



POLITECNICO DI MILANO

Scuola di Ingegneria civile, ambientale e territoriale - Dipartimento di
Elettronica e Informazione

Corso di Laurea in Ingegneria per l 'Ambiente ed il Territorio

Algoritmi evolutivi e reti neurali per la gestione delle risorse idriche

Relatore:

Prof. Andrea Castelletti

Correlatore:

Ing. Francesca Pianosi

Tesi di Laurea di:

Giulio Bonelli

Mat. 735928

Anno Accademico 2010-11

Indice

Introduzione	VIII
1 Inquadramento e formulazione del problema	1
1.1 L'acqua come risorsa	1
1.2 Inquadramento del processo gestionale	3
1.3 Il modello del sistema	3
1.3.1 Serbatoi	4
1.3.2 Traverse regolate	5
1.3.3 Bacini imbriferi	5
1.3.4 Indicatori per passo	6
1.3.5 Il modello globale	8
1.4 Formulazione del problema di controllo	8
1.4.1 Problema su orizzonte finito	9
1.4.2 Problema su orizzonte infinito	10
1.4.3 Problema a molti obiettivi	12
1.5 La soluzione tradizionale: l'approccio funzionale	15
1.6 La soluzione proposta: l'approccio parametrico	19
2 Metodi e strumenti utilizzati	21
2.1 Reti neurali	21
2.2 Algoritmi evolutivi	23
2.3 Gestire la ciclostazionarietà	26
2.4 Inizializzazione dell'algoritmo	27
2.5 Il seme	30
2.6 Indici di prestazione	31
2.6.1 Hypervolume	34
2.6.2 Soluzioni distinte di ranking 1	36
2.6.3 Crowding distance	36
2.6.4 Numero di convergenze	36
2.6.5 Conclusioni sugli indicatori utilizzati	37

3	Caso di Test	38
3.1	Descrizione del caso di Test	39
3.1.1	Parametri significativi	41
3.2	Scelta dell'architettura	43
3.3	Funzionamento dell'algoritmo	49
3.3.1	Caso SI: inizializzazione con regole semplici	50
3.3.2	Caso NO: inizializzazione completamente casuale	54
3.3.3	Simulazione Monte Carlo e confronto con la programmazione dinamica	58
3.3.4	Orizzonte temporale h	61
3.3.5	Prime conclusioni	64
3.4	Generalizzazione dei risultati	65
3.5	Validazione	73
3.6	Conclusioni sul caso di Test	74
4	Hoà Binh	76
4.1	Descrizione del sistema	76
4.1.1	Il serbatoio Hoà Binh	78
4.1.2	Il sistema a valle	79
4.1.3	Costi per passo	80
4.1.4	Il modello aggregato	82
4.2	Procedura seguita	82
4.3	Risultati della programmazione dinamica deterministica	83
4.4	Ingressi alla rete neurale: tempo e invaso	84
4.4.1	Scelta dell'architettura tramite addestramento sulle regole semplici	86
4.4.2	Ottimizzazione della rete con NsgaII	90
4.4.3	Analisi delle leggi di controllo	94
4.5	Ingressi alla rete neurale: tempo, invaso e afflussi del giorno precedente	97
4.5.1	Scelta dell'architettura tramite addestramento	97
4.5.2	Ottimizzazione della rete con NsgaII	100
4.5.3	Leggi di controllo	103
4.6	Confronto tra le reti con diversi ingressi	106
4.7	Conclusioni sul caso di studio	109
	Conclusioni	111
	Bibliografia	115

Elenco delle figure

1.1	Esempio di frontiera di Pareto (in azzurro) nel caso di problema a due obiettivi. I punti rossi rappresentano soluzioni dominate.	13
2.1	Schema di una generica rete neurale. Ogni freccia opera una modulazione del segnale attraverso dei pesi propri.	22
2.2	Esempio della ricerca del minimo durante la fase di addestramento, con una sola iterazione e 150 epoche. Il risultato potrebbe corrispondere a un minimo locale.	28
2.3	Esempio della ricerca del minimo durante la fase di addestramento, con tre iterazioni e 150 epoche. Il risultato finale è il miglior risultato trovato tra tutte le iterazioni.	29
2.4	Stocasticità nell'algoritmo.	31
2.5	Effetto del seme sulle prestazioni di diversi esperimenti.	32
2.6	Funzionamento di hypervolume.	35
3.1	Schema del sistema del caso di Test.	39
3.2	Leggi di controllo ottime per il solo obiettivo irriguo (a) e per l'obiettivo di riduzione delle piene (b).	40
3.3	Leggi di controllo attese.	41
3.4	Le leggi di controllo relative alle quattro regole semplici.	44
3.5	Approssimazione leggi di controllo con 2 neuroni tansig.	45
3.6	Approssimazione leggi di controllo con 5 neuroni purelin.	46
3.7	Approssimazione delle regole semplici con 3 neuroni tansig.	48
3.8	Frontiera di Pareto ottenuta con Caso SI, $pop = 100$ e $gen = 50$ zoomata (b) e non (a).	51
3.9	Andamento di hypervolume nelle generazioni (caso SI, $pop = 100$ e $gen = 50$), con $J_{max} = [5, 24]$	52
3.10	Range di parametri del cromosoma finale (caso SI, $pop = 100$ e $gen = 50$).	53
3.11	Frontiera di Pareto ottenuta con $pop = 100$ e $gen = 5$ (a) e $gen = 10$ (b) nel Caso SI.	53

3.12	Frontiera di Pareto (zoomata) ottenuta con $pop = 100$ e $gen = 10$, caso NO.	55
3.13	Range di parametri del cromosoma finale (Caso NO, $pop = 100$ e $gen = 10$).	55
3.14	Confronto tra Caso SI (soluzioni blu) e NO (soluzioni verdi). I cerchi gialli corrispondono alle inizializzazioni in comune, mentre quelli rossi a quelle del solo Caso SI (regole semplici 1 e 4).	56
3.15	Andamento di hypervolume nelle generazioni (Caso NO, $pop = 100$ e $gen = 50$).	57
3.16	Confronto tra Caso SI (soluzioni blu) e NO (soluzioni verdi). I cerchi gialli corrispondono alle inizializzazioni in comune, mentre quelli rossi a quelle del solo Caso SI (regole semplici 1 e 4). In questo caso (con un altro seme) si nota che il Caso NO non è riuscito a esplorare la frontiera.	57
3.17	Confronto tra la frontiera ottenuta mediante simulazione MC dei risultati dell'algoritmo evolutivo ($pop = 100$, $gen = 50$), e quelli della SDP.	59
3.18	Esempi con alcuni semi di confronti tra Caso SI (blu) e NO (verde) con simulazione MC, $pop = 100$ e $gen = 50$. I cerchi gialli sono le inizializzazioni in comune (casuali), quelle rosse le regole semplici.	60
3.19	Confronto con simulazione MC tra le popolazioni ottenute con $gen = 50$ e $gen = 7$ ($pop = 100$).	61
3.20	Frontiera di Pareto della popolazione Caso SI, $pop = 100$ e $gen = 50$ simulata con $h = 4990$ (a), $h = 550$ (b), $h = 100$ (c) e $h = 30$ (d). In blu le soluzioni, in cerchi rossi le inizializzazioni.	62
3.21	Esempio di legge di controllo con parti inefficienti.	63
3.22	Esempi di leggi di controllo insensate (confrontare con Figura 3.3), ottenute con notevole frequenza per limitati valori dell'orizzonte temporale h	64
3.23	Andamento dei valori medi di Hypervolume HD (a) e N_{soleff} (b) con le generazioni nei Casi SI e NO. $pop = 100$	72
3.24	Andamento dei valori medi di Hypervolume HD (a) e N_{soleff} (b) con le generazioni nei Casi SI e NO. $pop = 20$	72
4.1	Il bacino del fiume Rosso, Vietnam.	77
4.2	Schema del sistema idrico del fiume Rosso.	78

4.3	Soluzioni della programmazione dinamica deterministica (blu) che dominano la gestione storica (rosso). La grandezza dei cerchi è proporzionale al valore (negativo) dell'obiettivo idroelettrico.	84
4.4	Frontiera con $pop = 600$, $gen = 2000$ (ingressi della rete $I_t = [s_t, t]$). La grandezza dei cerchi è proporzionale al valore (negativo) dell'obiettivo J_{idr} . In rosso la gestione storica, in verde le soluzioni della DDP.	93
4.5	Confronto tra le soluzioni della SDP (verde) e dell'algoritmo evolutivo (ingressi della rete $I_t = [s_t, t]$) che dominano la storia. In rosso la gestione storica	94
4.6	Esempio di legge di controllo ottenuta con $pop = 600$, $gen = 2000$ (rete $I_t = [s_t, t]$). La funzione (a) è tipica dei primi e degli ultimi giorni dell'anno, mentre le altre del periodo centrale (afflussi elevati).	95
4.7	Esempi di leggi di controllo (ingressi della rete $I_t = [s_t, t]$) con sezioni fisicamente insensate (blu). In rosso le stesse leggi vincolate ai minimi e massimi valori ammissibili del controllo.	96
4.8	Andamento di Hypervolume (calibrazione) nelle generazioni, rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$ e $pop = 800$	101
4.9	Frontiera con $pop = 800$, $gen = 2400$ (ingressi alla rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$). La grandezza dei cerchi è proporzionale al valore (cambiato di segno) dell'obiettivo J_{idr} . In rosso la gestione storica.	103
4.10	Legge di controllo in funzione dell'invaso (a) e degli afflussi del giorno precedente (b). Il colore dei punti identifica l'istante temporale. In funzione dell'asse delle ascisse non si ha un andamento preciso, ma gli allineamenti monocolori per ogni livello del controllo indicano una forte dipendenza dal tempo.	104
4.11	Legge di controllo in funzione del tempo (giorno dell'anno). Il colore dei punti identifica l'invaso (a) e gli afflussi del giorno precedente (b). In questo caso si ha un andamento definito in funzione dell'ascissa (tempo).	104
4.12	Legge di controllo in funzione dell'invaso, con il colore che indica il valore degli afflussi (a), e viceversa (b). Gli afflussi si riferiscono a q_t^{YB} , ma l'andamento è simile per tutti e tre gli afflussi.	105
4.13	Unione delle soluzioni di ranking 1 che dominano la gestione storica (rosso): rete 1 (verde) e rete 2 (blu).	108

Elenco delle tabelle

3.1	Prestazioni delle diverse architetture, con $N_{iter} = 1$ per la seconda regola semplice e $N_{iter} = 100$ per le altre. R2 è la varianza spiegata relativa a ognuna delle quattro regole semplici.	47
3.2	Numero medio di iterazioni necessarie per ottenere valori di varianza spiegata $R2 > 0,9989$ (approssimazione perfetta) in tutte le regole semplici (tre neuroni).	49
3.3	Numero medio di iterazioni necessarie per ottenere valori di varianza spiegata $R2 > 0,9980$ (approssimazione buona) in tutte le regole semplici (tre neuroni).	49
3.4	Risultati medi relativi a $pop = 100$.	66
3.5	Risultati medi relativi a $pop = 20$.	69
4.1	Risultati in validazione della programmazione dinamica deterministica.	85
4.2	Prestazioni delle diverse architetture, in termini di varianza spiegata R2, nel riprodurre le diverse regole semplici (ingressi della rete $I_t = [s_t, t]$).	89
4.3	Risultati dei diversi esperimenti di ottimizzazione con valori crescenti di pop e gen , in calibrazione e validazione (ingressi della rete $I_t = [s_t, t]$).	91
4.4	Risultati in calibrazione e validazione (ingressi della rete $I_t = [s_t, t]$, $pop = 600$ e $gen = 2000$). In alto la gestione storica (solo val), poi le soluzioni N_{dom} che dominano la gestione storica, e le soluzioni che ottimizzano ognuno dei tre obiettivi (in validazione). In basso le regole semplici.	92
4.5	Risultati in validazione della programmazione dinamica stocastica.	93
4.6	Prestazioni delle diverse architetture, in termini di varianza spiegata R2 nel riprodurre le varie regole semplici (ingressi della rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$).	99

4.7	Risultati dell'esperimento $pop = 800$, $gen = 2400$, in calibrazione e validazione (rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$). In alto la gestione storica (solo val), poi le soluzioni N_{dom} che dominano la gestione storica, e le soluzioni che ottimizzano ognuno dei tre obiettivi (in validazione). In basso le prestazioni delle regole semplici.	102
4.8	Confronto tra le prestazioni delle reti 1 (ingressi $I_t = [s_t, t]$) e 2 (con anche gli afflussi in ingresso).	106
4.9	Confronto tra le soluzioni che dominano la gestione storica della rete 1 (ingressi $I_t = [s_t, t]$) e della rete 2 (ingressi $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$).	107

Introduzione

L'acqua è una risorsa naturale molto preziosa, ma anche limitata. In molte zone del mondo non c'è acqua a sufficienza per soddisfare i bisogni della popolazione e garantirle una qualità di vita sufficiente, e anche nelle regioni più fortunate i forti consumi della società moderna e il degrado degli ecosistemi acquatici a causa di fattori antropici di varia natura, rendono gli scenari futuri sempre più critici.

In un simile contesto, ottimizzare l'uso della risorsa appare fondamentale. A questo proposito, opere civili come canali e serbatoi permettono di riallocare la risorsa nello spazio e nel tempo, permettendo di razionalizzarne l'uso. La gestione di un sistema idrico è però un'attività molto delicata, dal momento che quasi sempre influisce sugli interessi di diverse categorie di soggetti, e molto spesso condiziona anche diverse realtà politico-territoriali. Tutt'ora, in molti casi la politica di gestione di un sistema idrico non tiene conto di questi due aspetti fondamentali, generando tensioni e conflitti, e in generale uno stato di inefficienza. Per questo motivo, risulta di fondamentale importanza adottare un modello gestionale che consideri in modo integrato tutte le componenti connesse al sistema idrico in esame.

Lo scopo di questo lavoro è quello di definire un approccio innovativo per ottenere una gestione efficiente e equa di un sistema idrico. Per fare ciò ci si avvarrà di due strumenti: le reti neurali e gli algoritmi evolutivi. Le prime rappresentano una particolare classe di funzioni che possiede dei vantaggi pratici considerevoli, che torneranno utili per la definizione di politiche di regolazione. I secondi sono invece strumenti di ottimizzazione che si basano su un approccio molto particolare per la ricerca di soluzioni efficienti, fondato sull'analogia con il processo di selezione naturale che si osserva in natura per le specie viventi. Gli algoritmi evolutivi sfruttano un'ottica a molti obiettivi, particolarmente adatta nell'ottimizzazione della gestione dei sistemi idrici, caratterizzati nella maggior parte dei casi da molteplici interessi conflittuali tra loro.

Nel primo capitolo verrà presentata una breve introduzione sulla delicata questione dell'acqua, mostrando tutte le principali tematiche connesse. Successivamente verrà mostrato un approccio sistemico per la definizione di quello che viene chiamato "problema di controllo", la cui soluzione mira a fornire una politica di regolazione del sistema idrico che sia il più possibile efficiente. Questo approccio si basa sull'introduzione di un modello matematico del sistema fisico, e sulla definizione di funzioni obiettivo che riflettono i diversi interessi connessi alla gestione del sistema. Verranno quindi presentati i principali strumenti utilizzati operativamente per la risoluzione di un problema di controllo, e ne verranno illustrati i principali pregi e difetti.

Nel secondo capitolo verranno descritti gli strumenti e la metodologia utilizzata in questo lavoro per la risoluzione di un problema di controllo, e conseguentemente per ottenere delle politiche di regolazione il più possibile efficienti. Dapprima verranno introdotte le reti neurali, di cui saranno presentate le principali caratteristiche e potenzialità. Quindi saranno trattati gli algoritmi evolutivi e in particolare NsgaII, l'algoritmo utilizzato in questa trattazione. Verranno quindi introdotte alcune innovazioni circa l'inizializzazione dell'algoritmo e la gestione della ciclostazionarietà, che permetteranno di incrementare le prestazioni degli strumenti utilizzati. Infine, sarà discussa una serie di indicatori atti a valutare la bontà dei risultati ottenuti nei diversi esperimenti di ottimizzazione.

Nel terzo capitolo, l'approccio proposto verrà testato su un caso fittizio, caratterizzato da un modello molto semplice e da soluzioni di forma nota. In tal modo sarà possibile valutare, attraverso gli indicatori introdotti nel capitolo 2, come variano le prestazioni modificando i parametri di tuning della rete neurale e dell'algoritmo evolutivo, e introducendo le innovazioni discusse nel capitolo 2. Verranno quindi paragonati i risultati con quelli della programmazione dinamica, uno strumento molto potente e largamente utilizzato nell'ambito dei problemi di ottimizzazione, descritto nel capitolo 1.

Nel quarto capitolo, l'approccio proposto sarà applicato su un caso reale, il serbatoio del'Hoa Binh in Vietnam. L'ottimizzazione verrà ripetuta per due configurazioni diverse della rete neurale con diversi ingressi, verificando quale può dare i risultati migliori. In ognuno dei due casi, le politiche di regolazione ottenute come soluzione saranno analizzate e confrontate con la politica storica.

Infine, nel quinto capitolo verranno avanzate delle conclusioni generali in merito al lavoro svolto e all'approccio utilizzato, discutendo a proposito di possibili vie di sviluppo in eventuali studi futuri.

Capitolo 1

Inquadramento e formulazione del problema

In questo primo capitolo verrà riportata un'introduzione sull'importanza e la criticità della risorsa acqua nei più diversi contesti politico - territoriali, e di conseguenza la necessità di una gestione razionale dei sistemi idrici. Verrà presentato un breve inquadramento del punto di vista dal quale si analizza la questione, nonché della metodologia seguita per formulare un modello gestionale che generi concreti benefici.

1.1 L'acqua come risorsa

Fin dall'antichità è stata compresa l'enorme importanza che ha l'acqua per la vita e la prosperità delle persone. Non è un caso, infatti, che le prime civiltà siano sorte proprio sulle sponde di importanti corsi d'acqua, come ad esempio i sumeri su quelle dei fiumi Tigri e Eufrate e il popolo egizio su quelle del Nilo. Tuttavia, specialmente col crescere della popolazione, la sola presenza della risorsa può non essere più sufficiente per garantire la qualità della vita, e da qui nasce la necessità di introdurre anche una corretta gestione. Così, già agli antichi romani risalgono importanti opere civili adibite a tale scopo, come un sofisticato sistema di acquedotti che permetteva un efficiente approvvigionamento dell'acqua a gran parte della città: esso divenne uno dei simboli della prosperità della civiltà romana, oltre a generare vantaggi dal punto di vista sociale ed economico.

L'acqua utilizzabile rappresenta purtroppo solo il 2,5% di quella totale presente nel globo terrestre, e inoltre di questa i 2/3 sono concentrati nei ghiacciai e non sono quindi utilizzabili. Con i consumi che si registrano oggi, questa quantità non sembra sufficiente per garantire una copertura totale

del fabbisogno. Oggi uno dei fattori di maggiore interesse sociopolitico è legato alla questione che l'acqua non è solo poca, ma anche distribuita in modo disomogeneo sulla superficie terrestre. La maggior parte di essa è concentrata in alcuni grandi complessi lacuali, come la regione dei Grandi Laghi in Nord America e i laghi Tanganika, Vittoria, Malawi in Africa, e nei maggiori sistemi fluviali: ad esempio il Rio delle Amazzoni, il Gange con il Bramaputra, il Congo, lo Yangtze e l'Orinco. Il risultato di questa situazione è che ad oggi più del 30% dell'umanità non ha acqua a sufficienza. Si consideri, inoltre, che nell'ultimo secolo la popolazione è triplicata, mentre i consumi sono aumentati di quasi 7 volte. Anche considerando una forte riduzione del tasso di crescita della popolazione mondiale, questi dati non promettono miglioramenti della situazione per il futuro.

Oggi la crisi della risorsa idrica è stata aggravata dall'interazione dinamica di molti processi sia a livello locale che a livello globale, ad esempio fattori ambientali (cambiamenti climatici, desertificazione, . . .); fattori economici (inquinamento da produzione, da movimentazione e smaltimento, . . .), sociali e culturali.

Il problema dell'acqua interessa ogni parte del mondo: per i paesi poveri la presenza di una sufficiente quantità d'acqua potrebbe permettere loro non solo di garantire la sopravvivenza delle persone, ma anche alle nazioni stesse di uscire dalla condizione di povertà. Nei paesi ricchi la qualità di laghi e fiumi è in molti casi fortemente declinata a causa di numerosi fattori antropici legati allo sviluppo economico, pertanto un loro recupero potrebbe garantire un miglioramento della qualità della vita. L'acqua è come detto una risorsa limitata, che si mantiene fundamentalmente costante nel tempo. Purtroppo, risulta anche particolarmente complicato acquisire una maggiore percentuale di quella totale disponibile nel pianeta. L'unica via percorribile è quindi aumentare la razionalizzazione nel suo utilizzo. La costruzione di dighe, serbatoi, canali di diversificazione e di altre opere civili ha permesso in molte zone del pianeta di riallocare la risorsa nello spazio e nel tempo secondo le esigenze (nei limiti del possibile). Tuttavia, la gestione da parte di una singola nazione di un sistema idrico che influenza anche altri stati ha generato in passato in molte parti del mondo, e genera tutt'ora, un conflitto di interessi che in più occasioni ha portato a un inasprimento dei rapporti tra le nazioni stesse. Inoltre, anche a livello di singola realtà politico-territoriale, nella gestione di un sistema idrico devono convivere diversi interessi che risultano spesso conflittuali tra di loro. Si intuisce quindi la notevole importanza di introdurre un modello gestionale che permetta di ottenere delle soluzioni che generino efficienza e garantiscano il consenso dei diversi portatori d'interesse. Questa metodologia si inserisce nel contesto del paradigma IWRM (Integrated Water Resources Management, GWP (2003)), che stabilisce l'im-

portanza (per i motivi esposti) di effettuare un processo decisionale integrato e partecipato.

1.2 Inquadramento del processo gestionale

In questo lavoro, considerando il contesto descritto, per ottenere una gestione positiva di un sistema idrico verrà utilizzato il seguente approccio. Si cercherà innanzitutto di comprendere il funzionamento e le relazioni che intercorrono tra le diverse componenti di un sistema idrico. Esse verranno rappresentate in termini modellistici, ovvero attraverso equazioni matematiche che permettono di descrivere l'andamento delle principali variabili di interesse di un sistema idrico. Questi modelli possono essere fisicamente basati (ovvero creati in modo da descrivere il processo che genera un fenomeno) oppure a scatola nera (ovvero senza descrivere il processo fisico che governa il fenomeno, ma riportando solo relazioni statistiche ingresso/uscita).

Nei prossimi paragrafi verranno dunque introdotti i modelli dei principali componenti, e verrà mostrata la procedura per aggregarli in un unico modello rappresentante il sistema idrico oggetto di studio. Come uscita di tale modello si considerano una serie di indicatori, che rappresentano modellisticamente gli interessi dei diversi portatori.

A partire dal modello del sistema è quindi possibile introdurre e risolvere quello che viene chiamato problema di controllo. La risoluzione di un problema di controllo permette di ottenere una politica di gestione del sistema idrico che risulti il più possibile efficiente, dal punto di vista degli obiettivi considerati (legati agli indicatori introdotti nel sistema).

In questo ambito il punto di vista che verrà assunto sarà dunque quello dell'analista, una figura professionale che possiede le conoscenze e gli strumenti necessari per poter garantire un supporto tecnico all'interno del processo decisionale (nel contesto della regolazione di un sistema idrico). È importante sottolineare nuovamente la necessità di garantire trasparenza e condivisione dell'informazione tra tutte le parti interessate, al fine di raggiungere un compromesso che generi il massimo consenso possibile.

1.3 Il modello del sistema

In questa sezione verrà mostrato come modellizzare correttamente un sistema idrico nelle principali componenti che più spesso si ritrovano in esso. In particolare si considera un sistema composto da un certo numero di serbatoi che ricevono afflussi da vari bacini imbriferi. I serbatoi sono collegati tra di

loro e con le utenze, quali per esempio centrali idroelettriche o distretti irrigui, tramite una rete di canali naturali e artificiali. Naturalmente i processi fisici che compongono il sistema sono tempo-continui, tuttavia il modello utilizzato è tempo-discreto poiché la decisione viene presa in istanti temporali discreti. Per una descrizione più dettagliata del sistema, ad esempio per i modelli dei canali, si rimanda a Soncini Sessa (2004).

1.3.1 Serbatoi

Il modello del j -esimo serbatoio della rete si basa sull'equazione di bilancio di massa

$$s_{t+1}^j = s_t^j + a_{t+1}^j - r_{t+1}^j + \varepsilon_{t+1} \quad (1.1)$$

dove s_t^j è l'invaso nel serbatoio j -esimo nell'istante t , a_{t+1}^j è il volume di afflusso nell'intervallo $[t, t+1)$, r_{t+1}^j è il rilascio nello stesso intervallo e ε_{t+1} rappresenta il disturbo casuale, ovvero un contributo esterno al sistema che non può essere noto all'istante t .

Si noti che nel modello del serbatoio si sono trascurati altri termini, quali la precipitazione, l'infiltrazione e l'evaporazione, nonché eventuali disturbi deterministici (contributi esterni al sistema modellizzato ma deterministicamente noti al tempo t). Queste componenti possono essere aggiunte al bilancio di massa se necessario. Il pedice temporale di ogni variabile denota l'istante in cui questa è nota deterministicamente. Così l'invaso misurato all'istante t è indicato con s_t mentre l'afflusso nell'intervallo $[t, t+1)$ si indica con a_{t+1} poiché si realizza e diviene noto deterministicamente solo alla fine di questo intervallo.

L'afflusso a_{t+1}^j del j -esimo serbatoio può rappresentare l'uscita di uno o più degli altri componenti del sistema, come bacini imbriferi a esso collegati o rilasci r_{t+1}^i ($i = 1, \dots, i \neq j$) di eventuali serbatoi modellizzati a monte.

Il rilascio r_{t+1}^j è funzione della variabile di controllo u_t^j , cioè la decisione di rilascio all'istante t per il serbatoio j , dell'invaso s_t^j e dell'afflusso a_{t+1}^j (e eventualmente di altri componenti considerati nel modello, come l'evaporazione specifica)

$$r_{t+1}^j = R_t^j(s_t^j, u_t^j, a_{t+1}^j, \varepsilon_{t+1})$$

$R_t^j(\cdot)$ rappresenta la funzione di rilascio ed è una funzione non lineare della seguente forma

$$R_t^j(s_t^j, u_t^j, a_{t+1}^j) = \begin{cases} v_t^j(s_t^j, a_{t+1}^j) & \text{se } u_t^j < v_t^j(s_t^j, a_{t+1}^j) \\ V_t^j(s_t^j, a_{t+1}^j) & \text{se } u_t^j > V_t^j(s_t^j, a_{t+1}^j) \\ u_t^j & \text{altrimenti} \end{cases} \quad (1.2)$$

dove $v_t^j(\cdot)$ e $V_t^j(\cdot)$ sono le funzioni di minimo e massimo rilascio che può prodursi nell'intervallo temporale $[t, t + 1)$ tenendo le paratoie completamente chiuse o completamente aperte, rispettivamente. Questa forma delle funzioni di rilascio permette di tenere conto dei vincoli fisici, in quanto può non essere fisicamente possibile che il rilascio verificatosi nell'intervallo $[t, t + 1)$ sia esattamente pari alla decisione di rilascio presa all'istante t . Le funzioni di minimo e massimo rilascio sono calcolabili integrando, nell'intervallo $[t, t + 1)$, l'equazione continua di bilancio di massa del serbatoio

$$\frac{ds^j}{d\zeta} = a^j(\zeta) - r^j(\zeta) \quad (1.3)$$

dove si ipotizza l'afflusso istantaneo $a^j(\zeta)$ costante e uguale a a_{t+1}^j/Δ , con Δ lunghezza del passo di modellizzazione, e il rilascio istantaneo $r^j(\zeta)$ è dato dalla minima $N^{\min,j}(s^j(\zeta))$ o massima $N^{\max,j}(s^j(\zeta))$ relazione vaso-rilascio delle paratie, derivabili dalle specifiche di costruzione del serbatoio stesso, quindi note.

1.3.2 Traverse regolate

In molti sistemi idrici capita che siano presenti delle traverse regolate, ovvero piccole dighe dotate di organi mobili tramite le quali è possibile diramare parte della portata di un corso d'acqua in un canale artificiale, che permette di servire determinate utenze. Sul corso d'acqua a valle della traversa è spesso imposto per legge un deflusso minimo vitale (DMV), per cui la portata derivata dalla traversa regolata è espressa dalla seguente equazione

$$q_{t+1}^d = \min \left\{ u_t, (q_{t+1}^m - DMV), q^{max} \right\} \quad (1.4)$$

in cui u_t è il controllo, q_{t+1}^m rappresenta la portata del corso d'acqua a monte e q^{max} la massima portata derivabile.

La portata del corso d'acqua a valle della traversa sarà invece

$$q_{t+1}^v = q_{t+1}^m - q_{t+1}^d \quad (1.5)$$

1.3.3 Bacini imbriferi

Per la formulazione di modelli dei deflussi dai bacini imbriferi naturali, esistono un'ampia letteratura e una notevole varietà di approcci disponibili, di cui per ragioni di spazio verranno sommariamente presentati solo pochi esempi. Lo stato dei modelli fisicamente basati in questo caso di solito non

è osservabile e può inoltre essere molto grande (centinaia di variabili quando il modello è distribuito spazialmente), provocando problemi computazionali: per tale motivo solitamente si usano semplici modelli statistici.

Per citare un esempio frequentemente utilizzato, il deflusso a_{t+1}^k dal k -esimo bacino non regolato può essere modellizzato come un processo stocastico ciclostazionario lognormale, con media e deviazione standard periodiche μ_t^k e σ_t^k e con dinamica

$$a_{t+1}^k = \exp(y_{t+1}^k \cdot \sigma_t^k + \mu_t^k) \quad (1.6a)$$

$$A^k(z^{-1})y_{t+1}^k = \varepsilon_{t+1}^k \quad (1.6b)$$

dove A^k è un operatore polinomiale nell'operatore z^{-1} ($z^{-1}y_{t+1}^k = y_t^k$) e ε_{t+1}^k è un rumore bianco (Barrow (1991)) gaussiano con media nulla e varianza costante.

In alternativa, si possono utilizzare direttamente le serie storiche registrate dei deflussi provenienti dai bacini considerati. Questo è il caso di alcuni metodi operativi di risoluzione utilizzati in quest'ambito, e sarà anche l'approccio seguito in questo lavoro.

Si noti che il deflusso proveniente da un bacino imbrifero, qualora ne rispetti i requisiti (o se non è richiesta una elevata precisione), può non essere modellizzato e essere trattato come un disturbo casuale.

1.3.4 Indicatori per passo

La presenza di diversi soggetti direttamente interessati dall'evoluzione del sistema può essere formalizzata mediante la definizione di costi per passo, che rappresentano il costo che il determinato portatore di interessi deve sopportare a ogni istante t per il fatto che il sistema si trova, in t , in un determinato stato, con l'utilizzo di un dato controllo e per una data realizzazione del disturbo. Un'altra soluzione può consistere nell'imposizione di vincoli su determinate variabili. Normalmente si cerca di evitare il secondo approccio in quanto i vincoli sono soluzioni più rigide, e generano problemi se per qualche motivo, in qualche istante, uno di essi non può essere soddisfatto, a causa ad esempio del manifestarsi di disturbi casuali di una certa entità (eventualmente il vincolo si può sostituire con un vincolo in probabilità). In questo lavoro si descriveranno gli interessi dei diversi portatori esclusivamente attraverso indicatori per passo.

Non è possibile affrontare una discussione generale dell'argomento a causa della grande varietà di interessi legati alla gestione dei sistemi idrici; ci si limiterà dunque a fornire qualche esempio delle più comuni categorie di portatori.

Un interesse comunemente molto importante nei sistemi idrici è certamente legato alle esondazioni. A questo obiettivo possono essere associate parecchie forme dei costi per passo, in base alla situazione, ma un esempio può essere fornito da

$$g_{t+1}^{es} = (h_{t+1} - \bar{h})^+ \quad (1.7)$$

nel caso dei territori adiacenti a un serbatoio, dove h rappresenta il livello del serbatoio (direttamente misurabile) e \bar{h} il livello di esondazione; mentre per i territori che si trovano a valle del serbatoio

$$g_{t+1}^{es} = (r_{t+1} - \bar{q})^+ \quad (1.8)$$

in cui r_t è il rilascio del serbatoio e \bar{q} la portata di esondazione. L'operatore $y = x^+$ rappresenta $y = \max(x, 0)$.

Per quanto riguarda la salvaguardia del fiume e dell'ecosistema ripario a valle del serbatoio, essa può essere garantita tramite l'introduzione di un vincolo di deflusso minimo vitale sul rilascio dal serbatoio. Un costo per passo allora può essere tipicamente

$$g_{t+1}^{amb} = (DMV - r_{t+1})^+ \quad (1.9)$$

Gli interessi di una generica compagnia idroelettrica che possiede l'impianto j -esimo sono rappresentati attraverso un costo per passo che esprime la produzione di potenza G_{t+1}^j nell'intervallo $[t, t+1)$

$$G_{t+1}^j = \eta^j \gamma g q_{t+1}^{d,j} H_t \quad (1.10a)$$

dove η^j è un fattore di conversione e di efficienza, γ è la densità dell'acqua, g è l'accelerazione di gravità, $q_{t+1}^{d,j}$ è la portata in condotta e H_t il carico idraulico. Quando la vasca di carico dell'impianto è il serbatoio stesso, H_t dipende dal livello della superficie idrica e quindi dall'invaso s_t^j . La portata $q_{t+1}^{d,j}$ non coincide sempre con il rilascio dal serbatoio a causa della presenza di una portata minima $q^{\min,j}$ e massima $q^{\max,j}$ turbinabile dall'impianto, e/o dal DMV imposto

$$q_{t+1}^{d,j} = \begin{cases} \min((r_{t+1}^j - \tilde{q}_t^j)^+, q^{\max,j}) & \text{se } (r_{t+1}^j - DMV)^+ \geq q^{\min,j} \\ 0 & \text{altrimenti} \end{cases} \quad (1.10b)$$

Gli interessi di un (l -esimo) distretto irriguo possono essere espressi attraverso un costo per passo che tenga conto del deficit di fornitura

$$d_{t+1}^l = (w_t^l - q_{t+1}^{d,l})^+ \quad (1.11)$$

dove $q_{t+1}^{d,l}$ è la portata fornita al distretto irriguo e w_t^l è la domanda idrica. Quest'ultima può essere espressa da uno scenario dato o tramite un modello dinamico di crescita delle colture (soluzione più precisa ma ovviamente più complessa).

1.3.5 Il modello globale

Per ottenere il modello globale del sistema si aggregano in modo appropriato i modelli dei vari componenti che lo compongono: i serbatoi, i bacini imbriferi, le utenze idriche, i canali, le traverse e le congiunzioni. Il risultato è un sistema tempo-discreto, periodico, non lineare, stocastico (o incerto) nella forma

$$x_{t+1} = f_t(x_t, u_t, \varepsilon_{t+1}) \quad (1.12)$$

dove $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$ e $\varepsilon_t \in \mathbb{R}^{n_\varepsilon}$ sono i vettori di stato, controllo e disturbo, rispettivamente. Lo stato è composto dalle variabili di stato degli N serbatoi (in generale i loro invasi), e degli altri componenti che richiedono una descrizione dinamica (per esempio bacini imbriferi, eventualmente canali, ...)

$$x_t = [s_t^1, \dots, s_t^N; y_t^1, \dots, y_{t-p_1}^1; \dots; y_t^M, \dots, y_{t-p_M}^M, \dots]^T$$

dove p_k è l'ordine del polinomio $A^k(z^{-1})$ nell'equazione (1.6b). Il controllo è un vettore composto dalle N decisioni di rilascio per gli N serbatoi, e di eventuali Tr traverse regolate

$$u_t = [u_t^1, \dots, u_t^{N+Tr}]^T$$

Il vettore dei disturbi è composto da tutte le L variabili casuali presenti nei modelli dei componenti del sistema: i disturbi stocastici che compaiono nei modelli dei bacini imbriferi non regolati, gli eventuali termini casuali nell'equazione di bilancio di massa del serbatoio (es. evaporazione, infiltrazione, ecc.) o nei modelli dei canali e delle utenze.

$$\varepsilon_{t+1} = [\varepsilon_{t+1}^1, \dots, \varepsilon_{t+1}^L]^T$$

A seconda di come sono modellizzati i singoli disturbi, il vettore ε_{t+1} può essere incerto o stocastico ed è descritto rispettivamente dall'insieme di appartenenza Ξ_t o dalla distribuzione di probabilità $\phi_t(\cdot)$. In ogni istante t , Ξ_t e $\phi_t(\cdot)$ sono funzioni dello stato e del controllo nello stesso istante temporale

$$\varepsilon_{t+1} \sim \phi_t(\cdot | x_t, u_t) \quad \text{o} \quad \varepsilon_{t+1} \in \Xi_t(x_t, u_t) \quad (1.13)$$

1.4 Formulazione del problema di controllo

Il problema di controllo che si vuole introdurre mira a produrre una regolazione del sistema che sia il più possibile efficiente rispetto a m obiettivi, corrispondenti ai punti di vista degli m portatori d'interesse considerati. La regolazione si può attuare attraverso le variabili su cui si ha il controllo (u_t)

per tutto il periodo di progetto considerato. Vista la stocasticità che permea il problema risulta conveniente definire come risultato, anziché una serie di controlli ottimi $u_t : t = 0, \dots, h - 1$ (h orizzonte del problema), una serie di *leggi di controllo*, ovvero di funzioni che associano la decisione di rilascio ottima al valore di alcune informazioni disponibili. La statistica sufficiente di una legge di controllo, che rappresenta la minima informazione necessaria per produrre la massima prestazione, è rappresentata dallo stato del sistema (e al limite, se presenti, dai disturbi deterministici) (Soncini Sessa (2004)). A valle della considerazione appena effettuata, le decisioni vengono rappresentate, in ogni istante temporale, dalla legge di controllo

$$u_t = m_t(x_t) \quad (1.14)$$

Una successione di leggi di controllo $m_t(\cdot) : t = 0, \dots, h - 1$ viene definita *politica*. Questa rappresenta proprio l'incognita del nostro problema, definito formalmente nei prossimi paragrafi.

