $\delta_2 \in \mathcal{U}(E_2)$. Dato che $E_1 \leq E_2$, si può verificare che $\delta_{12} = \min(\delta_1, \mathcal{U}^+(E_2))$ risulta la proiezione ortogonale di δ_1 su $\mathcal{U}(E_2)$, e che è tale per cui

$$\begin{cases} \delta_{12} \in \mathcal{U}(E_2) & \text{(per costruzione : si osservi la Figura 5.1a)} \\ \delta_{12} \leq \delta_1 & \text{(per definizione)} \\ E_2 + \delta_{12} = \min(E_2 + \delta_1, E_2 + \mathcal{U}^+(E_2)) = \min(E_2 + \delta_1, E_2 + \delta_{\max}, E_{\max}) \geq E_1 + \delta_1 \end{cases} \quad (5.26)$$

dove l'ultima disuguaglianza risulta verificata per ciascuno degli argomenti del minimo. Ad essere precisi, per dimostrare (in modo un po' pedante) la prima delle (5.26) bisogna ragionare osservando la forma della regione ammissibile $\mathcal{U}(E)$. Le ipotesi, quindi, sono che $E_1 \leq E_2$, $\delta_1 \in \mathcal{U}(E_1)$, $\delta_2 \in \mathcal{U}(E_2)$ e che $\delta_{12} = \min(\delta_1, \mathcal{U}^+(E_2)) = \min(E_1, \max(\delta_{\max}, E_{\max} - E_2))$. Si faccia quindi riferimento alla Figura 5.2: $\delta_{12} \leq \mathcal{U}^+(E_2)$ per come è stato esso stesso definito; inoltre, vale che $\mathcal{U}^+(E_i) \geq \mathcal{U}^-(E_j) \forall E_i, E_j \in \mathcal{E}$. Dato poi che $\mathcal{U}^+(E_1) \geq \mathcal{U}^+(E_2) \geq \mathcal{U}^-(E_2) \geq \mathcal{U}^-(E_1)$, δ_{12} , essendo uguale per definizione al minimo tra δ_1 e $\mathcal{U}^+(E_2)$, risulterà anch'esso $\geq \mathcal{U}^-(E_2)$, e quindi si è rigorosamente provato che $\delta_{12} \in \mathcal{U}(E_2)$.

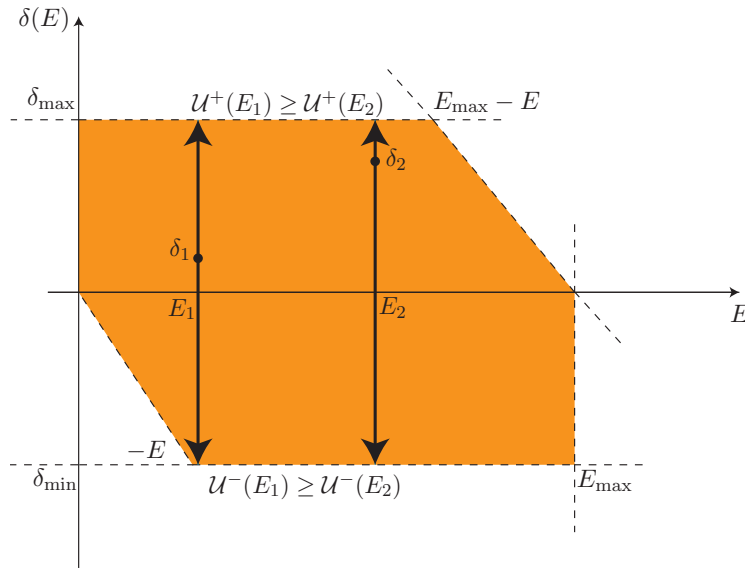


Figura 5.2: Rappresentazione di un generico livello energetico $E_1 \leq E_2$ e delle relative regioni di ammissibilità: $\mathcal{U}(E_1)$ e $\mathcal{U}(E_2)$.

Si ha poi che

$$\begin{aligned} V_n^*(E_2, \mathbf{x}) &= \min_{\delta \in \mathcal{U}(E_2)} \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_2 + \delta) \\ &\leq \bar{\gamma}(\delta_{12}, \mathbf{x}) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_2 + \delta_{12}) \\ &\leq \bar{\gamma}(\mathbf{x}, \delta_1) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_1 + \delta_1) \\ &\leq \bar{\gamma}(\mathbf{x}, \delta_1) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_2 + \delta_1) \\ &= V_n^*(E_1, \mathbf{x}) \end{aligned} \quad (5.27)$$

dove la prima disuguaglianza deriva dalla prima delle (5.26), la seconda disuguaglianza dalla seconda delle (5.26) e dal fatto che $\bar{\gamma}(\mathbf{x}, \delta)$ è non-decrescente in δ , e la terza disuguaglianza dalla

terza delle (5.26) e dal fatto che $\overline{G}_{n-1}(\mathbf{x}, \varepsilon)$ è non-crescente in ε dato che $V_{n-1}(E, \mathbf{x})$ lo è in E per ipotesi induttiva. \square

Quando poi la funzione di costo $\overline{\gamma}(\mathbf{x}, \delta)$ è anche convessa in δ si può dimostrare anche il seguente risultato sulla convessità, che generalizza quanto provato sempre in [2] (la cui prova di convessità, per inciso, non è corretta).

Lemma 2. Se la funzione di costo $\overline{\gamma}(\mathbf{x}, \delta)$ risulta anche convessa in δ , allora $V^*(\mathbf{s}) = V^*(E, \mathbf{x})$ è convesso in E .

Dimostrazione. Si segue la stessa linea di dimostrazione del Lemma 1 e si provi la convessità in E di $V_n^*(E, \mathbf{x})$ assumendo, per ipotesi induttiva, che $V_{n-1}^*(E, \mathbf{x})$ sia convessa in E . Sia $E_{12} = a E_1 + (1-a) E_2$, per un qualche $0 \leq a \leq 1$, una combinazione convessa dei due coefficienti. Si noti che, per $\delta_{12} = a \delta_1 + (1-a) \delta_2$ con δ_1 e δ_2 che risultino azioni ammissibili per i livelli di carica E_1 ed E_2 , vale che

$$\perp_{\mathcal{U}(E_{12})}(\delta_{12}) = \min(\max(\delta_{12}, \delta_{\min}, -E_{12}), \delta_{\max}, E_{\max} - E_{12}) = \delta_{12} \quad (5.28)$$

cioè $\delta_{12} = a \delta_1 + (1-a) \delta_2 \in \mathcal{U}(E_{12})$. Come già fatto nel Lemma 1, anche per convincersi di quest'ultima proprietà è meglio ragionare sfruttando una rappresentazione grafica delle regioni ammissibili: si faccia quindi riferimento alla Figura 5.3.

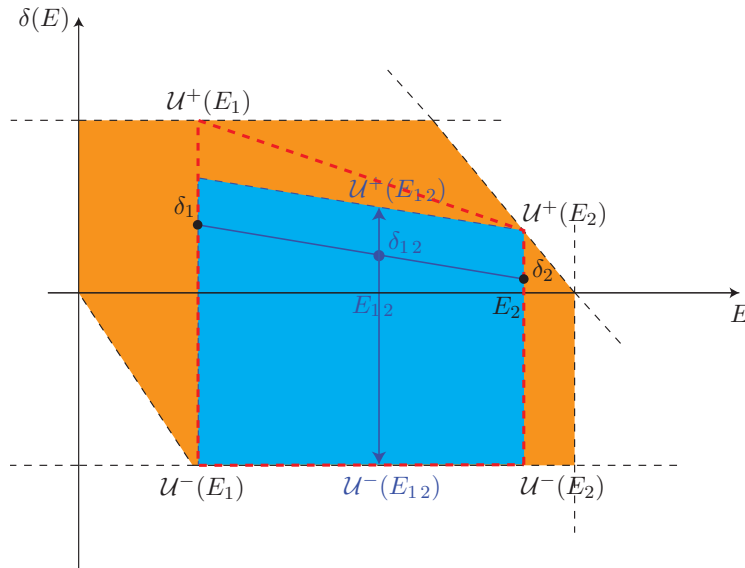


Figura 5.3: Rappresentazione della regione ammissibile $\mathcal{U}(E_{12})$ (con $E_{12} = a E_1 + (1-a) E_2$) e dell'azione δ_{12} appartenente ad essa.

Per ipotesi si ha che $\mathcal{U}^-(E_1) \leq \delta_1 \leq \mathcal{U}^+(E_1)$ e $\mathcal{U}^-(E_2) \leq \delta_2 \leq \mathcal{U}^+(E_2)$; inoltre, come già osservato precedentemente, $\mathcal{U}^+(E_1) \geq \mathcal{U}^+(E_2) \geq \mathcal{U}^-(E_1) \geq \mathcal{U}^-(E_2)$. A questo punto, per quanto riguarda il limite superiore della regione di ammissibilità si ha che

$$\delta_{12} = a \delta_1 + (1-a) \delta_2 \leq a \mathcal{U}^+(E_1) + (1-a) \mathcal{U}^+(E_2) \leq \mathcal{U}^+(E_{12}) \quad (5.29)$$

dove l'ultima uguaglianza vale per la concavità della funzione rappresentata nella zona superiore destra di Figura 5.3 (dove in questo caso vale la disuguaglianza stretta). In modo analogo si ha poi che

$$\delta_{12} = a \delta_1 + (1-a) \delta_2 \leq a \mathcal{U}^-(E_1) + (1-a) \mathcal{U}^-(E_2) \geq \mathcal{U}^-(E_{12}) \quad (5.30)$$

dove l'ultima uguaglianza è giustificata dalla la convessità della funzione rappresentata nella zona inferiore sinistra di Figura 5.3 (dove in questo caso vale il segno di uguaglianza). Da (5.29) e (5.30) risulta verificato che $\delta_{12} \in \mathcal{U}(E_{12})$.