1.4.1 Problema su orizzonte finito

In questo paragrafo verrà introdotto il problema di controllo considerando l'orizzonte finito $[0, h]$. Le funzioni obiettivo J^i (con $i = 1, \dots, m$) a cui è associata la bontà di una politica vengono definite a partire da ognuno degli m portatori di interesse individuati nel sistema (utenze idroelettriche e irrigue, limitazione esondazioni, qualità ambientale, ecc.):

$$J^i = \Psi_{\varepsilon_1^h} [Z^i(x_0^h, u_0^{h-1}, \varepsilon_1^h)] \quad (1.15)$$

dove $y_{t_1}^{t_2}$ è la traiettoria della variabile y_t da t_1 a t_2 , il funzionale $Z^i(\cdot)$ esprime il costo totale per il settore i -esimo in corrispondenza di una determinata traiettoria delle variabili del sistema e Ψ è la statistica usata per filtrare il disturbo. Le statistiche più comunemente utilizzate sono il valore atteso ($\Psi = E$, criterio di Laplace) e il massimo ($\Psi = \max$, criterio di Wald). Quest'ultimo rappresenta la situazione di massima avversione al rischio da parte dei portatori di interesse, in quanto restituisce la prestazione certa (nella condizione più sfortunata possibile). Per la risoluzione del problema, numerose tecniche (di cui alcune mostrate nei prossimi paragrafi) richiedono l'ipotesi che $Z^i(\cdot)$ sia separabile per ogni i , cioè

$$Z^i(x_0^h, u_0^{h-1}, \varepsilon_1^h) = \Phi [g_t^i(x_t, u_t, \varepsilon_{t+1}) : t = 0, \dots, h - 1; g_h^i(x_h)] \quad (1.16)$$

dove $g_t^i(\cdot)$ sono i costi per passo associati alla transizione da t a $t + 1$, $g_h^i(\cdot)$ è una penale associata allo stato finale e Φ è l'operatore di aggregazione temporale. Esempi delle funzioni dei costi per passo sono le equazioni (1.7), (1.8),

(1.9), (1.10a) e (1.11). La separabilità è una proprietà sempre raggiungibile a patto di allargare opportunamente lo stato del sistema (Soncini Sessa (2004)). Per mantenere questa proprietà anche attraverso il filtraggio dell'incertezza è necessario che si utilizzi come operatori di aggregazione temporale la somma ($\Phi = \Sigma$) quando il criterio di filtraggio è il valore atteso e il massimo ($\Phi = \max$) qualora si utilizzi invece il criterio di Wald. Ovviamente, il criterio di Laplace può essere utilizzato solo quando sia nota la distribuzione di probabilità dei disturbi ($\varepsilon_{t+1} \sim \phi_t$) mentre per utilizzare il criterio di Wald si possono utilizzare entrambe le forme del disturbo inserite nell'equazione (1.13) (purchè la $\phi(\cdot)$ sia limitata), in quanto ricerca il caso peggiore all'interno di una serie di possibili realizzazioni.

Il problema di controllo su orizzonte finito è dunque così formulato.

Problema P1. Problema di controllo su orizzonte finito. Trovare la politica $p = \{m_t(\cdot) : t = 0, \dots, h-1\}$ che minimizza J^1, J^2, \dots, J^m , con J^i dato da (1.15) per $i = 1, \dots, m$, sottoposto ai vincoli (1.12), (1.13), (1.14) per $t = 0, \dots, h-1$, e dato x_0 .

1.4.2 Problema su orizzonte infinito

Per quanto riguarda i sistemi ambientali la definizione dell'orizzonte temporale h e della penale $g_h(\cdot)$ risulta spesso critica, dal momento che essi hanno una vita illimitata. Per questo motivo può risultare conveniente utilizzare un orizzonte infinito (senza la necessità, quindi, di definire la penale). In tal modo la funzione obiettivo diventa

$$J^i = \lim_{h \rightarrow \infty} E_{\varepsilon_1^h} [Z^i(x_0^{h-1}, u_0^{h-1}, \varepsilon_1^h)] \quad (1.17a)$$

con

$$Z^i(x_0^{h-1}, u_0^{h-1}, \varepsilon_1^h) = \Sigma [g_t^i(x_t, u_t, \varepsilon_{t+1}) : t = 0, \dots, h-1] \quad (1.17b)$$

nel caso si utilizzi il criterio di Laplace (in ogni caso è possibile ricondurre il problema alla formulazione con criterio di Wald mediante la legge di dualità, si veda Soncini Sessa (2004)). Si noti che nell'equazione (1.17b), utilizzando come aggregatore temporale la somma, il costo totale $Z(\cdot)$ potrebbe divergere per h che tende a infinito. Per risolvere questo problema esistono due possibili approcci. Nel primo caso si definisce $Z(\cdot)$ come Total Discounted Cost (TDC)

$$Z^i(x_0^{h-1}, u_0^{h-1}, \varepsilon_1^h) = \sum_{t=0}^{h-1} \gamma^t g_t^i(x_t, u_t, \varepsilon_{t+1}) \quad (1.18)$$

con $0 < \gamma < 1$ che rappresenta il tasso di sconto. Attraverso questo parametro si può dare un peso inferiore ai costi più lontani nel tempo.

Il secondo approccio possibile è l'uso del Average Expected Value (AEV)

$$Z^i(x_0^{h-1}, u_0^{h-1}, \varepsilon_1^h) = \frac{1}{h} \sum_{t=0}^{h-1} g_t^i(x_t, u_t, \varepsilon_{t+1}) \quad (1.19)$$

L'AEV pesa in egual modo tutti i costi sostenuti dal sistema, e ne calcola il valor medio. Pertanto, essendo l'orizzonte infinito, al contrario del caso precedente si considera solamente la situazione a regime, senza dare peso al transitorio. Ovviamente, in base al problema considerato, può risultare più corretto utilizzare il primo o il secondo approccio.

Resta da definire il concetto di politica nel caso di orizzonte infinito, dal momento che non è ovviamente possibile determinare attraverso la risoluzione del problema un numero infinito di leggi di controllo. In caso però di sistema ciclostazionario (o eventualmente stazionario), cioè tale che $f_t(\cdot)$ nella (1.12), $g_t(\cdot)$ nella (1.16) e la distribuzione di probabilità dei disturbi $\phi_t(\cdot)$ (o il set-membership $\Xi_t(\cdot)$ nella (1.13) in caso di disturbo incerto) siano periodici di periodo T , è dimostrabile che anche la politica ottima è periodica dello stesso periodo, pertanto il risultato del problema è rappresentato da una politica formata da T leggi di controllo. I sistemi idrici, solitamente, sono considerabili periodici con periodo pari a un anno (se si utilizza come passo temporale il giorno, $T = 365$).

La formulazione del problema su orizzonte infinito presenta anche un altro problema. Infatti, la valutazione della (1.17) richiede in termini pratici la simulazione su un infinito numero di realizzazioni del disturbo, ognuno di infinita lunghezza. Anche in questo caso, però, se il disturbo è ergodico (come avviene praticamente sempre nelle trattazioni dei sistemi idrici), per $h \rightarrow \infty$ è dimostrabile che la (1.17) con Z definito dalla (1.19) è esattamente identica al costo medio relativo a un'unica realizzazione del disturbo. Considerando h sufficientemente grande, quindi, si può utilizzare la seguente approssimazione della funzione obiettivo (1.17):

$$\bar{J}^i = \frac{1}{h} \sum_{t=0}^{h-1} g^i(x_t, u_t, \bar{\varepsilon}_{t+1}) \quad (1.20)$$

in cui $\bar{\varepsilon}_{t+1}$ appartiene alla realizzazione finita del disturbo ε casualmente estratta da ϕ . Attraverso la (1.20), quindi, la valutazione della funzione obiettivo si riconduce a una simulazione deterministica relativa a una realizzazione sufficientemente lunga del disturbo.

La realizzazione del disturbo da utilizzare può essere presa dalle serie storiche (se si dispone di sufficienti dati), oppure può essere ottenuta da serie

sintetiche. Questa approssimazione sarà la forma della funzione obiettivo considerata nel presente lavoro.

Il problema di controllo ottimo su orizzonte infinito può essere quindi formalizzato come

Problema P2. Problema di controllo ottimo su orizzonte infinito.

Trovare la politica periodica $p = \{m_t(\cdot) : t = 0, 1, \dots, T - 1; m_{t+kT}(\cdot) = m_t(\cdot), k \in \mathbb{Z}\}$ che minimizza J^1, J^2, \dots, J^m , con J^i dato da (1.17) oppure dalla sua approssimazione (1.20) per $i = 1, \dots, m$, soggetto ai vincoli (1.12), (1.13), (1.14) per $t = 0, 1, \dots$, e dato x_0 .

Quale sia la lunghezza di h necessaria perchè la (1.20) sia valida è una questione aperta. Ovviamente, un limite superiore è fornito dalla disponibilità di dati o dalle necessità operative relative ai tempi di calcolo.

1.4.3 Problema a molti obiettivi

Come introdotto nella sezione 1.1, nei casi reali il problema di controllo solitamente è caratterizzato dalla presenza di molteplici interessi conflittuali tra loro. Così, non può esistere un'unica politica ottima che minimizzi contemporaneamente tutti gli obiettivi J^1, J^2, \dots, J^m considerati, bensì una molteplicità di politiche efficienti in senso paretiano. Il criterio di efficienza di Pareto stabilisce sostanzialmente che una soluzione è efficiente se non esiste un'altra soluzione che fornisca prestazioni migliori in tutti gli obiettivi considerati contemporaneamente, ovvero se essa non è *dominata*. Le soluzioni efficienti dal punto di vista paretiano, dunque, sono caratterizzate dal fatto che se a partire da esse si desidera migliorare un obiettivo, necessariamente occorre peggiorarne un altro. Se si mappano le soluzioni efficienti sullo spazio m -dimensionale degli obiettivi, si ottiene una superficie definita *frontiera di Pareto*. La frontiera di Pareto contiene dunque un insieme di soluzioni sostituibili dal punto di vista degli obiettivi considerati (trade-off). La scelta finale, che rappresenta la soluzione definita di miglior compromesso, sarà perciò compito del decisore, e può avvenire in seguito a una fase di negoziazione, mitigazione e compensazione tra le parti interessate secondo il già introdotto (sezione 1.1) paradigma IWRM (per approfondire si veda la procedura PIP, Soncini Sessa (2004)).

Dal punto di vista operativo i metodi principali di risoluzione di un problema a molti obiettivi agiscono in modo da ricondurlo a un problema mono-obiettivo. Di seguito se ne elencano alcuni con le loro principali caratteristiche (se si vuole approfondire, si può consultare sempre Soncini Sessa (2004)):

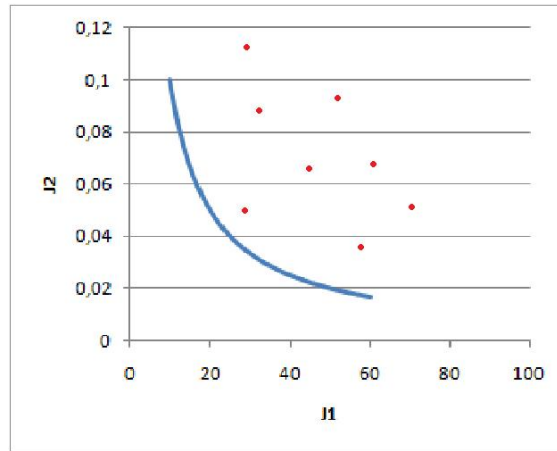


Figura 1.1: Esempio di frontiera di Pareto (in azzurro) nel caso di problema a due obiettivi. I punti rossi rappresentano soluzioni dominate.

- il *metodo lessicografico*, che consiste nella risoluzione del problema P1 o P2 (in base all'orizzonte considerato) per ogni obiettivo J^i , uno alla volta. In tal modo, è possibile ottenere gli estremi della frontiera di Pareto. In questo caso quindi la funzione obiettivo assume la forma

$$J = J^i \quad (1.21)$$

- il *metodo dei vincoli*, che si basa sulla risoluzione del problema P1 o P2 relativo a un obiettivo J^i , col vincolo che il valore degli altri obiettivi debba essere inferiore a una certa soglia:

$$J = J^i \quad J^k \leq L_k \quad k = 1, \dots, m; \quad k \neq i \quad (1.22)$$

Variando il valore della soglia (o delle soglie) L_k è possibile esplorare l'intera frontiera;

- il *metodo dei pesi*, che si basa sulla seguente funzione obiettivo:

$$J = \lambda_1 J^1 + \lambda_2 J^2 + \dots + \lambda_m J^m \quad (1.23)$$

con λ_i che corrisponde al peso dato all' i -esimo obiettivo con il vincolo $\Sigma(\lambda_1 + \lambda_2 + \dots + \lambda_m) = 1$. Variando opportunamente il set di pesi è possibile esplorare le soluzioni relative all'intera frontiera di Pareto. Purtroppo, questo metodo ha l'inconveniente di non riuscire a esplorare zone concave della frontiera;

- il *metodo del punto di riferimento*, che si basa appunto sulla determinazione di un riferimento nello spazio degli obiettivi, e sul calcolo del minimo, rispetto alla soluzione (la politica in tal caso), del massimo scarto tra gli m obiettivi dal punto di riferimento considerato

$$J = \max_i [J^i - R^i] \quad (1.24)$$

Ovviamente, se il punto di riferimento considerato dal decisore è dominato da qualche punto della frontiera (il decisore è stato modesto nelle sue aspettative) si otterrà comunque una soluzione efficiente, dal momento che gli scarti possono assumere anche valori negativi. Anche in questo caso, variando il punto di riferimento R , è possibile esplorare le soluzioni relative all'intera frontiera.

Per garantire la separabilità dell'obiettivo aggregato, è necessario adottare il metodo dei pesi solo se si utilizza il criterio di Laplace per il filtraggio dei disturbi, mentre il metodo del punto di riferimento solo se si adotta il criterio di Wald.

La formulazione del problema di controllo ottimo a un obiettivo aggregato è dunque la seguente.

Problema P3. Problema di controllo ottimo a un obiettivo. Trovare la politica p che minimizzi, in base al metodo utilizzato, la (1.21), la (1.22), la (1.23) o la (1.24), soggetta ai vincoli (1.12), (1.13), (1.14) per $t = 0, \dots, h-1$, e dato x_0 .

Ovviamente, il problema P3 può essere esteso al caso di orizzonte infinito. Purtroppo, in questo modo è possibile ottenere solo il valore dell'obiettivo aggregato e le soluzioni efficienti, ma non di valutare le prestazioni di esse secondo ognuno degli m obiettivi considerati. Per ottenere questo traguardo e mappare le soluzioni sulla frontiera di Pareto, bisogna passare a una successiva fase denominata *stima degli effetti*, fondamentale per fornire ai portatori di interesse le informazioni necessarie per la negoziazione (Soncini Sessa (2004)).

Infine, per la risoluzione del problema P3 è possibile ottenere solo una soluzione ad ogni ottimizzazione, dal momento che questo approccio si basa sull'aggregazione degli obiettivi in uno solo attraverso determinati parametri. Per ottenere una buona stima della frontiera di Pareto, quindi, è necessario risolvere il problema numerose volte utilizzando differenti valori dei parametri di aggregazione appena introdotti.

Nel prossimo paragrafo verrà mostrato uno dei principali metodi utilizzati per la risoluzione di problemi di controllo nell'ambito dei sistemi idrici, che si

basa su questa idea ed è quindi soggetto, tra l'altro, ai limiti esposti. In questo lavoro si utilizzerà invece un approccio differente (descritto successivamente), che non richiede l'agglomerazione degli obiettivi. In tal modo è possibile risolvere direttamente il problema senza dover ricondursi alla forma P3, e quindi evitare gli importanti problemi appena descritti.

1.5 La soluzione tradizionale: l'approccio funzionale

Per risolvere il problema P3 introdotto nel precedente paragrafo esistono operativamente varie soluzioni. Una delle più note è la *Programmazione Dinamica Stocastica* (SDP, Stochastic Dynamic Programming) introdotta da Bellman (1957). Essa è stata largamente applicata in passato a problemi idrici, in numerose varianti (Yeh (1985)). Essa permette di risolvere il problema nella sua formulazione originaria usando la funzione obiettivo (1.17) invece che la sua approssimazione (1.20), ovvero una statistica del costo totale invece che il costo rispetto a una realizzazione finita dei disturbi.

La programmazione dinamica si basa sulla considerazione che ogni scelta compiuta ad ogni istante nella regolazione del sistema, influisce sia in termini dei costi immediatamente generati da quella scelta, sia per lo stato che essa produrrà nell'istante successivo e che influenzerà le future scelte e i relativi costi futuri. Essa mira dunque a minimizzare il costo totale di tutti gli stadi dall'istante in cui si effettua la scelta in poi. Risulta quindi fondamentale il concetto di costo-futuro ottimo:

$$H_t(x_t) = \min_{u_t} E_{\varepsilon_{t+1}} [g_t(x_t, u_t, \varepsilon_{t+1}) + H_{t+1}(x_{t+1})] \quad (1.25)$$

nel caso in cui si usi il criterio di Laplace, o analogamente

$$H_t(x_t) = \min_{u_t} \max_{\varepsilon_{t+1}} (g_t(x_t, u_t, \varepsilon_{t+1}), H_{t+1}(x_{t+1})) \quad (1.26)$$

nel caso in cui si utilizzi il criterio di Wald. Il costo-futuro ottimo è il costo che si sosterrà dall'istante t alla fine dell'orizzonte, nell'ipotesi di usare sempre la politica ottima.

L'equazione (1.25) (o (1.26)) è un'equazione ricorsiva (detta *equazione di Bellman*) che rappresenta una relazione tra i costi sostenuti immediatamente, e quelli che verranno sopportati negli istanti successivi sempre utilizzando la miglior politica possibile.

In caso di orizzonte finito, il costo futuro ottimo $H_t(\cdot)$ per $t = 0, \dots, h-1$ può essere calcolato numericamente iniziando $H_h(x_h)$ con la penale $g_h(x_h)$ e

applicando ricorsivamente l'equazione (1.25) o (1.26) procedendo all'indietro dall'istante finale a quello iniziale. La legge di controllo ottimo all'istante $t = h - 1, \dots, 0$ è data da

$$m_t(x_t) = \arg \min_{u_t, \varepsilon_{t+1}} E [g_t(x_t, u_t, \varepsilon_{t+1}) + H_{t+1}(x_{t+1})] \quad (1.27)$$

(criterio di Laplace), o nel caso di criterio di Wald

$$m_t(x_t) = \arg \min_{u_t} \max_{\varepsilon_{t+1}} (g_t(x_t, u_t, \varepsilon_{t+1}), H_{t+1}(x_{t+1})) \quad (1.28)$$

e viene fornita nella forma di una tabella in cui ad ogni valore dello stato x_t è associato il valore del controllo ottimo u_t .

Nel caso di orizzonte temporale infinito, concettualmente l'algoritmo prevede ancora lo stesso approccio ricorsivo appena analizzato, ma viene inizializzato all'istante $t = 0$ ad un valore arbitrario, e poi si procede a ritroso nel tempo sperando che il costo-futuro ottimo converga ad una funzione costante o al limite periodica. Nel caso TDC la convergenza è soddisfatta sotto ipotesi molto ampie (Bertsekas (1976)), praticamente sempre soddisfatte nei sistemi idrici. Il relativo algoritmo si basa sull'equazione di Bellman in forma

$$H_t(x_t) = \min_{u_t, \varepsilon_{t+1}} E [g_t(x_t, u_t, \varepsilon_{t+1}) + \gamma H_{t+1}(x_{t+1})] \quad (1.29)$$

Se il sistema è periodico di periodo T , l'algoritmo prevede quindi l'inizializzazione arbitraria di T funzioni di Bellman e attraverso la (1.29), soggetta ai vincoli (1.12), (1.13) (caso stocastico) e (1.14), si procede a ritroso nel tempo fino a che le T funzioni non rispettino la condizione di convergenza

$$|H_{t+T}(x_t) - H_t(x_t)| < \alpha \quad (1.30)$$

con α che controlla la precisione della stima. Questo algoritmo si basa su tre teoremi che sostanzialmente provano che il risultato generato è una funzione di periodo T che corrisponde al costo ottimo atteso soluzione del sistema di equazioni di Bellman (condizione di ottimalità) (per approfondire si veda Bertsekas (1976)). La corrispondente politica sarà anch'essa periodica di periodo T e sarà dunque una buona approssimazione (in base al valore α scelto) della politica ottima.

Se invece si usa la formulazione AEV (1.19) non è detto che $H_t(\cdot)$ converga, per il semplice fatto che in assenza di sconto il costo-futuro su orizzonte infinito è in generale infinito. Per garantire la convergenza si può tuttavia utilizzare come indicatore, invece del costo-futuro ottimo, la differenza

$$h_{t+1} = H_{t+1}(x_{t+1}) - H_{t+1}(\bar{x}_{t+1}) \quad (1.31)$$

rispetto a un costo futuro associato a un arbitrario stato di riferimento \bar{x}_t . Su questa idea si basa l'Algoritmo delle Successive Approssimazioni ASA proposto per la prima volta da White (1963) per il caso stazionario e poi generalizzato da Su and Deininger (1972) per il caso periodico. Anche in questo caso la soluzione è una funzione periodica di periodo T che approssima il costo-ottimo atteso soluzione del sistema di equazioni di Bellman (Bertsekas (1976)), garantendo quindi l'ottimalità. La politica corrispondente inoltre è ancora quella ottima, dal momento che essa risulta invariante rispetto all'aggiunta di una costante additiva alla funzione di Bellman, come avvenuto nell'equazione (1.31).

Per i due algoritmi appena introdotti il criterio di Wald non è utilizzabile, dal momento che la sua filosofia è in contrasto con l'adozione del tasso di sconto e con il calcolo di un costo medio per passo su orizzonte infinito.

Per applicare gli algoritmi appena esaminati è necessario che gli insiemi di definizione S_{x_t} , S_{u_t} , e S_{ε_t} dello stato, del controllo e del disturbo siano finiti in ogni istante t (il sistema deve essere un automa). Se questa condizione non è verificata, essi devono essere discretizzati e il modello del sistema idrico deve essere sostituito dal corrispondente automa. Solitamente quando non si hanno informazioni a priori sulla forma del costo futuro ottimo $H_t(\cdot)$ si utilizza la discretizzazione uniforme. Ovviamente la numerosità degli insiemi di definizione (quindi della discretizzazione) influenza la precisione della soluzione, ma anche il corrispondente onere computazionale.

E' fondamentale introdurre inoltre che, perchè si possa applicare la programmazione dinamica, oltre all'ipotesi che il sistema sia un'automata devono essere rispettate le seguenti condizioni, che ne limitano la potenzialità di impiego:

1. il sistema non può avere in ingresso variabili esogene deterministiche (quali, ad esempio misure di precipitazione); gli ingressi possono essere controlli o disturbi casuali;
2. la distribuzione di probabilità ϕ (o l'insieme di validità Ξ) dei disturbi deve essere nella forma (1.13), non deve dipendere cioè esplicitamente dai valori che il disturbo ε_{t+1} assume negli istanti di tempo precedenti. In caso di disturbo bianco questa condizione è sempre verificata. Un disturbo è un rumore bianco se rappresenta una serie di dati non algoritmicamente compressibile (Barrow (1991)). Ciò equivale a dire che esso può essere descritto solo attraverso la sua distribuzione di probabilità, o se è incerto, attraverso l'insieme di valori Ξ_t che può assumere all'istante t . Se il disturbo non è un rumore bianco, bisogna a sua volta descriverlo con un modello, fino a che non si risalga a un disturbo che

sia effettivamente bianco (per approfondimenti, si veda Soncini Sessa (2004));

3. la funzione obiettivo e il sistema vincolare devono avere forma separabile (da qui le considerazioni in merito fatte nei precedenti paragrafi);
4. il problema deve essere mono-obiettivo, pertanto la programmazione dinamica può risolvere solo problemi del tipo P3. Questo, come si è visto al termine del paragrafo 1.4.3, comporta una serie di inconvenienti.

La programmazione dinamica è uno strumento molto potente, in grado di fornire delle prestazioni molto elevate, tuttavia presenta un limite notevole (oltre a quelli appena descritti): la sua complessità computazionale, che in certi casi rende impensabile un suo utilizzo con la potenza di calcolo oggi comunemente disponibile. Per comprendere la questione si chiamino N_{x_t} , N_{u_t} e N_{ε_t} la numerosità degli insiemi di stato, controllo e disturbo discretizzati in $S_{x_t} \subseteq \mathbb{R}^{n_x}$, $S_{u_t} \subseteq \mathbb{R}^{n_u}$ e $S_{\varepsilon_t} \subseteq \mathbb{R}^{n_\varepsilon}$. La soluzione ricorsiva della (1.25) (o (1.26)) con gli algoritmi introdotti richiede un tempo di calcolo

$$t_c \approx K \cdot (N_{x_t}^{n_x} \cdot N_{u_t}^{n_u} \cdot N_{\varepsilon_t}^{n_\varepsilon}) \quad (1.32)$$

in cui K è una costante di proporzionalità. Dunque la (1.32) mostra come l'onere computazionale cresca esponenzialmente con le dimensioni dello stato, del controllo e del disturbo. Ne consegue che la SDP non è applicabile a sistemi idrici complessi, per esempio con un numero di serbatoi superiore a poche unità. Tale limite è stato descritto da Bellman come “maledizione della dimensionalità”. Inoltre, come introdotto al termine del paragrafo 1.4.3, ogni risoluzione del problema fornisce una singola soluzione, pertanto in caso di problema a molti obiettivi per ottenere diverse soluzioni relative a vari punti della frontiera di Pareto è necessario risolvere il problema P3 più volte con gli algoritmi appena introdotti, generando quindi un ulteriore incremento degli oneri computazionali.

Per superare la maledizione della dimensionalità è necessario abbandonare l'approccio funzionale, che mira a determinare la politica ottima nello spazio di tutte le politiche ammissibili. Nel successivo paragrafo verrà introdotto quindi un approccio alternativo che si basa sulla ricerca della miglior politica in una classe di funzioni data.

1.6 La soluzione proposta: l'approccio parametrico

Per superare i limiti legati agli oneri computazionali della programmazione dinamica, una soluzione alternativa può essere la politica di classe fissata. L'idea è la seguente: se per ogni istante t si conoscesse la classe $\hat{m}_t(\cdot; \theta_t)$ di funzioni cui appartengono le leggi di controllo della politica ottima, con θ_t vettore di parametri incognito, il problema di controllo ottimo si ridurrebbe alla sola questione di trovare la sequenza $\{\theta_t : t = 0, \dots, T-1\}$ di parametri che minimizzano il costo totale per l' i -esimo obiettivo:

$$\bar{J}^i = \frac{1}{h} \sum_{t=0}^{h-1} g^i(x_t, u_t(x_t, \theta_t), \bar{\varepsilon}_{t+1}) \quad (1.33)$$

secondo la forma della funzione obiettivo approssimata introdotta nel caso di orizzonte infinito (con h sufficiente). Infatti, se il sistema è periodico di periodo T anche la sequenza dei parametri ottimi è periodica, cioè $\theta_{t+kT} = \theta_t, k \in \mathbb{Z}$.

E' quindi possibile definire il seguente problema, che sarà proprio la formulazione considerata nel presente lavoro.

Problema P4. Problema di controllo ottimo parametrico. Trovare all'interno di una determinata classe di funzioni la sequenza di parametri ottimi $\theta = \{\theta_t; t = 0, 1, \dots, T-1\}$ che minimizzano J^1, J^2, \dots, J^m , con J_i data dalla (1.33), soggetta ai vincoli (1.12), (1.13) e $u_t = \hat{m}_t(\cdot; \theta_t)$ per $t = 0, 1, \dots, T-1$, e dato x_0 .

Il problema P4 è concettualmente analogo a quelli introdotti nel paragrafo 1.4, ma la sua risoluzione si basa ora su tecniche di programmazione matematica, e risulta quindi molto più semplice da affrontare in termini computazionali rispetto a un problema funzionale. L'approccio può anche essere esteso ovviamente al caso di orizzonte finito. In questo caso è sufficiente trovare, risolvendo il problema P4 (con le funzioni obiettivo della forma (1.15)), il vettore $\{\theta_t : t = 0, 1, \dots, h-1\}$ che rappresenta la politica ottima.

Questo approccio comporta inoltre una serie di vantaggi, non conseguibili attraverso la programmazione dinamica:

1. sono ammesse variabili esogene in ingresso al modello (come ad esempio misure di precipitazione), e anzi esse possono addirittura divenire argomento stesso della politica;

2. non è richiesta necessariamente la bianchezza dei disturbi presenti nel modello;
3. non è strettamente necessaria la separabilità delle diverse componenti del modello, in relazione all'algoritmo operativamente utilizzato per risolvere il problema P4;
4. non è necessaria l'aggregazione degli obiettivi, ma è possibile operare il processo di ottimizzazione considerando separatamente tutti gli obiettivi considerati, permettendo di superare la serie di limitazioni descritta nel paragrafo 1.4.3. Anche in questo caso, ciò è dipendente dallo specifico algoritmo di programmazione matematica utilizzato in fase operativa.

Come si può notare, questo tipo di approccio permette di conseguire notevoli benefici. In particolare, nel presente lavoro la famiglia di algoritmi utilizzata, descritta nel successivo capitolo, è in grado di sfruttare tutti i vantaggi appena elencati.

Purtroppo il problema principale di questo approccio si trova a monte, ovvero nella stessa ipotesi (fatta inizialmente) di conoscere la classe di funzioni a cui appartiene la politica ottima. In generale, infatti, a priori non si è a conoscenza della forma che assumeranno le leggi di controllo che identificano la politica (dal momento che essa è ovviamente ignota), pertanto è necessario sceglierla solamente basandosi sulla propria intuizione. Se la classe di funzioni selezionata non contenesse la politica ottima, infatti, la risoluzione del problema P4 produrrebbe sicuramente la miglior politica interna a quella classe di funzioni, ma nulla vieta che essa sia pessima rispetto alla politica ottima.

Intuire la forma della politica ottima può non essere complicato nel caso di sistemi particolarmente semplici, caratterizzati da una singola relazione ingresso/uscita (invaso/rilascio, caso singolo serbatoio); tuttavia in molti casi reali più complessi, ad esempio relativi a una rete di molti serbatoi interconnessi, può risultare estremamente difficoltoso.

In questo lavoro verrà seguito questo tipo di approccio, utilizzando come classe di funzioni le reti neurali che, come si vedrà, hanno notevoli vantaggi, e un'ottima capacità di approssimare qualsiasi tipologia di funzione. In tal modo, si intende sfruttare i vantaggi forniti dall'approccio parametrico, sperando di ottenere comunque ottime soluzioni.

Capitolo 2

Metodi e strumenti utilizzati

Nel presente capitolo si introdurranno gli strumenti e la metodologia utilizzati in questo lavoro, e metterne in luce gli aspetti principali.

L'approccio seguito consiste nell'uso combinato delle reti neurali (ANN, Artificial Neural Network) e di una particolare famiglia di algoritmi, gli algoritmi evolutivi, per ottenere una corretta gestione di un sistema idrico. In particolare, si intende utilizzare le reti neurali come struttura delle leggi di controllo secondo il già citato approccio parametrico (1.6), e sfruttare le caratteristiche degli algoritmi evolutivi durante il processo di ottimizzazione, al fine di ricavare un insieme di politiche che permettono di gestire nel modo migliore possibile un sistema idrico, in un'ottica a molti obiettivi. Verranno inoltre introdotte all'interno di questo approccio alcune caratteristiche innovative.

2.1 Reti neurali

Il cervello umano è costituito da una rete biologica complessa composta da centinaia di milioni di speciali celle denominate neuroni. Tali neuroni si scambiano informazioni attraverso delle interconnessioni, e il risultato che si ottiene è un essere intelligente capace di apprendere, analizzare e riconoscere il mondo che lo circonda. Su questo principio si basano le reti neurali artificiali, sistemi introdotti a partire dagli anni '40 ma sviluppatasi a livelli interessanti per scopo ingegneristico solo negli ultimi due decenni del secolo scorso.

Le reti neurali rappresentano una struttura complessa di funzioni, sostanzialmente formata da uno strato d'ingresso, uno strato nascosto di neuroni che elaborano il segnale in ingresso, e uno strato di uscita. Questi strati sono formati ognuno da un certo numero di componenti in relazione al caso

esaminato e alla struttura (denominata *architettura*) scelta, e sono collegati attraverso un sistema complesso di interconnessioni che modificano il segnale mediante opportuni parametri, seguendo in modo semplificato lo schema del funzionamento del cervello umano. La struttura di una rete neurale è quindi

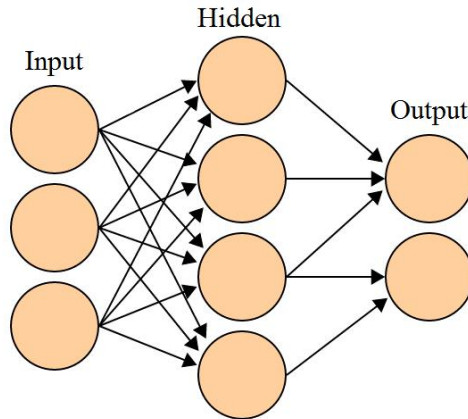


Figura 2.1: Schema di una generica rete neurale. Ogni freccia opera una modulazione del segnale attraverso dei pesi propri.

formalmente descrivibile attraverso l'equazione

$$u_t^j = a_j + \sum_{i=1}^{\nu} b_{ij} \psi_i(I_t \cdot \mathbf{c}_{ij} + d_{ij}) \quad (2.1)$$

dove $\psi_i(\cdot)$, $i = 1, \dots, \nu$ sono delle funzioni non lineari (neuroni), I_t rappresenta l'ingresso, u_t^j il j -esimo componente del controllo e $a_j, b_{ij}, d_{ij} \in \mathbb{R}$, $\mathbf{c}_{ij} \in \mathbb{R}^r$ sono i parametri della rete.

Le reti neurali hanno delle caratteristiche importanti che ne hanno determinato la scelta come classe di funzioni in questo lavoro. Il loro pregio principale risiede nella notevole flessibilità: anche delle architetture con un limitato numero di neuroni sono in grado di approssimare con buona precisione una vasta gamma di funzioni. Barron (1993) ha dimostrato che una rete neurale è in grado di approssimare con arbitraria precisione qualsiasi tipologia di funzione attraverso un numero di neuroni crescente con l'ingresso della rete in modo meno che esponenziale.

Il loro utilizzo come approssimatori delle leggi di controllo, d'altra parte, permette di risolvere il problema di controllo senza dover ricercare le politiche all'interno dello spazio di tutte le funzioni, ma stimando solo un numero limitato di parametri, precisamente il vettore: $\theta = (a_j, b_{ij}, \mathbf{c}_{ij}, d_{ij} :$

$i = 1, \dots, \nu; j = 1, \dots, n_u$) che appartiene a \mathbb{R}^{n_θ} , secondo l'approccio descritto nella sezione 1.6. Per una data architettura, il numero di parametri da stimare sarà quindi:

$$n_\theta = n_u(\nu(r + 2) + 1) \quad (2.2)$$

in cui r è il numero di ingressi e ν il numero di neuroni.

Il loro potenziale nell'ambito dei problemi di controllo nei sistemi idrici è testimoniato dal loro uso in precedenti lavori, come in Zoppoli et al. (2002), e Pianosi and Soncini-Sessa (2008).

Le reti neurali sono inoltre in grado, come le controparti biologiche, di sfruttare le informazioni ricevute per apprendere. Attraverso la fase definita addestramento è possibile fornire alla rete neurale delle traiettorie ingresso/uscita per ottenere il set di parametri che meglio le approssimano. Questa fase riveste una particolare importanza in questo lavoro, e verrà pertanto meglio discussa nella sezione 2.4.

2.2 Algoritmi evolutivi

Gli algoritmi evolutivi (anche chiamati algoritmi genetici) sono una famiglia di procedure euristiche di ricerca e ottimizzazione che basano il loro principio di funzionamento sul processo di selezione naturale proposto nella teoria evoluzionistica di Darwin (Goldberg (1989)).

In questi algoritmi le soluzioni (nel caso in esame la successione di parametri della rete neurale che approssima le leggi di controllo) vengono codificate in una stringa chiamata *cromosoma*. La codifica può avvenire in diversi modi, ad esempio in codice binario oppure utilizzando direttamente i valori reali dei parametri (come in questo caso). Le singole unità di cui è composto un cromosoma, ovvero in questa sede i singoli parametri della rete neurale, vengono chiamate in quest'ambito *geni*.

Partendo da un insieme di soluzioni inizializzate casualmente, questi metodi compiono il processo di ottimizzazione attraverso i cosiddetti operatori genetici: *crossover*, *mutazione* e *selezione*, in base al valore assunto dalla funzione obiettivo, comunemente chiamata *fitness*. Gli individui della popolazione a cui corrisponde un miglior valore di fitness avranno più probabilità di sopravvivere e riprodursi per dare vita a nuovi individui che faranno parte della popolazione della successiva iterazione, chiamata in questo ambito *generazione*. La nuova generazione include pertanto soluzioni ottenute attraverso gli operatori di crossover, che opera una ricombinazione dei geni di due cromosomi selezionati per formare due cromosomi figli; e la mutazione, che invece modifica (in modo casuale, con probabilità generalmente bassa) il singolo gene di un cromosoma. Attraverso questo processo si pilotano le

soluzioni verso regioni del loro dominio che corrispondono a valori sempre migliori di fitness come avviene in natura attraverso la selezione naturale, e si mira quindi a raggiungere l'ottimo in un numero finito di generazioni. In particolare, il crossover fa in modo che i migliori individui possano riprodursi e dare vita a una popolazione con fitness più elevato, mentre la mutazione dà comunque la possibilità di esplorare qualsiasi regione dello spazio dei parametri con una certa probabilità.

Gli algoritmi evolutivi sono stati concepiti da Holland (1975) e negli anni sono stati sviluppati in numerosissime direzioni anche molto differenti, segno dell'interesse che hanno destato in campo pratico e del fatto che non esiste (almeno fino a oggi) un algoritmo evolutivo che funzioni in modo evidentemente migliore degli altri: la scelta è pertanto fortemente dipendente dal caso oggetto di studio.

L'interesse che desta questa famiglia di algoritmi è giustificata da alcuni importanti pregi che essi hanno. Innanzitutto, il principio euristico su cui si basano fa sì che essi siano insensibili alla forma e alle proprietà matematiche delle funzioni obiettivo (Goldberg (1989)). Così questi algoritmi risultano strumenti molto flessibili e utilizzabili in una vasta gamma di problemi di ottimizzazione, dal momento che non vengono influenzati dalla formulazione del problema e non richiedono proprietà come la continuità o la separabilità (necessaria invece in molti altri strumenti utilizzati spesso nella gestione dei sistemi idrici, si veda la sezione 1.4). Allo stesso modo, risultano ovviamente indipendenti da minimi locali, visto che il loro carattere fa sì che possano esplorare qualunque regione del dominio delle soluzioni con una probabilità maggiore di zero. Infine, ma non ultimo in ordine di importanza, il fatto che come risultato producano non una singola soluzione ma una popolazione di soluzioni rende gli algoritmi evolutivi particolarmente adatti a problemi a molti obiettivi, dal momento che non si desidera ottenere un'unica soluzione ma una frontiera di Pareto (paragrafo 1.4.3). Infatti, al contrario di metodi come la programmazione dinamica, che ottengono come risultato una singola soluzione, ad ogni generazione essi evolvono una popolazione di soluzioni, che si presta molto meglio a riflettere i diversi compromessi di scelta tra i vari interessi in gioco.

Nel caso in esame verrà utilizzato l'algoritmo *Non-Dominated Sorting Genetic Algorithm-II*, NSGA II (Deb et al. (2002)). Esso nell'ambito dei problemi di controllo a molti obiettivi nei sistemi idrici è già stato applicato in passato, come per esempio in Kim et al. (2008).

NSGA II fa parte di un particolare settore degli algoritmi genetici, i MOGA (Multi-Objective Genetic Algorithm), che sono appunto particolarmente adatti per problemi caratterizzati da una molteplicità di interessi. Il suo fun-

zionamento, infatti, consiste (a ogni generazione) nell'ordinare i cromosomi di una popolazione tenendo conto di tutti gli obiettivi considerati separatamente, assegnando un valore di ranking pari a 1 per le soluzioni non dominate e poi valori via via crescenti per le altre; e quindi applicare gli operatori genetici ai cromosomi.

In NSGA II è stato inserito anche un altro indicatore per valutare la bontà delle soluzioni, la "crowding distance", ovvero un indice di dispersione che calcola statisticamente la distanza di una soluzione rispetto alle altre soluzioni della popolazione. Tra cromosomi con pari ranking, infatti, sono preferibili quelli più distanti dagli altri, al fine di assicurare la diversità nella popolazione (per esplorare porzioni maggiori nello spazio delle soluzioni ammissibili) e per garantire una migliore approssimazione della frontiera. Un'altra caratteristica fondamentale introdotta in NSGA II è l'elitismo, ovvero una particolare proprietà che fa sì che i migliori individui di una popolazione si conservino intatti anche nella generazione successiva. Mantenendo le soluzioni migliori, si intende velocizzare il processo di ottimizzazione e abbassare il numero di generazioni necessarie a convergere alla frontiera di Pareto (o a una buona approssimazione di essa).

In particolare, l'algoritmo implementato nel presente lavoro è costituito da due blocchi comunicanti: un blocco rappresenta NSGA II e l'altro opera la simulazione del sistema. A ogni generazione, il valore degli obiettivi corrispondenti a ciascun individuo della popolazione viene determinato passando i relativi parametri al blocco simulazione, che opera secondo le leggi di controllo corrispettive ai parametri dell'individuo esaminato. In termini computazionali, il contributo principale dell'algoritmo è dato proprio dal funzionamento di questo blocco, mentre NSGA II genera dei costi decisamente inferiori.

Il funzionamento dell'algoritmo, in definitiva, può essere schematizzato nel seguente modo:

Algoritmo NSGA II

1. alla generazione 0, vengono generate casualmente (nello spazio dei parametri) *pop* soluzioni;
2. le soluzioni vengono valutate secondo le funzioni obiettivo considerate;
3. gli individui vengono ordinati secondo il loro ranking e la loro crowding distance;

4. vengono attivati gli operatori genetici di selezione, crossover e mutazione (gli ultimi due con una certa probabilità prefissata) formando una nuova popolazione, e vengono quindi calcolati i corrispondenti valori degli obiettivi;
5. per garantire l'elitismo, la popolazione vecchia e quella nuova vengono combinate insieme formando una nuova popolazione di dimensioni $2pop$, le cui soluzioni sono valutate nuovamente secondo i criteri di ranking e crowding distance;
6. l'algoritmo quindi seleziona le pop soluzioni migliori che andranno a formare la popolazione della generazione successiva, quindi si incrementa il numero di generazioni di 1;
7. se il numero di generazioni è pari al numero massimo prefissato il procedimento si arresta e gli individui dell'ultima popolazione saranno le soluzioni finali del processo (di cui quelle con miglior ranking costituiranno la frontiera ottenuta), altrimenti si torna al passo 3.

In questo lavoro, all'interno del funzionamento dell'algoritmo sono state inserite alcune caratteristiche innovative che si spera possano incrementarne le prestazioni. Nelle prossime sezioni verranno discusse in dettaglio.

2.3 Gestire la ciclostazionarietà

Nella sezione 2.1 è stato introdotto il funzionamento di una rete neurale. Ora, resta da valutare a cosa corrispondono gli ingressi e le uscite dell'equazione (2.1). Le uscite rappresentano ovviamente il vettore dei controlli u_t . Tra gli ingressi I_t , dal momento che si vuole approssimare una legge di controllo, figura certamente lo stato x_t (comprendente anche passate osservazioni dei disturbi qualora essi fossero autocorrelati). Dal momento che le decisioni ottime dipendono dal tempo, e in particolare mostrano un andamento periodico come descritto nel paragrafo 1.4.2, in questo lavoro si è deciso di inserire nel vettore degli ingressi I_t anche le quantità $\sin(2\pi t/T)$ e $\cos(2\pi t/T)$, in cui T rappresenta il periodo del sistema (e quindi della politica ottima).

E' importante evidenziare dunque che l'approccio seguito intende stimare, invece di T vettori dei parametri $\{\theta_t; t = 0, 1, \dots, T - 1\}$, un solo vettore dei parametri θ , e inserire il fattore tempo come ingresso alla rete neurale. La forma della legge di controllo si sposta pertanto da $m_t(I_t, \theta_t)$ a $m_t(I_t, t, \theta)$.

2.4 Inizializzazione dell'algoritmo

Un'altra idea innovativa che verrà considerata nel presente lavoro modifica il processo di inizializzazione dell'algoritmo.

Come descritto nella sezione 2.2, il processo di ottimizzazione inizia generando in modo completamente casuale una serie di soluzioni. Se tra queste ce ne fossero alcune ragionevolmente “buone”, ovvero se fossero note a priori delle parametrizzazioni sufficientemente vicine all'ottimo, il processo di ottimizzazione potrebbe essere presumibilmente agevolato, riducendo il tempo necessario per convergere alla frontiera. Da qui nasce l'idea di inserire nell'inizializzazione una serie di soluzioni relative a determinate scelte di comportamento nella regolazione, chiamate “regole semplici”. Queste sono quindi leggi di controllo note corrispondenti a determinate soluzioni “ragionevoli”, come ad esempio:

- rilasciare sempre la minima/massima portata possibile;
- rilasciare esattamente la domanda delle utenze quando possibile;
-

Inserendo nell'inizializzazione queste soluzioni ragionevoli è possibile pertanto pilotare la popolazione attraverso regioni in cui le soluzioni saranno probabilmente abbastanza vicine all'ottimo, pur preservando la possibilità di esplorare le altre regioni dello spazio dei parametri.

Le soluzioni relative alle regole semplici possono essere ottenute attraverso una fase preliminare di addestramento della rete neurale che permette di ottenere le parametrizzazioni che meglio approssimano una determinata serie di coppie ingresso/uscita, secondo l'equazione (2.1). Tali serie sono ottenute simulando il sistema su un certo orizzonte temporale, sottoposto alla legge di controllo della regola semplice

- $u_t = v_t(s_t, a_{t+1})$ (minimo rilascio) o $u_t = V_t(s_t, a_{t+1})$ (massimo rilascio);
- $u_t = \min(s_t, w_t)$;
-

in cui v_t e V_t sono le funzioni di minimo e massimo rilascio, e w_t è la domanda idrica (si veda sezione 1.3). Eseguita la simulazione, l'addestramento usa come serie di ingresso alla rete la serie del vettore I_t (sezione 2.1) che contiene gli invasi dei serbatoi (e le eventuali altre variabili di stato del sistema) oltre che il tempo; e come uscita la corrispondente traiettoria dei controlli u_t .

La fase di addestramento è sostanzialmente un processo di ricerca nello spazio dei parametri del set che minimizza la distanza delle funzioni ottenute dalla determinata serie di ingresso/uscita che viene fornita in ingresso. Questa distanza viene calcolata in termini di scarto quadratico medio

$$Q = \sum (\hat{y}_t - \bar{y}_t)^2 \quad (2.3)$$

in cui \hat{y}_t rappresenta le uscite stimate e \bar{y}_t quelle fornite alla rete neurale per l'addestramento.

Alternativamente, si può ricorrere alla varianza spiegata

$$R2 = 1 - \frac{COV(\hat{y}_t - \bar{y}_t)}{COV(\bar{y}_t)} \quad (2.4)$$

Si noti però che questo indicatore è da massimizzare.

Per ridurre i tempi di addestramento si considera un numero di epoche limitato, ovvero un numero di passi per cui verrà ricercata la parametrizzazione ottima: questo, purtroppo, comporta alcuni problemi.

La ricerca del minimo produrrà infatti senza dubbio come risultato la para-

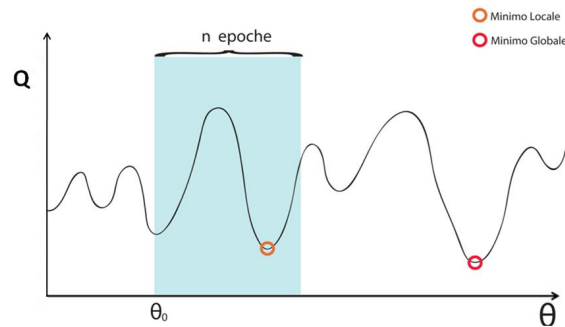


Figura 2.2: Esempio della ricerca del minimo durante la fase di addestramento, con una sola iterazione e 150 epoche. Il risultato potrebbe corrispondere a un minimo locale.

metrizzazione ottima all'interno delle epoche considerate, ma nulla assicura che ricercando oltre non si sarebbe potuta trovare una parametrizzazione migliore: il risultato potrebbe corrispondere quindi a un minimo locale (Figura 2.2). E' pertanto evidente che la bontà del risultato ottenuto nell'addestramento può dipendere fortemente dal punto θ_0 in cui si inizia la ricerca della parametrizzazione ottima; tale punto è completamente casuale. Per ovviare a questo inconveniente si può aumentare il numero N_{iter} di iterazioni in cui si compie la fase di addestramento: in questo modo la parametrizzazione

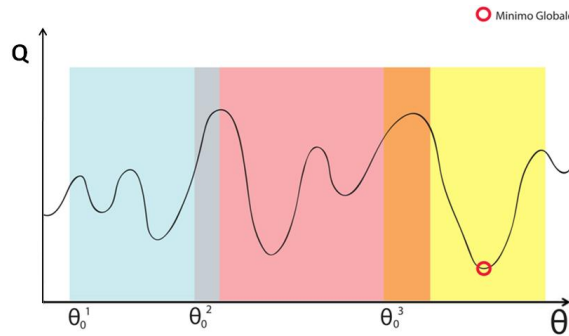


Figura 2.3: Esempio della ricerca del minimo durante la fase di addestramento, con tre iterazioni e 150 epoche. Il risultato finale è il miglior risultato trovato tra tutte le iterazioni.

ricavata sarà la migliore tra quelle ottenute in N_{iter} tentativi, ognuno corrispondente a un punto iniziale di ricerca θ_0 differente (ma sempre con lo stesso numero di epoche fissato). Un numero sufficientemente elevato di iterazioni assicura con buona probabilità che la parametrizzazione ottenuta corrisponda al minimo globale (Figura 2.3). Ovviamente l'aumento di questo parametro genera un incremento dei costi computazionali della fase di addestramento. Il numero di iterazioni da considerare è pertanto un parametro importante da definire con cura in fase operativa.

La fase di addestramento, inoltre, possiede altre due importanti funzioni. La prima è quella di valutare con che precisione una determinata architettura neurale sia in grado di effettuare un'approssimazione delle leggi di controllo. Infatti, se un'architettura non sarà in grado di funzionare bene nella fase di addestramento, quasi sicuramente sarà pessima anche durante il funzionamento dell'algoritmo evolutivo vero e proprio, e verrà pertanto scartata. Attraverso la valutazione delle prestazioni nella fase di addestramento, è perciò possibile definire la scelta dell'architettura che riflette il miglior compromesso tra precisione e complessità.

Il secondo scopo è quello di fornire come ingresso all'algoritmo evolutivo un range limitato nello spazio dei parametri entro cui ricercare le soluzioni, dal momento che non sarebbe certamente efficiente l'esplorazione di uno spazio infinito anche con una popolazione molto grande e un elevato numero di generazioni. Si auspica infatti che le parametrizzazioni ricavate nella fase di addestramento possano già fornire un'idea dell'area in cui i set di parametri ottimi possano ragionevolmente trovarsi, nel caso in cui le regole semplici introdotte riflettano a dovere la varietà delle possibili scelte di comportamento

del regolatore, in relazione agli obiettivi considerati nel problema in esame. Per ogni parametro θ_i della rete neurale viene calcolato il minimo θ_i^{min} , il massimo θ_i^{max} e la deviazione standard S_i tra tutti i valori assunti nelle n regole semplici. Il range di variazione dell' i -esimo parametro è quindi compreso tra $\theta_i^{min} - 5S_i$ e $\theta_i^{max} + 5S_i$. Esso, ovviamente, è diverso per ciascuno dei parametri che caratterizzano la rete neurale.

In definitiva, la procedura dell'algoritmo rimane quella mostrata per l'algoritmo NSGA II, con la sola differenza: al passo 1, per n regole semplici definite, la popolazione iniziale sarà formata da n soluzioni ricavate nella fase di addestramento e $pop - n$ soluzioni completamente casuali.

2.5 Il seme

La metodologia seguita nel presente lavoro, ampiamente descritta nella sezione 2.2, si basa sostanzialmente su un blocco che esegue la simulazione e un blocco che applica gli operatori genetici di selezione, crossover e mutazione. Mentre la simulazione è deterministica, la fase in cui sono attivi gli operatori genetici (oltre che l'inizializzazione) è governata da un processo casuale. Per questo motivo, due run consecutivi dello stesso esperimento daranno molto probabilmente risultati differenti.

La casualità evidentemente compromette la possibilità di valutare e paragonare efficacemente le prestazioni di diversi esperimenti (ad esempio, diverse architetture neurali o diversi parametri dell'algoritmo evolutivo): se i risultati di un esperimento sono influenzati dal caso, una loro valutazione in senso assoluto perde di significato. In realtà la generazione casuale di numeri compiuta dal calcolatore attraverso i linguaggi di programmazione non è veramente aleatoria, al contrario è ottenuta attraverso un algoritmo pseudo-casuale, il cui risultato dipende deterministicamente dall'inizializzazione (*seme*). Fissando il seme, due generazioni di numeri casuali, e quindi due run di uno stesso esperimento di ottimizzazione produrranno gli stessi risultati. Se la metodologia di randomizzazione è univoca, allora ha finalmente senso paragonare le prestazioni di diversi esperimenti.

Tuttavia, in seguito a una più attenta riflessione si potrebbe obiettare che i risultati potrebbero dipendere dal particolare seme utilizzato. Infatti, fissando il seme è possibile paragonare i risultati ottenuti in diversi esperimenti e affermare che le prestazioni di uno di essi sono migliori o meno di quelle degli altri, ma tale considerazione è vera solo in relazione al seme utilizzato, mentre potrebbe non esserlo più per altri semi. Si può quindi in definitiva

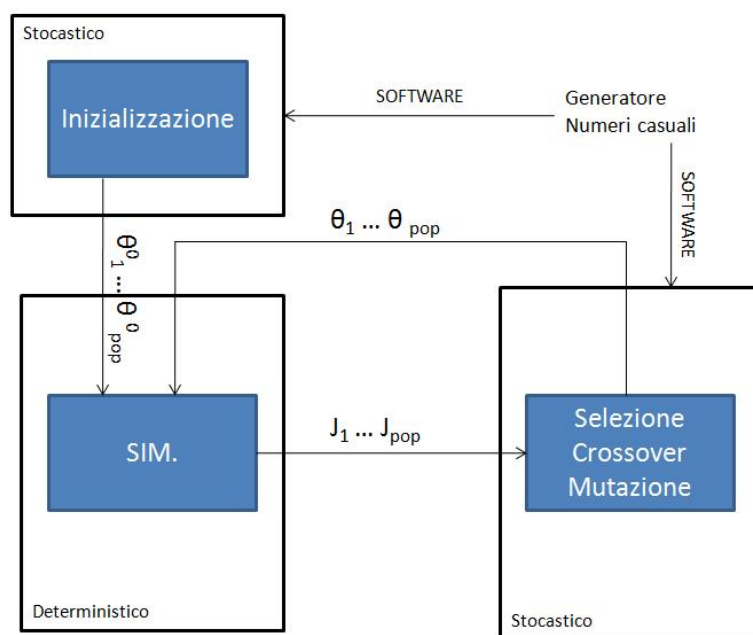


Figura 2.4: Stocasticità nell'algorithm.

asserire che i risultati ottenuti da un esperimento dipendono in parte dalle sue caratteristiche, e in parte dal seme utilizzato (Figura 2.5).

Per valutare in senso assoluto le prestazioni dei diversi esperimenti sarà dunque necessario ripetere lo stesso esperimento con semi differenti e ricavare una media dei risultati.

2.6 Indici di prestazione

Negli ultimi anni, oltre a NSGA II, sono stati proposti numerosi altri algoritmi di ottimizzazione a molti obiettivi. Per questa ragione, è divenuta sempre più importante l'introduzione di efficaci indicatori per la misura delle loro prestazioni. Se, infatti, in un problema mono-obiettivo (come il problema P3, paragrafo 1.4.3) è semplice determinare le prestazioni di un algoritmo, in base al valore dell'obiettivo ottenuto, non è così per la valutazione di un'intera frontiera di Pareto caratterizzata da una molteplicità di soluzioni efficienti. Per questa ragione, recentemente sono stati proposti molti indicatori per il giudizio dei risultati degli algoritmi a molti obiettivi. Deb (2001) ha mostrato che sostanzialmente gli indicatori possono essere divisi in tre gruppi:

- indicatori per la misura del grado di convergenza;

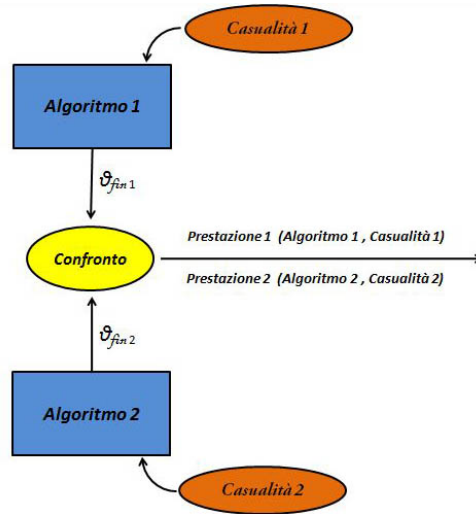


Figura 2.5: Effetto del seme sulle prestazioni di diversi esperimenti.

- indicatori per la misura della diversità;
- indicatori per la misura di convergenza e diversità.

Nella prima categoria, vale la pena citare l'indicatore C , proposto da Zitzler and Thiele (1998) e sviluppato in Zitzler (1999). Se $A, B \subseteq X$ sono due set di soluzioni, la funzione $C(A, B)$ mappa le coppie ordinate (A, B) e calcola la frazione di coppie in cui A domina B :

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \succeq b\}|}{|B|} \quad (2.5)$$

in cui l'operatore $a \succeq b$ denota la condizione di dominanza di a su b e $|B|$ rappresenta il numero totale di soluzioni in B .

Si noti che non necessariamente $C(A, B) = 1 - C(B, A)$. Nel caso in cui risulti $C(A, B) > C(B, A)$ si può ragionevolmente dire che A è migliore di B , fino al caso $C(A, B) = 1$ in cui A domina completamente B . Il problema principale risiede nel fatto che questo indicatore (come molti altri indicatori comunemente utilizzati, non menzionati per ragioni di spazio) può solo operare un confronto tra due set di soluzioni, mentre non è in grado di fornire un punteggio al risultato di un singolo algoritmo.

Un'altra soluzione molto interessante è stata proposta da Deb and Jain (2002). Considerando una frontiera di riferimento P^* , il grado di convergenza di una frontiera F (ottenuta dai migliori membri della popolazione P fornita dall'algoritmo) può essere calcolato a partire dalla minima distanza euclidea

di ogni i -esimo punto in F da P^* :

$$d_i = \min_{j=1, \dots, |P^*|} \sqrt{\sum_{k=1}^m \left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}} \right)^2} \quad (2.6)$$

dove M è il numero degli obiettivi considerati e f_k^{max} e f_k^{min} sono rispettivamente il massimo e il minimo tra i valori assunti in P^* nel k -esimo obiettivo. A partire dalle distanze euclidee, il grado di convergenza è calcolato come media delle distanze normalizzate per tutti i punti in F :

$$C(P) = \frac{\sum_{i=1}^{|F|} d_i}{|F|} \quad (2.7)$$

Questo indicatore si presta in particolare molto bene alla valutazione dell'andamento della convergenza generazione per generazione.

Esso può essere molto efficace se adottato in contemporanea con indicatori per il calcolo della diversità, di cui se ne cita in particolare uno molto interessante proposto sempre da Deb and Jain (2002). Esso si basa sulla suddivisione dello spazio degli obiettivi in celle di dimensioni $m - 1$, di grandezza anche differente. Utilizzando la stessa simbologia del caso precedente, per ogni cella (i, j, \dots) vengono determinati due valori:

- $H(i, j, \dots) = 1$ se la cella contiene un punto di P^* , $H(i, j, \dots) = 0$ altrimenti;
- $h(i, j, \dots) = 1$ se $H(i, j, \dots) = 1$ e se la cella contiene un punto di F , $h(i, j, \dots) = 0$ altrimenti.

A entrambi i due valori viene associata una funzione $m(\cdot)$, maggiore se il valore H (nel primo caso) o h (nel secondo) della relativa cella è pari a 1, e crescente con il numero di celle vicine occupate; dunque viene calcolato il rapporto

$$D(P) = \frac{\sum_{H(i,j,\dots) \neq 0}^{i,j,\dots} m(h(i, j, \dots))}{\sum_{H(i,j,\dots) \neq 0}^{i,j,\dots} m(H(i, j, \dots))} \quad (2.8)$$

che stabilisce quindi il grado di dispersione della frontiera F nelle celle considerate.

L'approccio costituito dall'uso combinato della (2.7) e della (2.8) è particolarmente brillante ed efficace, tuttavia presenta un inconveniente: si basa sull'ipotesi di conoscere a priori la frontiera di Pareto ottima P^* . Ciò nella maggior parte dei casi reali non è ovviamente vero. Una soluzione proposta è quella di utilizzare come P^* la combinazione delle migliori soluzioni generate nell'arco delle t generazioni effettuate, tuttavia essendo questo risultato

fortemente dipendente dall'algoritmo stesso di cui si vogliono monitorare le prestazioni, non appare essere una soluzione molto convincente.

Gli esempi mostrati, inerenti a metodi particolarmente rappresentativi, riflettono molto bene il fatto che la scelta di uno o più indicatori per la valutazione delle prestazioni di un algoritmo non è assolutamente semplice. Ad oggi infatti non esiste una serie di indicatori che funzioni assolutamente meglio di un'altra, e la scelta è fortemente dipendente dal caso particolare oggetto di studio.

Nel presente lavoro, a valle dei difetti riscontrati in molti metodi analoghi ai precedenti, si cercherà di seguire un approccio basato sulla valutazione assoluta del singolo risultato di un esperimento, che non richieda la conoscenza della frontiera di Pareto P^* . Esso si basa sulla valutazione di quattro indicatori selezionati dal sottoscritto:

- hypervolume HD per la valutazione della convergenza;
- numero N_{soleff} di soluzioni distinte di ranking 1 per la misura della ricchezza della frontiera;
- crowding distance media CD per la valutazione della diversità della popolazione;
- numero N_{conv} di run in cui è stata raggiunta la convergenza, per la generalizzazione dei risultati rispetto al seme.

2.6.1 Hypervolume

Il primo indicatore è Hypervolume, proposto anch'esso da Zitzler e Thiele (1998) e sviluppato in Zitzler (1999). Questo indicatore misura il volume (o l'area nel caso di due obiettivi) compreso fra le soluzioni della frontiera F e un punto di riferimento (comunemente il punto corrispondente alla gestione storica). Inizialmente, lo spazio viene normalizzato attraverso il punto utopia $J_{min} = [J_{min}^1 \quad J_{min}^2 \quad \dots \quad J_{min}^m]$ e il punto di riferimento $J_{max} = [J_{max}^1 \quad J_{max}^2 \quad \dots \quad J_{max}^M]$ in modo che tutti gli obiettivi assumano valori adimensionali compresi tra 0 e 1. Successivamente, le soluzioni di F vengono interpolate nel modo mostrato in Figura 2.6 e viene calcolato il volume tra la frontiera e il punto di riferimento. Questo calcolo avviene per integrazione numerica, con metodo Monte Carlo. Esso consiste nella generazione di una quantità n molto grande di punti in modo casuale nello spazio degli obiettivi normalizzato, e di questi viene calcolata la frazione di punti interna al volume generato

$$HD = \frac{n_{puntiinterni}}{n} \quad (2.9)$$

Valori di hypervolume prossimi a 1 significano che la frontiera si trova in prossimità del punto utopia, mentre valori pari a 0 implicano che non c'è stato alcun miglioramento rispetto alla gestione storica (o al punto di riferimento J_{max}).

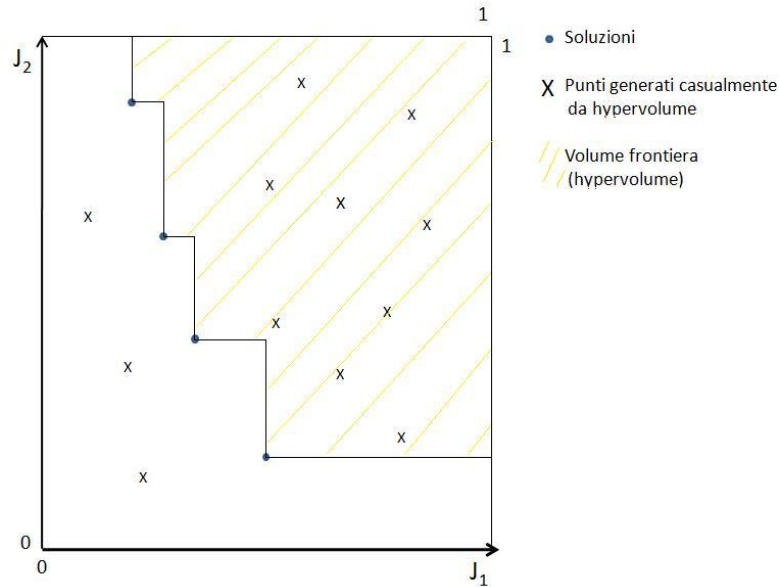


Figura 2.6: Funzionamento di hypervolume.

Questo indicatore è molto interessante in quanto anche con molti obiettivi è in grado di fornire una buona stima della bontà di una frontiera in termini assoluti, cioè senza la necessità di effettuare paragoni con altri risultati. E' inoltre particolarmente adatto agli algoritmi evolutivi in quanto attraverso un suo calcolo in tutte le generazioni si può valutare l'andamento delle prestazioni dell'algoritmo. Teoricamente, questo andamento dovrebbe mostrare valori di hypervolume crescenti in quanto la frontiera si sposta sempre più verso l'utopia. In realtà, a causa della stocasticità nel calcolo numerico dell'integrale può avvenire che in qualche generazione il valore di hypervolume si abbassi leggermente anche se ciò non corrisponde ovviamente a un peggioramento nelle prestazioni. Pertanto, è importante valutare l'andamento di hypervolume considerando una soglia di errore.

Più in generale, l' HD non permette di discriminare fra due approssimazioni della frontiera di Pareto, quando esse differiscano per un volume dell'ordine dell'errore di integrazione numerica.

2.6.2 Soluzioni distinte di ranking 1

Per arricchire l'informazione fornita da HD , si è deciso di inserire come secondo indicatore il numero N_{soleff} di punti distinti con ranking 1, ovvero il numero di soluzioni che realmente costituiscono l'approssimazione della frontiera. Attraverso l'eliminazione dei punti ridondanti l'indicatore risulta decisamente più immediato per l'analista (o il decisore).

Attraverso questo indicatore, frontiere con simile valore di hypervolume sono discriminabili in base al numero di punti che ne costituiscono l'approssimazione.

2.6.3 Crowding distance

Qualora anche il numero di punti che caratterizzano la frontiera sia simile, sono preferibili frontiere caratterizzate da maggiore dispersione dei punti, pertanto come terzo indicatore si utilizza la crowding distance media, che oltretutto possiede la qualità non indifferente di essere uno degli stessi criteri considerati nell'ottimizzazione dell'algoritmo evolutivo. E' infatti buona norma che gli obiettivi che governano il processo di ottimizzazione siano il più possibile conformi agli indicatori con cui se ne valutano poi le prestazioni.

La crowding distance CD dell' i -esimo individuo rappresenta la distanza media tra i suoi vicini in tutti gli m obiettivi considerati. Agli estremi, invece, viene assegnato un valore infinito. L'indicatore finale viene quindi calcolato come media dei valori di crowding distance di tutti gli individui della frontiera esclusi gli estremi.

Come detto, il numero di punti della frontiera viene calcolato eliminando i doppi, in quanto solo i punti unici contribuiscono a costituire una frontiera bene esplorata. Per non perdere coerenza, la crowding distance viene quindi ricalcolata a partire dai soli punti distinti.

2.6.4 Numero di convergenze

L'ultimo indicatore N_{conv} serve solo nella fase di generalizzazione dei risultati nel caso di Test (capitolo 3), attraverso simulazione con elevato numero di semi, come descritto nella sezione 2.5. Questo parametro dice sostanzialmente per quanti semi quel particolare esperimento è arrivato alla convergenza. Per convergenza si intende quella condizione in cui si è raggiunta la frontiera finale ed essa è stata poi totalmente esplorata, ed è quindi verificabile attraverso l'analisi del ranking e della crowding distance: il rispettivo cromosoma possiede infatti tutte (o quasi) soluzioni con ranking 1 (in cui sono state eliminate quindi le soluzioni dominate), e almeno una buona percen-

tuale di esse sono soluzioni distinte: in questo lavoro si utilizza la condizione $N_{soleff} = 0,7 * pop$. In questa fase, inoltre, è doveroso specificare che gli altri indicatori sono calcolati come media del valore assunto nelle diverse ottimizzazioni.

2.6.5 Conclusioni sugli indicatori utilizzati

Il set di indicatori proposto sembra funzionare bene all'atto pratico, dando le giuste dimensioni del potenziale di un algoritmo. Essi riescono infatti a dare varie informazioni sulle soluzioni, che risultano complementari fra loro. In particolare, hypervolume permette di valutare in prima istanza il livello di convergenza di una frontiera, e N_{soleff} descrive il grado con cui essa è approssimata. La crowding distance fornisce quindi l'informazione sull'omogeneità nella distribuzione dei punti, e infine N_{conv} permette di comprendere in termini statistici il potenziale di un esperimento effettuato con diversi semi.

Il loro uso combinato, inoltre, permette di valutare le prestazioni di un esperimento in termini assoluti e senza richiedere la conoscenza della frontiera di Pareto ottima P^* .

La loro lettura però non è banale, in quanto il singolo valore numerico (senza la visualizzazione della frontiera) di un solo indicatore deve essere contestualizzato in funzione del valore assunto dagli altri. La loro corretta manipolazione richiede quindi una certa sensibilità.

Capitolo 3

Caso di Test

Prima di utilizzare l'algoritmo in una situazione reale si ritiene opportuno testarlo su un caso ideale più semplice, al fine di verificarne il corretto funzionamento, il comportamento in diverse situazioni e l'effetto delle variazioni di alcuni parametri significativi. In tal modo è possibile valutarne l'effettivo potenziale e comprenderne tutti gli aspetti e le caratteristiche. E' infatti di fondamentale importanza conoscere approfonditamente l'algoritmo per utilizzarlo efficacemente su un caso reale: in tal modo sarà possibile sfruttarlo al massimo e sarà molto più semplice capire le cause di eventuali criticità o problemi. Al fine di garantire il successo di questa fase, il caso di Test considerato possiede due caratteristiche fondamentali: è molto semplice (rispetto a un comune caso reale) e le sue soluzioni, in termini di leggi di controllo, hanno struttura nota. Analizzando i risultati dell'algoritmo evolutivo è quindi immediatamente possibile valutare se essi sono coerenti con le aspettative. Inizialmente verrà presentata una breve descrizione del sistema (sezione 3.1), il cui simulatore è sviluppato dal gruppo di Modellistica e gestione dei sistemi ambientali del Dipartimento di Elettronica e Informazione del Politecnico di Milano, in Matlab (versione 7.7.1). Successivamente verrà mostrata la fase di ottimizzazione e analisi dei risultati svolta dal sottoscritto. Essa consiste inizialmente nell'implementazione delle regole semplici e nell'analisi delle prestazioni delle diverse architetture neurali (sezione 3.2); dopodichè nella valutazione dei comportamenti e dei risultati dell'algoritmo evolutivo al variare dei parametri di tuning (sezione 3.3); infine nella generalizzazione dei risultati attraverso numerose ottimizzazioni con diversi semi (sezione 3.4) e nella procedura di validazione (sezione 3.5).

L'algoritmo evolutivo NsgaII è stato ottenuto da Matlab Central (Mat (2009)), modificato e manipolato da me per introdurre le personali innovazioni e svolgere le opportune analisi.

3.1 Descrizione del caso di Test

Il caso di Test su cui verrà applicato l'algoritmo evolutivo è costituito da un semplice sistema comprendente un singolo serbatoio, che fornisce acqua a un distretto irriguo (Figura 3.1). Al fine di rendere il caso di Test semplice e immediato, si considera il sistema stazionario.

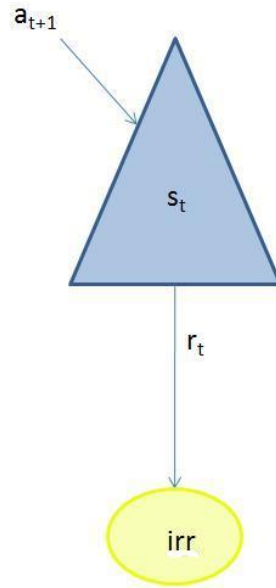


Figura 3.1: Schema del sistema del caso di Test.

Il modello del serbatoio, della forma dell'equazione (1.1) è il seguente

$$s_{t+1} = s_t + a_{t+1} - r_t \quad (3.1)$$

in cui a_{t+1} corrisponde all'afflusso e r_t al rilascio

$$r_t = \max(v_t, \min(V_t, u_t)) \quad (3.2)$$

u_t corrisponde alla decisione di rilascio all'istante t ed è la variabile di controllo del sistema. v_t e V_t sono invece le funzioni di minimo e massimo rilascio, identificate dalle seguenti equazioni

$$v_t = \max(s_t - 100, 0) \quad (3.3)$$

$$V_t = s_t \quad (3.4)$$

Il dominio dei valori ammissibili delle variabili è $0 \leq s_t \leq 155$ per l'invaso e $0 \leq u_t \leq 160$ per la decisione di rilascio.

In questo test sono considerati due obiettivi: minimizzare le piene nei territori adiacenti al serbatoio, e minimizzare il deficit al distretto irriguo. Il primo è associato al costo per passo

$$g_t^1 = \max(h_t - \bar{h}, 0) \quad (3.5)$$

corrispondente al dislivello tra il lago e una quota di riferimento \bar{h} corrispondente al limite di esondazione. h_t si trova dividendo l'invaso s_t per la superficie del serbatoio S . In questo sistema $S = 1$, pertanto $h_t = s_t$. Il limite di esondazione \bar{h} è invece fissato a 50.

Il secondo obiettivo è invece associato al costo per passo

$$g_t^2 = \max(\bar{r} - r_t, 0) \quad (3.6)$$

che corrisponde al deficit irriguo, in cui la domanda \bar{r} è fissata a 50. Per quanto riguarda gli afflussi a_{t+1} , è disponibile una serie di 5000 dati ottenuti per generazione casuale da una normale (40;0,001). Per la simulazione durante il processo di ottimizzazione verrà utilizzata una parte di essi, di lunghezza h sufficientemente lunga perchè si possa utilizzare correttamente la forma delle funzioni obiettivo (1.20).

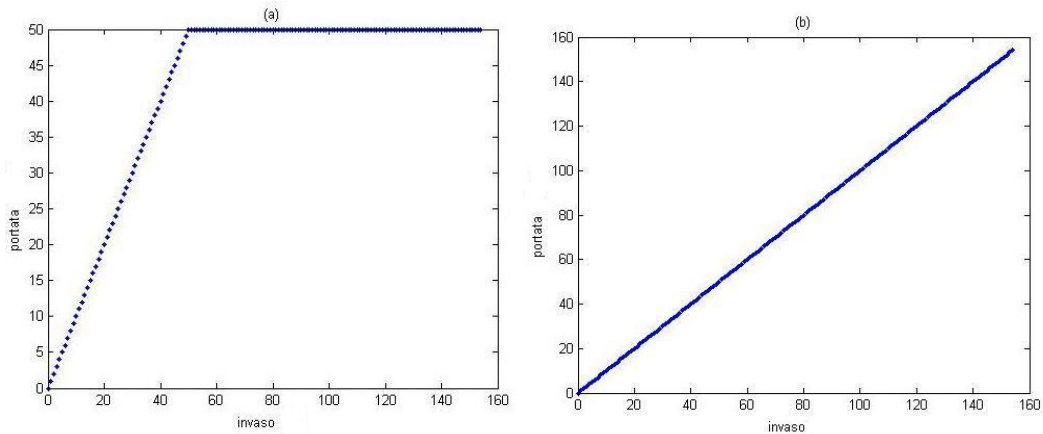


Figura 3.2: Leggi di controllo ottime per il solo obiettivo irriguo (a) e per l'obiettivo di riduzione delle piene (b).