Indicando sempre con δ_1 e δ_2 le azioni ottime da eseguire in E_1 ed E_2 rispettivamente, si ha infine che

$$\begin{aligned} V_n^*(E_{12}, \mathbf{x}) &= \min_{\delta \in \mathcal{U}(E_{12})} \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_{12} + \delta) \\ &\leq \bar{\gamma}(\mathbf{x}, \delta_{12}) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_{12} + \delta_{12}) \\ &\leq a [\bar{\gamma}(\mathbf{x}, \delta_1) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_1 + \delta_1)] + (1-a) [\bar{\gamma}(\mathbf{x}, \delta_2) + \alpha \bar{G}_{n-1}(\mathbf{x}, E_2 + \delta_2)] \\ &= a V_n^*(E_1, \mathbf{x}) + (1-a) V_n^*(E_2, \mathbf{x}) \end{aligned} \quad (5.31)$$

grazie alla convessità di $V_{n-1}^*(E, \mathbf{x})$ in E (che vale per ipotesi induttiva), e quindi quella di $\bar{G}_{n-1}(\mathbf{x}, \varepsilon)$ in ε , insieme alla convessità di $\bar{\gamma}(\mathbf{x}, \delta)$ in δ . \square

5.3 Azioni ottime con funzione di costo lineare

Quando il costo γ è convesso, allora V^* (e quindi G) sono anch'essi convessi. Quindi $Q(\mathbf{x}, \delta)$ risulta convessa, e (5.14) risulta un *problema di minimizzazione convessa* il cui minimo può essere individuato analizzando i sub-differenziali. Prima di arrivare a ciò, tuttavia, conviene fare un passo indietro, ricordando le basi su cui si poggia il teorema successivamente presentato. Quello che ci si prefigge di ricavare è la strategia ottima π^* tale che

$$\pi^*(E, \mathbf{x}) = \operatorname{argmin}_{\delta \in \mathcal{U}(E)} \bar{\gamma}(\mathbf{x}, \delta) + \alpha \bar{G}(\mathbf{x}, E + \delta) \quad (5.32)$$

dove $\bar{\gamma}(\mathbf{x}, \delta) = f(\mathbf{x}) \cdot \left(L - R + \frac{1}{\eta_c} (\delta)^+ + \eta_d (\delta)^- \right)$ è una funzione convessa (e lineare) in δ e non-decrescente in δ , mentre \bar{G} è una funzione convessa in δ , E e non-crescente in δ , E .

Teorema 1. La strategia che risolve il problema di minimizzazione convessa dato da (5.32) quando la funzione di costo $\bar{\gamma}$ risulta lineare rispetto a δ ha la forma $\pi = \pi^*(E, \mathbf{x})$ tale che

$$\pi^*(E, \mathbf{x}) = \begin{cases} \min \{ \varepsilon^-(\mathbf{x}) - E, \mathcal{U}^+(E) - E \}, & E \leq \varepsilon^-(\mathbf{x}) \\ \max \{ \varepsilon^+(\mathbf{x}) - E, \mathcal{U}^-(E) - E \}, & E \geq \varepsilon^+(\mathbf{x}) \\ 0, & \text{altrimenti.} \end{cases} \quad (5.33)$$

Quindi, se $E \leq \varepsilon^-(\mathbf{x})$, la strategia ottima è quella di caricare la batteria ad un livello energetico tanto più vicino possibile ad $\varepsilon^-(\mathbf{x})$ quanto la regione ammissibile $\mathcal{U}(E)$ lo consente. Se $E \geq \varepsilon^+(\mathbf{x})$, invece, la batteria dovrebbe essere scaricata ad un livello energetico tanto più prossimo a $\varepsilon^+(\mathbf{x})$ rispettando i limiti della regione ammissibile $\mathcal{U}^+(E)$. Infine, nel caso in cui $\varepsilon^+(\mathbf{x}) < E < \varepsilon^-(\mathbf{x})$, non risulta ottimo nè caricare nè scaricare la batteria: conviene cioè lasciarne il livello energetico invariato.

Dimostrazione. Trattandosi, come già menzionato, di un problema di minimizzazione convessa, per la ricerca di $\pi^*(E, \mathbf{x}) = \delta^*$ si può sfruttare il fatto che quest'ultima deve essere tale che

$$0 \in \left. \frac{\partial \bar{\gamma}(\mathbf{x}, \delta)}{\partial \delta} \right|_{\delta=\delta^*} + \alpha \left. \frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta} \right|_{\delta=\delta^*} \quad (5.34)$$

di cui conviene prendere in considerazione i due addendi separatamente. Nel caso del primo addendo si ha che

$$\frac{\partial \bar{\gamma}}{\partial \delta} = \begin{cases} f(\mathbf{x}) \eta_d, & \delta < 0 \\ f(\mathbf{x}) \left[\eta_d, \frac{1}{\eta_c} \right], & \delta = 0 \\ f(\mathbf{x}) \frac{1}{\eta_c}, & \delta > 0 \end{cases} \quad (5.35)$$

il cui andamento è rappresentato in Figura 5.4.

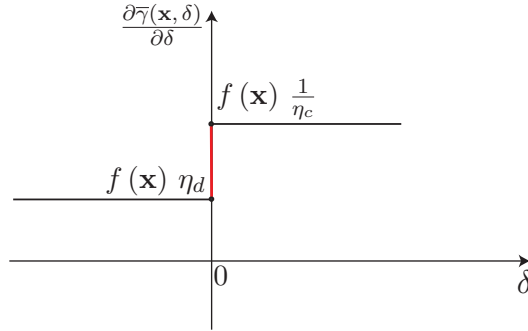


Figura 5.4: Esempio di rappresentazione del sub-differenziale $\frac{\partial \bar{\gamma}(\mathbf{x}, \delta)}{\partial \delta}$ quando la funzione $\gamma(\mathbf{s}, \delta)$ è lineare rispetto a δ .

Andando poi ad analizzare il sub-differenziale di $\frac{\partial \bar{G}(\mathbf{x}, \varepsilon)}{\partial \delta} = \frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta}$, ricordandosi della convessità rispetto a δ della funzione \bar{G} , esso risulterà un operatore non-decrescente, e quindi è lecito aspettarsi un comportamento (nel caso più generale possibile) simile a quello rappresentato nella Figura 5.5.

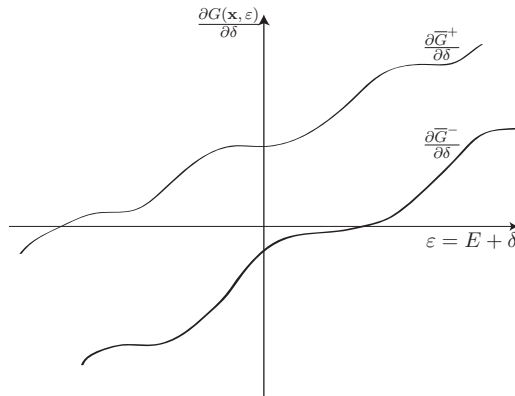


Figura 5.5: Esempio di rappresentazione del sub-differenziale $\frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta}$.

Sommando il primo addendo al secondo moltiplicato per α e considerando prima il caso con $\delta > 0$, si sta quindi cercando la strategia ottima $\pi^*(E, \mathbf{x})$ tale che

$$0 \in f(\mathbf{x}) \frac{1}{\eta_c} + \alpha \frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta} \Big|_{\delta = \delta^*} \quad (5.36)$$

ovvero, espresso in modo più chiaro

$$f(\mathbf{x}) \frac{1}{\eta_c} + \alpha \frac{\partial \bar{G}^-(\mathbf{x}, E + \delta^*)}{\partial \delta} \leq 0 \leq f(\mathbf{x}) \frac{1}{\eta_c} + \alpha \frac{\partial \bar{G}^+(\mathbf{x}, E + \delta^*)}{\partial \delta}. \quad (5.37)$$

Il significato dell'equazione (5.37) è illustrato in Figura 5.6, dove $\mathcal{D}^-(\mathbf{x})$ e $\mathcal{D}^+(\mathbf{x})$ rappresentano, rispettivamente, l'estremo inferiore e superiore dell'intervallo di azioni ammissibili rispetto alla seconda componente soltanto, \mathbf{x} , dello stato $\mathbf{s} = (E, \mathbf{x})$.

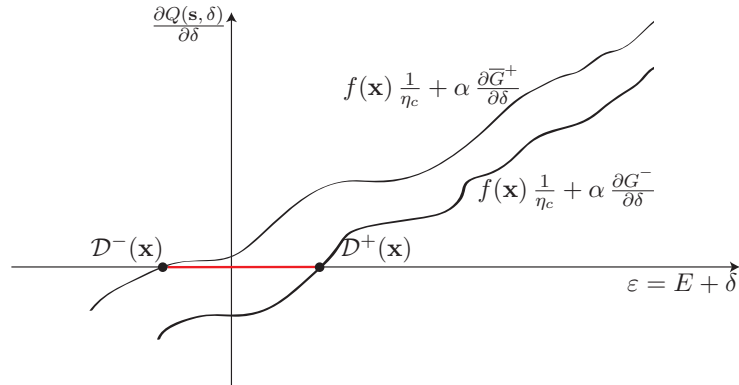


Figura 5.6: Esempio di rappresentazione del sub-differenziale $\frac{\partial Q(\mathbf{s}, \delta)}{\partial \delta}$ per $\delta > 0$.

Per definizione, quindi, dovrà valere che

$$\begin{cases} \mathcal{D}^-(\mathbf{x}) \leq E + \delta \leq \mathcal{D}^+(\mathbf{x}) \\ \mathcal{D}^-(\mathbf{x}) - E \leq \delta \leq \mathcal{D}^+(\mathbf{x}) - E \end{cases} \quad (5.38)$$

come rappresentato nella Figura 5.7.

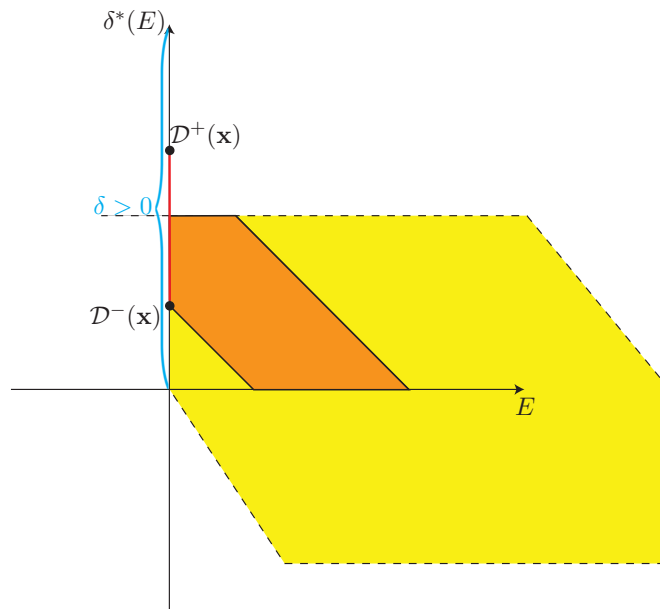


Figura 5.7: Esempio di rappresentazione della regione ottima per la scelta delle azioni da applicare alla batteria in funzione del livello di carica in essa presente: si tratta cioè di $\delta^*(E)$ per $\delta > 0$.

Quanto appena discusso vale per $\delta > 0$: estendere il ragionamento a $\delta < 0$ è abbastanza banale e qui non viene riportato per brevità. Risulta più interessante rappresentare l'andamento

complessivo di $\delta^*(E) \geq 0$ e < 0 quando il sub-differenziale $\frac{\partial \bar{G}(\mathbf{x}, E+\delta)}{\partial \delta}$ ha anche andamenti diversi da quello rappresentato in Figura 5.5 (per $\delta > 0$) e in Figura 5.8 (andamento complessivo): esempi a tale proposito, al variare del sub-differenziale di \bar{G} , sono rappresentati in Figura 5.9.

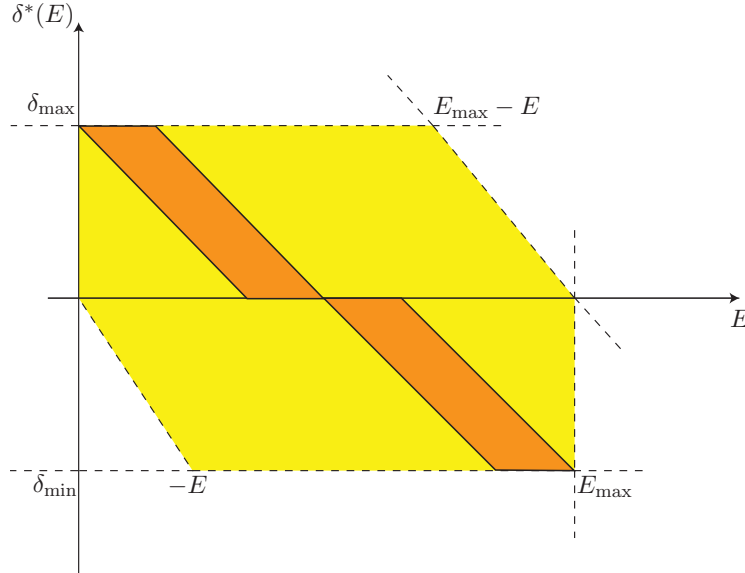


Figura 5.8: Esempio di rappresentazione della regione ottima (colorata in arancione) da cui possono essere scelte le azioni $\delta^*(E)$ in funzione del livello di carica presente nella batteria.

A conclusione della dimostrazione, si consideri la funzione che definisce il nuovo livello energetico ottimo che la batteria deve raggiungere, dato da quello precedente più gli effetti dell'azione intrapresa, ovvero

$$\beta^*(E, \mathbf{x}) = \pi^*(E, \mathbf{x}) + E = \begin{cases} \min \{ \varepsilon^-(\mathbf{x}), \mathcal{U}^+(E) \}, & E \leq \varepsilon^-(\mathbf{x}) \\ \max \{ \varepsilon^+(\mathbf{x}), \mathcal{U}^-(E) \}, & E \geq \varepsilon^+(\mathbf{x}) \\ E, & \text{altrimenti.} \end{cases} \quad (5.39)$$

Questa funzione risulta, cioè, una rotazione di -45° rispetto a quella che rappresenta l'azione ottima da eseguire. Per rendere più chiaro il significato della formula (5.39) è stato eseguito un esempio di tale trasformazione per arrivare ad una rappresentazione di $\beta^*(E, \mathbf{x})$, riportata in Figura 5.10: se dalla regione ottima si sceglie un'azione opportuna tra tutte quelle ottime si può osservare lo stesso andamento presente anche nell'articolo [2] più volte citato, ed illustrato nella Figura 5.11.

Ad essere più precisi, la regione rappresentata nelle Figure 5.7, 5.8 e 5.9 è legata alla quella rappresentata in Figura 5.10 da qualcosa in più che una semplice una rotazione: bisogna anche andare ad intersecare il risultato, infatti, con l'insieme dei livelli energetici ammissibili, che non è stato definito in modo esplicito, ma risulta legato univocamente a $\mathcal{U}(E)$ al variare del livello energetico E . Dato che, a meno della suddetta rotazione, le equazioni (5.33) e (5.39) coincidono perfettamente, la dimostrazione del teorema risulta così conclusa e costituisce un risultato di grande importanza nell'interpretazione dei risultati numerici riportati in questa tesi. \square

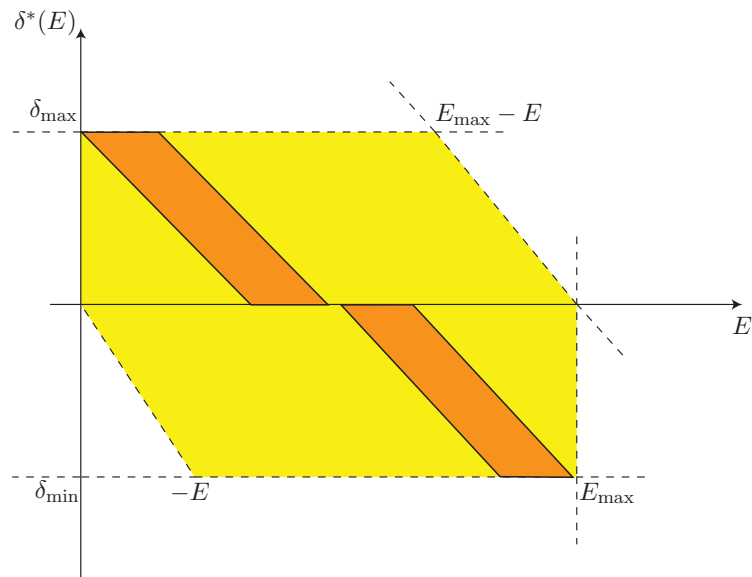


Figura 5.9: Altro esempio di rappresentazione della regione descritta in Figura 5.9.

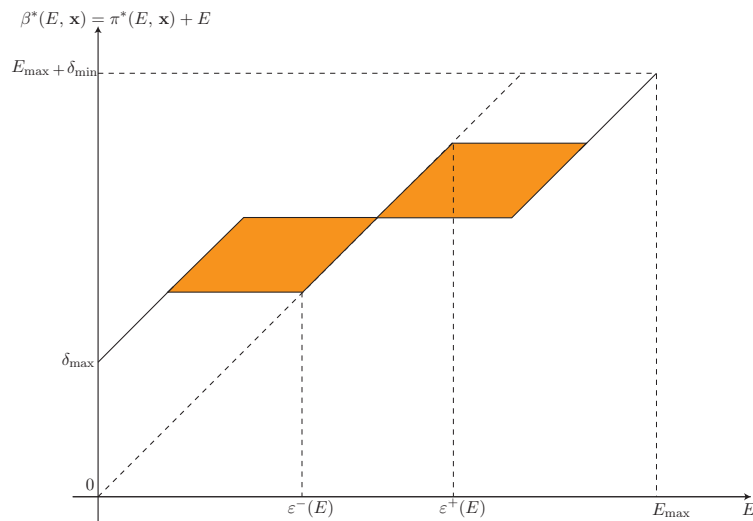


Figura 5.10: Esempio di rappresentazione della regione ottima da cui scegliere le azioni, $\beta^*(E, \mathbf{x})$, in funzione del livello di carica presente nella batteria.

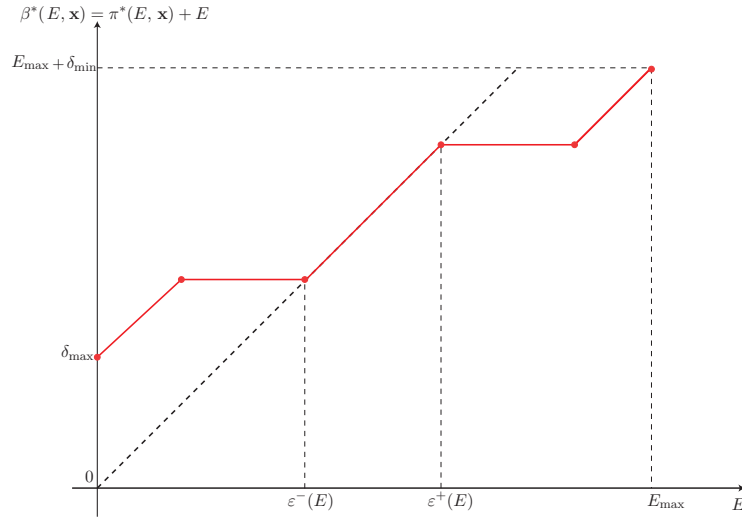


Figura 5.11: Esempio di rappresentazione della strategia ottima relativa al nuovo livello energetico che la batteria deve assumere, $\beta^*(E, \mathbf{x})$, in funzione del suo livello di carica (quando la funzione di costo $\bar{\gamma}$ risulta lineare rispetto a δ).