In questo caso di Test è nota a priori la forma delle leggi di controllo che ci si aspetta di ottenere come soluzione del processo di ottimizzazione. Le due leggi di controllo presentate in Figura 3.2 sono infatti le leggi di controllo

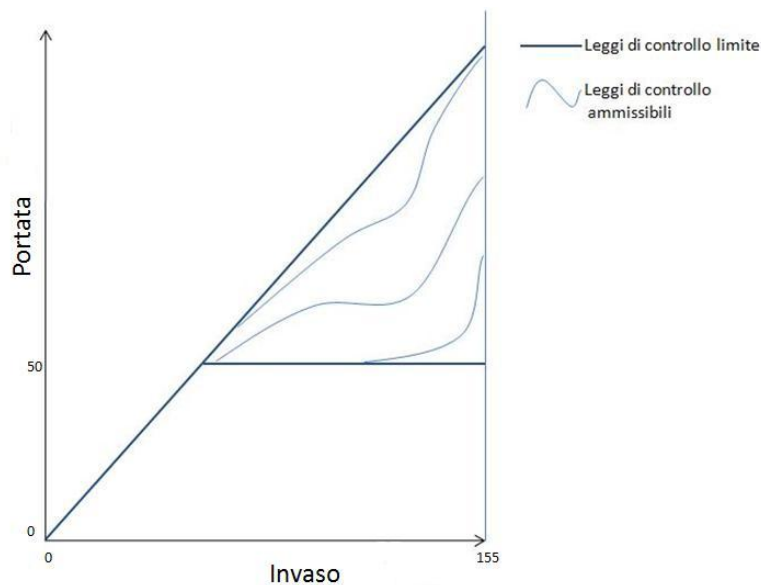


Figura 3.3: Leggi di controllo attese.

ottime dal punto di vista dei due obiettivi presi singolarmente.

Per gli irrigui sarà importante rilasciare una portata sempre il più possibile vicina alla domanda \bar{r} : per invasi superiori a essa la decisione di rilascio ottima sarà proprio \bar{r} , mentre per invasi inferiori, sarà conveniente rilasciare tutto il possibile, originando la Figura 3.2 (a).

Per i rivieraschi del serbatoio, al fine di minimizzare il rischio di piene, sarà invece desiderabile rilasciare sempre tutto il possibile, per fare in modo di avere sempre i minimi livelli possibili (Figura 3.2 (b)).

Queste due regole semplici corrispondono quindi ai casi limite in cui viene considerato solo uno dei due obiettivi del caso di Test, pertanto le leggi di controllo che ci si aspetta di trovare come risultato del processo di ottimizzazione saranno intermedie a esse, e quindi della forma presentata in Figura 3.3.

3.1.1 Parametri significativi

Rete neurale

Nella sezione 2.1 è stato ampiamente descritto il funzionamento delle reti neurali. La scelta dell'architettura è, come si può facilmente intuire, determinante nella generazione di soluzioni efficienti in quanto da essa dipende la bontà dell'approssimazione delle leggi di controllo. L'architettura della rete

utilizzata nell'algoritmo è modificabile attraverso due parametri: il numero e la tipologia dei neuroni dello strato nascosto. E' immediato osservare che un numero crescente di neuroni comporta generalmente un aumento nelle prestazioni della rete, ma d'altra parte ciò genera anche un aumento del numero di parametri da stimare. Sarà dunque molto importante valutare quale sia l'architettura che permette di ottenere leggi di controllo efficienti minimizzando il numero di neuroni. Nel caso in esame le tipologie dei neuroni dello strato nascosto considerate sono:

- tansig;
- logsig;
- purelin.

Algoritmo evolutivo

Per quanto riguarda l'algoritmo evolutivo NsgaII (con le nostre modifiche), i parametri su cui intervenire sono principalmente quattro: la grandezza della popolazione (abbreviata *pop*), il numero di generazioni (*gen*), l'introduzione o meno delle parametrizzazioni delle regole semplici nell'inizializzazione e l'influenza dell'orizzonte di simulazione *h*.

Per quanto riguarda *pop*, ci si aspetta che le prestazioni dell'algoritmo crescano con esso, essendo più numerosi a ogni iterazione i punti disponibili per esplorare lo spazio dei parametri. Per quanto riguarda invece le generazioni, è interessante valutare per ogni esperimento quante ne servano per convergere a una determinata frontiera a cui corrisponde un certo valore degli indicatori utilizzati (sezione 2.6).

Riveste infine una particolare importanza l'introduzione delle parametrizzazioni relative alle regole semplici nell'inizializzazione dell'algoritmo. Come ampiamente discusso nella sezione 2.4, ci si aspetta che questa innovazione generi notevoli benefici. D'ora in avanti, la configurazione per cui nell'inizializzazione siano incluse anche tali parametrizzazioni sarà abbreviata: "Caso SI", mentre la situazione opposta (inizializzazione completamente casuale) "Caso NO".

In questo caso di Test sono state considerate quattro regole semplici per l'inizializzazione (Figura 3.4):

1. rilasciare esattamente la domanda irrigua quando possibile (regola semplice che coincide, nel caso specifico, con la regola ottima per l'obiettivo degli irrigui di ottenere sempre una portata il più possibile vicina al loro fabbisogno);

2. rilasciare tutto il volume possibile (regola semplice che coincide, nel caso specifico, con la regola ottima per l'obiettivo "Minimizzare le piene a monte");
3. rilasciare il minimo possibile, seguendo quindi il vincolo di minimo rilascio;
4. soluzione intermedia di compromesso tra le prime due.

In questo caso l'ingresso I_t della rete è costituito dal solo stato del sistema, ovvero l'invaso del serbatoio, dal momento che il sistema è stazionario e la variabile tempo non avrebbe quindi significato. L'uscita è invece rappresentata dalla decisione di rilascio. Le leggi di controllo delle regole semplici hanno in questo test il notevole vantaggio di avere forma nota analiticamente (Figura 3.4) per cui oltre a essere più semplice la loro implementazione è possibile raggiungere la massima precisione nella fase di addestramento.

Infine, sarà interessante valutare l'impatto dell'orizzonte di simulazione h . Come descritto nel paragrafo 1.4.2, è necessario che esso sia sufficientemente lungo perchè si possa considerare una singola traiettoria del disturbo (in questo caso a_t) invece di una statistica di tutte le sue possibili realizzazioni. Dal momento che non esiste ad oggi un'indicazione univoca sul quale sia effettivamente il valore di h minimo da utilizzare, in questo caso di Test si valuterà l'impatto di questo parametro all'interno del processo di ottimizzazione.

3.2 Scelta dell'architettura

L'obiettivo principale di questa prima fase è valutare la capacità delle diverse architetture neurali di approssimare le leggi di controllo relative alle regole semplici, secondo la metodologia introdotta nella sezione 2.4. Questa fase è realizzata effettuando l'addestramento della rete neurale secondo le quattro regole semplici considerate, e paragonando i risultati con le relative leggi di controllo note analiticamente.

A partire dai primi esperimenti si nota subito che per i tipi tansig e logsig due neuroni sono già sufficienti per approssimare discretamente le funzioni delle regole semplici (Figura 3.5), anche con numero di iterazioni $N_{iter} = 1$. Il tipo purelin invece non è in grado di riprodurre le forme delle leggi di controllo (a), (c) e (d), nemmeno con un numero elevato di neuroni (Figura 3.6).

In particolare, le difficoltà principali della rete neurale risiedono nell'approssimazione dei punti angolosi (le regole semplici sono delle spezzate) dal momento che la struttura delle reti neurali non permette di riprodurre perfettamente questo tipo di geometrie. La regola semplice due (rilasciare sempre tutto), che corrisponde alla retta bisettrice del piano invasore-decisione

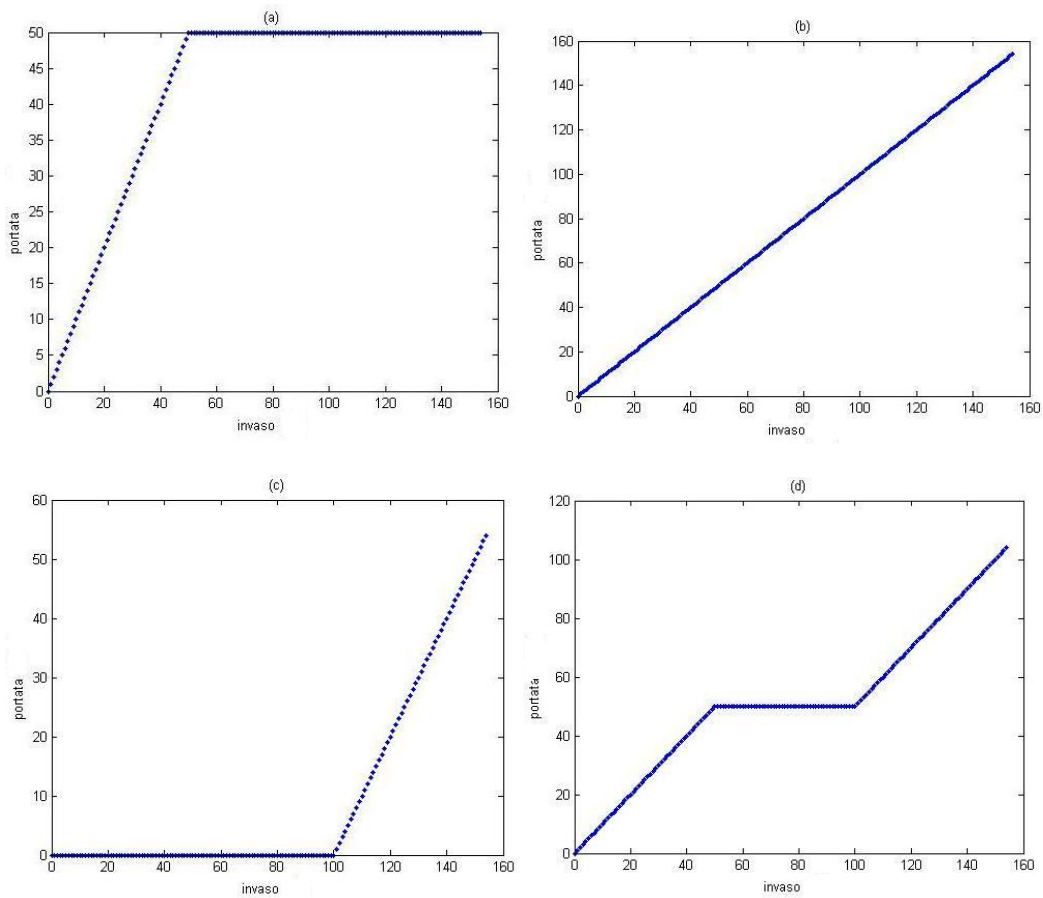


Figura 3.4: Le leggi di controllo relative alle quattro regole semplici.

di rilascio, essendo geometricamente molto semplice è invece perfettamente approssimata con qualunque numero di neuroni (valori di varianza spiegata $R2 \approx 1$), anche attraverso la tipologia purelin.

Per numeri di neuroni crescenti, come ci si aspettava, l'approssimazione è sempre migliore, mentre le differenze tra le tipologie tansig e logsig risultano piccole e fluttuanti in base al numero di neuroni e alla regola semplice considerata.

Attraverso l'analisi incrociata delle figure con i valori di varianza spiegata $R2$, si nota che un'approssimazione buona e sicuramente soddisfacente si ha già per $R2 > 0,990$. La precisione diventa ottima per $R2 = 0,9980$, e pressochè perfetta quando $R2 = 0,9990$: questi valori saranno pertanto adottati come riferimenti nella selezione dell'architettura migliore.

Con 5 neuroni, già con numero di iterazioni $N_{iter} = 1$, la riproduzione del-

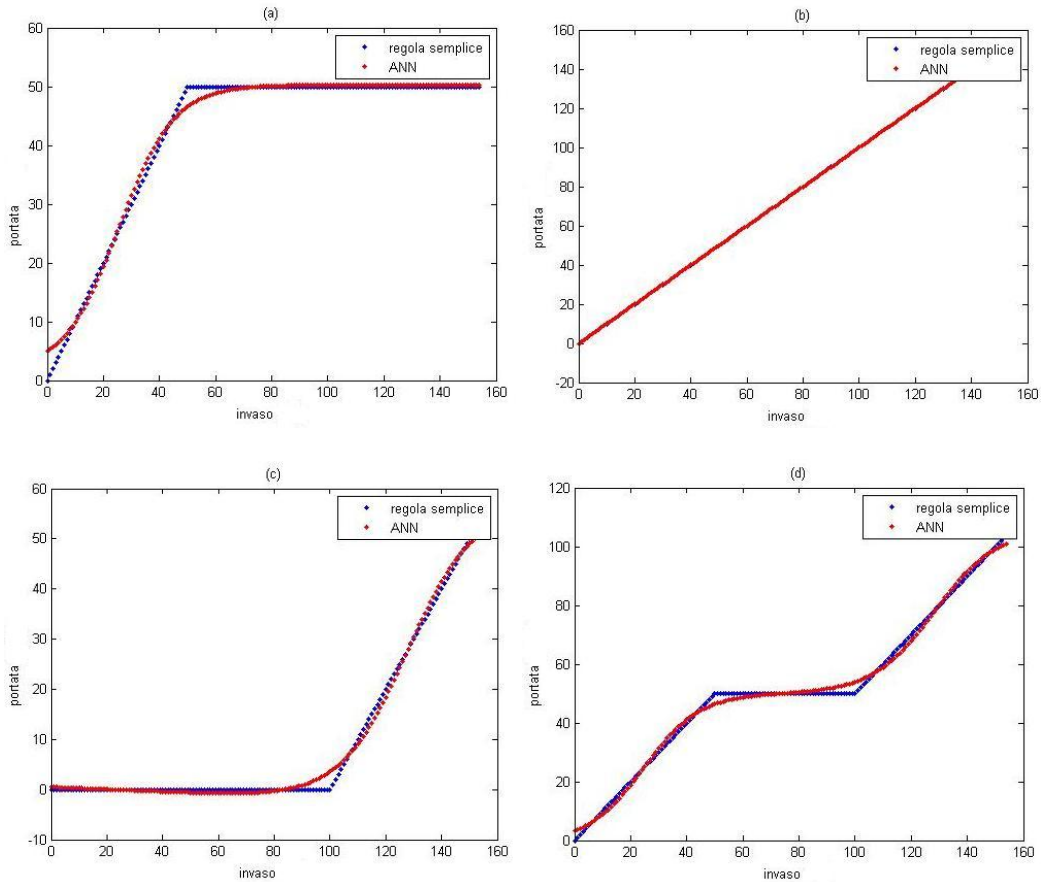


Figura 3.5: Approssimazione leggi di controllo con 2 neuroni tansig.

le leggi di controllo è pressochè perfetta, con valori di $R^2 > 0,9990$ per entrambe le tipologie tansig e logsig.

La scelta definitiva dell'architettura si compie implementando un ciclo che esegue l'addestramento della rete neurale secondo configurazioni diverse. Le combinazioni tenute in considerazione sono, visti i risultati appena discussi, tutti gli incroci possibili tra le tipologie tansig e logsig e numero di neuroni da 1 a 5. A ogni iterazione vengono salvati i parametri e i range ottenuti dall'addestramento (saranno utili come input per la successiva fase di ottimizzazione con l'algoritmo evolutivo vero e proprio), e vengono registrate le varianze spiegate R^2 relative a tutte e quattro le regole semplici. Per questa fase è necessario effettuare, per i motivi mostrati nella sezione 2.4, un'analisi di sensitività sul numero N_{iter} di ripetizioni dell'addestramento per architettura della rete. Come si è visto la seconda regola semplice (rilasciare tutto il volume possibile) è approssimata perfettamente con qualunque architettura

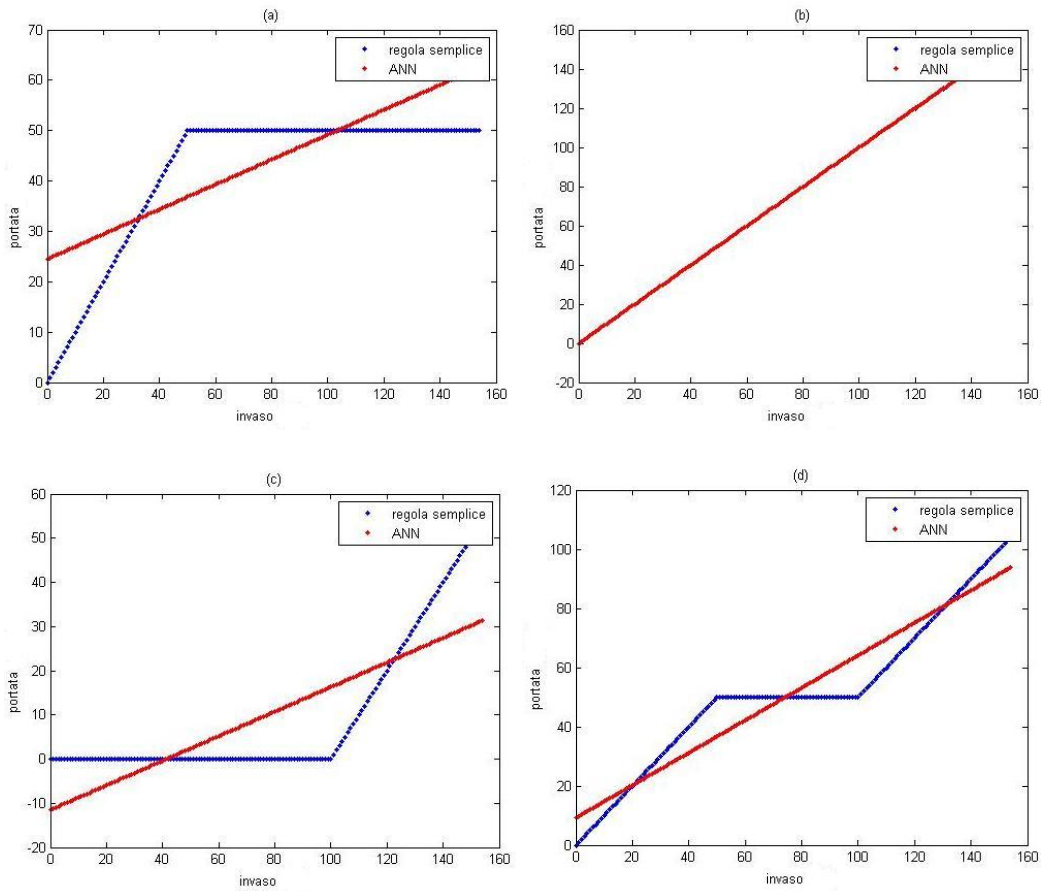


Figura 3.6: Approssimazione leggi di controllo con 5 neuroni purelin.

quindi, per ridurre i tempi di calcolo, per questa legge di controllo N_{iter} è fissato a 1. Per gli altri tre casi, le architetture sono state testate per $N_{iter} = 1$, $N_{iter} = 5$, $N_{iter} = 20$, $N_{iter} = 50$, $N_{iter} = 100$ e $N_{iter} = 500$. Come ci si aspettava i risultati sono sempre leggermente variabili, a testimonianza del fatto che comunque la casualità del punto iniziale di ricerca del minimo ha sempre un peso concreto. Tuttavia, per $N_{iter} \geq 20$, diverse simulazioni mostrano un trend definito. La Tabella 3.1 contiene i risultati relativi a $N_{iter} = 100$, che rappresentano in modo molto preciso le prestazioni ottenibili dalle diverse architetture.

Si nota, come anticipato dalle prime osservazioni, che le architetture con un neurone non sono in grado di approssimare adeguatamente le leggi di controllo delle regole semplici. Con due neuroni (a patto di usare un sufficiente numero N_{iter}) si possono invece ottenere delle prestazioni per la verità piuttosto buone. Aumentando il numero di neuroni la precisione aumenta

Tabella 3.1: Prestazioni delle diverse architetture, con $N_{iter} = 1$ per la seconda regola semplice e $N_{iter} = 100$ per le altre. R2 è la varianza spiegata relativa a ognuna delle quattro regole semplici.

Numero neuroni	Tipo neuroni	R2 1	R2 2	R2 3	R2 4
1	logsig	0,9948	1,0000	0,9947	0,9141
2	logsig	0,9993	1,0000	0,9984	0,9959
3	logsig	0,9999	1,0000	0,9988	0,9997
4	logsig	0,9999	1,0000	0,9995	0,9992
5	logsig	0,9999	1,0000	0,9996	0,9993
1	tansig	0,9948	1,0000	0,9947	0,9141
2	tansig	0,9993	1,0000	0,9984	0,9985
3	tansig	0,9998	1,0000	0,9994	0,9990
4	tansig	0,9999	1,0000	0,9994	0,9991
5	tansig	1,0000	1,0000	0,9999	0,9995

sempre più, tuttavia si può notare che un'architettura con tre neuroni tansig è già in grado di fornire in tutte le regole semplici un valore di varianza spiegata $R2 \geq 0,9990$, comportando un'approssimazione delle leggi di controllo pressochè perfetta, pertanto risulta inutile ricercare soluzioni con maggior numero di neuroni. In definitiva, la scelta dell'architettura ricade su tre neuroni tansig: questa sarà dunque quella che verrà d'ora in poi utilizzata nel caso di Test (Figura 3.7).

A partire dalle prove appena compiute, che permettono di conoscere le prestazioni ottenibili dalle diverse architetture, è stata quindi effettuata una valutazione per determinare quale valore di N_{iter} sia generalmente da utilizzare. Questa è basata sostanzialmente su un metodo statistico, che su un numero elevato di prove (5000) per ogni regola semplice calcola il numero di volte in cui (con 150 epoche) il risultato dell'addestramento ha registrato una varianza spiegata maggiore di una certa quantità preimpostata. In tal modo, è possibile valutare per ogni regola semplice e per ogni architettura quante iterazioni N_{iter} servono per ottenere in media determinate prestazioni.

A titolo esemplificativo si mostrano i risultati relativi alle architetture con tre neuroni (Tabelle 3.2 e 3.3), in cui il numero di iterazioni è arrotondato per eccesso all'intero più vicino. I risultati sono molto variabili, in base al caso esaminato si passa da poche iterazioni a un numero pari addirittura a 1000. Ciò testimonia che il parametro N_{iter} da utilizzare è fortemente dipendente dalle leggi di controllo da approssimare e dall'architettura. In ogni caso, risulta a queste condizioni praticamente impossibile definire univocamente un

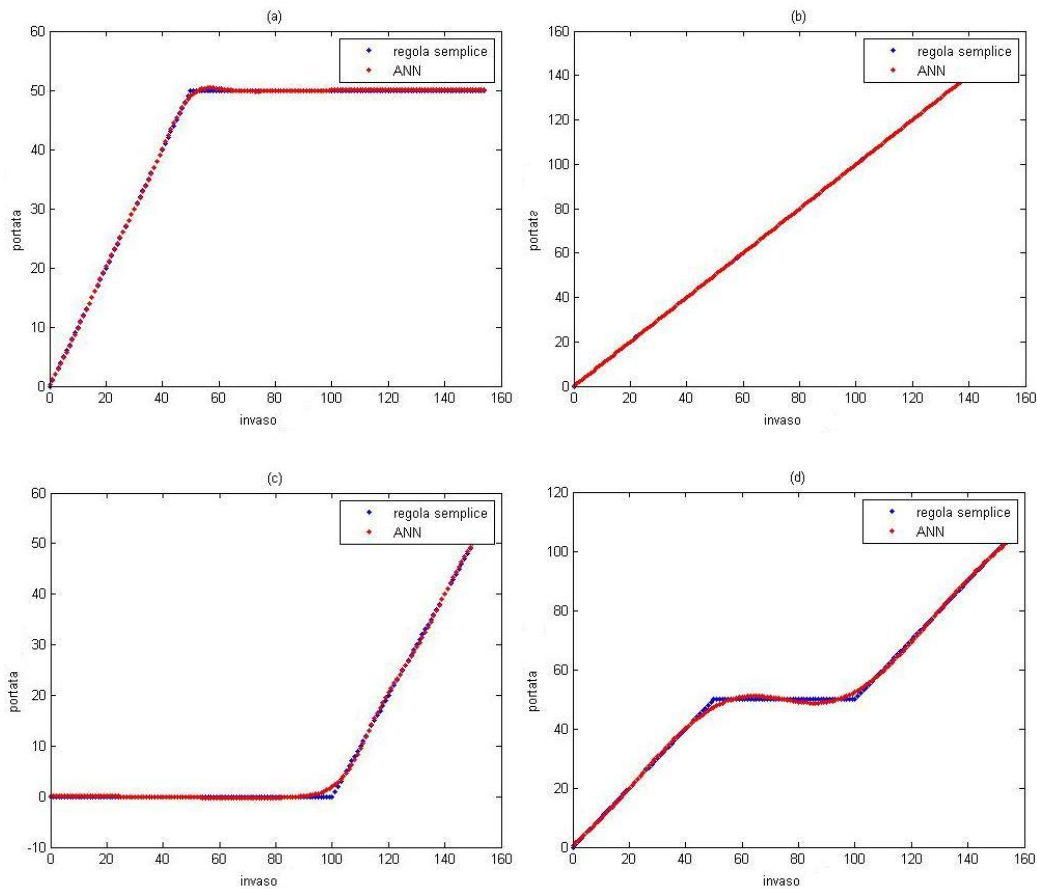


Figura 3.7: Approssimazione delle regole semplici con 3 neuroni tansig.

valore del parametro da utilizzare, tuttavia, a partire dai risultati fino a qui mostrati si possono considerare alcune indicazioni:

- non utilizzare mai una sola iterazione: questa situazione va bene solo per leggi di controllo estremamente semplici, mentre in tutti gli altri casi servono varie iterazioni per ottenere un risultato ottimale (se è conseguibile);
- anche se raggiungere determinate prestazioni è possibile, a patto di utilizzare un numero enorme di iterazioni, non è affatto conveniente avere un N_{iter} esageratamente grande dal momento che, oltre ad aumentare notevolmente i tempi richiesti dall'addestramento, anche in fase di ottimizzazione l'architettura utilizzata potrebbe avere delle difficoltà a garantire buoni risultati. Infatti se una bassissima percentuale di

Tabella 3.2: Numero medio di iterazioni necessarie per ottenere valori di varianza spiegata $R^2 > 0,9989$ (approssimazione perfetta) in tutte le regole semplici (tre neuroni).

Tipo neuroni	Regola1	Regola2	Regola3	Regola4
logsig	3	1	1000	2
tansig	4	1	107	109

Tabella 3.3: Numero medio di iterazioni necessarie per ottenere valori di varianza spiegata $R^2 > 0,9980$ (approssimazione buona) in tutte le regole semplici (tre neuroni).

Tipo neuroni	Regola1	Regola2	Regola3	Regola4
logsig	1	1	1	2
tansig	1	1	2	2

parametrizzazioni comporta dei buoni risultati nell'addestramento, allo stesso modo in fase di ottimizzazione potrebbero esserci pochissime regioni nello spazio dei parametri che siano in grado di generare delle buone soluzioni, e esse potrebbero essere raggiunte solo con un gran numero di generazioni o addirittura rimanere inesplorate.

3.3 Funzionamento dell'algoritmo

In questa fase ci si propongono principalmente due obiettivi: valutare se l'algoritmo è in grado di giungere alla frontiera di Pareto, e comprendere l'impatto dei parametri di tuning (pop , gen , h , tipo di inizializzazione) sulle soluzioni. Si cercherà di capire quindi in modo esauriente il comportamento dell'algoritmo.

Tutte le operazioni effettuate sono riferite a un singolo seme (dove non specificato il contrario). Questa parte ha infatti lo scopo di rappresentare una fase di studio preliminare.

L'algoritmo, teoricamente descritto nel capitolo 2, dal punto di vista operativo principalmente riceve in ingresso:

- i parametri pop e gen . Come descritto nella sezione 2.2, pop rappresenta la grandezza della popolazione, ovvero il numero di soluzioni che viene generato dall'algoritmo a ogni generazione; mentre gen è il numero

effettivo di generazioni (ovvero di iterazioni) per il quale l'algoritmo effettuerà il processo di ottimizzazione;

- il numero di obiettivi considerati nel caso di Test q (in questo caso pari a due);
- i set di parametri ottenuti dall'addestramento della rete neurale (solo nel Caso SI), e i vettori relativi al range di variazione dei parametri;
- i parametri del sistema, necessari per la simulazione.

Per quanto riguarda la serie di afflussi, come detto si possiede un file di testo che contiene una serie di cinquemila dati, e agendo opportunamente sul codice, è possibile modificare la lunghezza h della serie da fornire in ingresso all'algoritmo evolutivo e valutare le differenze. Sarà interessante in questo test effettuare anche un'analisi di sensitività sulla lunghezza h della serie di afflussi.

Visti i risultati ottenuti nella fase di addestramento della rete neurale, l'architettura utilizzata è 3 neuroni tansig.

3.3.1 Caso SI: inizializzazione con regole semplici

Le prime configurazioni dell'algoritmo evolutivo esaminate sono:

- Caso SI, $pop = 200$, $gen = 100$;
- Caso SI, $pop = 100$, $gen = 50$;
- Caso SI, $pop = 20$, $gen = 15$.

Inizialmente la lunghezza dell'orizzonte di simulazione h è fissato a 500, un valore ragionevolmente elevato. In tutti e tre i casi, il risultato ottenuto è una frontiera di Pareto molto simile, esplorata molto bene (ovviamente in modo più fitto nei casi con valori maggiori di pop).

La frontiera ottenuta nel caso $pop = 100$ e $gen = 50$ è mostrata in Figura 3.8. Gli estremi sono prossimi alle inizializzazioni (corrispondenti ai cerchi rossi) relative alle prime due regole semplici (che, come ci si aspettava, sono infatti proprio quelle che corrispondono ai due obiettivi considerati nel problema). In Figura 3.8 (b) sembra addirittura che delle soluzioni migliorino l'obiettivo "deficit irriguo" rispetto alla regola semplice 1: ciò è dovuto ad errori di approssimazione (le differenze, ovviamente, sono comunque minime). Le soluzioni del cromosoma finale sono ben distribuite in tutto l'arco della

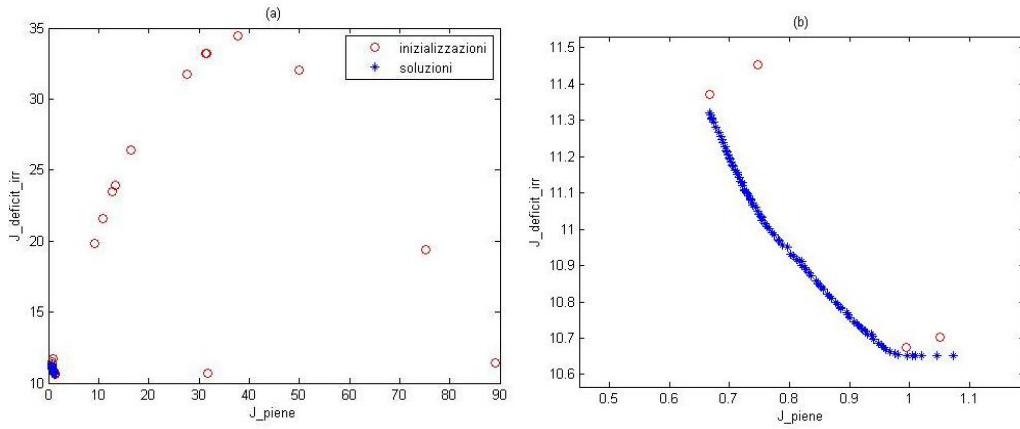


Figura 3.8: Frontiera di Pareto ottenuta con Caso SI, $pop = 100$ e $gen = 50$ zoomata (b) e non (a).

frontiera. Come si può notare nella Figura 3.8 (b), l’inizializzazione (marchiata con il cerchio rosso più a destra) relativa alla regola semplice quattro, che dovrebbe rappresentare una ragionevole soluzione di compromesso, è dominata da qualche soluzione ottenuta nel risultato finale, segno che essa non era in realtà ottima, e l’algoritmo evolutivo è stato in grado di trovare delle parametrizzazioni migliori. In questo caso gli errori di approssimazione, il particolare stato iniziale utilizzato e la serie di afflussi in ingresso hanno fatto sì che essa fosse addirittura leggermente dominata dalla regola semplice 1. Interessante notare, inoltre, come i punti relativi all’inizializzazione (cerchi rossi) siano molto pochi in confronto ai valori di pop . Questo è essenzialmente dovuto al fatto che l’algoritmo durante l’inizializzazione esplora in modo casuale lo spazio dei parametri interno al range fornito in ingresso. Delle pop soluzioni generate, tuttavia, molte di esse corrisponderanno a leggi di controllo fisicamente non eseguibili, perchè non rispettano i vincoli di massimo e minimo rilascio. Ne consegue che tutte le leggi di controllo che si trovano al di sopra della funzione di massimo rilascio (3.4) saranno tradotte, durante la simulazione, tutte nella stessa soluzione che corrisponde proprio al massimo rilascio secondo il vincolo: $r = \min(u, V)$. Allo stesso modo, tutte le inizializzazioni che corrispondono a leggi di controllo inferiori alla funzione di minimo rilascio (3.3) saranno tradotte tutte nella stessa legge di controllo corrispondente proprio al minimo rilascio, per via del vincolo: $r = \max(u, v)$. Tutte le inizializzazioni che non rispettano i vincoli per nessun valore ammissibile dell’invaso, pertanto, saranno mappate in soli due punti nello spazio degli obiettivi (uno positivo, corrispondente al massimo rilascio, e l’altro pessimo, corrispondente al minimo), pur avendo parametrizzazioni totalmente

differenti.

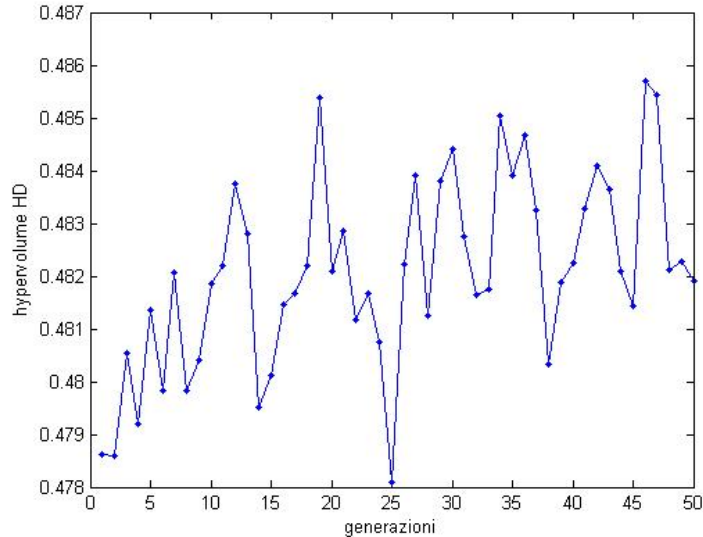


Figura 3.9: Andamento di hypervolume nelle generazioni (caso SI, $pop = 100$ e $gen = 50$), con $J_{max} = [5, 24]$.

Dai risultati si nota che anche gli esperimenti con valori bassi di pop forniscono comunque una frontiera di Pareto ottima, generando piccolissime differenze anche nei valori di hypervolume finali (punto di riferimento $J_{max} = [5, 24]$; $J_{min} = [0, 0]$).

La dinamica di hypervolume nelle generazioni non mostra affatto l'andamento crescente che ci si aspetterebbe normalmente, rimane anzi pressochè costante dall'inizio alla fine (Figura 3.9). Ciò è probabilmente dovuto alla bontà dell'inizializzazione: nel Caso SI, le parametrizzazioni ottenute dall'addestramento secondo le regole semplici 1,2 e 4 sono già molto vicine all'ottimo proprio per il principio con il quale esse vengono create, perciò il valore di hypervolume è già pressochè al massimo fin dall'inizio.

Infine, nella Figura 3.10, vengono mostrati i set di parametri relativi a tutte le soluzioni del cromosoma finale. Interessante notare che esse corrispondono tutte a un range particolarmente ristretto.

I risultati ottenuti utilizzando un valore di $gen = 5$ sono decisamente diversi da quelli mostrati in Figura 3.8: sono presenti poche soluzioni prossime all'ottimo. Aumentando progressivamente le generazioni la situazione migliora, e intorno a $gen = 10$ ci si trova già di fronte a una buona frontiera. I due casi sono mostrati in Figura 3.11. Sebbene il valore di Hypervolume sia già elevato fin da subito per i motivi prima esposti, quindi, inizialmente

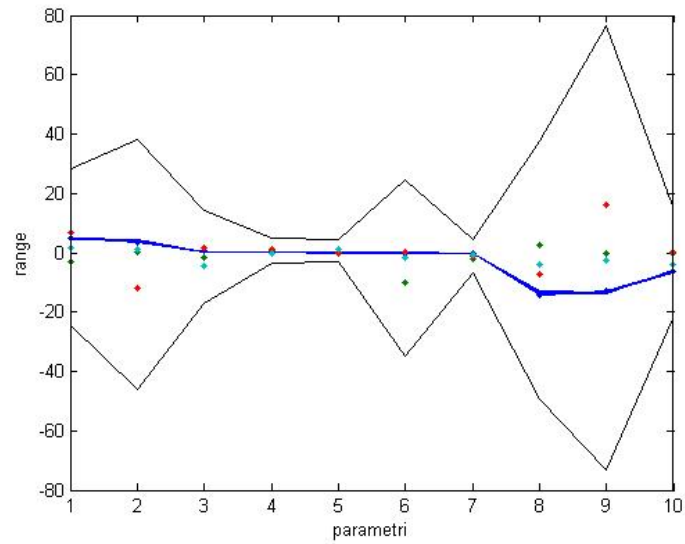


Figura 3.10: Range di parametri del cromosoma finale (caso SI, $pop = 100$ e $gen = 50$).

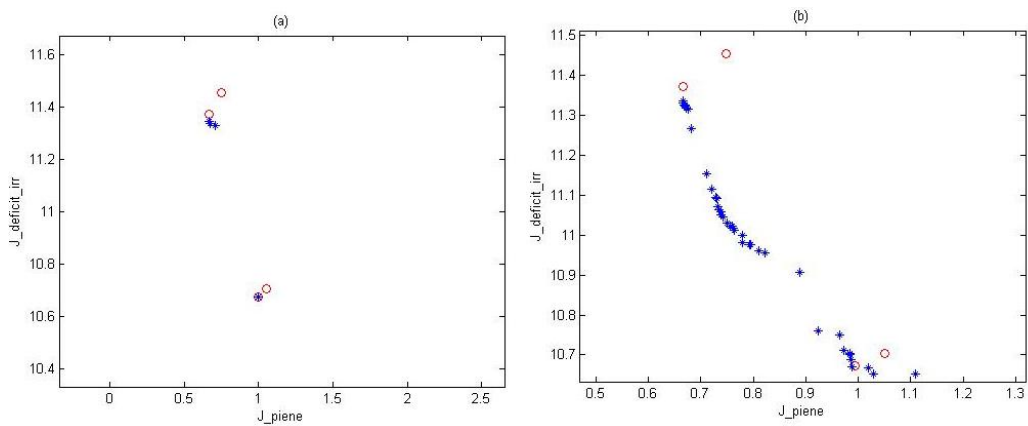


Figura 3.11: Frontiera di Pareto ottenuta con $pop = 100$ e $gen = 5$ (a) e $gen = 10$ (b) nel Caso SI.

il numero di soluzioni efficienti N_{soleff} è piuttosto basso, ma tende a crescere molto rapidamente con le generazioni, fino a valori pari alla totalità della popolazione.

3.3.2 Caso NO: inizializzazione completamente casuale

A fronte degli ottimi risultati ottenuti anche con bassi valori di pop e delle considerazioni fatte al termine del precedente paragrafo legate all'apparente importanza delle regole semplici nell'inizializzazione, sembra interessante valutare come cambiano le cose nel caso di inizializzazione puramente casuale (Caso NO). Per correttezza nei prossimi esperimenti si uniforma il seme con la stessa configurazione utilizzata nei tre casi precedenti.

Nel caso di inizializzazione puramente casuale i progressi con l'aumentare delle generazioni sembrano inferiori.

A titolo esemplificativo, nella Figura 3.12 viene mostrato il risultato relativo a 10 generazioni nel Caso NO: si notano pochi punti molto vicini, prossimi all'inizializzazione relativa al massimo rilascio (che viene ottenuta praticamente sempre, per la questione dei vincoli discussa nel precedente paragrafo). Esso è, come detto, già praticamente sulla frontiera di Pareto finale, perciò in così poche generazioni l'algoritmo è stato in grado solo di eliminare le soluzioni peggiori e proiettare tutta la popolazione finale intorno a quel punto.

Curiosa anche la valutazione del range di parametri post ottimizzazione, mostrata nella Figura 3.13. Al contrario del caso precedente (Figura 3.10), si nota che nella popolazione finale i diversi cromosomi hanno un valore dei parametri che copre praticamente l'intero range stabilito. Questa osservazione è interessante, perchè significa che nonostante l'algoritmo abbia cercato in tutto lo spazio disponibile (a partire dall'inizializzazione), non è riuscito a convergere alla frontiera.

Il valore di hypervolume in questo caso è abbastanza elevato perchè il punto di massimo rilascio (a cui corrisponde pressochè la totalità della popolazione) è come detto già prossimo all'ottimo; tuttavia è comunque inferiore al caso SI perchè la frontiera è stata esplorata solo in una sua frazione minima.

Incrementando la popolazione, i risultati non variano in modo significativo. Aumentando invece il numero di generazioni, sembra che anche nel Caso NO sia possibile ottenere risultati simili, anche se leggermente inferiori, a quanto osservato nel Caso SI (in termini di frontiera e di Hypervolume). In Figura 3.14 viene mostrato un confronto tra Caso SI e NO con $pop = 100$ e $gen = 50$: si può notare che il Caso NO risulta leggermente peggiore, tuttavia le differenze sono minime al punto da non generare concrete differenze di hypervolume.

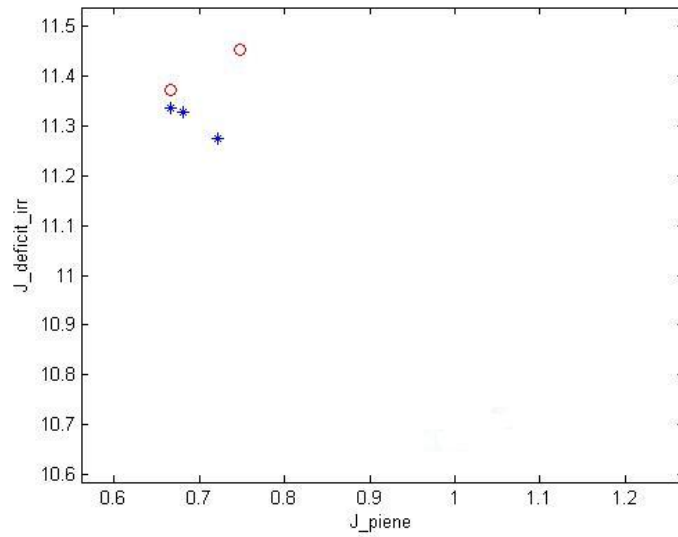


Figura 3.12: Frontiera di Pareto (zoomata) ottenuta con $pop = 100$ e $gen = 10$, caso NO.

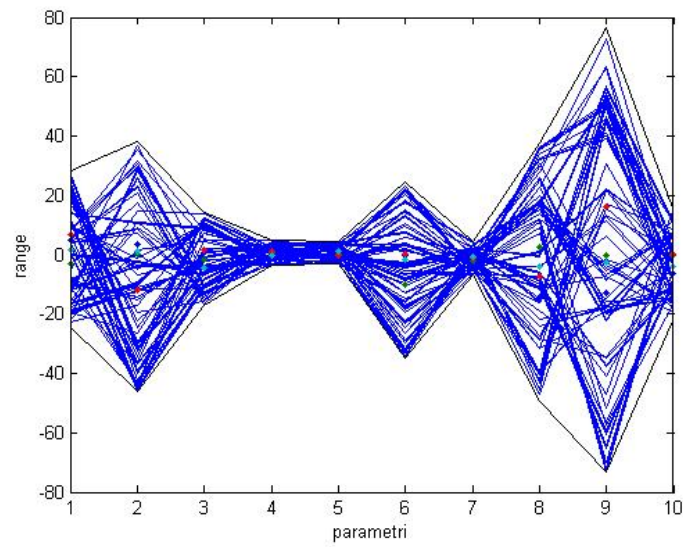


Figura 3.13: Range di parametri del cromosoma finale (Caso NO, $pop = 100$ e $gen = 10$).

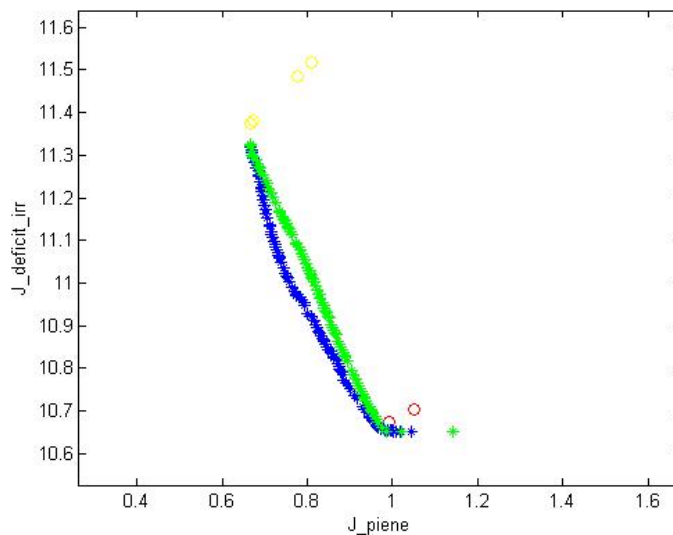


Figura 3.14: Confronto tra Caso SI (soluzioni blu) e NO (soluzioni verdi). I cerchi gialli corrispondono alle inizializzazioni in comune, mentre quelli rossi a quelle del solo Caso SI (regole semplici 1 e 4)).

E' dunque interessante il grafico relativo all'andamento di hypervolume con 50 generazioni nel Caso NO, Figura 3.15. L'andamento si mantiene costante su un valore leggermente basso, e poi intorno alla quindicesima generazione sale rapidamente fino a raggiungere un valore elevato. Evidentemente in tale generazione è apparsa una soluzione significativamente migliore.

Ripetendo gli esperimenti cambiando il seme, si ottengono risultati in linea con quanto mostrato in questi due paragrafi.

Per alcuni semi, tuttavia, si osserva che il Caso NO può non essere mai in grado di raggiungere buoni risultati (Figura 3.16). In questi casi, il comportamento dell'algorithm è tale da non riuscire, dopo l'inizializzazione, a ottenere ulteriori punti efficienti che permettano di effettuare il salto di Hypervolume mostrato in Figura 3.15 e di esplorare la frontiera. Come risultato, la popolazione si stabilizza su un punto (o pochi), prossimo all'inizializzazione di massimo rilascio. Il caso SI invece, indipendentemente dal seme, sembra sempre in grado di ottenere ottimi risultati, come si vede nella stessa Figura 3.16.

Quest'ultima considerazione è fondamentale, perchè getta luce sul potenziale che hanno le regole semplici sull'ottimizzazione.

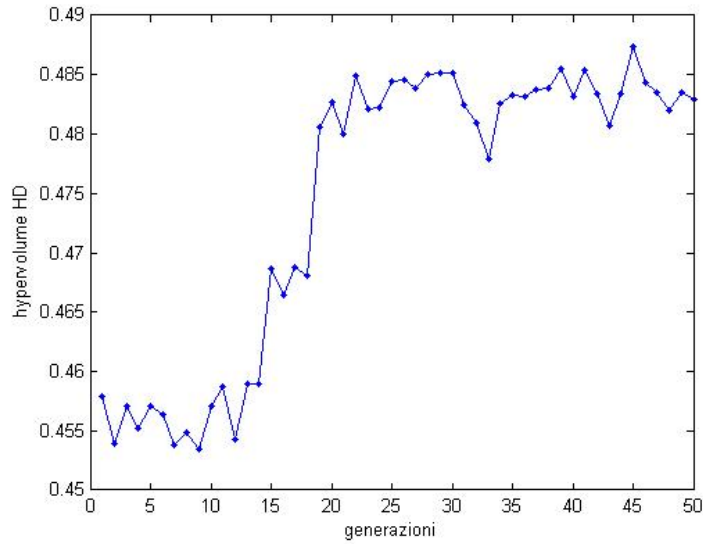


Figura 3.15: Andamento di hypervolume nelle generazioni (Caso NO, $pop = 100$ e $gen = 50$).

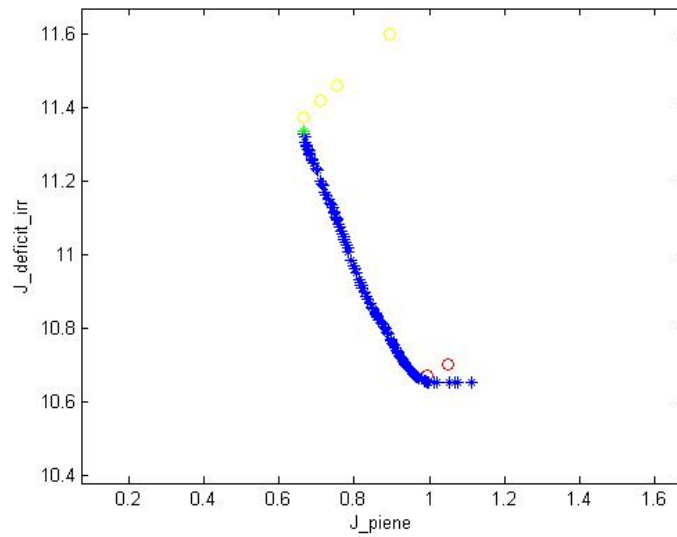


Figura 3.16: Confronto tra Caso SI (soluzioni blu) e NO (soluzioni verdi). I cerchi gialli corrispondono alle inizializzazioni in comune, mentre quelli rossi a quelle del solo Caso SI (regole semplici 1 e 4). In questo caso (con un altro seme) si nota che il Caso NO non è riuscito a esplorare la frontiera.

3.3.3 Simulazione Monte Carlo e confronto con la programmazione dinamica

A fronte delle osservazioni mostrate nei precedenti paragrafi, si valideranno le politiche ottenute nei diversi casi attraverso simulazione Monte Carlo. Questa simulazione, invece che considerare un unico stato iniziale e una sola realizzazione dei disturbi (gli afflussi a_t al serbatoio), come fatto finora, opera un certo numero di simulazioni deterministiche, considerando quindi diverse possibili traiettorie del disturbo. In questo modo la simulazione terrà ampiamente conto della stocasticità dei disturbi (in misura superiore che con una singola simulazione deterministica con h limitato, si veda paragrafo 1.4.2). La Monte Carlo, ovviamente, comporta oneri computazionali decisamente superiori rispetto a una simulazione deterministica.

Il test maggiormente interessante riguarda le politiche relative all'ottimizzazione con Caso SI, $pop = 100$ e $gen = 50$ (che sembra poter generare ottimi risultati, paragrafo 3.3.1). In Figura 3.17 viene pertanto mostrato un esempio di frontiera ottenuta con tale popolazione mediante simulazione Monte Carlo, e viene confrontata con i risultati della SDP. Questo confronto è molto significativo perchè, come visto nella sezione 1.5, la programmazione dinamica è uno strumento molto potente che, per sua stessa natura, garantisce l'ottimo (è un algoritmo *greedy*).

Dalla Figura 3.17 si può notare come i risultati dell'algoritmo evolutivo siano assolutamente paragonabili a quelli della programmazione dinamica. La differenza principale tra questi due metodi, tuttavia, riguarda il numero di punti che compongono la frontiera. Infatti, se con la programmazione dinamica per ottenere un numero di punti pari a 7, piuttosto esiguo, sono state necessarie 7 ottimizzazioni, con l'algoritmo evolutivo una singola ottimizzazione ha permesso di generare un elevato numero di punti efficienti. In definitiva, da questo test si può confermare che l'algoritmo evolutivo NsgaII utilizzato è in grado di approssimare notevolmente meglio una frontiera di Pareto rispetto alla Programmazione Dinamica, pur non incorrendo in prestazioni inferiori.

Rispetto alla stessa frontiera relativa alla simulazione deterministica (Figura 3.8) si può notare la mancanza di alcuni punti: ciò è dovuto all'approssimazione della simulazione deterministica con orizzonte h limitato. Alcune politiche ritenute ottime con tale simulazione (utilizzata in fase di ottimizzazione), infatti, potrebbero risultare inefficienti attraverso la Monte Carlo.

A questo punto, è utile validare con la Monte Carlo l'effettiva importanza delle regole semplici e del parametro gen . Inizialmente, lo stesso esperimento appena utilizzato (Caso SI, $pop = 100$ e

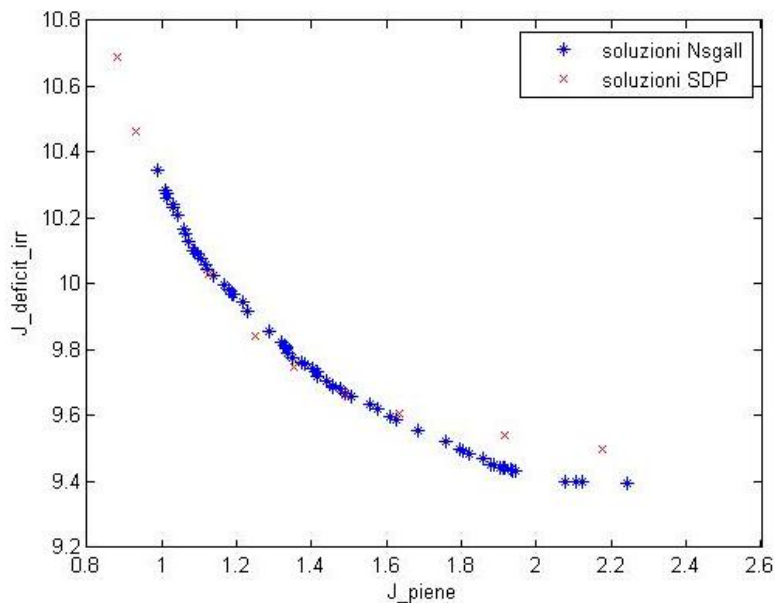


Figura 3.17: Confronto tra la frontiera ottenuta mediante simulazione MC dei risultati dell’algoritmo evolutivo ($pop = 100$, $gen = 50$), e quelli della SDP.

$gen = 50$) sarà paragonato al corrispettivo nel Caso NO (ovviamente con lo stesso seme). In Figura 3.18 ci sono tre esempi (esaustivi dei totali analizzati con vari semi) che mostrano nettamente la superiorità del Caso SI.

Una cosa interessante da notare è che gli errori di approssimazione della simulazione deterministica si riflettono in modo negativo soprattutto sul Caso NO: basti fare un confronto tra la Figura 3.14 e 3.18 (a), che sono proprio relative allo stesso seme, per rendersi conto della questione. Questo significa che grazie alle regole semplici l’algoritmo è in grado di trovare soluzioni oggettivamente migliori in una più vasta gamma di condizioni di simulazione. Nella Figura 3.18 si può osservare che al contrario di quanto emerso in precedenza (paragrafo 3.3.1), con la MC la regola semplice 1 non domina più la 4. Queste considerazioni mostrano che la simulazione deterministica può erroneamente valutare le prestazioni delle soluzioni.

In Figura 3.19 vengono presentati i risultati dell’ottimizzazione nel Caso SI con $pop = 100$ nei casi $gen = 50$ e $gen = 7$. Anche in questo caso viene confermato l’impatto che hanno le generazioni sulla bontà della frontiera.

In definitiva, la simulazione Monte Carlo ha confermato, e anzi addirittura

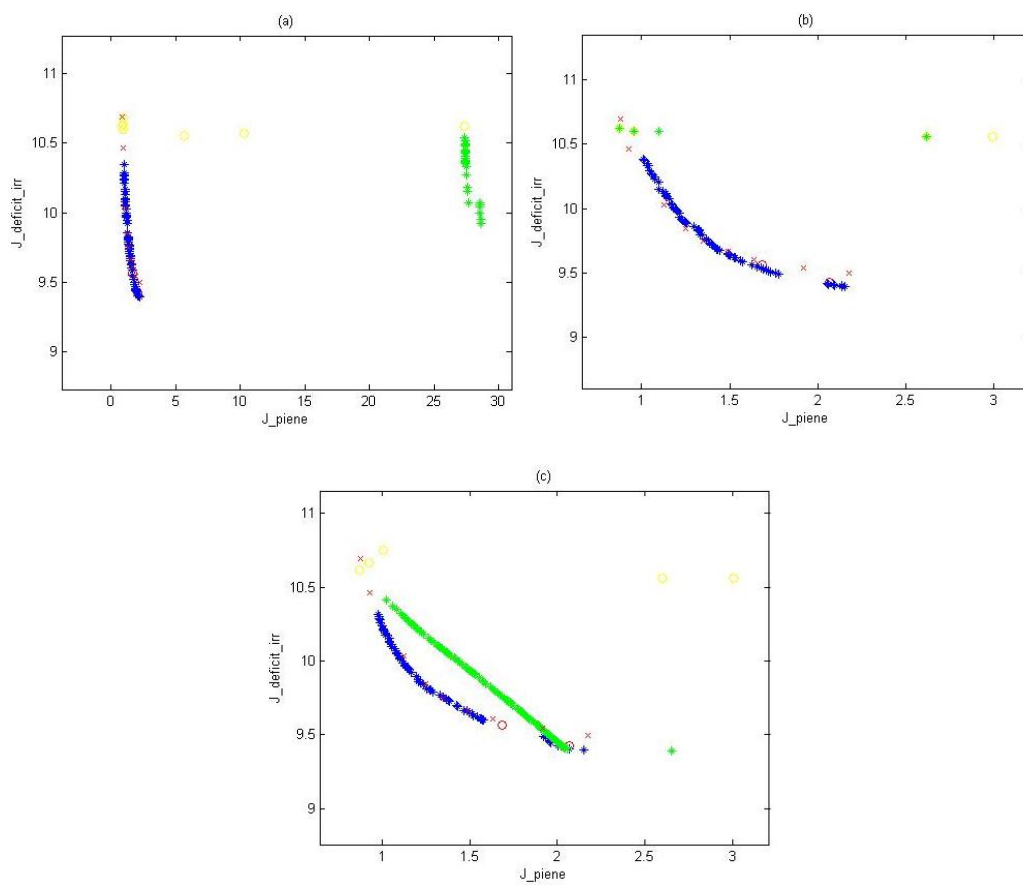


Figura 3.18: Esempi con alcuni semi di confronti tra Caso SI (blu) e NO (verde) con simulazione MC, $pop = 100$ e $gen = 50$. I cerchi gialli sono le inizializzazioni in comune (casuali), quelle rosse le regole semplici.

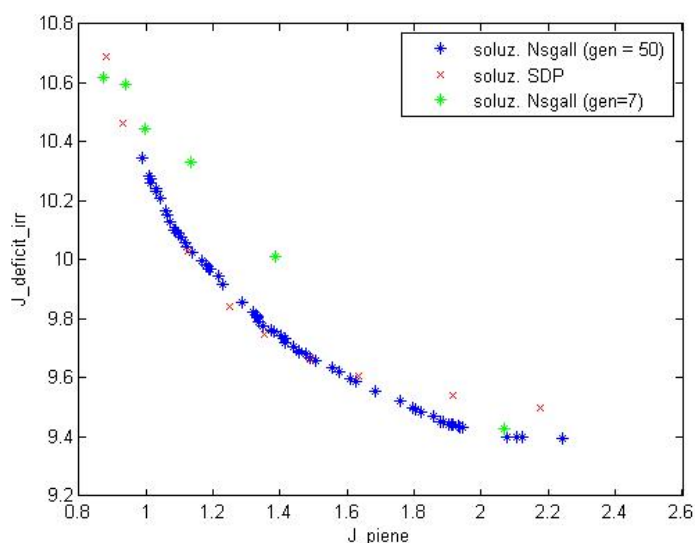


Figura 3.19: Confronto con simulazione MC tra le popolazioni ottenute con $gen = 50$ e $gen = 7$ ($pop = 100$).

tura rafforzato, le considerazioni precedentemente effettuate.

Tuttavia, le differenze riscontrate tra alcuni risultati della simulazione deterministica e della Monte Carlo, evidenziano i limiti legati alla simulazione deterministica, che rimane una forte approssimazione anche con valori di h molto elevati.

3.3.4 Orizzonte temporale h

A questo punto, si è ritenuto interessante effettuare una breve analisi di sensitività sull'orizzonte temporale della serie di afflussi h , effettuando varie simulazioni sulla popolazione ottenuta nel Caso SI, $pop = 100$ e $gen = 50$. La simulazione deterministica, come evidenziato anche nel paragrafo precedente, costituisce un'approssimazione e quindi il risultato può dipendere dallo stato iniziale e dalla serie di afflussi utilizzata.

Come ci si aspettava, si è riscontrato che con il crescere di h i valori degli obiettivi associati alle stesse soluzioni cambiano, e quindi anche la loro bontà rispetto alla frontiera. In effetti, anche utilizzando tutta la serie disponibile ($h = 5000$) la forma della funzione obiettivo (1.20) continua comunque a rimanere un'approssimazione.

Sebbene questa analisi sia comunque dipendente dalla serie di afflussi utilizzata, si è osservato che a partire da valori di h compresi tra 200 e 550 in avanti la forma della frontiera e la bontà delle soluzioni comincia a

stabilizzarsi e quindi (almeno per questo caso di Test) un valore di $h = 550$ si può ritenere ragionevolmente un'approssimazione accettabile (nonostante, come evidenziato nel precedente paragrafo, ci siano comunque delle differenze rispetto alla MC). Anche i valori assoluti degli obiettivi assunti dalle soluzioni iniziano a convergere verso $h = 200$.

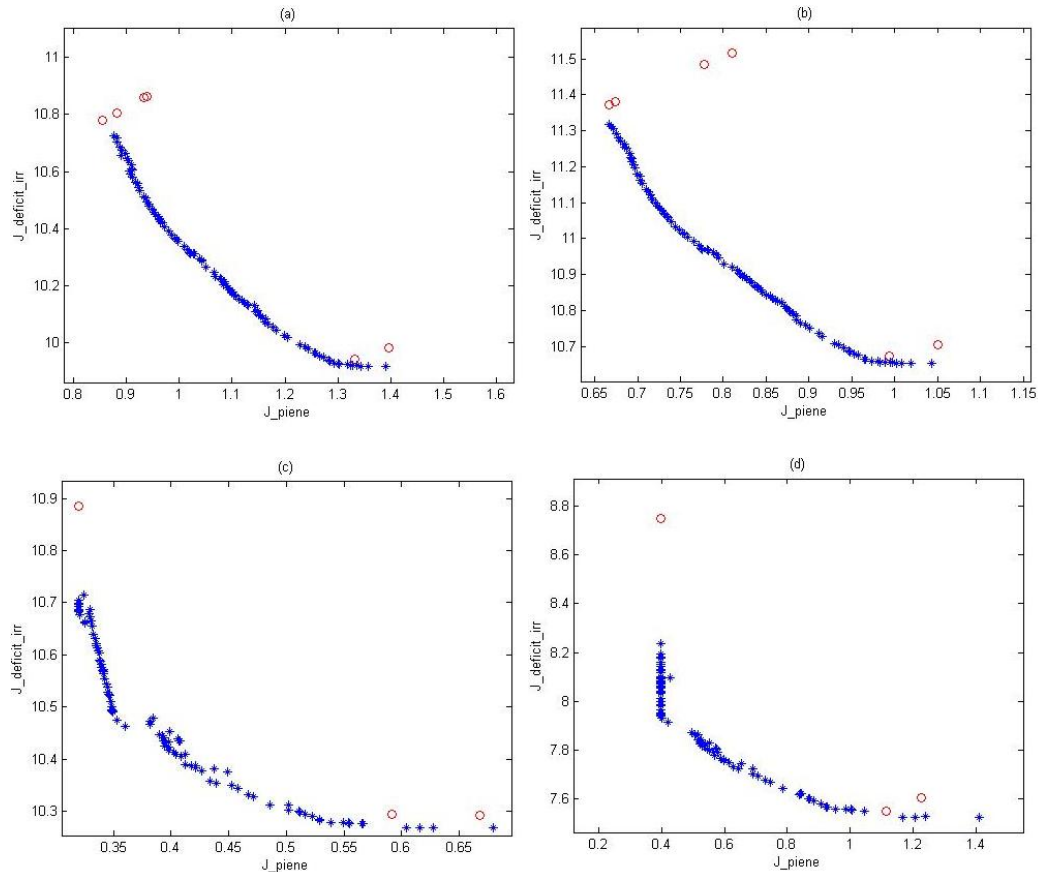


Figura 3.20: Frontiera di Pareto della popolazione Caso SI, $pop = 100$ e $gen = 50$ simulata con $h = 4990$ (a), $h = 550$ (b), $h = 100$ (c) e $h = 30$ (d). In blu le soluzioni, in cerchi rossi le inizializzazioni.

In Figura 3.20 si nota come le frontiere con $h = 4990$ (quasi tutta la serie) e $h = 550$ siano molto simili: $h = 550$ può costituire una buona approssimazione. Già con $h = 100$ la frontiera cambia vistosamente. Il numero di soluzioni efficienti N_{soleff} cala da valori prossimi a 100 (tutta la popolazione) riscontrati con $h = 4990$ e $h = 550$ (si ricorda che l'ottimizzazione è avvenuta con $h = 500$) a $N_{soleff} = 35$ con $h = 100$: quasi il 70% di differenza nei risultati. Nel caso (d) ($h = 30$), addirittura, anche il valore assoluto

degli obiettivi cambia sensibilmente (soprattutto dell'obiettivo J_{irr} , che cala di quasi il 30%).

In questo caso il sistema è stazionario, ma in un caso reale (in cui il periodo T è tipicamente pari a un anno) anche utilizzare “solo” $h = 200$ significherebbe che occorrerebbero 200 anni di dati sugli afflussi. Ciò è ovviamente impensabile, sia a livello computazionale che di disponibilità dei dati stessi.

Pertanto, invece del valore degli obiettivi, si sono osservate le leggi di controllo. Ripetendo l'ottimizzazione con $pop = 100$ e $gen = 50$ abbassando progressivamente h (e in varie condizioni di simulazione), si è valutata la presenza di eventuali leggi di controllo non efficienti (non in linea con la Figura 3.3).

Leggi di controllo leggermente non conformi alle aspettative si possono

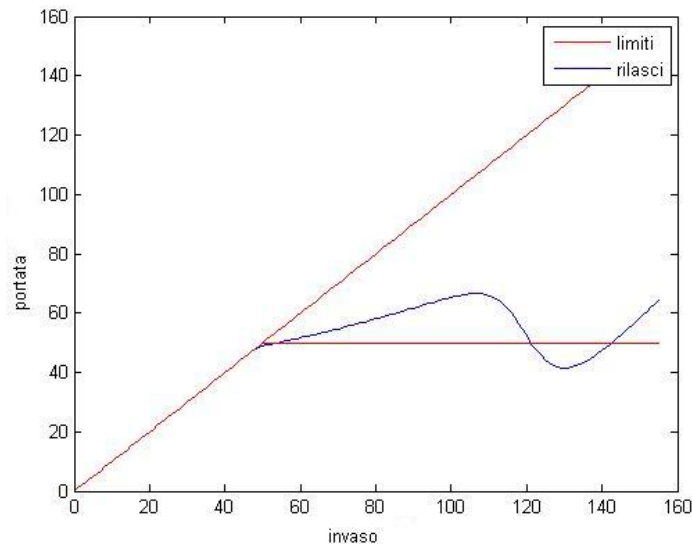


Figura 3.21: Esempio di legge di controllo con parti inefficienti.

trovare, in relazione alle particolari condizioni di simulazione, anche con h superiori a 100, come nel caso della Figura 3.21.

Definire un valore limite di h risulta estremamente complesso, dal momento che più tale valore è ridotto, più è forte l'influenza che esercitano i particolari afflussi considerati e lo stato iniziale sull'ottimizzazione, generando risultati fortemente variabili. Modificando varie volte le condizioni di simulazione, si è notato che leggi particolarmente insensate ricadono con frequenza a partire da valori di h compresi tra 2 e 6, e in qualche raro caso anche sotto $h = 15$. Esempi di queste leggi di controllo sono mostrate in Figura 3.22.

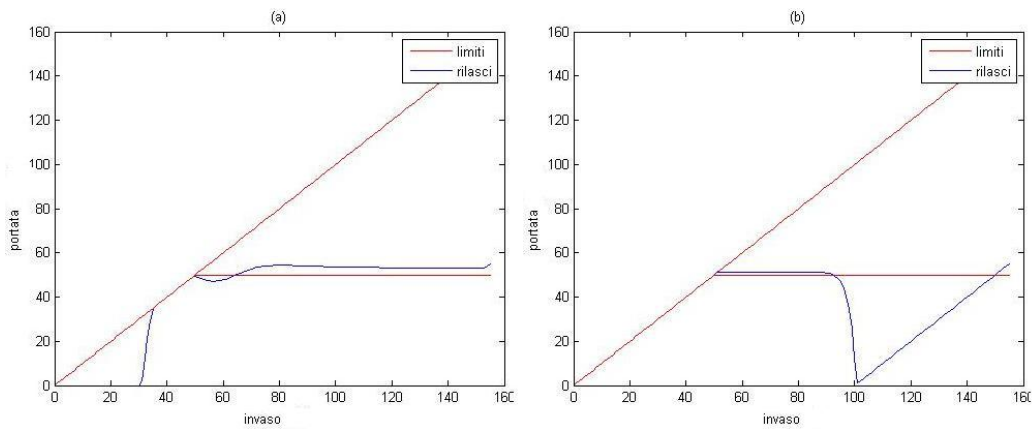


Figura 3.22: Esempi di leggi di controllo insensate (confrontare con Figura 3.3), ottenute con notevole frequenza per limitati valori dell'orizzonte temporale h .

3.3.5 Prime conclusioni

L'obiettivo di questa prima fase è stato ampiamente raggiunto; si possono infatti già effettuare alcune importanti considerazioni. Innanzitutto, l'algoritmo evolutivo è in grado di convergere efficacemente alla frontiera, approssimandola in modo esauriente con tempi di calcolo contenuti (si ricordi che per ottenere la stessa frontiera di Figura 3.8, con la programmazione dinamica occorrerebbe risolvere il problema 100 volte).

Attendendo una generalizzazione dei risultati attraverso l'analisi su un elevato numero di semi, si può avanzare inoltre che la grandezza della popolazione ha un impatto limitato sulla soluzione, mentre critico è il ruolo delle generazioni. In aggiunta, sembra già a un primo impatto che il Caso SI comporti benefici concreti. Il Caso NO, infatti, non sempre è in grado di ottenere buoni risultati, mentre il Caso SI sembra portare benefici all'ottimizzazione e poter sempre approssimare la frontiera in modo esauriente (con gen sufficiente).

Infine, anche se in questo caso di Test è possibile utilizzare orizzonti di simulazione h molto elevati per via della stazionarietà, le analisi effettuate mostrano un primo limite legato alla simulazione deterministica, che potrebbe divenire influente in un caso reale, qualora la disponibilità di dati non fosse sufficiente.

3.4 Generalizzazione dei risultati

Avendo avuto riscontri positivi sul potenziale dell'algoritmo, si cercherà ora di ottenere i risultati definitivi sull'influenza che hanno i principali parametri considerati.

Come anticipato, per la questione del seme (sezione 2.5) questa fase prevede l'esecuzione di un numero sufficientemente elevato di esperimenti di ottimizzazione con seme diverso, per ogni esperimento. In particolare, il numero di semi attraverso cui si intende generalizzare i risultati è pari a 10. La bontà dei risultati verrà giudicata in base ai valori medi, sulle dieci ottimizzazioni, degli indicatori introdotti nella sezione 2.6.

Nel dettaglio, questa fase verrà realizzata nel seguente modo:

- viene fissato $pop = 100$ e un numero iniziale di generazioni $gen = 5$;
 - viene effettuata l'ottimizzazione nel Caso SI per dieci valori diversi del seme, e si calcola N_{conv} e la media dei valori di HD , N_{soleff} e CD ;
 - se $N_{conv} \geq 7$, cioè almeno 7 ottimizzazioni su 10 hanno raggiunto la convergenza si passa al seguente step 2, altrimenti si ripete il punto precedente con un numero di generazioni incrementato di 1.
2. Si ripete il punto 1 per il caso NO, con il cromosoma inizialmente casuale. Il procedimento è analogo tranne la condizione di terminazione, che è la seguente: il valore di N_{conv} deve essere almeno pari a quello ottenuto al termine del Caso SI (sarà quindi almeno pari a 7), e lo stesso anche per il valore medio di hypervolume. Se questa condizione non è mai verificata, si arresta il procedimento a $gen = 100$;
3. Il procedimento illustrato nei primi due punti viene ripetuto utilizzando $pop = 20$.

Il senso è dunque valutare quante generazioni servano al Caso NO per raggiungere lo stesso numero di successi ottenuti al termine del Caso SI, e le stesse prestazioni in termini di hypervolume medio. Ciò che ci si aspetta è che questo numero di generazioni sia maggiore rispetto al Caso SI. Si noti che alla condizione sul valore di hypervolume non viene posto alcun margine di errore per tenere conto del suo carattere stocastico (sezione 2.6). Ci si attende infatti che calcolando la media di 10 risultati di hypervolume, i contributi positivi e negativi della casualità nella generazione dei punti si compensino.

Le Tabelle 3.4 e 3.5 mostrano i risultati ottenuti.

Tabella 3.4: Risultati medi relativi a $pop = 100$.

Caso SI/NO	Pop	Gen	HD medio	N_{soleff} medio	CD media	N_{conv}
SI	100	5	0,4805	6,1	0,0502	0
SI	100	6	0,4806	8,9	0,0496	0
SI	100	7	0,4808	12,9	0,0476	0
SI	100	8	0,4809	16,4	0,0449	0
SI	100	9	0,4809	23,5	0,0441	1
SI	100	10	0,4811	30,5	0,0414	1
SI	100	11	0,4812	40,3	0,0397	1
SI	100	12	0,4812	52,4	0,0388	3
SI	100	13	0,4813	59	0,0381	4
SI	100	14	0,4814	65,7	0,0386	6
SI	100	15	0,4814	70,1	0,0384	6
SI	100	16	0,4815	75,6	0,0375	6
SI	100	17	0,4815	80	0,0380	7
NO	100	5	0,4613	1,8	0,0600	0
NO	100	6	0,4615	2,2	0,0589	0
NO	100	7	0,4643	2,7	0,0475	0
NO	100	8	0,4671	4	0,0481	0
NO	100	9	0,4681	6	0,0475	0
NO	100	10	0,4685	11,1	0,0464	0
NO	100	11	0,4687	20,7	0,0434	0
NO	100	12	0,4704	34,3	0,0399	2
NO	100	13	0,4704	37,7	0,0411	3
NO	100	14	0,4707	45,5	0,0413	4
NO	100	15	0,4707	50,8	0,0408	5
NO	100	16	0,4707	50,7	0,0409	4
NO	100	17	0,4708	51	0,0409	5
NO	100	18	0,4708	51	0,0420	5
NO	100	19	0,4708	51	0,0420	5
NO	100	20	0,4708	51	0,0419	5
NO	100	21	0,4708	51	0,0418	5
NO	100	22	0,4708	51	0,0419	5
NO	100	23	0,4708	51	0,0419	5
NO	100	24	0,4708	51	0,0420	5
NO	100	25	0,4708	51	0,0420	5
NO	100	26	0,4708	51	0,0410	5
NO	100	27	0,4708	51	0,0411	5
NO	100	28	0,4708	51	0,0411	5

Caso SI/NO	Pop	Gen	HD medio	N_{soleff} medio	CD media	N_{conv}
NO	100	29	0,4708	51	0,0411	5
NO	100	30	0,4708	51	0,0418	5
NO	100	31	0,4708	51	0,0418	5
NO	100	32	0,4708	51	0,0415	5
NO	100	33	0,4708	51	0,0416	5
NO	100	34	0,4708	51	0,0416	5
NO	100	35	0,4708	51	0,0416	5
NO	100	36	0,4708	51	0,0416	5
NO	100	37	0,4708	51	0,0415	5
NO	100	38	0,4708	51	0,0416	5
NO	100	39	0,4708	51	0,0416	5
NO	100	40	0,4708	51	0,0414	5
NO	100	41	0,4708	51	0,0412	5
NO	100	42	0,4708	51	0,0413	5
NO	100	43	0,4708	51	0,0413	5
NO	100	44	0,4708	51	0,0413	5
NO	100	45	0,4708	51	0,0413	5
NO	100	46	0,4708	51	0,0413	5
NO	100	47	0,4708	51	0,0411	5
NO	100	48	0,4708	51	0,0411	5
NO	100	49	0,4708	51	0,0412	5
NO	100	50	0,4708	51	0,0411	5
NO	100	51	0,4708	51	0,0410	5
NO	100	52	0,4708	51	0,0411	5
NO	100	53	0,4708	51	0,0410	5
NO	100	54	0,4708	51	0,0411	5
NO	100	55	0,4708	50,9	0,0411	5
NO	100	56	0,4708	50,9	0,0410	5
NO	100	57	0,4708	50,8	0,0409	5
NO	100	58	0,4708	50,9	0,0409	5
NO	100	59	0,4708	50,9	0,0410	5
NO	100	60	0,4708	50,9	0,0410	5
NO	100	61	0,4708	50,8	0,0410	5
NO	100	62	0,4708	50,9	0,0410	5
NO	100	63	0,4708	50,9	0,0409	5
NO	100	64	0,4708	50,9	0,0418	5
NO	100	65	0,4708	50,9	0,0416	5
NO	100	66	0,4708	50,9	0,0416	5
NO	100	67	0,4708	50,9	0,0417	5

Caso SI/NO	Pop	Gen	HD medio	N_{soleff} medio	CD media	N_{conv}
NO	100	68	0,4708	50,8	0,0416	5
NO	100	69	0,4708	50,9	0,0417	5
NO	100	70	0,4708	50,9	0,0417	5
NO	100	71	0,4708	50,9	0,0415	5
NO	100	72	0,4708	50,9	0,0415	5
NO	100	73	0,4708	50,9	0,0414	5
NO	100	74	0,4708	50,9	0,0414	5
NO	100	75	0,4708	50,9	0,0414	5
NO	100	76	0,4708	50,9	0,0414	5
NO	100	77	0,4708	50,9	0,0414	5
NO	100	78	0,4708	50,9	0,0424	5
NO	100	79	0,4708	50,9	0,0425	5
NO	100	80	0,4708	50,9	0,0425	5
NO	100	81	0,4708	50,9	0,0425	5
NO	100	82	0,4708	50,9	0,0432	5
NO	100	83	0,4708	50,9	0,0432	5
NO	100	84	0,4708	50,9	0,0433	5
NO	100	85	0,4708	50,9	0,0432	5
NO	100	86	0,4708	50,9	0,0433	5
NO	100	87	0,4708	50,9	0,0433	5
NO	100	88	0,4708	50,9	0,0433	5
NO	100	89	0,4708	50,9	0,0432	5
NO	100	90	0,4708	50,9	0,0432	5
NO	100	91	0,4708	50,9	0,0432	5
NO	100	92	0,4708	50,9	0,0431	5
NO	100	93	0,4708	50,9	0,0431	5
NO	100	94	0,4708	50,8	0,0431	5
NO	100	95	0,4708	50,7	0,0430	5
NO	100	96	0,4708	50,8	0,0430	5
NO	100	97	0,4708	50,8	0,0429	5
NO	100	98	0,4708	50,9	0,0429	5
NO	100	99	0,4708	50,9	0,0428	5

Tabella 3.5: Risultati medi relativi a $pop = 20$.