5.4 Azioni ottime con funzione di costo lineare a tratti

L'unica differenza di questa sezione rispetto alla 5.3 è che ora la funzione f , rispetto all'azione δ , non risulta più lineare ma lineare a tratti (con un numero finito di pendenze), come rappresentato dalla seguente equazione ²

$$\gamma(\mathbf{s}, \delta) = \begin{cases} k_1 f(\mathbf{x}) \cdot (L - R + F(\delta)), & L - R + F(\delta) \leq P_{\text{thr}, 1} \\ \vdots \\ k_i f(\mathbf{x}) \cdot (L - R + F(\delta)), & P_{\text{thr}, i-1} < L - R + F(\delta) \leq P_{\text{thr}, i} \\ \vdots \\ k_n f(\mathbf{x}) \cdot (L - R + F(\delta)), & L - R + F(\delta) > P_{\text{thr}, n-1} \end{cases} \quad (5.40)$$

Bisogna prestare attenzione al fatto che la funzione $L - R + F(\delta)$ è sì lineare a tratti, ma rispetto al suo intero argomento, con delle generiche soglie date da $P_{\text{thr}, 1}, \dots, P_{\text{thr}, i}, \dots, P_{\text{thr}, n-1}$: per ricavarne l'andamento in funzione di δ soltanto conviene innanzitutto eseguire una traslazione di $-(L - R) \triangleq P$. Considerando poi la diversa pendenza di $F(\delta)$ per $\delta \geq 0$ e $\delta < 0$, già rappresentata in Figura 5.1, si avrà che la composizione di funzioni $P + F(\delta)$, plottata rispetto a δ , può essere rappresentata come in Figura 5.12.

Le soglie presenti nella definizione di $\bar{\gamma}$ nell'equazione (5.40) possono poi essere riscritte solo

²Nella quale si assume che $k_j \geq 0 \forall j = 1, 2, \dots, n$ e con dei valori tali da garantire la *convessità* della funzione f .

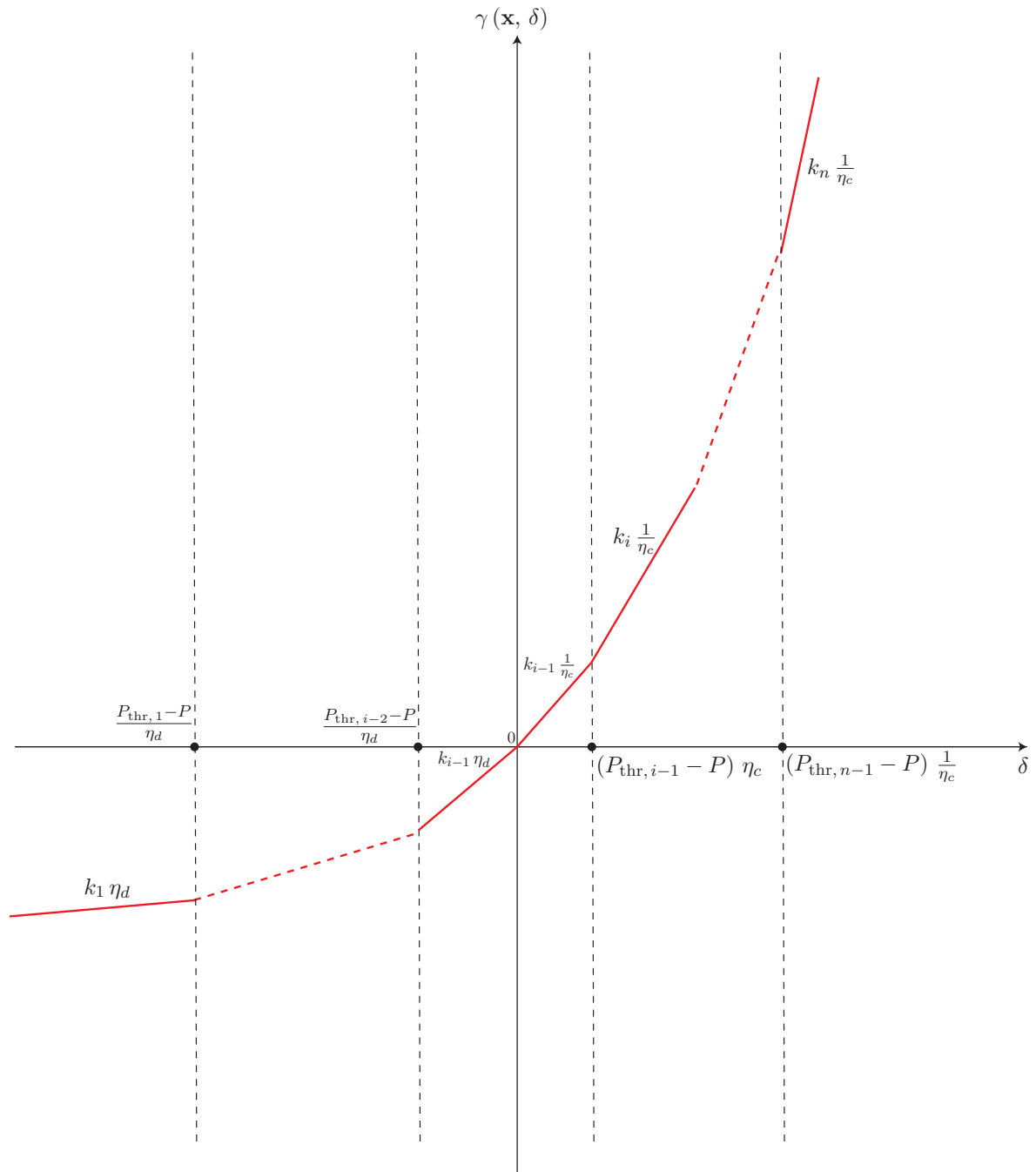


Figura 5.12: Esempio di rappresentazione della funzione $\gamma(\mathbf{x}, \delta)$ (quando la funzione di costo $\bar{\gamma}$ risulta lineare a tratti rispetto a δ).

in funzione di δ nel modo seguente

$$\gamma(\mathbf{x}, \delta) = \begin{cases} k_1 f(\mathbf{x}) \cdot (P + F(\delta)), & F(\delta) \leq P_{\text{thr},1} + P \triangleq \delta_{\text{thr},1} \\ \vdots \\ k_i f(\mathbf{x}) \cdot (P + F(\delta)), & P_{\text{thr},i-1} + P \triangleq \delta_{\text{thr},i-1} < F(\delta) \leq P_{\text{thr},i} + R - L \triangleq \delta_{\text{thr},i} \\ \vdots \\ k_n f(\mathbf{x}) \cdot (P + F(\delta)), & F(\delta) > P_{\text{thr},n-1} + P \triangleq \delta_{\text{thr},n-1}. \end{cases} \quad (5.41)$$

supponendo che $\delta_{\text{thr},1} < \delta_{\text{thr},2} < \dots < \delta_{\text{thr},n-1}$. Si ricordi poi che $F(\delta) = \frac{1}{\eta_c}(\delta)^+ + \eta_d(\delta)^-$: conviene quindi considerare separatamente il caso con $\delta > 0$, quello con $\delta = 0$ e quello con $\delta < 0$. Supponendo, WLOG, che $\delta_{\text{thr},1}, \dots, \delta_{\text{thr},i-2}$ siano tutti < 0 e che $\delta_{\text{thr},i-1}, \dots, \delta_{\text{thr},n-1}$ siano tutti > 0 , il sub-differenziale di $\bar{\gamma}$ rispetto a δ assumerà un'espressione del tipo

$$\frac{\partial \bar{\gamma}}{\partial \delta} = \begin{cases} k_1 \eta_d f(\mathbf{x}), & \delta < \delta_{\text{thr},1} \\ \vdots \\ k_{i-1} \eta_d f(\mathbf{x}), & \delta_{\text{thr},i-1} < \delta \leq 0 \\ f(\mathbf{x}) \left[k_{i-1} \eta_d, k_{i-1} \frac{1}{\eta_c} \right], & \delta = 0 \\ k_{i-1} \frac{1}{\eta_c} f(\mathbf{x}), & 0 < \delta \leq \delta_{\text{thr},i} \\ \vdots \\ k_n \frac{1}{\eta_c} f(\mathbf{x}), & \delta > \delta_{\text{thr},n-1} \end{cases} \quad (5.42)$$

Seguendo la stessa linea guida del Teorema 2, si riporta in Figura 5.13 la rappresentazione grafica dell'equazione (5.36), che la rende più chiara.

Il calcolo e la rappresentazione del sub-differenziale $\frac{\partial \bar{G}(\mathbf{x}, \varepsilon)}{\partial \delta} = \frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta}$ risultano del tutto equivalenti a quelli riportati nel Teorema 1. Nella determinazione del sub-differenziale complessivo, $\frac{\partial \bar{\gamma}(\mathbf{x}, \delta)}{\partial \delta}$, si consideri quindi il caso con $\delta > 0$ e la somma dei primi addendi al secondo moltiplicato per α : si sta quindi cercando la strategia ottima $\pi^*(E, \mathbf{x})$ tale che

$$0 \in k_1 \eta_d f(\mathbf{x}) + \dots + k_n \frac{1}{\eta_c} f(\mathbf{x}) + \alpha \frac{\partial \bar{G}(\mathbf{x}, E + \delta)}{\partial \delta} \Bigg|_{\delta=\delta^*} \quad (5.43)$$

che risulta equivalente a

$$\begin{aligned} k_1 \eta_d f(\mathbf{x}) + \dots + k_n \frac{1}{\eta_c} f(\mathbf{x}) + \alpha \frac{\partial \bar{G}^-(\mathbf{x}, E + \delta^*)}{\partial \delta} &\leq \\ \leq 0 &\leq k_1 \eta_d f(\mathbf{x}) + \dots + k_n \frac{1}{\eta_c} f(\mathbf{x}) + \alpha \frac{\partial \bar{G}^+(\mathbf{x}, E + \delta^*)}{\partial \delta}. \end{aligned} \quad (5.44)$$

Il significato dell'equazione (5.44) è illustrato (parzialmente: per $j = i - 2, i - 1$ soltanto) in Figura 5.14, dove $\mathcal{D}_j^-(\mathbf{x})$ e $\mathcal{D}_j^+(\mathbf{x})$ per $j = i - 1, \dots, n - 1$ rappresentano, rispettivamente, gli estremi inferiori e superiori degli intervalli relativi alle azioni ammissibili rispetto alla seconda componente soltanto, i quali ora dipendono anche dalla pendenza della funzione f al variare di δ all'interno delle zone definite dall'equazione (5.42).

Si consideri poi direttamente il risultato finale, che viene di seguito riportato sotto forma di lemma.