Caso SI/NO	Pop	Gen	HD medio	$N_{soleff\ medio}$	CD media	N_{conv}
SI	20	5	0,4805	6,6	0,2663	0
SI	20	6	0,4805	7,8	0,2588	1
SI	20	7	0,4806	10	0,2503	1
SI	20	8	0,4806	10,4	0,2471	4
SI	20	9	0,4807	10,8	0,2472	4
SI	20	10	0,4807	10,8	0,2433	4
SI	20	11	0,4808	11,4	0,2434	4
SI	20	12	0,4808	12,7	0,2318	4
SI	20	13	0,4808	13,1	0,2313	5
SI	20	14	0,4808	13,5	0,2305	5
SI	20	15	0,4809	13,7	0,2292	6
SI	20	16	0,4809	13,9	0,2322	6
SI	20	17	0,4810	14	0,2275	6
SI	20	18	0,4810	14,3	0,2292	6
SI	20	19	0,4810	14,8	0,2270	7
NO	20	5	0,4615	2	0,2543	0
NO	20	6	0,4641	2,7	0,2423	0
NO	20	7	0,4642	3,7	0,2507	1
NO	20	8	0,4643	4,1	0,2317	1
NO	20	9	0,4643	5,2	0,2145	1
NO	20	10	0,4644	5,8	0,2038	2
NO	20	11	0,4644	6,8	0,1952	3
NO	20	12	0,4645	6,8	0,1966	3
NO	20	13	0,4645	6,8	0,1956	3
NO	20	14	0,4649	6,8	0,1885	3
NO	20	15	0,4649	6,8	0,1897	3
NO	20	16	0,4649	6,8	0,1873	3
NO	20	17	0,4649	6,8	0,2019	3
NO	20	18	0,4650	6,8	0,2017	3
NO	20	19	0,4650	6,8	0,2005	3
NO	20	20	0,4650	6,8	0,2010	3
NO	20	21	0,4650	6,8	0,2025	3
NO	20	22	0,4650	6,8	0,2127	3
NO	20	23	0,4650	6,8	0,2087	3
NO	20	24	0,4650	6,8	0,2112	3
NO	20	25	0,4650	6,8	0,2115	3
NO	20	26	0,4650	6,8	0,2117	3

Caso SI/NO	Pop	Gen	HD medio	N_{soleff} medio	CD media	N_{conv}
NO	20	27	0,4651	6,8	0,2083	3
NO	20	28	0,4651	6,8	0,2066	3
NO	20	29	0,4650	6,8	0,2048	3
NO	20	30	0,4650	6,8	0,2066	3
NO	20	31	0,4650	6,8	0,2048	3
NO	20	32	0,4650	6,7	0,2064	3
NO	20	33	0,4650	6,6	0,2063	3
NO	20	34	0,4651	6,7	0,2076	3
NO	20	35	0,4651	6,8	0,2041	3
NO	20	36	0,4650	6,8	0,2069	3
NO	20	37	0,4650	6,7	0,2041	3
NO	20	38	0,4650	6,7	0,2081	3
NO	20	39	0,4650	6,8	0,2131	3
NO	20	40	0,4650	6,7	0,2117	3
NO	20	41	0,4650	6,7	0,2111	3
NO	20	42	0,4650	6,7	0,2110	3
NO	20	43	0,4650	6,7	0,2121	3
NO	20	44	0,4650	6,8	0,2088	3
NO	20	45	0,4650	6,8	0,2106	3
NO	20	46	0,4651	6,8	0,2138	3
NO	20	47	0,4651	6,8	0,2138	3
NO	20	48	0,4651	6,8	0,2123	3
NO	20	49	0,4651	6,8	0,2123	3
NO	20	50	0,4651	6,8	0,2109	3
NO	20	51	0,4651	6,8	0,2100	3
NO	20	52	0,4651	6,8	0,2115	3
NO	20	53	0,4651	6,8	0,2112	3
NO	20	54	0,4651	6,8	0,2081	3
NO	20	55	0,4651	6,8	0,2106	3
NO	20	56	0,4651	6,8	0,2125	3
NO	20	57	0,4651	6,8	0,2128	3
NO	20	58	0,4651	6,8	0,2146	3
NO	20	59	0,4651	6,8	0,2135	3
NO	20	60	0,4651	6,8	0,2135	3
NO	20	61	0,4651	6,8	0,2123	3
NO	20	62	0,4651	6,8	0,2092	3
NO	20	63	0,4651	6,8	0,2090	3
NO	20	64	0,4651	6,8	0,2117	3
NO	20	65	0,4651	6,8	0,2131	3

Caso SI/NO	Pop	Gen	HD medio	N_{soleff} medio	CD media	N_{conv}
NO	20	66	0,4651	6,7	0,2127	3
NO	20	67	0,4651	6,7	0,2157	3
NO	20	68	0,4651	6,7	0,2157	3
NO	20	69	0,4650	6,7	0,2158	3
NO	20	70	0,4650	6,7	0,2158	3
NO	20	71	0,4650	6,6	0,2104	3
NO	20	72	0,4650	6,7	0,2161	3
NO	20	73	0,4650	6,8	0,2141	3
NO	20	74	0,4650	6,8	0,2139	3
NO	20	75	0,4650	6,8	0,2115	3
NO	20	76	0,4650	6,8	0,2162	3
NO	20	77	0,4650	6,8	0,2139	3
NO	20	78	0,4650	6,8	0,2119	3
NO	20	79	0,4650	6,8	0,2137	3
NO	20	80	0,4651	6,8	0,2155	3
NO	20	81	0,4651	6,8	0,2137	3
NO	20	82	0,4651	6,8	0,2118	3
NO	20	83	0,4651	6,8	0,2121	3
NO	20	84	0,4651	6,8	0,2143	3
NO	20	85	0,4651	6,8	0,2109	3
NO	20	86	0,4651	6,8	0,2153	3
NO	20	87	0,4651	6,8	0,2131	3
NO	20	88	0,4651	6,8	0,2129	3
NO	20	89	0,4651	6,7	0,2111	3
NO	20	90	0,4651	6,7	0,2093	3
NO	20	91	0,4651	6,7	0,2100	3
NO	20	92	0,4651	6,8	0,2097	3
NO	20	93	0,4651	6,8	0,2095	3
NO	20	94	0,4651	6,8	0,2122	3
NO	20	95	0,4651	6,8	0,2133	3
NO	20	96	0,4651	6,8	0,2148	3
NO	20	97	0,4651	6,8	0,2126	3
NO	20	98	0,4651	6,8	0,2106	3
NO	20	99	0,4651	6,8	0,2116	3
NO	20	99	0,4651	6,8	0,2116	3

Come si può vedere, sia per quanto riguarda i valori medi di hypervolume, sia per il numero medio di soluzioni efficienti, che per il numero di ottimizzazioni che hanno raggiunto la convergenza, a parità di generazioni il Caso SI

funziona nettamente meglio del caso NO. Inoltre, i valori di hypervolume del caso NO, anche dopo moltissime generazioni, non riescono mai a raggiungere quelli del Caso SI già nelle prime generazioni. Questo è probabilmente dovuto soprattutto al contributo di quei run che, anche con molte generazioni, nel Caso NO non riescono mai a generare buoni risultati, producendo bassi valori di hypervolume. Infine, si può notare che nel Caso SI i valori medi di hypervolume sono piuttosto simili sia per $pop = 100$ che per $pop = 20$, mentre nel Caso NO si notano valori leggermente più elevati nella situazione in cui $pop = 100$. Si deduce quindi che il Caso SI risulta meno sensibile al parametro pop .

In Figura 3.23 e 3.24 sono presentati dei riassunti grafici dei risultati appena raccolti, in cui si vede in modo immediato la superiorità del Caso SI.

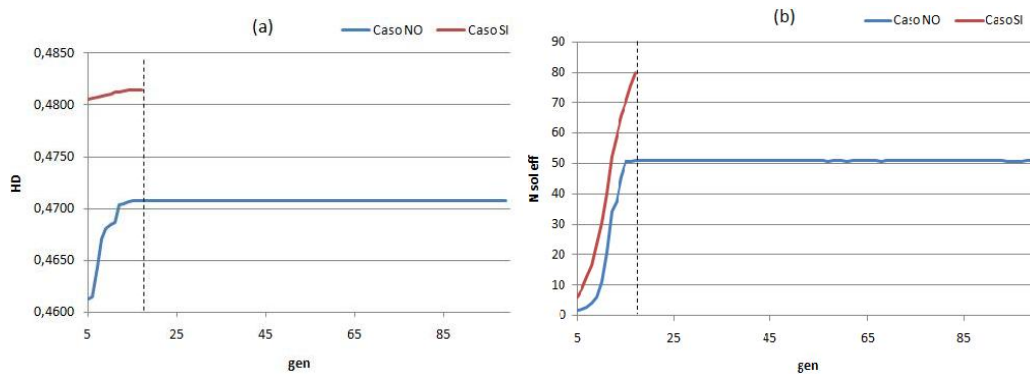


Figura 3.23: Andamento dei valori medi di Hypervolume HD (a) e N_{soleff} (b) con le generazioni nei Casi SI e NO. $pop = 100$.

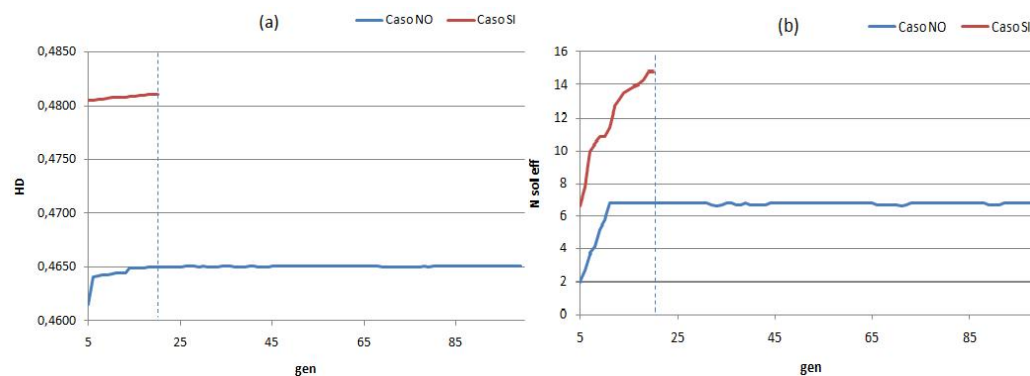


Figura 3.24: Andamento dei valori medi di Hypervolume HD (a) e N_{soleff} (b) con le generazioni nei Casi SI e NO. $pop = 20$.

3.5 Validazione

In questa sezione si intende validare i risultati ottenuti. A valle di quanto mostrato nella sezione 3.4, l'esperimento che ha dato i migliori risultati con il minore costo computazionale è:

Caso SI, $pop = 100$, $gen = 17$.

Questo esperimento verrà pertanto validato sottoponendolo a ottimizzazioni con differenti stati iniziali e traiettorie dell'afflusso (con h sempre pari a 550). In particolare, saranno considerati 9 stati iniziali s_0 uniformemente distribuiti in tutto il range dei valori ammissibili dell'invaso s , e 9 traiettorie dell'afflusso a_t , per un totale di 81 combinazioni. A valle di queste simulazioni, saranno calcolati, analogamente a come fatto nella sezione precedente, il valore medio di hypervolume HD , il valore medio delle soluzioni efficienti N_{soleff} il valore medio della crowding distance CD e infine il numero totale N_{conv} di soluzioni arrivate alla convergenza.

I risultati hanno fornito le seguenti indicazioni:

HD medio	N_{soleff} medio	CD media	N_{conv}
0,4862	79,3	0,0399	47

Questi valori possono essere quindi confrontati con quelli della Tabella 3.4, che erano:

HD medio	N_{soleff} medio	CD media	N_{conv}
0,4815	80	0,0380	7

Come si può vedere, in tutti i primi tre indicatori i risultati sono coerenti. Per quanto riguarda l'indicatore N_{conv} , i due valori per essere confrontati possono essere trasposti in percentuale. Così, il valore ottenuto dalla validazione indica che circa il 58% delle ottimizzazioni ha raggiunto la convergenza, leggermente meno del 70% ottenuto nella sezione precedente, ma comunque piuttosto elevato e in linea con le aspettative (considerando il basso numero di generazioni).

In definitiva, la validazione ha confermato i risultati precedentemente ottenuti, e pertanto la configurazione Caso SI, $pop = 100$, $gen = 17$ risulta ottimale.

3.6 Conclusioni sul caso di Test

Il caso di Test esaminato in questo capitolo ha fornito parecchie informazioni interessanti.

1. L'algoritmo è in grado di raggiungere la frontiera di Pareto con un numero non eccessivamente elevato di generazioni (quindi con tempi di calcolo accettabili). Le prestazioni sono assolutamente paragonabili con quelle della Programmazione Dinamica, con il vantaggio di un'ottima approssimazione della frontiera, con un numero elevato di punti ben distribuiti, in un singolo run. Si è notato inoltre come l'algoritmo già a partire dall'inizializzazione è in grado di esplorare ottimamente lo spazio dei parametri in virtù del suo carattere stocastico, e di convergere nelle generazioni a un range limitato di parametri ottimi.
2. Le generazioni hanno un impatto notevole sulla qualità delle soluzioni di una popolazione.
Grazie alle regole semplici, nel Caso SI è più probabile avere soluzioni migliori anche a generazioni più basse.
3. La popolazione ha invece un impatto inferiore. Nel Caso NO la sua importanza è comunque abbastanza rilevante; lo è meno nel Caso SI.
4. L'effetto dell'introduzione delle parametrizzazioni relative alle regole semplici nell'inizializzazione genera concreti benefici, migliorando le prestazioni a generazioni più basse, e garantendo buoni risultati anche con popolazioni meno numerose. Non sempre (in relazione al seme) il Caso NO è in grado di convergere alla frontiera (nemmeno con tante generazioni), al contrario del Caso SI che con il crescere delle generazioni sembra poter sempre ottenere ottimi risultati. Inoltre, i valori medi di hypervolume riscontrati nel caso di inizializzazione puramente casuale sono sempre inferiori anche con un gran numero di generazioni, in parte a causa proprio di quelle ottimizzazioni che nel Caso NO non sono in grado di convergere alla frontiera. Le differenze riscontrate nei valori di hypervolume, comunque, non sono mai particolarmente elevate a causa dei vincoli del sistema, che fanno in modo che anche nei casi peggiori ci siano molte soluzioni casuali generate nell'inizializzazione corrispondenti a un punto ottimo (prossimo alla legge di controllo di massimo rilascio). Per lo stesso motivo, si riscontrano piccole differenze nei valori di hypervolume con il procedere delle generazioni, al contrario di quello che dovrebbe essere l'andamento tipico (crescente con saturazione). Nel Caso SI, addirittura, l'andamento di hypervolume è pressochè

costante, proprio per il fatto che le regole semplici inserite sono già in grado di portare la frontiera molto vicina all'ottimo.

5. L'orizzonte temporale h utilizzato in questo caso di Test non sembra poter generare instabilità nei risultati. Tuttavia, alcune valutazioni effettuate su valori di h piuttosto bassi fanno intuire un possibile limite legato alla simulazione deterministica.

Capitolo 4

Hoa Binh

A fronte dei risultati ottenuti nella fase di Test, si sfrutteranno ora gli strumenti utilizzati nel presente lavoro per la progettazione di una serie di politiche di regolazione del serbatoio Hoa Binh, in Vietnam.

Inizialmente sarà inquadrato il sistema, e verrà descritto nelle sue principali componenti (sezione 4.1).

Successivamente, saranno presentati i risultati ottenuti con la simulazione dinamica deterministica dal gruppo di Modellistica e gestione dei sistemi ambientali del Dipartimento di Elettronica e Informazione del Politecnico di Milano (sezione 4.3).

Sarà quindi mostrato il lavoro di ottimizzazione vero e proprio svolto dal sottoscritto. Esso è composto dalla definizione delle regole semplici, dall'addestramento della rete neurale e dalla conseguente scelta dell'architettura; e successivamente dall'uso dell'algoritmo evolutivo, i cui risultati verranno elaborati e analizzati. La procedura sarà ripetuta utilizzando ingressi diversi della rete neurale: prima la sola coppia $I_t = [s_t, t]$ (sezione 4.4) e successivamente introducendo le informazioni sugli afflussi $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$ (sezione 4.5); per una descrizione più approfondita del procedimento vedere la sezione 4.2. A livello operativo si utilizzerà l'algoritmo NsgaII su piattaforma C prodotto dal *Kanpur Genetic Algorithms Laboratory* (Kan (2005)), manipolato da me per introdurre le personali innovazioni e modifiche, e per adattarlo al simulatore. Quest'ultimo, su piattaforma C++, è prodotto da Marcello Restelli del Dipartimento di Elettronica e Informazione.

4.1 Descrizione del sistema

Il Bacino del Fiume Rosso (Figura 4.1) è il secondo bacino più grande del Vietnam. Esso occupa una superficie totale di 169000 km², di cui il 51%

(86600 km²) in Vietnam, il 49% (81240 km²) in Cina e il resto in Laos. I tre principali affluenti sono il fiume Da, il fiume Thao e il fiume Lo. Di questi, quello che fornisce il maggior contributo di acqua è il fiume Da. Il bacino è caratterizzato dalla presenza di una stagione secca e una stagione particolarmente piovosa, che dura all'incirca da maggio a ottobre, il cui apporto è di circa l'80% della pioggia totale dell'anno (di cui circa il 20% ad agosto). Il risultato di questa situazione è la presenza di grandi eventi di piena in agosto e la mancanza di acqua durante la stagione secca.



Figura 4.1: Il bacino del fiume Rosso, Vietnam.

Sul corso del fiume Da è stato realizzato il serbatoio Hoa Binh proprio per il controllo degli eventi di piena e per immagazzinare acqua per la stagione secca. Inoltre, l'Hoa Binh è utilizzato anche per la produzione idroelettrica: dal 1989 (data di entrata in funzione delle turbine) al 2004 ha prodotto 8 GWh/anno, circa il 15% dell'energia elettrica totale del Vietnam.

Gli interessi principali legati alla regolazione dell'Hoa Binh sono quindi tre:

- l'agricoltura nelle zone irrigate connesse alle traverse di Sontay, a valle del serbatoio. In particolare, è critico il deficit irriguo che si registra durante la stagione secca. L'apporto di acqua a Sontay è dipendente sia dal rilascio dell'Hoa Binh che dagli afflussi dei fiumi Thao e Lo;
- la minimizzazione dei danni nella città di Hanoi, a valle, legati agli eventi di piena che si registrano durante la stagione piovosa (e soprattutto ad agosto);
- la produzione di energia idroelettrica nell'impianto dell'Hoa Binh.

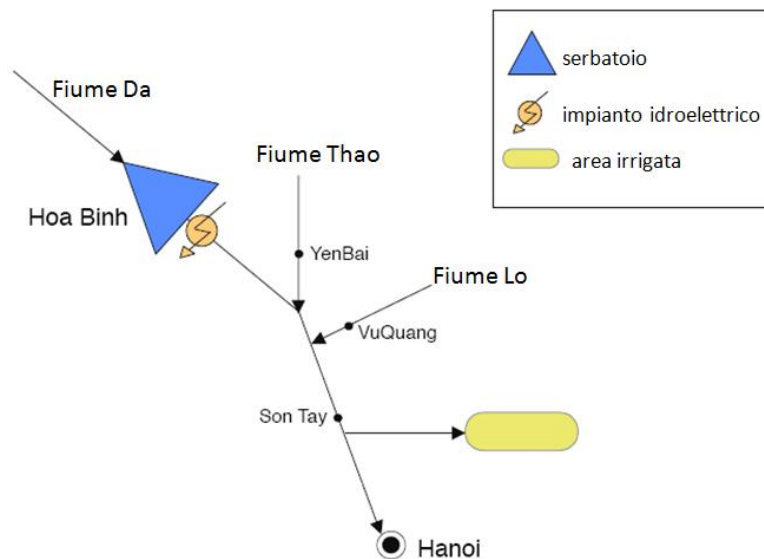


Figura 4.2: Schema del sistema idrico del fiume Rosso.

Attualmente la regolazione dell'Hoa Binh segue due linee distinte. Durante la stagione delle piogge si segue la regola del controllo delle piene, una curva che descrive le decisioni di rilascio in funzione dei livelli dell'acqua nel serbatoio e ad Hanoi. Durante la stagione secca, invece, non c'è una linea precisa. La regolazione è affidata al gestore dell'impianto idroelettrico dell'Hoa Binh che segue la sua personale strategia, anche se deve sottostare al vincolo di minimo rilascio imposto dal Ministero dell'Agricoltura e dello Sviluppo Rurale per tutelare gli interessi agricoli.

L'attuale modello di regolazione si basa quindi su una netta separazione tra stagione secca e stagione piovosa. Sarebbe probabilmente molto più vantaggioso adottare un approccio integrato, che permetta di definire una politica di regolazione su base annuale considerando tutti e tre gli obiettivi contemporaneamente. In questo modo è presumibile il raggiungimento di soluzioni che producano benefici in tutti gli obiettivi considerati contemporaneamente.

4.1.1 Il serbatoio Hoa Binh

Il serbatoio Hoa Binh ha una capacità di invaso di 10,5 miliardi di m^3 . Il range entro cui è possibile operare va da un livello minimo di 80m al livello di 122m. Il serbatoio possiede 8 turbine con capacità massima di $2360 m^3/s$, 12 scarichi di fondo con capacità di massimo rilascio pari a $22000 m^3/s$ e 6 sfioratori con capacità di $14000 m^3/s$. Vista la sua collocazione l'Hoa Binh riceve in afflusso la portata del fiume Da. Il modello del serbatoio può essere

quindi espresso mediante l'equazione di bilancio di massa:

$$s_{t+1} = s_t + q_{t+1}^{HB} - e_{t+1}S(s_t) - r_{t+1}(s_t, q_{t+1}^{HB}, u_t) \quad (4.1)$$

dove s_t è l'invaso al giorno t ; q_{t+1}^{HB} è l'afflusso dal fiume Da; e_{t+1} è l'evaporazione specifica (che segue un andamento periodico lungo l'anno); $S(\cdot)$ è la superficie del serbatoio, che è funzione dell'invaso; e r_{t+1} è il rilascio dal serbatoio. Come descritto nel paragrafo 1.3.1, il rilascio coincide con la decisione di rilascio u_t solo se questa è fisicamente realizzabile.

L'impianto idroelettrico, collocato a valle del serbatoio, ha una capacità installata totale di 1920 MW. La portata attraverso le turbine è data da:

$$q_{t+1}^{turb} = \begin{cases} 0 & \text{se } r_{t+1} \leq q^{\min} \\ \min(r_{t+1}, q^{\max}) & \text{altrimenti} \end{cases} \quad (4.2)$$

in cui r_{t+1} è il rilascio dall' Hoa Binh, e q^{\min} e q^{\max} sono rispettivamente la minima (38 m³/s) e la massima (2360 m³/s) portata turbinabile.

Il salto motore H_{t+1} è dato dalla differenza tra il livello del pelo libero a monte h^{up} , che dipende dall'invaso del serbatoio, e il livello del pelo libero a valle h^{do} , che dipende invece dal rilascio dal serbatoio:

$$H_{t+1} = h^{up}(s_t) - h^{do}(r_{t+1}) \quad (4.3)$$

La produzione giornaliera di energia è data infine da:

$$P_{t+1} = \varphi \cdot g \cdot \gamma \cdot \eta(H_{t+1}) \cdot q_{t+1}^{turb} \cdot H_{t+1} \quad (4.4)$$

dove φ è un coefficiente di conversione dimensionale, g è l'accelerazione di gravità (9.81 m/s²), γ è la densità dell'acqua (1000 kg/m³), e $\eta(\cdot)$ è l'efficienza delle turbine.

4.1.2 Il sistema a valle

Il rilascio dell'Hoa Binh, insieme agli afflussi dei fiumi Thao e Lo, influenzano lo scarico a Sontay e il livello dell'acqua ad Hanoi (Figura 4.2), due variabili molto importanti per gli interessi rispettivamente degli agricoltori e degli abitanti di Hanoi (minimizzazione delle piene). Pertanto, è necessario introdurre dei modelli che descrivano le interazioni esistenti tra queste grandezze. Ovviamente, questi modelli dovrebbero cercare di produrre il più fedelmente possibile le serie storiche minimizzando le variabili di stato, che aumentano notevolmente gli oneri computazionali in fase di ottimizzazione. In questo caso specifico modelli fisicamente basati che siano sufficientemente accurati

non soddisfano il secondo requisito. Pertanto, si è deciso di utilizzare dei modelli empirici, e precisamente le reti neurali artificiali. In questi due modelli gli ingressi sono proprio il rilascio dall’Hoa Binh r_{t+1} e gli afflussi dai fiumi Thao e Lo registrati rispettivamente a YenBai (q_{t+1}^{YB}) e a VuQuang (q_{t+1}^{VQ}). Le uscite saranno rispettivamente la portata a Sontay q_{t+1}^{ST} e il livello del pelo libero ad Hanoi h_{t+1}^{HN} .

La scelta dell’architettura e delle parametrizzazioni è avvenuta attraverso il confronto delle serie di ingresso/uscita simulate e quelle storicamente registrate. In particolare, a queste reti neurali è stata richiesta una particolare accuratezza nelle previsioni di bassi valori di portata nel caso di Sontay e di alti valori del livello dell’acqua ad Hanoi, dal momento che questi sono i casi più critici per gli agricoltori e per gli abitanti di Hanoi.

In definitiva, la scelta è ricaduta in entrambi i casi su un’architettura con un solo strato nascosto di neuroni di tipo tansig. Nel modello della portata a Sontay la scelta è ricaduta su 6 neuroni, nel modello del livello dell’acqua ad Hanoi su 8 neuroni.

4.1.3 Costi per passo

Irrigazione

Normalmente gli interessi degli agricoltori sono nel massimizzare il ricavo dalla vendita del raccolto. Per utilizzare un simile indicatore bisognerebbe introdurre complessi modelli di crescita delle colture, che genererebbero un aumento delle variabili di stato. Pertanto si utilizza un indicatore proxy della tipologia descritta nel paragrafo 1.3.4, che modella il deficit di fornitura idrica. Questo indicatore può essere valido nell’ipotesi che la nuova politica di regolazione del serbatoio non sia drasticamente diversa da quella storicamente utilizzata, in modo da poter ancora utilizzare la traiettoria nominale della domanda idrica. Il costo per passo associato all’irrigazione sarà pertanto:

$$g_{t+1}^{irr} = \begin{cases} 0 & \text{se } q_{t+1}^{ST} > w_t \\ (w_t - q_{t+1}^{ST})^2 w_t^{irr} & \text{altrimenti} \end{cases} \quad (4.5)$$

La portata q_{t+1}^{ST} (nota solo al tempo $t+1$) fornita agli irrigui è stimata a Sontay, appena prima delle traverse che incanalano l’acqua al distretto irriguo. Questa portata viene ottenuta mediante il modello descritto nel paragrafo 4.1.2. w_t è il valore nominale della domanda idrica per il giorno t -esimo dell’anno. Si noti che il deficit è elevato al quadrato: gli agricoltori sono più sensibili ai grandi deficit dal momento che questi sono estremamente dannosi per le colture. Attraverso l’elevamento a potenza alti valori di deficit saranno

maggiormente presi in considerazione. Infine, w_t^{irr} è un coefficiente stagionale. Infatti, nei mesi di gennaio, febbraio e marzo, quando la coltivazione del riso necessita di acqua per la preparazione della terra, la sua mancanza è particolarmente critica, obbligando in alcuni casi a cambiare migliaia di ettari di terra con altre tipologie di colture e generando ingenti danni economici. Per questo motivo, w_t^{irr} è posto uguale a 2 durante questi tre mesi critici, uguale a 1 nel resto dell'anno.

Controllo delle piene

Nel delta del fiume Rosso gli eventi di piena si manifestano quando avviene la rottura degli argini o quando si registrano fortissime precipitazioni all'interno del delta. Il secondo caso è ovviamente indipendente dalla politica di regolazione adottata.

L'indicatore ideale per rappresentare gli interessi di Hanoi sarebbero i danni economici legati alle piene. Tuttavia, è molto difficile stimare i danni, poichè ciò richiederebbe di tenere conto non solo delle condizioni idrauliche ma anche dello stato delle infrastrutture e dello stato economico della città. Per questa ragione, si utilizza anche in questo caso un indicatore proxy della tipologia mostrata nel paragrafo 1.3.4, basato sul livello del pelo libero h_{t+1}^{HN} ad Hanoi. Infatti, c'è una forte correlazione tra alti livelli e il rischio della rottura degli argini (con conseguenti ingenti danni).

Il costo per passo è dunque:

$$g_{t+1}^{flo} = \begin{cases} 0 & \text{se } h_{t+1}^{HN} \leq \bar{h} \\ (h_{t+1}^{HN} - \bar{h})^2 w_t^{flo} & \text{altrimenti} \end{cases} \quad (4.6)$$

dove \bar{h} è la soglia di esondazione (9,5 m) e w_t^{flo} è un coefficiente stagionale, pari a 2 in agosto (il periodo più critico per i danni da esondazione) e 1 durante il resto dell'anno.

Produzione idroelettrica

L'obiettivo associato alla produzione idroelettrica è di massimizzare l'energia prodotta. Per questa ragione, il costo per passo sarà il seguente:

$$g_{t+1}^{hyd} = -P_{t+1} w_t^{hyd} \quad (4.7)$$

in cui P_{t+1} è data dall'equazione (4.4). Il cambio di segno è legato a esigenze tecniche, dal momento che l'algoritmo evolutivo tende a minimizzare gli obiettivi. w_t^{hyd} è un coefficiente stagionale, uguale a 2 nel periodo tra aprile e giugno e 1 nel resto dell'anno. Infatti, negli ultimi anni il Vietnam è diventato

un forte importatore di energia elettrica, e la mancanza di elettricità, ovviamente, si registra in particolar modo negli ultimi mesi della stagione secca (aprile - giugno). Di conseguenza, in questo periodo si valuta un maggiore valore marginale dell'energia disponibile.

4.1.4 Il modello aggregato

Nel sistema dell'Hoa Binh quindi:

- lo stato corrisponde all'invaso del serbatoio: $x_t = s_t$;
- il controllo è la decisione di rilascio dal serbatoio: u_t ;
- i disturbi sono gli afflussi dai fiumi Da, Thao e Lo: $\varepsilon_{t+1} = [q_{t+1}^{HB} \quad q_{t+1}^{YB} \quad q_{t+1}^{VQ}]$.
Sono disponibili due serie storiche di dati di afflusso che saranno utilizzate rispettivamente in fase di calibrazione e validazione: la prima dal 1/11/1957 al 31/12/1977 e la seconda dal 1/11/1994 al 31/12/2004.

Gli obiettivi considerati J_{irr} , J_{piene} e J_{idr} sono il valore medio su un certo orizzonte temporale (vedi equazione (1.20)) dei tre costi per passo (4.5), (4.6) e (4.7).

4.2 Procedura seguita

Oltre ai parametri di tuning dell'algoritmo evolutivo e le regole semplici utilizzate, riveste come noto una fondamentale importanza, per perseguire un buon risultato, la scelta di una buona architettura della rete neurale che approssima la politica. In particolare, vista la complessità del caso in esame, sarà importante anche valutare con cura gli input da inserire. In questo caso infatti il sistema non è più stazionario, per cui oltre all'invaso s_t del serbatoio verranno introdotte le quantità $t_1 = \sin(2\pi t/T)$ e $t_2 = \cos(2\pi t/T)$, dove t è il giorno dell'anno. $T = 365$ è il periodo di ciclostazionarietà (vedi sezione 2.3 per dettagli).

Lo stato del sistema è tuttavia fortemente influenzato anche dall'andamento degli afflussi, sia a monte sia a valle del serbatoio: potrebbe essere pertanto utile inserire in ingresso alla rete neurale anche gli afflussi del giorno prima q_t^{HB} , q_t^{YB} e q_t^{VQ} .

L'ottimizzazione e la validazione dei risultati verrà dunque effettuata in entrambi i casi, che saranno quindi confrontati per valutare se l'introduzione degli afflussi in ingresso comporta benefici concreti.

Per ognuno dei due casi, analogamente a quanto fatto nel Test, verrà dapprima effettuata una fase di addestramento della rete, al fine di determinare

l'architettura che offre il miglior compromesso tra complessità e prestazioni, e di determinare i range dei parametri da esplorare in fase di ottimizzazione e le parametrizzazioni relative alle regole semplici. Successivamente, sarà presentata la fase vera e propria di ottimizzazione e validazione dei risultati. In questo caso sarà effettuato un solo run per ogni esperimento, utilizzando quindi un singolo seme. Infatti nel caso di studio, più che valutare oggettivamente l'effetto dei parametri di tuning (che interessava nel caso di Test) si desidera ottenere semplicemente delle buone soluzioni, qualunque sia l'esperimento che le genera. Inoltre, un gran numero di run di uno stesso esperimento comporterebbe un significativo dispendio in termini di tempo, vista la complessità del sistema.

Per la simulazione durante il processo di ottimizzazione, verrà utilizzata la serie storica degli afflussi dal 1/11/1957 al 31/12/1977. Lo stato iniziale utilizzato sarà $s_0 = 9,871 * 10^9 m^3$. Importante sottolineare che in fase di validazione, invece di simulare con il metodo Monte Carlo come fatto nel caso di Test (capitolo 3), sarà effettuata una simulazione deterministica a partire dallo stato iniziale $s_0 = 9,8567 * 10^9 m^3$ e con la serie storica di afflussi del periodo 1/11/1994 - 31/12/2004. Infatti si possiede l'informazione sul valore degli obiettivi ottenuti dalla gestione storica durante questo periodo, ed è in questo modo possibile fare un confronto diretto con i risultati ottenuti dall'algoritmo.

4.3 Risultati della programmazione dinamica deterministica

Prima di partire con la scelta dell'architettura e l'ottimizzazione, si presentano i risultati ottenuti attraverso la *Programmazione Dinamica Deterministica* (DDP, Deterministic Dynamic Programming) con gli afflussi del periodo di validazione. Essi sono infatti molto interessanti, dal momento che la programmazione dinamica deterministica, proprio come la PD stocastica tende a identificare la miglior politica possibile (sezione 1.5), con in più l'ipotesi di conoscere a priori tutti gli afflussi che si realizzeranno per tutto l'orizzonte temporale considerato. Questo significa che per il periodo di validazione, particolarmente interessante perchè in tal caso si conosce la prestazione della gestione storica, le soluzioni ottenute dalla DDP sono le migliori politiche che sarebbe possibile realizzare, e rappresentano quindi un riferimento molto significativo per tutti i risultati che verranno d'ora in poi prodotti. La Tabella 4.1 riporta tutte le soluzioni della DDP ottenute con diversi set di pesi, e la Figura 4.3 rappresenta le soluzioni della programmazione dinamica

deterministica che dominano la gestione storica. In Tabella 4.1 l'obiettivo di produzione idroelettrica J_{idr} viene rappresentato cambiato di segno per rendere il valore (in termini di energia prodotta) più significativo. Allo stesso modo, in Figura 4.3 la grandezza dei cerchi è proporzionale all'energia prodotta (cerchi più grandi sono soluzioni migliori).

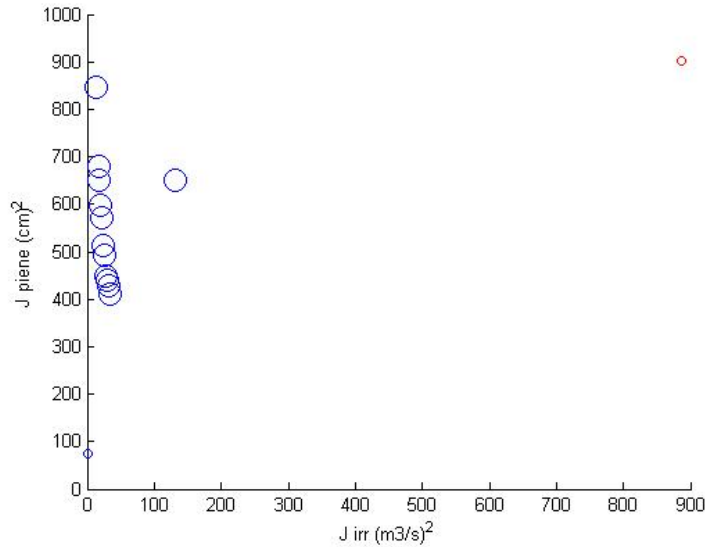


Figura 4.3: Soluzioni della programmazione dinamica deterministica (blu) che dominano la gestione storica (rosso). La grandezza dei cerchi è proporzionale al valore (negativo) dell'obiettivo idroelettrico.