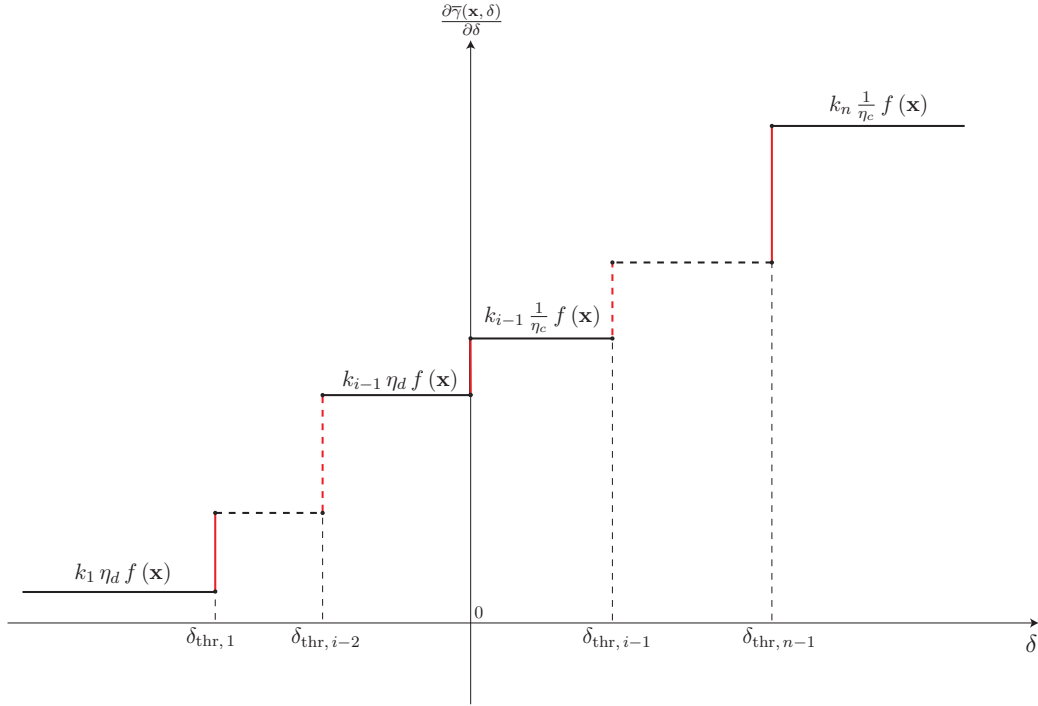


Figura 5.13: Esempio di rappresentazione del sub-differenziale $\frac{\partial \bar{\gamma}(\mathbf{x}, \delta)}{\partial \delta}$ quando la funzione $\gamma(\mathbf{s}, \delta)$ è lineare a tratti rispetto a δ .

Lemma 3. La strategia che risolve il problema di minimizzazione convessa dato da (5.32) quando la funzione di costo $\bar{\gamma}$ risulta lineare a tratti rispetto a δ ha la forma $\pi = \pi^*(E, \mathbf{x})$ tale che

$$\pi^*(E, \mathbf{x}) = \begin{cases} \min \{ \varepsilon_1(\mathbf{x}) - E, \mathcal{U}^+(E) - E \}, & E < \varepsilon_1(\mathbf{x}) \\ \vdots \\ \min \{ \varepsilon_{i-1}(\mathbf{x}) - E, \mathcal{U}^+(E) - E \}, & \varepsilon_{i-1}(\mathbf{x}) < E \leq 0 \\ 0, & 0 < E \leq \varepsilon_{i-1}(\mathbf{x}) \\ \max \{ \varepsilon_i(\mathbf{x}) - E, \mathcal{U}^-(E) - E \}, & \varepsilon_{i-1} < E \leq \varepsilon_i(\mathbf{x}) \\ \vdots \\ \max \{ \varepsilon_{n-1}(\mathbf{x}) - E, \mathcal{U}^-(E) - E \}, & E > \varepsilon_{n-1}(\mathbf{x}) \end{cases} \quad (5.45)$$

e la funzione che definisce il nuovo livello energetico ottimo che la batteria deve raggiungere, dato da quello precedente più gli effetti dell'azione intrapresa, si può esprimere come

$$\beta^*(E, \mathbf{x}) = \pi^*(E, \mathbf{x}) + E = \begin{cases} \min \{ \varepsilon_1(\mathbf{x}), \mathcal{U}^+(E) \}, & E < \varepsilon_1(\mathbf{x}) \\ \vdots \\ \min \{ \varepsilon_{i-1}(\mathbf{x}), \mathcal{U}^+(E) \}, & \varepsilon_{i-1}(\mathbf{x}) < E \leq 0 \\ E, & 0 < E \leq \varepsilon_{i-1}(\mathbf{x}) \\ \max \{ \varepsilon_i(\mathbf{x}), \mathcal{U}^-(E) \}, & \varepsilon_{i-1} < E \leq \varepsilon_i(\mathbf{x}) \\ \vdots \\ \max \{ \varepsilon_{n-1}(\mathbf{x}), \mathcal{U}^-(E) \}, & E > \varepsilon_{n-1}(\mathbf{x}) \end{cases} \quad (5.46)$$

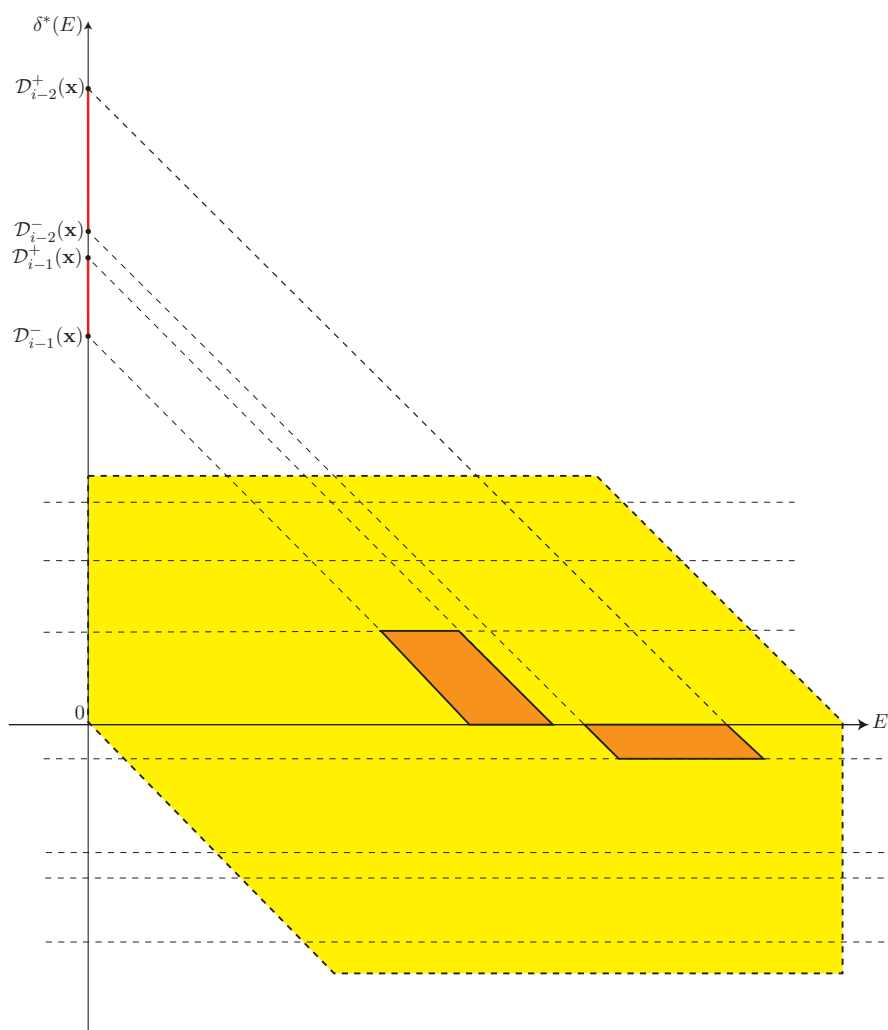


Figura 5.14: Esempio di rappresentazione (di 2 parti) della regione ottima per la scelta delle azioni da applicare alla batteria in funzione del livello di carica in essa presente quando la funzione $\gamma(\mathbf{s}, \delta)$ è lineare a tratti rispetto a δ .

La funzione lineare a tratti definita dall'equazione (5.46) è rappresentata in Figura 5.15: si può notare come il suo andamento sia una banale generalizzazione di quello rappresentato in Figura 5.11 e riportato anche nel paper [2].

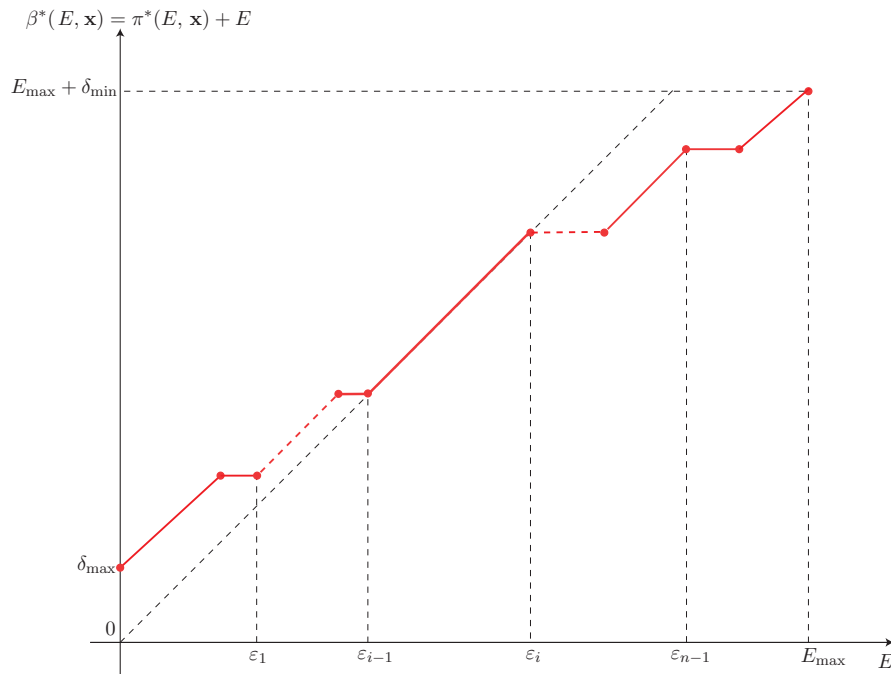


Figura 5.15: Esempio di rappresentazione della strategia ottima relativa al nuovo livello di energia che la batteria deve assumere, $\beta^*(E, \mathbf{x})$, in funzione del suo livello di carica quando la funzione di costo $\bar{\gamma}$ risulta lineare a tratti rispetto a δ .