In particolare, le ultime tre soluzioni mostrate in Tabella 4.1 riportano i migliori risultati conseguibili per ciascuno dei tre obiettivi considerati. La combinazione dei valori ottimi in ogni obiettivo rappresenta il punto utopia: $[0 \quad 74 \quad 31,92]$. Come si può notare è impossibile azzerare i costi dovuti alle piene ad Hanoi: evidentemente in alcuni casi gli afflussi provenienti dai fiumi Lo e Thao sono stati così rilevanti da generare livelli superiori alla soglia di esondazione in ogni caso, a prescindere dalla politica adottata. Le traiettorie realizzate simulando il sistema con le politiche della DDP potranno inoltre essere sfruttate per l'addestramento della rete neurale.

4.4 Ingressi alla rete neurale: tempo e invaso

In questa sezione saranno descritti i risultati ottenuti utilizzando una rete neurale che sfrutta in ingresso il tempo ($t_1 = \sin(2\pi t/T)$ e $t_2 = \cos(2\pi t/T)$) e l'invaso s_t , senza utilizzare le informazioni sugli afflussi.

Tabella 4.1: Risultati in validazione della programmazione dinamica deterministica.

	pesi			soluzioni		
	J_{irr}	J_{piene}	J_{idr}	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$
g. storica	-	-	-	887	902	26,30
1	0,55	0,35	0,1	28	460	31,69
2	0,58	0,32	0,1	26	494	31,70
3	0,61	0,29	0,1	23	514	31,71
4	0,64	0,26	0,1	22	571	31,72
5	0,67	0,23	0,1	19	598	31,72
6	0,7	0,2	0,1	18	652	31,73
7	0,73	0,17	0,1	17	695	31,74
8	0,8	0,1	0,1	14	846	31,76
9	0,9	0	0,1	10	1895	31,77
10	0,4	0,4	0,2	132	651	31,76
11	0,5	0,5	0	0	74	26,32
12	0,54	0,36	0,1	28	449	31,69
13	0,72	0,18	0,1	17	681	31,74
14	0,46	0,44	0,1	35	410	31,68
15	0,49	0,41	0,1	32	429	31,68
16	0,52	0,38	0,1	30	440	31,69
17	1	0	0	0	1463	27,28
18	0	1	0	0	74	26,30
19	0	0	1	9920	1895	31,92

4.4.1 Scelta dell'architettura tramite addestramento sulle regole semplici

Analogamente a quanto fatto nel caso di Test (sezione 3.2) si determinerà l'architettura, meno complessa possibile, che meglio riesce ad approssimare le regole semplici che verranno inserite. Visti i risultati ottenuti durante la fase di test, che hanno mostrato come le prestazioni dei neuroni di tipo *tansig* e *logsig* siano molto simili (e nettamente migliori, in quest'ambito, del tipo *purelin*), in questo caso verrà sempre utilizzata la tipologia *tansig*, e l'architettura verrà quindi scelta solo in termini di numero di neuroni nello strato nascosto della rete.

L'addestramento permetterà inoltre di ottenere vari set di parametri (relativi alle regole semplici) da fornire in ingresso a NsgaII. Nel capitolo 3 si è visto come l'introduzione di queste soluzioni all'interno dell'inizializzazione possa migliorare notevolmente le prestazioni dell'algoritmo. Come descritto nella sezione 2.4, sarà in questo modo possibile ottenere poi i range dei parametri all'interno del quale l'algoritmo cercherà le soluzioni in fase di ottimizzazione. Vista la complessità di questo caso di studio, estremamente maggiore di quella del caso di Test precedentemente analizzato, ci si aspetta che questa fase sia cruciale per la buona riuscita del processo di ottimizzazione.

Regole semplici

Nel caso di Test è emersa con evidenza l'importanza delle regole semplici. Vista la complessità del sistema dell'Hoa Binh, e ricordando che a partire dall'equazione (2.2) il maggior numero di ingressi (3 in questo caso) genera molti più parametri da stimare, si intuisce l'importanza di introdurre delle regole semplici sensate, possibilmente varie, e in numero sufficientemente elevato. In questo caso le regole semplici utilizzate sono le seguenti.

Regole semplici predeterminate Le prime tre regole semplici sono state preparate discretizzando tempo e invaso, e associando a ogni combinazione di tempo e invaso delle decisioni di rilascio predeterminate (vedi elenco seguente). L'invaso è stato discretizzato in modo più fitto nelle zone in cui a ogni istante temporale è più probabile che esso si trovi. Tale distribuzione di probabilità è stata ricavata a partire dai dati storicamente disponibili.

- Regola 1 - Obiettivo irrigazione: rilasciare sempre la domanda irrigua corrispondente all'istante t considerato. Visto che dal giorno 1 al 90 i deficit hanno peso doppio, nei 15 giorni finali dell'anno se l'invaso è superiore a una certa soglia, dipendente dalla domanda irrigua dei giorni

immediatamente successivi, viene rilasciata una quantità inferiore alla domanda irrigua in modo proporzionale all'invaso, fino a un minimo di 0.

- Regola 2 - Obiettivo piene: tra il giorno $t = 212$ e $t = 244$ il costo delle piene ha peso doppio. Nel periodo immediatamente precedente, pertanto, se l'invaso supera una determinata quantità la decisione di rilascio sarà $u_t = 0$. Le piene hanno una forte dipendenza dal tempo, in quanto sono più frequenti nella stagione delle piogge. Se non è verificata la condizione precedente, quindi, la decisione di rilascio è nulla nel periodo in cui le piene sono più frequenti, mentre nel resto dell'anno si rilascia una quantità sufficientemente elevata (al fine di mantenere il livello del serbatoio basso), pari alla capacità turbinabile (per favorire anche la produzione di energia).
- Regola 3 - Obiettivo idroelettrico: per massimizzare quest'obiettivo, sarebbe auspicabile rilasciare sempre la capacità massima delle turbine utilizzate dall'impianto idroelettrico. Questa, pari a $2360m^3/s$, è però molto elevata, pertanto nei periodi in cui gli afflussi non sono consistenti una simile condotta finirebbe per svuotare il serbatoio prima del periodo in cui il valore marginale dell'energia è superiore, ovvero dal giorno $t = 91$ al giorno $t = 181$. Pertanto, la regola consiste nel rilasciare sempre la capacità turbinabile in questo periodo. Nel resto dell'anno, viene effettuata una simulazione dal giorno t considerato fino al giorno $t = 90$ successivo, a partire dall'invaso s_0 corrispondente all'ingresso alla rete, con la decisione di rilascio sempre nulla, e considerando ogni giorno gli afflussi medi realizzatisi nella serie storica utilizzata poi per l'ottimizzazione. Dalla traiettoria dei rilasci \tilde{r} così ottenuta, viene quindi calcolato il totale della portata che sarebbe "sprecata", ovvero ogni volta che il rilascio \tilde{r}_{t+1} ha superato la massima capacità turbinabile q^{\max} (a causa di invasi o afflussi troppo elevati) viene sommata la differenza tra questi due valori. Questa somma totale diviene quindi la decisione di rilascio:

$$u_t = \sum_{k=t}^{90} (\max(\tilde{r}_{k+1} - q^{\max}, 0)) \quad (4.8)$$

Ovviamente, se esso è superiore alla capacità turbinabile, il controllo diverrà proprio la capacità turbinabile.

Traiettorie programmazione dinamica deterministica La regola semplice 4 utilizza come serie di ingresso e uscita le traiettorie realizzate simulando il sistema con una delle politiche ottenute con la programmazione dinamica deterministica (sezione 4.3). Questa, infatti, sarà per lo meno una soluzione ragionevolmente positiva, e potrà pertanto contribuire efficacemente durante la fase di ottimizzazione. In questo caso la soluzione utilizzata è quella relativa al vettore dei pesi: $[0, 72 \quad 0, 18 \quad 0, 1]$.

Regole semplici da simulazione Le ultime regole semplici utilizzano come serie di ingresso (tempo e invasore) e uscita (rilascio) le traiettorie ottenute simulando il sistema con la serie storica di calibrazione degli afflussi e utilizzando una decisione di rilascio pari alla domanda irrigua (regola 5), sempre pari a Infinito (regola 6) e sempre pari a 0 (regola 7). I rilasci ottenuti con le regole 5 e 6, corrisponderanno ovviamente al massimo rilascio V_{t+1} e al minimo rilascio v_{t+1} ammissibile. La loro importanza sarà relativa soprattutto alla determinazione dei range, in quanto grazie a esse sarà possibile ottenere delle parametrizzazioni che rappresentano praticamente i “casi limite” dei rilasci fisicamente possibili.

Scelta del numero di neuroni

L’addestramento della rete neurale sulle serie di dati corrispondenti alle varie regole semplici è stato svolto con crescente numero di neuroni con valori di N_{iter} (numero di ripetizioni dell’addestramento, sezione 2.4) sufficientemente elevati (in base alla regola e al numero di neuroni, da 50 a diverse centinaia). Vista la complessità del sistema e delle regole semplici implementate, è ovviamente impensabile che le precisioni dell’addestramento siano della stessa entità di quelle ottenute nel caso di Test: l’obiettivo è quindi di ottenere delle approssimazioni ragionevoli. I risultati, in termini di varianza spiegata R^2 , sono mostrati in Tabella 4.2.

Nella regola semplice DDP si nota un iniziale aumento della varianza spiegata, la quale rimane tuttavia pressochè costante su precisioni molto basse anche sopra i 3 neuroni. Leggermente superiori, ma comunque abbastanza scarse, anche le varianze spiegate riscontrate nelle tre regole semplici di simulazione. Anche in questi casi, un aumento del numero di neuroni genera vantaggi complessivamente trascurabili. Evidentemente, questo tipo di architettura della rete neurale con ingressi t e s_t non è in grado di seguire le traiettorie che vengono generate dalle simulazioni.

Per quanto riguarda le regole semplici predeterminate, l’aumento di neuroni genera dei benefici concreti. In particolare, nella regola semplice 1 relativa all’obiettivo irriguo si ottiene a partire da 6 neuroni una precisione decisiva-

Tabella 4.2: Prestazioni delle diverse architetture, in termini di varianza spiegata $R2$, nel riprodurre le diverse regole semplici (ingressi della rete $I_t = [s_t, t]$).

Neuroni	Regole semplici						
	1.Irr	2.Piene	3.Idr	4.DDP	5.Sim irr	6.Sim V_{t+1}	7.Sim v_{t+1}
1	0,6011	0,5723	0,8433	0,4648	0,7619	0,7744	0,7453
2	0,8539	0,8930	0,9460	0,5818	0,7682	0,7946	0,7666
3	0,9502	0,9145	0,9636	0,6094	0,7847	0,7790	0,7706
4	0,9720	0,9488	0,9756	0,6172	0,7878	0,8001	0,7769
5	0,9753	0,9179	0,9803	0,6210	0,7894	0,8024	0,7735
6	0,9916	0,9575	0,9840	0,6275	0,7872	0,8002	0,7741
7	0,9927	0,9383	0,9869	0,6293	0,7886	0,7943	0,7710
8	0,9872	0,9395	0,9867	0,6302	0,7907	0,7963	0,7758

mente soddisfacente, superiore a 0,99; e lo stesso vale anche per la regola 3 relativa all'obiettivo idroelettrico, seppur con precisioni leggermente inferiori. Nella regola semplice 2 relativa all'obiettivo piene, invece, questi valori di varianza spiegata non vengono mai raggiunti, anche se le precisioni sono comunque nettamente superiori a quelle riscontrate nelle regole semplici di simulazione. Anche in questo caso, dopo un iniziale aumento della varianza spiegata $R2$, un successivo aumento del numero di neuroni non comporta notevoli vantaggi, e le precisioni continuano a fluttuare intorno a valori di circa $R2 = 0,93 - 0,94$.

Complessivamente, analizzando l'andamento delle precisioni, si è optato per un'architettura con 6 neuroni, che rappresenta il miglior compromesso tra semplicità e prestazioni. Sebbene un aumento (magari consistente) del numero di neuroni possa generare ancora un incremento delle prestazioni ottenute in una o due regole semplici, l'aumento della complessità della rete non sarebbe comunque sufficientemente compensato dalle precisioni ottenibili, sia nella fase di addestramento che conseguentemente durante la fase di ottimizzazione. Si tenga presente che, ad esempio, in una rete come questa con tre ingressi ($t_1 = \sin(2\pi t/T)$, $t_2 = \sin(2\pi t/T)$) e s_t) il passaggio da 6 a 10 neuroni genererebbe un aumento del numero di parametri da stimare da 31 a 51, rendendo il processo di ottimizzazione molto più complesso.

4.4.2 Ottimizzazione della rete con NsgaII

In questa fase sono state effettuate ottimizzazioni con valori di pop e gen via via crescenti, in modo da valutare quei valori dei parametri di tuning che possono garantire il miglior compromesso tra prestazioni e tempi di calcolo. L'esperienza maturata nel caso di Test (capitolo 3) ha aiutato a definire poche ottimizzazioni mirate che mostrassero dove si trova questo compromesso. Gli esperimenti analizzati vanno da un minimo di $pop = 20$, $gen = 15$ a $pop = 1000$, $gen = 2000$, oltre il quale i tempi di calcolo diverrebbero troppo elevati, seguendo un'analisi progressiva delle prestazioni dei diversi esperimenti.

In fase di calibrazione, l'indicatore principale su cui si farà riferimento è hypervolume (paragrafo 2.6.1). In fase di validazione sarà interessante valutare anche N_{soleff} (paragrafo 2.6.2), ma in questo caso verrà calcolato in percentuale rispetto alla popolazione totale, dal momento che saranno considerati valori di pop molto diversi tra loro. Infine, verrà introdotto un ultimo indicatore di valutazione, ovvero il numero di soluzioni N_{dom} che dominano il punto della gestione storica, disponibile per la serie di afflussi usata in validazione. Questo indicatore è molto importante in un caso reale, dal momento che questi punti sono quelli realmente di interesse per il decisore, in quanto oggettivamente vantaggiosi. Sarà quindi calcolato nuovamente hypervolume (questa volta, ovviamente, relativamente alla validazione). Per il calcolo di hypervolume i punti di riferimento sono il punto della gestione storica J^{max} e il punto utopia J^{min} identificato nella sezione 4.3 per la serie di validazione (con l'obiettivo di produzione idroelettrica negativo per ragioni pratiche).

I risultati sono presentati in Tabella 4.3.

Dai risultati viene confermata la maggiore importanza che riveste il parametro gen nelle prestazioni dell'algoritmo, ampiamente dimostrata nel caso di Test. Generalmente la percentuale di soluzioni efficienti in validazione N_{soleff} cresce con l'aumentare delle generazioni e con il diminuire della popolazione (popolazioni più grandi richiedono più iterazioni). Si può notare come i risultati concordino tra calibrazione e validazione tranne nel caso $pop = 1000$, $gen = 2000$. Questo caso, migliore in fase di calibrazione rispetto al $pop = 600$, $gen = 2000$, è infatti inferiore in fase di validazione. Questo fatto sottolinea la possibilità di divergenze nei risultati legate all'utilizzo di serie storiche di afflussi diverse, quando esse non sono sufficientemente lunghe: questo problema, come evidenziato nel caso di Test, rimane probabilmente l'aspetto più critico del metodo proposto.

Dal momento che per il decisore risulta fondamentale il confronto con le prestazioni della gestione storica, è stata scelta la combinazione più vantaggiosa in fase di validazione: $pop = 600$, $gen = 2000$. Dai risultati prece-

Tabella 4.3: Risultati dei diversi esperimenti di ottimizzazione con valori crescenti di pop e gen , in calibrazione e validazione (ingressi della rete $I_t = [s_t, t]$).

		Calibrazione	Validazione		
Pop	Gen	Hypervolume	N_{soleff} [%]	N_{dom}	Hypervolume
20	15	0,0087	55%	0	/
100	50	0,0893	37%	0	/
300	200	0,1957	54%	3	0,0050
200	800	0,2424	61%	8	0,1341
600	200	0,1297	24%	6	0,0161
600	800	0,2516	42%	23	0,1657
600	2000	0,2577	49%	28	0,2192
1000	2000	0,2988	38%	26	0,1884

denti si può comunque intuire che anche un'ottimizzazione con parametri di tuning leggermente inferiori avrebbe probabilmente avuto prestazioni non significativamente peggiori.

Dal momento che la rappresentazione di una popolazione di 600 soluzioni risulterebbe particolarmente caotica e sarebbe quindi di difficile interpretazione, si è scelto quindi di mostrare solo le soluzioni N_{dom} che dominano la gestione storica, quelle realmente di interesse per il decisore. I risultati sono mostrati in Tabella 4.4 e in Figura 4.4. Anche in questo caso l'obiettivo J_{idr} è cambiato di segno quindi valori più alti indicano maggiore energia prodotta.

In Tabella 4.4 sono presenti anche le soluzioni che corrispondono alle massime prestazioni ottenibili all'interno della popolazione per ognuno dei tre obiettivi considerato singolarmente.

Sempre in Tabella 4.4 si può notare come le regole semplici "irr" e "idr" siano già particolarmente positive per i rispettivi obiettivi.

I risultati sono decisamente positivi, dal momento che si sono ottenute numerose soluzioni che dominano nettamente la gestione storica. Negli obiettivi J_{idr} e J_{irr} le prestazioni ottenute dall'algoritmo evolutivo sono assolutamente paragonabili a quelle della programmazione dinamica deterministica (sezione 4.3). Nell'obiettivo J_{piene} invece le prestazioni sono leggermente inferiori, ma comunque decisamente migliori rispetto a quelle della gestione storica.

A questo punto è interessante fare un confronto con i risultati ottenuti con la Programmazione Dinamica Stocastica SDP (sezione 1.5) dal gruppo

Tabella 4.4: Risultati in calibrazione e validazione (ingressi della rete $I_t = [s_t, t]$, $pop = 600$ e $gen = 2000$). In alto la gestione storica (solo val), poi le soluzioni N_{dom} che dominano la gestione storica, e le soluzioni che ottimizzano ognuno dei tre obiettivi (in validazione). In basso le regole semplici.

		cal			val		
		J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} 10^6 [kWh]	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} 10^6 [kWh]
g. storica		-	-	-	887	902	26,30
Algoritmo evolutivo	1	62	399	28,27	24	756	28,87
	2	70	261	26,64	33	442	27,25
	3	278	246	2,681	319	443	27,68
	4	311	250	27,10	568	430	27,95
	5	297	260	27,38	522	480	28,44
	6	2	336	26,74	29	752	27,65
	7	108	347	27,81	264	581	28,40
	8	192	278	27,12	325	491	28,03
	9	373	328	28,13	520	674	29,08
	10	63	402	28,28	23	758	28,85
	11	325	243	26,79	571	424	27,51
	12	238	280	27,32	307	500	28,08
	13	0	329	26,57	15	758	27,60
	14	2	349	26,80	27	608	27,10
	15	102	253	26,30	320	419	27,03
	16	884	270	27,85	879	506	28,74
	17	255	274	27,26	489	478	27,96
	18	348	272	27,56	372	591	28,78
	19	125	320	27,66	315	569	28,51
	20	220	240	26,00	646	385	26,41
	21	42	272	26,98	30	507	28,07
	22	246	279	27,19	297	497	27,97
	23	6	363	27,86	64	673	28,27
	24	74	339	27,68	110	637	28,28
	25	17	336	27,23	27	624	27,59
	26	249	241	26,17	642	389	26,65
	27	227	243	26,49	562	419	26,95
	28	219	247	26,91	580	426	27,75
	min irr	0	630	29,34	0	1383	30,10
	min piene	2444	232	27,00	2028	374	27,50
	max idr	7696	644	31,10	9928	1387	31,90
Regole semplici	irr	0	782	26,67	0	1461	27,40
	piene	15372	674	25,08	15984	1509	26,20
	idr	7843	647	30,34	9838	1392	31,12
	DDP	3433	551	27,28	10432	1404	28,93
	Sim irr	5	780	25,49	10	1463	26,31
	Sim V_{t+1}	15555	1263	16,55	15987	2238	16,98
	Sim v_{t+1}	14903	1019	16,67	15975	1668	17,00

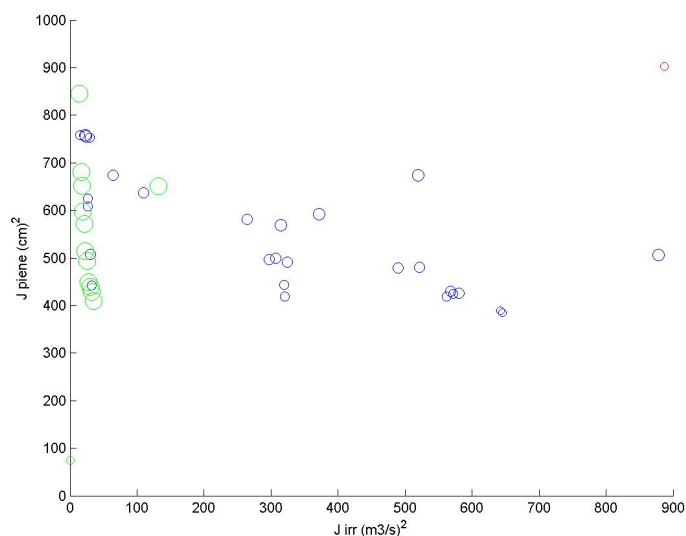


Figura 4.4: Frontiera con $pop = 600$, $gen = 2000$ (ingressi della rete $I_t = [s_t, t]$). La grandezza dei cerchi è proporzionale al valore (negativo) dell'obiettivo J_{idr} . In rosso la gestione storica, in verde le soluzioni della DDP.

di Modellistica e gestione dei sistemi ambientali del Dipartimento di Eletttronica e Informazione del Politecnico di Milano. Le leggi di controllo sfruttate in questo caso sono di classe fissata (sezione 1.6), precisamente della forma di reti neurali con ingressi $I_t = [s_t, t]$: i risultati possono essere quindi efficacemente paragonati con quelli (relativi agli stessi ingressi) dell'algorithm evolutivo. La SDP ha fornito, per il periodo di validazione, risultati mostrati in Tabella 4.5 e in Figura 4.5.

Nessuna delle soluzioni della SDP domina quelle dell'algorithm evolutivo e

Tabella 4.5: Risultati in validazione della programmazione dinamica stocastica.

	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$
g storica	887	902	26,30
SDP 1	102	869	29,66
SDP 2	617	877	29,80
SDP 3	178	896	29,70

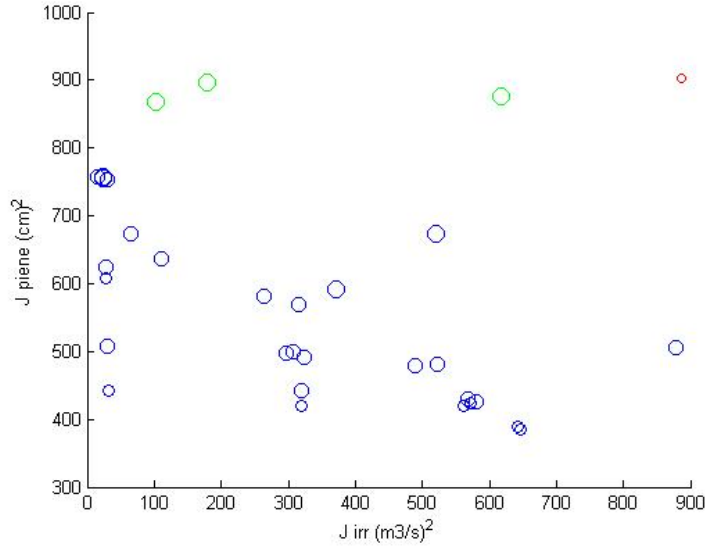


Figura 4.5: Confronto tra le soluzioni della SDP (verde) e dell’algoritmo evolutivo (ingressi della rete $I_t = [s_t, t]$) che dominano la storia. In rosso la gestione storica

viceversa, pertanto i risultati sono dello stesso ranking. Dalla Figura 4.5 si nota però come la bontà e la varietà delle politiche ottenute con NsgaII offra un compromesso di scelta tutto sommato migliore di quanto non faccia la SDP. Ciò è confermato anche dal valore di Hypervolume della SDP, pari a 0,0213, di molto inferiore a quello dell’algoritmo evolutivo (0,2192, Tabella 4.3, $pop = 600$ e $gen = 2000$).

In questo caso l’ipotesi che gli afflussi siano disturbi bianchi (vedere Soncini Sessa (2004)) è un’approssimazione rilevante, e per incrementare le prestazioni della SDP occorrerebbe pertanto introdurre dei modelli degli afflussi, aumentando fortemente gli oneri computazionali. Questo confronto conferma il grande potenziale del nostro algoritmo evolutivo, in quanto la SDP è un metodo che garantisce per sua stessa natura l’ottimo, e inoltre considera una statistica dei disturbi invece che una singola realizzazione.

4.4.3 Analisi delle leggi di controllo

A questo punto risulta interessante valutare la forma delle leggi di controllo relative alle soluzioni presentate nel paragrafo precedente. Infatti queste sono gli strumenti attraverso cui il regolatore opererà fisicamente, ed è pertanto importante valutare se, come si spera, abbiano un andamento sensato. Generalmente, per esempio, ci si aspetta che a parità degli altri ingressi la

funzione sia crescente con l'invaso, o per lo meno non monotona decrescente: appare abbastanza strano immaginare che una politica che richieda di rilasciare sempre una portata elevata con un invaso scarso e viceversa possa essere efficiente.

I risultati sulle N_{dom} (28) soluzioni presentate nel precedente paragrafo hanno mostrato una grande maggioranza di leggi di controllo che si conciliano con il buon senso, della forma presentata in Figura 4.6.

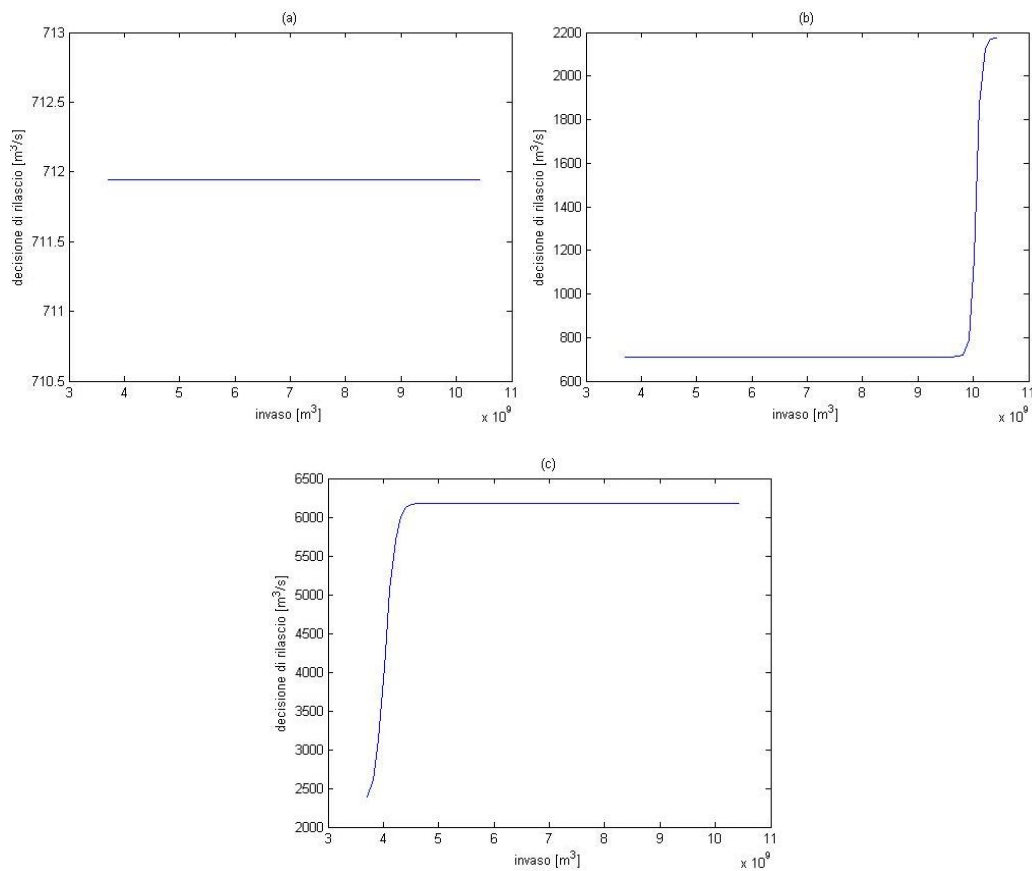


Figura 4.6: Esempio di legge di controllo ottenuta con $pop = 600$, $gen = 2000$ (rete $I_t = [s_t, t]$). La funzione (a) è tipica dei primi e degli ultimi giorni dell'anno, mentre le altre del periodo centrale (afflussi elevati).

L'andamento è costante o quasi durante i primi e gli ultimi mesi dell'anno, in cui gli afflussi sono scarsi. Via via che si passa in periodi più piovosi le funzioni diventano sempre più variabili, quasi sempre crescenti con l'invaso.

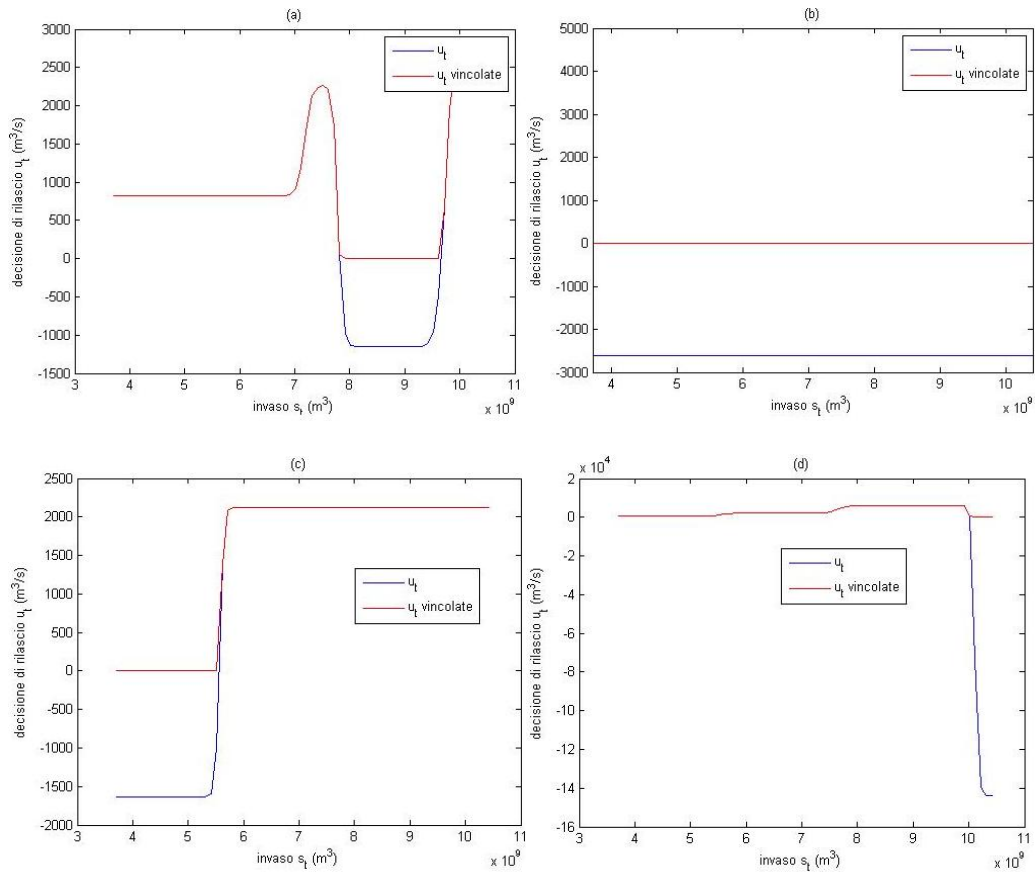


Figura 4.7: Esempi di leggi di controllo (ingressi della rete $I_t = [s_t, t]$) con sezioni fisicamente insensate (blu). In rosso le stesse leggi vincolate ai minimi e massimi valori ammissibili del controllo.

In varie loro parti alcune leggi di controllo toccano però purtroppo valori insensati fisicamente (in Figura 4.7 valori negativi), che durante la simulazione generano un rilascio pari al minimo (o massimo). Queste leggi non vengono pertanto necessariamente eliminate durante l'ottimizzazione. Anzi, se per una particolare combinazione s_t, t risulta efficiente rilasciare proprio il minimo, l'algoritmo genererà con massima probabilità parametrizzazioni che corrispondono a valori negativi, dal momento che attraverso la simulazione non discrimina i valori del controllo che siano inferiori ad esso (lo stesso vale per il massimo rilascio).

Al di là del semplice problema estetico (peraltro eliminabile mettendo un vincolo a posteriori sui controlli ottenuti, vedi linea rossa in Figura 4.7), l'assenza di vincoli sulle parametrizzazioni ammissibili può ostacolare il rag-

giungimento di buoni risultati in fase di ottimizzazione. Implementando un sistema che impone a NsgaII di non accettare parametrizzazioni relative a leggi di controllo fortemente insensate, infatti, si restringerebbe notevolmente il campo di ricerca dei parametri, e sarebbe presumibilmente possibile raggiungere buoni risultati con meno generazioni. Questa potrebbe essere una interessante questione da approfondire in studi futuri.

4.5 Ingressi alla rete neurale: tempo, invaso e afflussi del giorno precedente

In questa fase si ripeteranno le operazioni precedenti inserendo tra gli ingressi della rete neurale anche le informazioni degli afflussi del giorno precedente: q_t^{HB} , q_t^{YB} e q_t^{VQ} . Si spera che queste informazioni possano aiutare la rete a generare leggi di controllo più efficienti.

4.5.1 Scelta dell'architettura tramite addestramento

Regole semplici

Le regole semplici introdotte nel caso precedente sono state mantenute, adattandole ovviamente alle informazioni disponibili sugli afflussi.

La regola semplice 2, per la minimizzazione delle piene, è stata leggermente modificata. Infatti, invece che l'informazione sul tempo (non si rilasciava durante il periodo delle piene), ora si sfrutta l'informazione sugli afflussi del giorno precedente q_t^{YB} e q_t^{VQ} visto che, come descritto nel paragrafo 4.1.2, gli afflussi dai fiumi Thao e Lo contribuiscono insieme al rilascio dal serbatoio a determinare l'altezza del pelo libero ad Hanoi. Ora pertanto la decisione di rilascio u_t avrà un valore inversamente proporzionale ai due afflussi q_t^{YB} e q_t^{VQ} . E' invece mantenuto il vincolo sul rilascio nei giorni precedenti a $t = 212$ quando l'invaso è elevato.

Inoltre, sfruttando le nuove informazioni fornite alla rete sono state implementate tre nuove regole semplici. Queste sfruttano le due reti neurali che descrivono il sistema a valle dell'Hoa Binh e permettono di ottenere i valori di q_{t+1}^{ST} e h_{t+1}^{HN} . Il rilascio è stato discretizzato in 155 valori, e tra questi la decisione di rilascio è quella che, insieme agli afflussi del giorno precedente, fornisce determinati valori delle due variabili q_{t+1}^{ST} e h_{t+1}^{HN} :

- Regola 8 - Obiettivo irrigazione: la decisione u_t è il minimo valore del rilascio a cui corrisponde, insieme agli afflussi q_t^{YB} e q_t^{VQ} , un deficit di fornitura nullo:

$$\begin{aligned}
u_t &= \min(r) : q_{t+1}^{ST} \geq w_t \quad \text{con } q_{t+1}^{ST} = f(r, q^{YB}, q^{VQ}) \\
q^{YB} &= q_t^{YB} \\
q^{VQ} &= q_t^{VQ}
\end{aligned} \tag{4.9}$$

In questo modo, si cerca di annullare il deficit mantenendo il serbatoio più pieno possibile per la stagione secca.

- Regola 9 - Obiettivo piene: la decisione u_t è il massimo valore del rilascio a cui corrisponde, insieme agli afflussi q_t^{YB} e q_t^{VQ} , un livello h^{HN} del pelo libero ad Hanoi inferiore alla soglia di esondazione:

$$\begin{aligned}
u_t &= \min(r) : h_{t+1}^{HN} \leq \bar{h} \quad \text{con } h_{t+1}^{HN} = f(r, q^{YB}, q^{VQ}) \\
q^{YB} &= q_t^{YB} \\
q^{VQ} &= q_t^{VQ}
\end{aligned} \tag{4.10}$$

In questo modo, si cerca di annullare le esondazioni mantenendo il serbatoio più vuoto possibile per la stagione delle piene.

- Regola 10 - Obiettivo piene - produzione idroelettrica: questa regola semplice è analoga alla precedente, ma si limita il rilascio alla capacità turbinabile qualora la decisione u_t ottenuta nel caso precedente sia superiore ad essa. In questo modo, si cerca innanzitutto di salvaguardare gli abitanti di Hanoi, pur non sprecando acqua per la produzione idroelettrica.

Queste regole semplici si basano su un'ipotesi molto forte: gli afflussi che si realizzeranno nell'istante $t + 1$ sono gli stessi del giorno precedente, noti al tempo t . Ovviamente non sarà quasi mai così, specialmente nella stagione delle piene in cui l'irregolarità degli afflussi è massima. Esiste però una forte correlazione tra gli afflussi in giorni successivi. Visto il loro scopo, che non è quello di raggiungere l'ottimo ma semplicemente di indicare forme "ragionevoli" della legge di controllo, queste regole semplici possono presumibilmente essere abbastanza buone.

Scelta del numero di neuroni

I risultati, in termini di varianza spiegata R2, dell'addestramento con diverso numero di neuroni (tipologia tansig) sono espressi in Tabella 4.6.

La prima cosa che colpisce sono le precisioni delle regole semplici di simulazione, non elevatissime ma comunque superiori di ben 0,2 rispetto all'addestramento della rete con ingressi $I_t = [s_t, t]$ (Tabella 4.2). Evidentemente le informazioni sugli afflussi aumentano notevolmente la capacità di seguire le

Tabella 4.6: Prestazioni delle diverse architetture, in termini di varianza spiegata R2 nel riprodurre le varie regole semplici (ingressi della rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$).