5.5 Azioni ottime con funzione di costo lineare a tratti (con due pendenze) e ammortamento batteria

Come già stato fatto notare nel Capitolo 4, un possibile modo per tenere conto della spesa iniziale sostenuta per l'acquisto della batteria è quello di spalmare tale costo sugli intervalli temporali in cui viene presa una qualche azione di carica/scarica $|\delta| \neq 0$, perchè essa andrà a diminuire il tempo di vita medio della batteria, la quale andrà riacquistata quando le sue prestazioni non risulteranno più soddisfacenti rispetto ad una qualche metrica. L'equazione (4.7) suggerisce una maniera per modellizzare quanto appena citato facendolo rientrare nella funzione di costo: basta adattarla alla notazione della Sezione 5.5. Si assuma che la funzione di costo $\bar{\gamma}$ risulti lineare rispetto a δ con due sole pendenze per quanto riguarda la parte che dipende dalla somma tra P e $F(\delta)$ (come avviene con una tariffazione che prezza in modo diverso l'energia a seconda che venga comprata o venduta) per non appesantire troppo la successiva trattazione. Calcolando il

sub-differenziale di tale funzione rispetto a δ si ha che

$$\frac{\partial \bar{\gamma}}{\partial \delta} = \begin{cases} k_1 \eta_d - \rho f(\mathbf{x}), & \delta < \delta_{\text{thr},1} \triangleq -\frac{P}{\eta_d} \\ k_2 \eta_d - \rho f(\mathbf{x}), & \delta_{\text{thr},1} < \delta \leq \delta_{\text{thr},2} \triangleq 0 \\ f(\mathbf{x}) \left[k_2 \eta_d - \rho, k_2 + \frac{1}{\eta_c} + \rho \right], & \delta_{\text{thr},1} < \delta \leq \delta_{\text{thr},2} \triangleq 0 \\ k_2 \frac{1}{\eta_c} + \rho f(\mathbf{x}), & \delta > \delta_{\text{thr},2} \end{cases} \quad (5.47)$$

da cui appare una scelta plausibile considerare come condizione sufficiente ma non necessaria³ affinché risulti conveniente comprare una batteria per immagazzinare energia all'interno di una certa grid che $k_1 + \eta_d - \rho \geq 0 \Rightarrow \rho \leq k_1 + \eta_d$, considerando la derivata come un indicatore del costo per unità di energia includendo l'ammortamento dell'acquisto iniziale della batteria. Il sub-differenziale dell'equazione (5.47) è rappresentato in Figura 5.16.

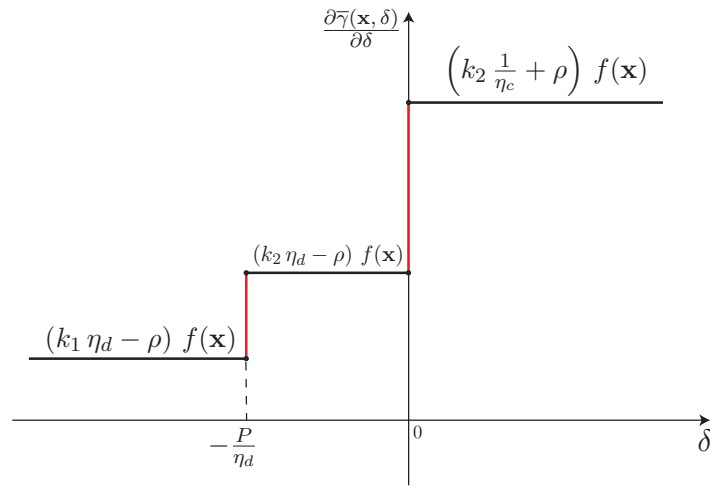


Figura 5.16: Esempio di rappresentazione del sub-differenziale $\frac{\partial \bar{\gamma}(\mathbf{x}, \delta)}{\partial \delta}$ quando la funzione $\gamma(\mathbf{s}, \delta)$ è lineare con due pendenze (alle quali corrisponde una prezzazione che differenzia il costo dell'energia a seconda che sia comprata o venduta) rispetto a δ e viene aggiunto in essa un termine relativo all'ammortamento del costo della batteria.

Seguendo poi lo stesso schema di ragionamento del Teorema 1, si riporta direttamente sotto forma di lemma la funzione che definisce il nuovo livello energetico ottimo che la batteria deve raggiungere, dato da quello precedente più gli effetti dell'azione intrapresa.

Lemma 4. La strategia che risolve il problema di minimizzazione convessa dato da (5.32) quando la funzione di costo $\bar{\gamma}$ risulta lineare a tratti (con due pendenze) rispetto a δ avendo un termine che tiene conto dell'ammortamento per l'acquisto della batteria ha la forma $\pi = \pi^*(E, \mathbf{x})$ tale che

$$\beta^*(E, \mathbf{x}) = \pi^*(E, \mathbf{x}) + E = \begin{cases} \min \{ \varepsilon_1(\mathbf{x}), \mathcal{U}^+(E) \}, & E < \varepsilon_1(\mathbf{x}) \\ \min \{ \varepsilon_2(\mathbf{x}), \mathcal{U}^+(E) \}, & \varepsilon_1(\mathbf{x}) \leq E \leq \varepsilon_2(\mathbf{x}) \\ E, & \varepsilon_2(\mathbf{x}) < E \leq \varepsilon_3(\mathbf{x}) \\ \max \{ \varepsilon_3(\mathbf{x}), \mathcal{U}^-(E) \}, & E > \varepsilon_3(\mathbf{x}) \end{cases} \quad (5.48)$$

³Tale affermazione meriterebbe un maggior approfondimento dal punto di vista analitico.

La funzione lineare a tratti definita dall'equazione (5.48), con tre diverse soglie al posto delle due del Teorema 1, è rappresentata in Figura 5.17: si può notare come il suo andamento sia ancora una volta una minima generalizzazione di quello rappresentato in Figura 5.11 e riportato anche nel paper [2].

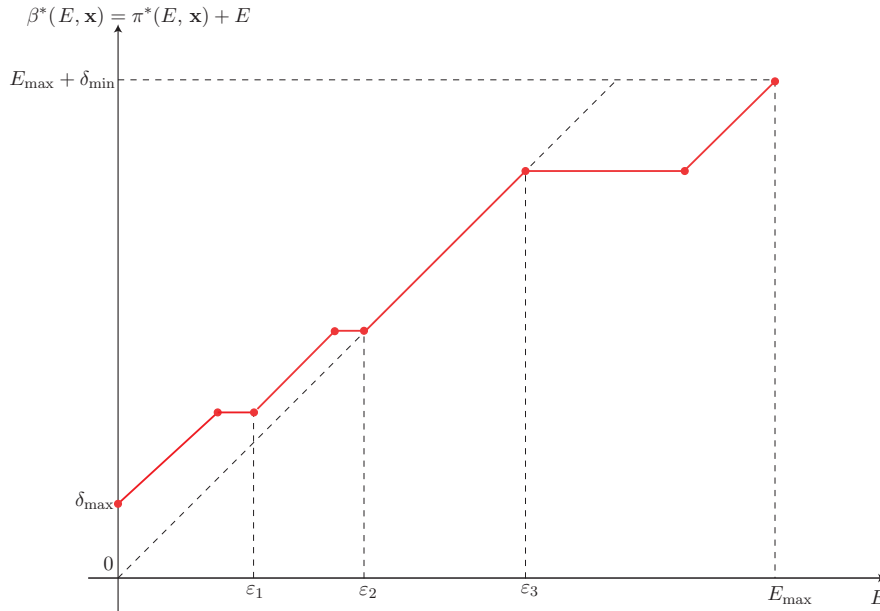


Figura 5.17: Esempio di rappresentazione della strategia ottima relativa al nuovo livello di energia che la batteria deve assumere, $\beta^*(E, \mathbf{x})$, in funzione del suo livello di carica quando la funzione di costo $\bar{\gamma}$ risulta lineare a tratti (con due pendenze) rispetto a δ avendo un termine che tiene conto dell'ammortamento per l'acquisto della batteria.

5.6 Azioni ottime con funzione di costo lineare a tratti, ammortamento batteria e perdita carica nel tempo

L'ultimo passo teorico che si ritiene interessante cercare di effettuare è quello di includere nel modello di sistema, oltre a tutto quello discusso nella sezione 5.5, anche il fatto che la batteria perda una (piccola) frazione della propria energia al passare nel tempo indipendentemente dall'azione che viene intrapresa, come era già stato formalizzato nel Capitolo 4. In quest'ottica l'equazione (5.1) diventa

$$E_{k+1} = E_k + \delta_k - \theta (1 - \exp(-\vartheta E_k)) \quad (5.49)$$

dove, come già accennato, θ e ϑ sono due generici parametri che impattano sull'ordine di grandezza del fenomeno della perdita di carica nel tempo da parte della batteria. L'equazione (5.49) deve ovviamente continuare a rispettare i vincoli sull'escursione del livello energetico presente nella batteria e sui suoi rate di carica/scarica finiti, esprimibili rispettivamente come $0 \leq E_k \leq E_{\max}$ e $\delta_{\min} \leq \delta_k \leq \delta_{\max}$. Mettendo insieme tali vincoli con la nuova equazione di aggiornamento del

livello di carica si ottiene

$$\max(\delta_{\min}, -E_k + \theta(1 - \exp(-\vartheta E_k))) \leq \delta_k \leq \min(\delta_{\max}, E_{\max} - E_k + \theta(1 - \exp(-\vartheta E_k))) \quad (5.50)$$

ovvero

$$E_{k+1} = E_k + \delta_k \quad \text{con } \delta_k \in \mathcal{U}_{\text{new}}(E_k) \quad (5.51)$$

dove

$$\begin{aligned} \mathcal{U}_{\text{new}}(E) &= [\mathcal{U}_{\text{new}}^-(E), \mathcal{U}_{\text{new}}^+(E)] \\ &= [\max(\delta_{\min}, -E + \theta(1 - \exp(-\vartheta E))), \min(\delta_{\max}, E_{\max} - E + \theta(1 - \exp(-\vartheta E)))] \end{aligned} \quad (5.52)$$

come rappresentato in Figura 5.18.