Neuroni	Regole semplici				
	1.Irr	2.Piene	3.Idr	4.DDP	5.Sim irr
1	0,6014	0,6056	0,8053	0,4662	0,9301
2	0,8453	0,9113	0,9051	0,6078	0,9442
3	0,9572	0,928	0,9496	0,6399	0,9492
4	0,9795	0,9406	0,9601	0,6519	0,9502
5	0,9923	0,9286	0,9698	0,6606	0,9505
6	0,9946	0,9415	0,9752	0,6611	0,951
7	0,9942	0,943	0,9798	0,6654	0,9511
8	0,9961	0,9396	0,9814	0,6722	0,9517
9	0,997	0,9578	0,9832	0,6704	0,9511
10	0,9972	0,9649	0,9851	0,6725	0,9512

Neuroni	Regole semplici				
	6.Sim V_{t+1}	7.Sim v_{t+1}	8.Irr2	9.Piene2	10.Piene-idro2
1	0,9477	0,9459	0,6542	0,8321	0,7657
2	0,9503	0,9486	0,8532	0,9262	0,8234
3	0,9518	0,9491	0,9466	0,9854	0,8905
4	0,9523	0,9502	0,9559	0,9901	0,9281
5	0,9528	0,9505	0,9611	0,9935	0,9552
6	0,9535	0,9506	0,9608	0,9959	0,9645
7	0,9531	0,9508	0,9612	0,9966	0,9655
8	0,9537	0,9517	0,9629	0,9968	0,9707
9	0,9533	0,9511	0,9622	0,9973	0,9744
10	0,9537	0,9514	0,9637	0,9977	0,9794

traiettorie ottenute nelle simulazioni. Molto simili invece le precisioni ottenute nella regola “DDP”.

L’andamento di R2 col numero di neuroni nelle regole predeterminate non cambia significativamente con i nuovi ingressi (anche se la regola 2 relativa alle piene è leggermente diversa da prima).

Tra le nuove regole semplici, si può vedere che l’unica che con un numero sufficiente di neuroni raggiunge ottime precisioni è la 9.

In generale nella maggior parte delle regole semplici, dopo un iniziale forte incremento di R2 la crescita rallenta fino a stabilizzarsi intorno a 8 neuroni. Nelle regole semplici 3, 8 e 10 invece, le precisioni continuano lentamente ad aumentare, tuttavia per avere miglioramenti significativi bisognerebbe scegliere un architettura con molti neuroni. A partire da queste considerazioni, dunque, la scelta ricade su 8 neuroni.

4.5.2 Ottimizzazione della rete con NsgaII

A valle dei risultati ottenuti con gli ingressi $I_t = [s_t, t]$, ma considerando anche il fatto che ora i parametri da stimare sono molti di più (65), si è deciso di effettuare l’ottimizzazione con $pop = 800$, $gen = 2400$. Questo esperimento richiede già un tempo abbastanza consistente. Di seguito sono riportati i risultati:

		Calibrazione	Validazione		
Pop	Gen	Hypervolume	N_{soleff} [%]	N_{dom}	Hypervolume
800	2400	0,3027	35%	12	0,2636

Come si può vedere si ha un discreto aumento nei valori di Hypervolume rispetto ai casi esaminati con la rete (s_t, t) , ma valori inferiori N_{soleff} e curiosamente anche di N_{dom} . Evidentemente, le soluzioni generate con gli afflussi in ingresso sono complessivamente migliori, tuttavia casualmente la frontiera è composta da meno soluzioni che dominano la gestione storica.

Dal momento che al contrario del caso precedente non si dispone di dati su ulteriori esperimenti, può risultare utile valutare l’andamento di Hypervolume (in calibrazione) nelle generazioni (Figura 4.8).

Si può notare che il valore di Hypervolume cresce rapidamente fino a circa 500 generazioni. Successivamente, il valore continua ad aumentare lentamente fino alla generazione finale. Si suppone quindi che le generazioni adottate siano largamente sufficienti per ottenere i migliori risultati, almeno con questa popolazione.

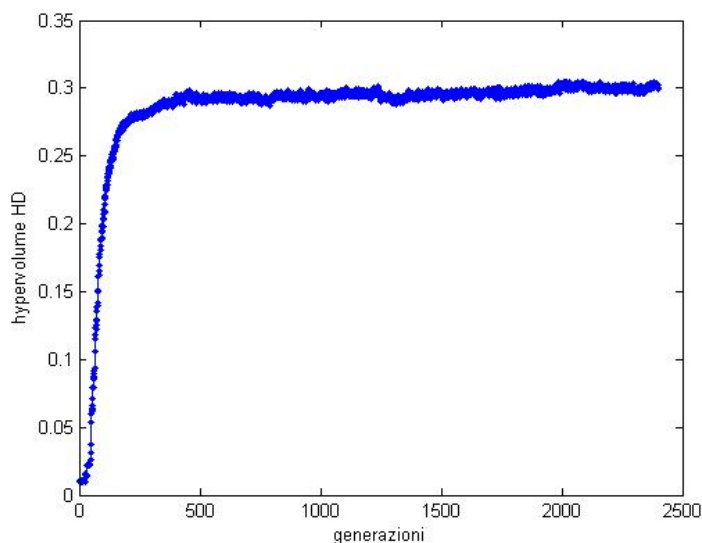


Figura 4.8: Andamento di Hypervolume (calibrazione) nelle generazioni, rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$ e $pop = 800$.

Come nel caso precedente verranno ora presentate le N_{dom} soluzioni che dominano la gestione storica, i valori ottimi ottenuti per ognuno dei tre obiettivi considerati e le prestazioni delle regole semplici (Tabella 4.7 e Figura 4.9).

Anche qui, oltre alle 12 soluzioni che dominano la gestione storica, il decisore può adottare altre 788 politiche efficienti, se è il caso.

I risultati sono anche in questo caso decisamente positivi, con una moltitudine di soluzioni che domina abbondantemente la gestione storica, e non molto lontani dalle soluzioni ideali fornite dalla programmazione dinamica deterministica (Figura 4.9). A un primo impatto sembra anche che i risultati siano migliori di quelli ottenuti con la rete $I_t = [s_t, t]$. Per un'analisi più approfondita del confronto tra rete $I_t = [s_t, t]$ e rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$ si rimanda alla sezione 4.7.

Dalla Tabella 4.7 si può notare che le regole semplici rappresentano già un ottimo punto di partenza per ognuno dei tre obiettivi (regole "Irr", "Idr" e "Piene2"): questo è indubbiamente un buon vantaggio. Altre regole, invece, non hanno prodotto i risultati sperati, alcune per le scarse precisioni in fase di addestramento, alcune a causa dell'ipotesi (in realtà non vera) che gli afflussi siano gli stessi del giorno precedente, o a causa di un'idea di base evidentemente non efficace.

Tabella 4.7: Risultati dell'esperimento $pop = 800$, $gen = 2400$, in calibrazione e validazione (rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$). In alto la gestione storica (solo val), poi le soluzioni N_{dom} che dominano la gestione storica, e le soluzioni che ottimizzano ognuno dei tre obiettivi (in validazione). In basso le prestazioni delle regole semplici.

		cal			val		
		J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$
g. storica		-	-	-	887	902	26,30
Algoritmo evolutivo	1	109	253	27,33	254	382	28,61
	2	101	249	27,17	209	403	28,30
	3	133	261	27,62	401	474	28,93
	4	30	270	27,25	127	575	28,44
	5	50	254	27,19	209	523	28,31
	6	92	281	27,84	304	611	29,47
	7	18	306	27,27	85	643	28,38
	8	118	305	27,97	246	834	29,19
	9	13	303	27,22	60	633	28,34
	10	3	325	26,91	4	792	28,00
	11	3	324	26,90	4	791	27,99
	12	70	294	27,34	143	777	29,04
	min irr	0	392	27,16	0	1048	28,50
	min piene	953	242	27,42	1645	348	28,20
max idr	7198	658	31,10	7231	1414	31,80	
Regole semplici	Irr	0	782	26,67	0	1461	27,40
	Piene	15702	626	25,60	16049	1397	26,51
	Idr	7683	646	30,30	9778	1389	31,04
	DDP	4387	540	26,98	6641	1401	29,51
	Sim irr	2502	792	24,08	3374	1498	25,90
	Sim V_{t+1}	15796	971	16,73	15899	1728	17,05
	Sim v_{t+1}	7556	815	27,53	9650	1548	28,22
	Irr2	578	782	27,52	451	1461	28,25
	Piene2	15448	246	16,94	15847	258	17,55
	Piene-idro2	15772	645	25,82	15491	1384	26,70

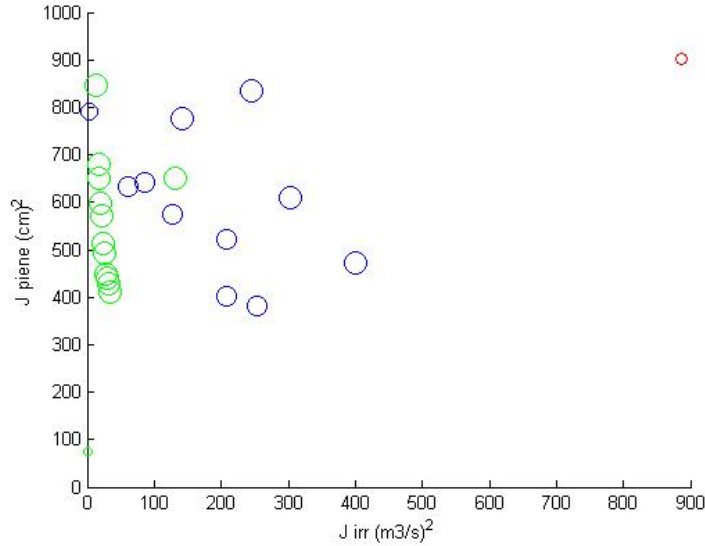


Figura 4.9: Frontiera con $pop = 800$, $gen = 2400$ (ingressi alla rete $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$). La grandezza dei cerchi è proporzionale al valore (cambiato di segno) dell'obiettivo J_{idr} . In rosso la gestione storica.

4.5.3 Leggi di controllo

Visto il maggior numero di ingressi alla rete neurale, in questo caso la visualizzazione delle leggi di controllo risulta molto più complessa. Per questo motivo, verranno presentate figure che riflettono diversi punti di vista della stessa legge di controllo. Tutte le leggi di controllo relative alle N_{dom} soluzioni presentate in Figura 4.9 hanno, salvo piccole differenze, lo stesso andamento.

In Figura 4.10 viene mostrato un esempio di queste leggi di controllo in funzione dell'invaso s_t (a) e degli afflussi del giorno precedente (in questo caso q_t^{YB} , ma l'andamento è simile con tutti e tre gli afflussi) (b), con il colore dei punti che indica l'istante temporale. In funzione dell'invaso e degli afflussi il controllo non mostra un trend definito, ma si può chiaramente notare che a ogni valore del controllo tutti i punti corrispondono a uno specifico (o pochi) istante temporale: i punti sono di colore bianco e nero per bassi valori del controllo (inizio e fine dell'anno), mentre il massimo valore del controllo corrisponde sempre a punti grigio scuro (prossimità della stagione delle piene). Questo dovrebbe suggerire una forte dipendenza del controllo dal tempo, e una minore rilevanza di invasore e afflussi.

L'intuizione è confermata anche dalla Figura 4.11.

In questo caso si nota un trend assolutamente ben definito in funzione dell'asse delle ascisse, che corrisponde al tempo, con decisioni di rilascio basse

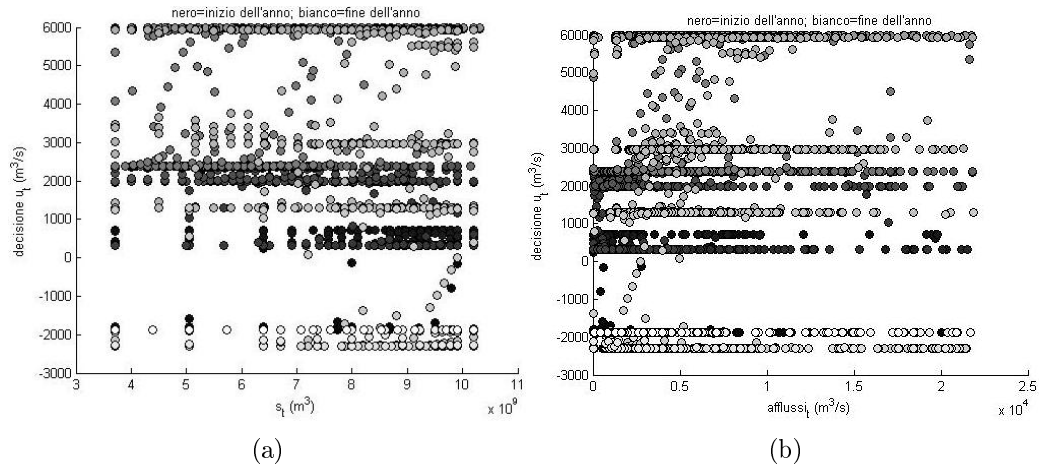


Figura 4.10: Legge di controllo in funzione dell'invaso (a) e degli afflussi del giorno precedente (b). Il colore dei punti identifica l'istante temporale. In funzione dell'asse delle ascisse non si ha un andamento preciso, ma gli allineamenti monocolori per ogni livello del controllo indicano una forte dipendenza dal tempo.

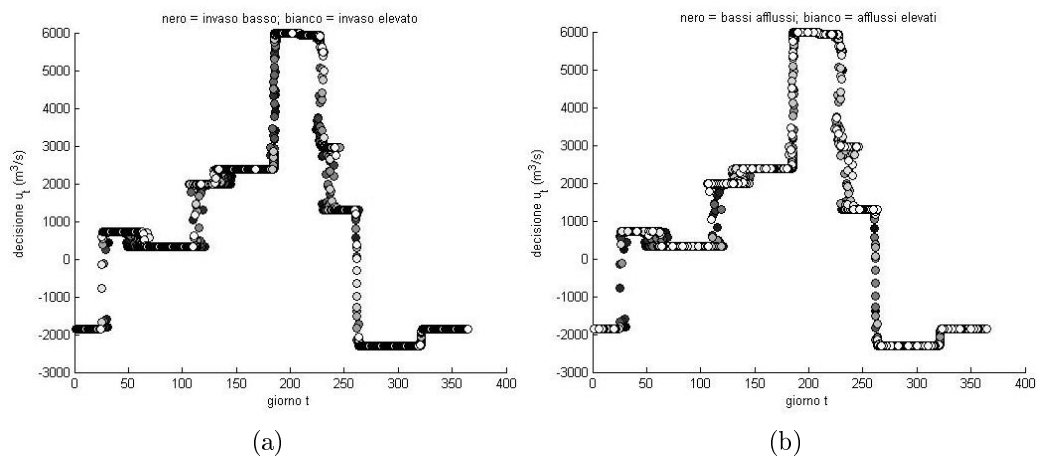


Figura 4.11: Legge di controllo in funzione del tempo (giorno dell'anno). Il colore dei punti identifica l'invaso (a) e gli afflussi del giorno precedente (b). In questo caso si ha un andamento definito in funzione dell'ascissa (tempo).

all'inizio e alla fine dell'anno e valori via via crescenti in prossimità della stagione delle piene. In questo caso, il colore dei punti (che indica il valore dell'invaso s_t nel pannello (a) e dell'afflusso q_t^{YB} nel pannello (b)) non ha un pattern preciso, ma semplicemente compaiono più punti di una certa tonalità nell'istante temporale in cui la discretizzazione (dell'invaso o dell'afflusso) è più fitta nel relativo range di valori. Si noti inoltre in entrambe le Figure la presenza di valori negativi della decisione che, come discusso nel paragrafo 4.4.3, saranno poi messi a zero in simulazione grazie alla funzione di rilascio.

Appurato che la variabile dominante nella legge di controllo è il tempo, può risultare interessante verificare com'è invece l'andamento della legge in funzione di invasore e afflussi (Figura 4.12). In questo caso non è presente un

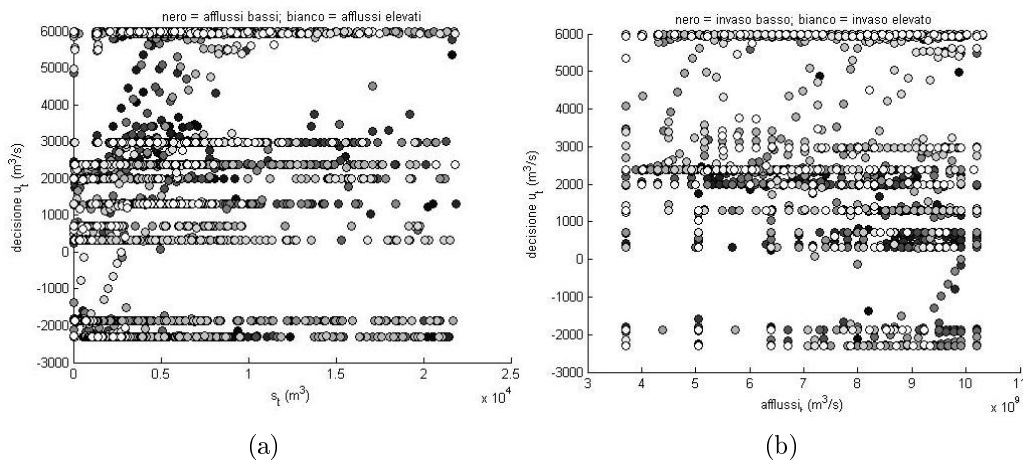


Figura 4.12: Legge di controllo in funzione dell'invasore, con il colore che indica il valore degli afflussi (a), e viceversa (b). Gli afflussi si riferiscono a q_t^{YB} , ma l'andamento è simile per tutti e tre gli afflussi.

trend definito. Semplicemente, è più probabile che il controllo assuma valori elevati quando l'invasore o gli afflussi sono alti (maggiore presenza di punti bianchi per alti valori del controllo).

In definitiva l'algoritmo evolutivo ha raggiunto la massima efficienza esplorando leggi di controllo che hanno una forte dipendenza dal tempo, segno che la forte stagionalità delle variabili di interesse del sistema è la caratteristica più rilevante per una politica di regolazione. A parità di istante temporale, l'entità della decisione di rilascio u_t dipenderà poi dalla particolare combinazione di invasore e afflussi che si manifesta in quel dato istante. L'unico piccolo difetto riscontrato è che, come anche nelle leggi di controllo ottenute con ingressi $I_t = [s_t, t]$, quando risulta efficiente che il controllo segua la

scala di minimo rilascio si generano parametrizzazioni fisicamente insensate (valori negativi), dal momento che attraverso la simulazione l’algoritmo non discrimina valori del controllo che siano inferiori al minimo valore rilasciabile (e viceversa nel caso di massimo rilascio). Come descritto nel paragrafo 4.4.3 ciò probabilmente può ostacolare il raggiungimento di buoni risultati.

4.6 Confronto tra le reti con diversi ingressi

Per concludere, si valuterà ora più in dettaglio la bontà delle frontiere ottenute con le due reti: $I_t = [s_t, t]$ e $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$. Per convenzione, il primo caso sarà associato al numero 1, il secondo al 2. Come già mostrato, anche se il numero di soluzioni che dominano la gestione storica N_{dom} è casualmente maggiore nella rete 1 (e così anche N_{soleff}), i valori di Hypervolume sono nettamente superiori sia in calibrazione che in validazione nella rete 2, e sembra quindi che le informazioni sugli afflussi generino concreti benefici.

Tabella 4.8: Confronto tra le prestazioni delle reti 1 (ingressi $I_t = [s_t, t]$) e 2 (con anche gli afflussi in ingresso).

	Calibrazione	Validazione		
Rete	Hypervolume	N_{soleff} [%]	N_{dom}	Hypervolume
1	0,2577	49%	28	0,2192
2	0,3027	35%	12	0,2636

Per avere un’idea più chiara, si è deciso di unire le popolazione delle due reti e ordinarle secondo i criteri di Ranking e Crowding Distance (CD). In calibrazione sono risultate di ranking 1 il 35% delle soluzioni della rete 1 e ben il 76% della rete 2. In validazione, invece, sono risultate di ranking 1 il 77% delle soluzioni efficienti della rete 1 e solo il 46% della rete 2. Questo testimonia la variabilità dei risultati con diverse serie temporali degli afflussi. In Tabella 4.9 è stata ripetuta questa operazione per le sole soluzioni che dominano la gestione storica (in validazione). Si vede in questo caso che nella popolazione totale tutte le soluzioni della rete 2 sono di ranking 1, mentre della rete 1 solo 9 (il 32%). Probabilmente la bontà di queste soluzioni della rete 2 è una delle cause del superiore valore di Hypervolume della rete 2 anche in validazione.

Tabella 4.9: Confronto tra le soluzioni che dominano la gestione storica della rete 1 (ingressi $I_t = [s_t, t]$) e della rete 2 (ingressi $I_t = [s_t, t, q_t^{HB}, q_t^{YB}, q_t^{VQ}]$).

Rete	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$	Ranking	CD
2	254	382	28,61	1	Inf
2	209	403	28,31	1	0,326861
2	401	474	28,93	1	Inf
2	127	575	28,44	1	0,39363
2	209	523	28,31	1	0,330166
2	304	611	29,47	1	Inf
2	85	643	28,38	1	0,469364
2	246	834	29,19	1	Inf
2	60	633	28,34	1	0,203295
2	4	792	28,00	1	Inf
2	4	792	27,99	1	0,206144
2	143	777	29,04	1	0,389202
1	24	756	28,87	1	0,054715
1	33	442	27,25	1	0,43527
1	29	752	27,65	1	0,421036
1	23	758	28,85	1	0,067186
1	15	758	27,60	1	0,116246
1	27	608	27,10	1	Inf
1	372	591	28,78	1	0,421928
1	30	507	28,07	1	0,24816
1	27	624	27,59	1	0,206564
1	319	443	27,68	2	0,263933
1	568	430	27,95	2	0,151244
1	522	480	28,44	2	0,136582
1	264	581	28,40	2	0,523591
1	325	491	28,03	2	0,304245
1	520	674	29,08	2	Inf
1	571	424	27,51	2	0,282123
1	307	500	28,08	2	0,142799
1	320	419	27,03	2	0,234458
1	879	506	28,74	2	Inf
1	489	478	27,96	2	0,375179
1	315	569	28,51	2	0,390021
1	646	385	26,41	2	Inf
1	297	497	27,97	2	0,111126

Rete	J_{irr} [m^3/s] ²	J_{piene} [cm^2]	J_{idr} $10^6[kWh]$	Ranking	CD
1	64	673	28,27	2	Inf
1	110	637	28,28	2	0,612592
1	642	389	26,65	2	0,398442
1	562	419	26,95	2	0,303457
1	580	426	27,75	2	0,208329

Una rappresentazione grafica dell'unione di queste soluzioni è mostrata in Figura 4.13.

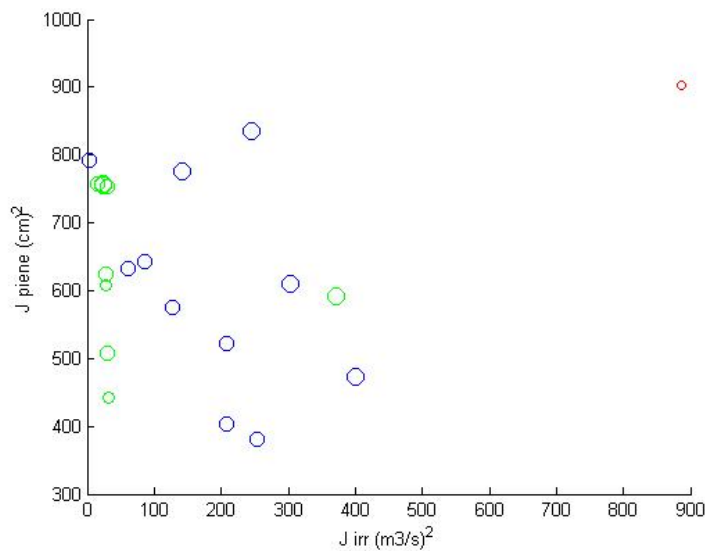


Figura 4.13: Unione delle soluzioni di ranking 1 che dominano la gestione storica (rosso): rete 1 (verde) e rete 2 (blu).

In definitiva sembra proprio che, nonostante alcuni indizi contrastanti in termini di numerosità di soluzioni efficienti in validazione, l'introduzione delle informazioni sugli afflussi generi un netto miglioramento delle prestazioni, soprattutto durante l'ottimizzazione stessa (calibrazione). Anche in validazione le soluzioni di maggiore interesse, ovvero quelle che dominano la gestione storica, risultano migliori nella rete 2. I risultati sono confermati anche in termini di Hypervolume.

4.7 Conclusioni sul caso di studio

Le buone indicazioni ottenute nel caso di Test sono state confermate anche nel caso di studio della regolazione dell’Hoa Binh.

1. Anche in questo caso reale, l’algoritmo evolutivo NsgaII con le nostre innovazioni ha dimostrato di essere un valido strumento. Con entrambe le reti con diversi ingressi, esso è stato in grado di ottenere una frontiera di soluzioni discretamente varie che dominano nettamente la gestione storica. Le politiche così realizzate possono pertanto costituire dei risultati economicamente e socialmente vantaggiosi nella regolazione del serbatoio Hoa Binh da parte degli organi competenti.
2. I risultati ottenuti con la programmazione dinamica stocastica con leggi di controllo della forma di reti neurali con ingressi $I_t = [s_t, t]$, si sono dimostrati dello stesso ranking di quelli ottenuti con NsgaII con gli stessi ingressi alla rete. La numerosità di soluzioni efficienti ricavabili con NsgaII in un singolo run permette però di riflettere molto meglio i diversi compromessi tra gli interessi dei portatori, con un costo computazionale minore.
3. L’introduzione delle informazioni sugli afflussi come ingressi della rete neurale si è dimostrata complessivamente vantaggiosa, anche se non ha comunque rivoluzionato le prestazioni dell’algoritmo. Le leggi di controllo ottenute nel processo di ottimizzazione hanno infatti dimostrato di sentire una fortissima dipendenza dal tempo, e una minore influenza dagli altri fattori, che forniscono comunque delle informazioni utili alla rete neurale per migliorare le prestazioni della politica.
4. L’analisi delle leggi di controllo ha fornito delle indicazioni sensate e interessanti, come il fortissimo impatto del tempo, probabilmente a causa della forte stagionalità del sistema che permette di anticipare determinati comportamenti vantaggiosi. Quando in ingresso non sono presenti gli afflussi, in un determinato istante temporale la decisione di rilascio u_t ha mostrato un andamento generalmente crescente (salvo rari casi) con l’invaso s_t nei periodi di piena, in cui gli afflussi sono elevati. Un difetto riscontrato è la tendenza delle leggi di controllo ad assumere andamenti fisicamente insensati (per esempio con valori negativi) dal momento che all’algoritmo interessa solo il valore degli obiettivi che produce la simulazione, e non la forma delle leggi di controllo. Questo difetto probabilmente può costituire un ostacolo alla convergenza dell’algoritmo.

5. Le perplessità riguardanti l'orizzonte temporale di simulazione h sono state confermate anche in questo caso di studio. I risultati sono infatti leggermente variabili da calibrazione ($h \approx 20 * 365$ giorni) a validazione ($h \approx 10 * 365$ giorni), e molte soluzioni che in un caso risultavano efficienti possono non esserlo più per un altro. I risultati non vengono comunque complessivamente rivoluzionati, e anzi mantengono sempre una certa coerenza di base. La simulazione deterministica risulta quindi, in questo caso, un'approssimazione tutto sommato accettabile.

Conclusioni

Lo scopo del presente lavoro, ovvero introdurre e valutare un approccio innovativo, costituito dall'uso integrato di algoritmi evolutivi (nella fattispecie NsgaII) e reti neurali, per la risoluzione dei problemi di controllo nell'ambito dei sistemi idrici, è stato ampiamente soddisfatto.

Dopo avere sottolineato l'importanza di gestire efficacemente le risorse idriche e aver formalmente introdotto il problema di controllo (capitolo 1), si è descritto l'approccio seguito in questa trattazione, presentando le caratteristiche dei principali strumenti utilizzati: reti neurali (sezione 2.1) e algoritmi evolutivi a Molti Obiettivi (sezione 2.2). Dopodichè sono state mostrate alcune importanti innovazioni inserite appositamente nel presente lavoro con lo scopo di incrementare le prestazioni degli strumenti utilizzati (sezioni 2.3 e 2.4), ed è stato presentato brevemente uno studio da me effettuato sugli indici di prestazione per gli algoritmi a molti obiettivi, al fine di definire una serie di indicatori per determinare nella maniera più completa e efficace possibile la bontà dei risultati raggiunti operativamente (sezione 2.6).

Si sono quindi mostrati i risultati ottenuti in un caso di Test (capitolo 3), che ha permesso di valutare il comportamento dell'algoritmo e di valutare l'effetto dei parametri di tuning. Questo caso di Test ha fornito indicazioni molto positive sul potenziale dell'algoritmo, confermate poi successivamente nel caso di studio, sulla regolazione del serbatoio Hoa Binh in Vietnam (capitolo 4).

L'utilizzo integrato delle reti neurali per approssimare la politica, e dell'algoritmo evolutivo NsgaII per il processo di ottimizzazione, ha dimostrato di poter generare buoni risultati, sia nel caso di Test che nel caso reale del serbatoio Hoa Binh. In particolare, nel caso di Test è stata prodotta una frontiera di Pareto potenzialmente paragonabile a quella ottenuta con la programmazione dinamica, ma approssimata molto meglio. Infatti, uno dei principali vantaggi dell'algoritmo evolutivo è quello di poter realizzare a ogni generazione una popolazione di soluzioni che riflette molto bene i possibili compromessi tra gli interessi in gioco. L'algoritmo si presta pertanto particolarmente a essere utilizzato nei problemi di ottimizzazione a molti

obiettivi, come quasi sempre lo sono i problemi di controllo nell'ambito dei sistemi idrici. Strumenti tradizionali come la programmazione dinamica sono fortemente penalizzati, in quest'ambito, dalla loro natura mono-obiettivo, e pagano le loro elevate prestazioni con forti oneri computazionali che li rendono inapplicabili a casi troppo complessi. Per approssimare discretamente una frontiera, infatti, con questi strumenti il numero di ottimizzazioni necessarie cresce esponenzialmente col numero degli obiettivi, rendendo costi e tempi spesso proibitivi. Il principio euristico e la notevole flessibilità degli algoritmi evolutivi, che li rende utilizzabili in una grandissima varietà di problemi di ottimizzazione, insieme alle loro buone prestazioni fanno in modo che questi strumenti possano divenire una valida alternativa ai classici strumenti di ottimizzazione.

E' bene ricordare che NsgaII è comunque solo uno dei tanti algoritmi evolutivi realizzati negli ultimi anni, e utilizzati nei più disparati campi di applicazione. Uno studio lungo ma probabilmente proficuo, può essere quello di approfondire le caratteristiche degli altri algoritmi, e magari sfruttarle integrando le più vantaggiose.

Anche nel caso dell'Hoa Binh l'approccio utilizzato ha prodotto una buona popolazione di soluzioni, di cui una parte domina fortemente la gestione storica. L'applicazione pratica delle politiche prodotte nel caso di studio potrebbe pertanto generare notevoli benefici sociali ed economici ai diversi portatori di interesse che sono direttamente influenzati dalla gestione dell'Hoa Binh.

L'utilizzo delle reti neurali con l'invaso e il tempo in ingresso ha permesso di approssimare le politiche di regolazione riducendo notevolmente il numero di parametri da stimare. In questo modo è infatti sufficiente un singolo set di parametri (dipendente dall'architettura adottata) per identificare un'intera politica, mentre normalmente sarebbe stato necessario stimare un numero di T leggi di controllo pari al periodo di ciclostazionarietà (solitamente 365). Grazie all'approccio utilizzato, è stato possibile ridurre immensamente la complessità del processo di ottimizzazione, sfruttando le innovative e flessibili caratteristiche delle reti neurali.

L'introduzione delle regole semplici è risultata essere una grande intuizione. Essa permette infatti di ottenere molteplici risultati: comprendere a priori quale architettura neurale sarà in grado di funzionare bene durante l'ottimizzazione, e restringere notevolmente il range di ricerca nello spazio dei parametri. Inoltre, il caso di Test ha mostrato inequivocabilmente che l'introduzione delle parametrizzazioni delle regole semplici nell'inizializzazione migliora le prestazioni dell'algoritmo. Questi vantaggi sono conseguibili grazie alla capacità di apprendimento delle reti neurali.

Tra i parametri di tuning dell'algoritmo evolutivo quello che sembra avere un impatto maggiore è sicuramente il numero di generazioni. E' infatti necessario che esse siano in numero sufficiente per garantire all'algoritmo la convergenza a buoni risultati. La grandezza della popolazione ha un impatto inferiore, ma può comunque diventare rilevante, specie quando la rete neurale diventa più complessa e quindi aumenta il numero di parametri da stimare.

La valutazione dei risultati di un algoritmo a molti obiettivi è assai complessa, dal momento che non si tratta più di valutare una singola soluzione ma una intera frontiera di soluzioni efficienti. In letteratura è presente una grande varietà di indicatori, che sono però soggetti a diversi difetti (per esempio, alcuni richiedono la conoscenza a priori della frontiera di Pareto ottima; altri possono solo confrontare due frontiere senza dare un giudizio in senso assoluto). La combinazione di indicatori sfruttata in questo lavoro ha permesso di valutare a 360 gradi le prestazioni di diversi esperimenti di ottimizzazione. Essi infatti forniscono una molteplicità di informazioni complementari sulla bontà di un set di soluzioni: il grado di convergenza verso l'ottimo (Hypervolume) e il grado di approssimazione della frontiera in termini di numero di soluzioni efficienti (N_{soleff}) e di dispersione delle soluzioni (Crowding Distance). Infine, il numero esperimenti giunti a buon fine N_{conv} permette di generalizzare i risultati rispetto al seme (sezione 2.5), questione molto importante nel caso di algoritmi che sfruttano la generazione casuale di numeri da parte del calcolatore.

La valutazione dei risultati nel presente lavoro è stata in definitiva chiara e sensibilmente agevolata grazie agli indicatori utilizzati.

Le leggi di controllo ottenute nel caso di Test hanno evidenziato nella quasi totalità dei casi coerenza con le aspettative. Anche nel caso dell'Hoa Binh le leggi di controllo hanno mostrato un trend sensato, e hanno fornito alcune informazioni interessanti, come la netta dipendenza dalla variabile temporale. Il processo di ottimizzazione tuttavia non discrimina quelle parametrizzazioni che corrispondono a leggi di controllo a tratti fisicamente insensate, dal momento che in fase di simulazione entrano in gioco i vincoli fisici. Queste parametrizzazioni possono dunque risultare efficienti, nonostante il loro andamento. Ad esempio, nella sezione 3.3.1 si è visto come nell'inizializzazione parametrizzazioni estremamente diverse fossero rappresentate tutte da un unico punto corrispondente alla scala di massimo rilascio. Se fosse stato possibile eliminare da subito tutte quelle con andamenti fisicamente assurdi, sarebbe stato presumibilmente più agevole esplorare l'intera frontiera. Un tema che sarebbe interessante approfondire in futuri studi è pertanto

quello dell'imposizione di vincoli nelle leggi di controllo. Anzichè farlo a valle (modificando i controlli non fisicamente ammissibili) sarebbe utile implementare all'interno dell'algoritmo un vincolo sulle parametrizzazioni, eliminando durante l'ottimizzazione stessa tutte quelle soluzioni che non soddisfano i requisiti del buon senso. Ovviamente questa è un'operazione molto delicata, in quanto in casi complessi la definizione di quale sia il limite del buon senso può risultare molto complicata. Un simile sviluppo merita quindi approfonditi studi, ma potrebbe agevolare il processo di ottimizzazione, scartando sul nascere leggi di controllo insensate ma efficaci a causa dei vincoli fisici del simulatore.

Alcuni risultati contrastanti tra calibrazione e validazione sia nel Test che nell'Hoa Binh, mettono in evidenza che la forma della funzione obiettivo (1.20) è una forte approssimazione. Nel Test si è visto infatti come per convergere con precisione a un certo risultato servano valori di h molto grandi (almeno $200 * T$, con T periodo di ciclostazionarietà). La coerenza nelle leggi di controllo ottenute, tuttavia, viene quasi sempre mantenuta anche con valori di h molto bassi (fino a circa $6 * T$). La stabilità di un risultato è comunque fortemente variabile con l'orizzonte temporale h , in relazione al caso in esame e alla particolare serie di afflussi utilizzata. In ogni caso, anche i risultati del caso di studio hanno dimostrato di mantenere sempre una certa coerenza di fondo.

In futuro, approfondire gli studi in questo senso può comunque essere proficuo. Tenere in maggiore considerazione la variabilità dei disturbi (rispetto a una semplice simulazione deterministica), può infatti garantire un enorme aumento della stabilità dei risultati. Raggiungere un simile obiettivo non è comunque facile, a patto di mantenere bassi gli oneri computazionali.

Bibliografia

- Kanpur genetic algorithms laboratory, 2005. URL <http://www.iitk.ac.in/kangal/codes.shtml>.
- Matlab central, NSGA - II: A multi-objective optimization algorithm, 2009. URL <http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii-a-multi-objective-optimization-algorithm>.
- A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(33):930–945, 1993.
- J.D. Barrow. *Theories of everything. The quest for ultimate explanation*. Clarendon Press, Oxford, EN, 1991.
- R.E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957.
- D.P. Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, New York, 1976.
- K. Deb. *Multiobjective optimization using evolutionary algorithms*. John Wiley & Sons, Chichester, EN, 2001.
- K. Deb and S. Jain. Running performance metrics for evolutionary multiobjective optimization. Technical report, KanGAL, 2002.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-III. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Reading, MA, 1989.
- GWP. *ToolBox Integrated Water Resource Management*. Global Water Partnership. Stockholm, S., 2003.

- J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- T. Kim, J.H. Heo, D.H. Bae, and J.H. Kim. Single-reservoir operating rules for a year using multiobjective genetic algorithm. *Journal of Computing in Civil Engineering*, 10(2):163–179, 2008.
- F. Pianosi and R. Soncini-Sessa. Extended Ritz method for reservoir management over an infinite horizon. In *17th IFAC World Congress, 6-11 July, Seoul, KR, 2008*.
- R. Soncini Sessa. *MODSS*. McGraw Hill, Milano, IT, 2004.
- Y.S. Su and R.A. Deininger. Generalization of Whites method of successive approximations. *Operations research*, 20(2):318–326, 1972.
- D.J. White. Dynamic programming, Markov chains, and the method of successive approximations. *Journal of Mathematical Analysis and Applications*, 6:373–376, 1963.
- W.W.G. Yeh. Reservoir management and operation models: A state-of-the-art review. *Water Resour. Res.*, 21(12):1797–1818, 1985.
- E. Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*. PhD thesis, Swiss Federal Institute of Technology Zurich, 1999.
- E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Berlin, D, 1998.
- R. Zoppoli, M. Sanguineti, and T. Parisini. Approximating networks and extended Ritz method for the solution of functional optimization problems. *Journal of Optimization Theory and applications*, 112(2):403–440, 2002.