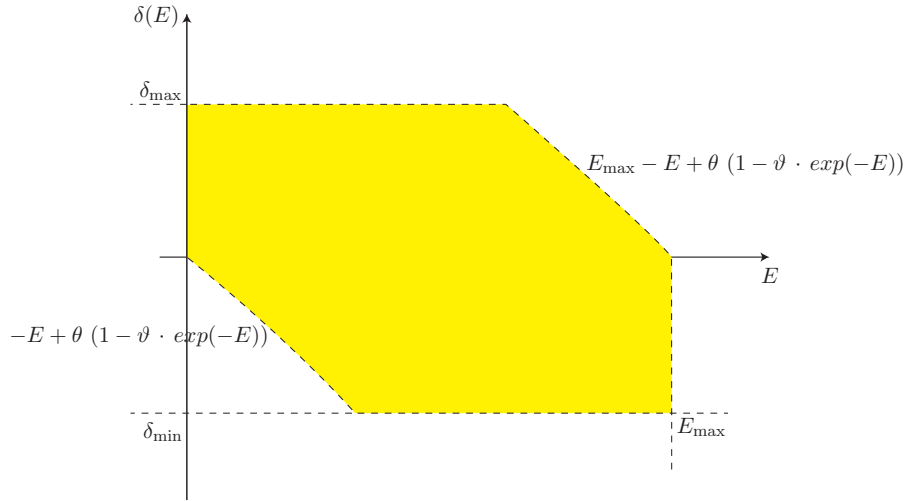


Figura 5.18: Esempio di rappresentazione della nuova regione ammissibile $\mathcal{U}_{\text{new}}(E)$

Non è però la strada più conveniente per risolvere il nuovo problema di ottimizzazione convessa quella di ragionare per via grafica, sulla base della diversa forma della regione ammissibile, come è stato fatto nel Teorema 1. Risulta importante osservare invece che il termine dell'equazione (5.49) che modella la perdita di energia nel tempo da parte della batteria risulta *convesso* rispetto ad E : si può dimostrare che la convessità, rispetto ad E e a δ , è verificata anche per la funzione $\bar{G}(\mathbf{x}, h(E + \delta))$, dove $h(E + \delta) = E_{k+1} = E_k + \delta_k - \theta(1 - \exp(-\vartheta E_k))$. Dato che la somma è una particolare combinazione lineare, la quale preserva la convessità, si avrà che \bar{G} è convessa rispetto al suo secondo argomento $\varepsilon = E + \delta$. Per arrivare infine a determinare la struttura della strategia ottima tenendo conto della perdita di carica nel tempo, si può a questo punto seguire sempre la stessa identica tecnica del Teorema 1, andando ad analizzare separatamente il sub-differenziale di \bar{G} rispetto ad ε , $\frac{\partial \bar{G}(\mathbf{x}, h(\varepsilon))}{\partial \varepsilon} = \frac{\partial \bar{G}(\mathbf{x}, h(E + \delta))}{\partial (E + \delta)}$, che risulterà ancora un operatore non-decrescente. La strategia π^* che risolve il nuovo problema di ottimizzazione che tiene conto della perdita di energia nel tempo da parte della batteria risulterà ancora una volta analoga a quella riportata in Figura 5.15, con un numero di soglie che dipende dal numero di pendenze rispetto a δ della funzione di costo $\bar{\gamma}$.

5.7 Verifica delle dimostrazioni analitiche tramite una simulazione numerica ad hoc

Per concludere la sezione, si è scelto di riportare un'ultima simulazione numerica, cercando di interpretarla alla luce di quanto illustrato nelle sezioni di questo capitolo. A tale scopo si è usata come funzione di costo $C_3(\cdot)$, definita nella Sezione 3.3: si tratta di una funzione crescente e lineare a tratti con *tre pendenze*, che quindi ricade perfettamente nei casi coperti dalla Sezione 5.4 di questo capitolo, i cui risultati prendono in considerazione un generico numero n di pendenze. Secondo l'equazione (5.45), con un numero di pendenze della funzione di costo pari a tre e considerando la differenziazione tra $\delta > 0$ e $\delta < 0$ insita nella definizione di $F(\delta)$, la strategia ottima per quanto riguarda l'azione di carica/scarica da intraprendere nei confronti della batteria dovrebbe avere quattro soglie, le quali, WLOG, possono anche coincidere tra loro. Ciò viene verificato nella Figura 5.19, che mostra la strategia ottima π (plottata in funzione del livello energetico della batteria, parametrizzata per alcuni valori della richiesta di potenza) ricavata tramite l'algoritmo DPM, col setting simulativo descritto nella Sezione 3.3.

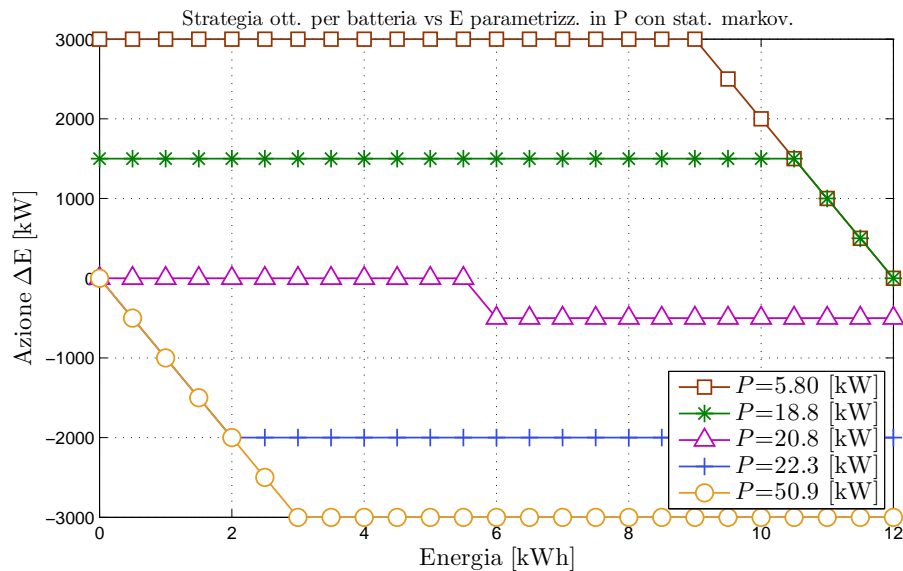


Figura 5.19: Strategia ottima di carica/scarica nei confronti della batteria ricavata tramite DPM applicata alla funzione di costo $C_3(\cdot)$, lineare a tratti e con tre pendenze.

Dalla Figura 5.19 si osserva come il numero di soglie dell'azione ottima da eseguire, in funzione del livello di energia presente nella batteria, siano *al massimo due* per qualunque valore della richiesta di potenza. Questo risultato si sposa perfettamente con la dimostrazione della Sezione 5.4, la quale prende in considerazione il caso in cui la funzione (e quindi il suo sub-differenziale) $\gamma(\mathbf{s}, \delta)$, lineare a tratti rispetto a δ , sia la la più generale possibile. Per la particolare funzione di costo analizzata, dipendente solo da $L + F(\delta)$ e da nessun *modulating state*, è ragionevole che il numero di soglie rispetto al livello energetico della batteria sia ridotto a due.

Per concludere, un limite di tutto questo capitolo può sembrare quello di aver considerato solo funzioni di costo lineari a tratti e mai con un andamento quadratico, come quello della coda della funzione di costo utilizzata nel Capitolo 3. La ragione è che il caso quadratico, per la forma del sub-differenziale di $\gamma(\mathbf{s}, \delta)$ rispetto a δ ad esso associata, non si presta a trovare (almeno con metodi analoghi a quelli qui presentati) risultati in forma chiusa. Va

notato, per inciso, che, a causa della distribuzione di probabilità della sequenza $\{P_k\}$ (riportata in Figura 1.6), fortemente sbilanciata verso valori bassi della richiesta di potenza, l'influenza di quelli più elevati, che cadrebbero nella zona della funzione di costo definita dall'equazione (1.4) con pendenza quadratica, hanno *pochissima influenza* sul calcolo della strategia ottima tramite DP (che va ad effettuare iterativamente operazioni di integrali rispetto alla PDF e minimizzazioni). Quest'ultima affermazione si può verificare in tutti i risultati del Capitolo 3 (rispettivamente nelle Figure 3.2, 3.4 e 3.13) che mostrano la strategia ottima in termini dell'azione da intraprendere nei confronti della batteria rispetto al suo livello energetico parametrizzata per alcuni valori della richiesta di potenza: in tutti i casi π risulta sempre una funzione non-crescente, lineare a tratti e descrivibile tramite un numero finito di soglie rispetto al livello energetico della batteria.

Sviluppi futuri

Il primo passo che si ritiene costituisca la naturale continuazione del lavoro di tesi svolto sarebbe l'*ottimizzazione del processo di identificazione della strategia ottima di carica/scarica della batteria*. Si è provato per via simulativa che DP si comporta molto bene rispetto al lower bound sul costo corrispondente all'algoritmo AOS, però risulta parecchio costosa dal punto di vista computazionale, soprattutto se si pensa di diminuire il passo di quantizzazione del livello energetico della batteria (facendo in tal modo aumentare la cardinalità dello spazio degli stati) o di estenderla all'ottimizzazione congiunta dell'*energy storage* su più batterie. Nel Capitolo 5 si è dimostrato analiticamente quale forma abbia la funzione che definisce il livello energetico che la batteria deve assumere all'inizio di un certo slot temporale dato quello corrente e un modello di stato del sistema molto generale, ma non si è trovato un modo per calcolare le soglie relative al livello di carica, indicate come ε_i , se non conoscendo il sub-differenziale della funzione \bar{G} , la cui stima necessita appunto della Programmazione Dinamica. Il punto di arrivo ideale sarebbe quello di raffinare la strategia proposta dagli autori di [1]: l'algoritmo ST ha bisogno, come unico parametro di ingresso, del valore della richiesta media di potenza che arriva al controllore della grid. Un approccio che sfrutti una più dettagliata conoscenza statistica del sistema (ad esempio la matrice delle probabilità di transizione ad un passo di P) per inserirla all'interno di una strategia di *energy storage* più complessa, presumibilmente approssimando π come una funzione lineare a tratti a più soglie in modo che siano necessari un limitato numero di parametri per memorizzarla, porterebbe verosimilmente a delle prestazioni migliori in termini di costo complessivo da pagare da parte degli utenti della grid.

Un secondo passo che andrebbe a proseguire la presente trattazione sarebbe la *gestione ottima di più di una batteria in maniera centralizzata*. Nella Sezione 1.9 si è potuta constatare la complessità computazionale che presenta tale problema di ottimizzazione, il quale, definito come nell'equazione (1.32), risulta ancora trattabile come problema offline usando l'algoritmo di Viterbi come tecnica risolutiva, ma già con due batterie DP potrebbe presentare dei problemi di convergenza. Sarebbe interessante verificare, in primo luogo, se si manifesta una differenza di prestazioni in termini di costo risolvendo il problema di ottimizzazione (anche solo tramite AOS) rispetto ad un unico ingresso che accorpa tutte le richieste di potenza degli utenti all'interno della grid o a più ingressi, considerando *separatamente* le singole richieste di potenza: intuitivamente non dovrebbe esserci una grossa differenza di risultati a causa della forte correlazione presente nei consumi energetici degli utenti all'interno della micro-grid, ma questa affermazione andrebbe approfondita. Per rendere poi trattabile il problema di ottimizzazione con più batterie dal punto di vista centralizzato, bisognerebbe trovare una relazione che legghi l'identificazione della strategia ottima rispetto ad una sola batteria a quella rispetto ad un numero generico di esse: se si riuscisse a trovare una sorta di mappa tra le due funzioni, il secondo caso si potrebbe ricondurre ad una generalizzazione del primo, passando da un problema di ottimizzazione vettoriale congiunta a più problemi indipendenti di ottimizzazione scalare.

L'ultima strada che sarebbe affascinante cercare di percorrere è l'individuazione di una

soluzione ottima distribuita del problema di minimizzazione del costo energetico rispetto a più batterie, magari prendendo in considerazione la matrice di impedenza della grid e quindi le perdite lungo le linee di trasmissione. Da una ricerca sulla rete approfondita condotta in data 12/10/2012 questo argomento risulta ancora un terreno inviolato: soltanto in [21] si affronta l'argomento *Distributed Energy Storage Systems for Applications in Future Smart Grids*, ma vengono soltanto discusse le tecnologie di *energy storage* distribuito più adatte dal punto di vista hardware ad applicazioni per smart micro-grid. Non viene neanche accennato al problema di ottimizzazione del loro utilizzo in termini di azioni di carica/scarica che le batterie dovrebbero intraprendere per minimizzare la spesa energetica.

Bibliografia

- [1] I. Koutsopoulos, V. Hatzi, and L. Tassiulas, “Optimal energy storage control policies for the smart power grid,” in *Proc. of Second Int. Conf. on Smart Grid Commun.*, Brussels (B), oct 2011.
- [2] P. Van de Ven, N. Hegde, L. Massoulié, and T. Salonidis, “Optimal control of residential energy storage under price fluctuations,” in *Proc. of IARIA ENERGY 2011: The First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, Venice (I), may 2011, pp. 159–162.
- [3] S. Caproni, “Sistemi di accumulo per impianti fotovoltaici domestici: le batterie long-life di panasonic.” [Online]. Available: <http://newsenergia.com/sistemi-accumulo-fotovoltaico-accumulatori-0608.html>
- [4] “Smart grid: risparmio energetico ed efficienza.” [Online]. Available: <http://www.fotovoltaiconorditalia.it/mondo-fotovoltaico/smart-grid-risparmio-energetico-ed-efficienza>
- [5] J. Eyer and G. Corey, “Energy storage for the electricity grid: Benefits and market potential assessment guide,” Sandia National Laboratories, Tech. Rep., 2010.
- [6] M. Albadi and E. F. El-Saadany, “A summary of demand response in electricity markets,” *Electric Power Systems Research*, vol. 78, no. 11, pp. 1989–1996, 2008.
- [7] H. Allcott, “Rethinking real-time electricity pricing,” *Resource and Energy Economics*, vol. 33, no. 4, pp. 820–842, 2011.
- [8] G. Barbose and C. Goldman, “A survey of utility experience with real time pricing,” Berkeley National Laboratory, Tech. Rep., 2004.
- [9] M. Lijensen, “The real-time price elasticity of electricity,” *Energy Economics*, vol. 29, no. 2, pp. 249,258, 2007.
- [10] B. Li, S. Gangadhar, S. Cheng, and P. Verma, “Maximize user rewards in distributed generation environments using reinforcement learning,” in *Energytech, 2011 IEEE*, may 2011, pp. 1 –6.
- [11] PJM, “Pjm monthly locational marginal pricing.” [Online]. Available: <http://www.pjm.com/markets-and-operations/energy/real-time/monthlylmp.aspx>
- [12] Y. Li and P. Flynn, “Electricity deregulation, spot price patterns and demand-side management,” *Energy*, vol. 31, no. 6-7, pp. 908–922, 2006.
- [13] I. Richardson, M. Thomson, D. Infield, and C. Clifford, “Domestic electricity use: A high-resolution energy demand model,” *Energy and Buildings*, vol. 42, no. 10, pp. 1878–1887, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378778810001854>

- [14] M. R. Jongerden and B. Haverkort, “Which battery model to use?” *IET Software*, vol. 3, no. 6, pp. 445–457, 2009.
- [15] C. Chiasserini and R. Rao, “Energy efficient battery management,” *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1235–1245, 2001.
- [16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, 1998.
- [17] R. A. Howard, *Dynamic Probabilistic Systems, volume 2: Semi-Markov and Decision Processes*. John Wiley & Sons, 1972.
- [18] *Office for National Statistics, Social Trends No. 37*. Crown, table 2.1 Household by Size.
- [19] N. Benvenuto and G. Cherubini, *Algorithms for Communication Systems and their Applications*. John Wiley and Sons Ltd, 2002.
- [20] S. by EAS, “Li-ion batteries - the application engineer’s perspective,” university of Padova, 3rd May 2012.
- [21] M. G. Molina, “Distributed energy storage systems for applications in future smart grids,” in *Transmission and Distribution: Latin America Conference and Exposition (T D-LA), 2012 Sixth IEEE/PES*, sept. 2012, pp. 1 –7.

Ringraziamenti

Come continuazione della seconda pagina, voglio ringraziare innanzitutto i miei genitori a cui dedico questa tesi, i quali mi hanno sostenuto (oltre che da quando non ho neanche ricordi!) dal primo all'ultimo giorno del mio percorso di studi universitari, non solo condividendo la mia felicità di fronte al raggiungimento dei buoni risultati, ma anche dandomi sostegno e conforto nei momenti più negativi e difficili da vivere da soli, facendomi sempre sentire ripagato dell'impegno messo nello studio. **GRAZIE.**

Grazie davvero alla mia ragazza Valentina, che mi sopporta e mi rende *felice* ogni giorno dal secondo anno di Ingegneria (quindi, avendo una bassa considerazione delle matricole, dall'inizio della mia carriera universitaria!). Grazie per aver portato pazienza nelle diverse occasioni in cui ti ho negato la possibilità di trascorrere del tempo in cui avresti voluto avere la mia compagnia per stare semplicemente insieme a me o andare da qualche parte, sentendoti rispondere non posso “perchè devo studiare per l'esame xxx!” o “perchè devo lavorare alla tesi che sono preso con le bombe!”. Grazie per avermi tirato su il morale e fatto ridere e stare bene anche quando il mio umore tendeva a 0^+ e grazie per non essere iscritta ad ingegneria: in tal caso penso che i momenti della mia vita in cui parlerei di qualcosa di non tecnico si conterebbero sulle dita di una mano monca, e io diventerei ancora più nerd di quanto non sia già! **GRAZIE AMORE.**

Grazie a tutti i miei compagni di università (di cui più di uno è diventato anche mio collega alla Telecom), prima di tutto per l'amicizia davvero importante per me che ci lega, ma anche per tutto quello che mi avete insegnato, in alcuni casi anche senza rendervene conto. In particolare ci tengo a ringraziare Nicola Belli ed Alex Pegoraro per tutti le *fondamentali* consulenze tecniche che mi avete dato nel corso di questa tesi: grazie per tutta la gentilezza con cui avete risposto alla sorgente inesauribile di domande più o meno sensate costituita da me stesso. Ho iniziato a scrivere queste righe quasi un anno fa che non sapevo neanche cosa fossero nè \TeX nè un'immagine vettoriale: se il lavoro finale risulta esteticamente accettabile, è soprattutto merito vostro. Colgo qui l'occasione per ringraziare anche mio cugino Enrico e Marco Nicolosi: a voi ho rotto le scatole solo in un'occasione, ma mi avete rispettivamente insegnato ad “usare” Adobe Illustator (almeno per quello che serve ad un ingegnere perfezionista!) e passato il template fighissimo che ho usato nelle slide (ci avrei impiegato una settimana per crearne uno simile da zero!). **GRAZIE.**

Grazie a *tutti* i miei amici, che spero riescano ad essere presenti alla mia laurea nel maggior numero possibile (un pensiero va ad Eleonora e Silvia che sono all'estero), da quelli che vedo ogni giorno e con i quali condivido la mia quotidianità a quelli con cui riesco a stare insieme con frequenza molto minore, ma quando li incontro riescono a fare in modo che tutto il tempo passato dall'ultima volta che ci siamo visti/sentiti *sembri non essere mai trascorso*, perchè ad essere speciale è il rapporto che ci lega, e non il numero di volte che riusciamo ad uscire insieme. **GRAZIE.**

E a chiunque (anche sconosciuto) sia riuscito a leggere, non dico la tesi intera (spero sinceramente tu abbia di meglio da fare!), ma solo tutta quest'ultima pagina scassa-maroni che tenta di snocciolare chissà quali insegnamenti morali. **GRAZIE.**