POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Energetica e Nucleare

Ottimizzazione dei consumi energetici di una rete idrica in uno stabilimento industriale

Tutor

Prof. Marco Badami

Candidata

Fulvia Piccinnu

A.A. 2020/2021

ABSTRACT

Il tema del risparmio delle risorse è ormai preponderante nelle agende politiche in tutto il mondo e il recente intensificarsi dei cambiamenti climatici ha richiesto un aumento delle risorse sia economiche che tecnologiche per permettere una evoluzione in questa direzione. L'ottimizzazione energetica nelle aziende produttive permette il risparmio di una ingente quantità di energia ma anche di materie prime, come l'acqua.

Nel presente lavoro si persegue l'obiettivo di simulare la rete idrica di una industria tessile tramite reti neurali, per poi ottimizzare il suo funzionamento tramite l'utilizzo degli algoritmi genetici. In particolare il lavoro si sofferma sulla ottimale regolazione delle pompe per il sollevamento delle acque da alcuni pozzi sotterranei.

La creazione del modello fisico avviene inizialmente tramite il software EPANET, che permette di simulare qualsiasi rete idrica. Grazie agli output di EPANET è stato possibile allenare le reti neurali, che simulano il comportamento della rete sottoposta a diversi tipi di controlli delle pompe presenti. Una volta creato il modello grazie alle reti neurali, si è cercato di ottimizzarlo attraverso l'utilizzo degli algoritmi genetici.

Oltre a EPANET, l'altro software utilizzato per sviluppare reti neurali e algoritmi genetici è MATLAB.

Indice

Indice delle figure 4
Indice delle tabelle 8
INTRODUZIONE 9
1 Disponibilità e consumi idrici 11
2 Simulazione di una rete idrica tramite EPANET 17
2.1 Il software EPANET 17
2.2 Gli elementi della rete 20
2.3 Costruzione di un modello 23
3 Reti neurali 28
3.1 Introduzione alle reti neurali artificiali 32
3.2 Reti neurali su MATLAB 37
4 Algoritmi genetici 41
4.1 Introduzione agli algoritmi genetici 41
4.2 Algoritmi genetici su MATLAB 45
5 Analisi del caso studio 49
5.1 Simulazione della rete tramite EPANET 51
5.2 Ottimizzazione della regolazione del sistema tramite algoritmi genetici 57
5.3 Sviluppo del modello surrogato tramite reti neurali 80
5.4 Ottimizzazione tramite algoritmi genetici del modello surrogato 97
6 Conclusioni 103
BIBLIOGRAFIA e SITOGRAFIA 108

Indice delle figure

Figura 1. Confronto tra popolazione mondiale e prelievo di acqua dal 1900 al 2010 11
Figura 2. Prelievo mondiale di acqua da parte dei principali settori dal 1900 al 2010 12
Figura 3. Confronto tra i vari continenti del prelievo di acqua da parte dei principali settori 13
Figura 4. Domanda di acqua per uso industriale per continente, confronto 2010 e 2050 13
Figura 5. Impatto ambientale dell'industria tessile nel mondo nel 2015 15
Figura 6. Consumo delle risorse non rinnovabili da parte dell'industria tessile. Fonte: Circular Fibres initiative analysis..... 16
Figura 7. Schermata iniziale del software EPANET 18
Figura 8. Tabella con coefficienti per calcolo perdita di carico nei tubi su EPANET 20
Figura 9. Tabella dei coefficienti di rugosità (EPANET) 21
Figura 10. Simbolo dell'elemento tubo nel software EPANET..... 21
Figura 11. Simbolo dell'elemento pompa nel software EPANET 22
Figura 12. Simbolo dell'elemento valvola nel software EPANET 22
Figura 13. Simboli degli elementi nodi nel software EPANET 23
Figura 14. Interfaccia grafica di EPANET per iniziare..... 23
Figura 15. Esempio costruzione rete 24
Figura 16. Property Editor dei vari elementi su EPANET 25
Figura 17. Inserimento della curva caratteristica della pompa su EPANET 25
Figura 18. Options su EPANET 26
Figura 19. Proprietà idrauliche su EPANET 26
Figura 20. Time Options nelle Options di EPANET..... 27
Figura 21. Esempio di Machine Learning..... 28
Figura 22. Rete neurali biologiche e reti neurali artificiali 30
Figura 23. Neurone al microscopio ottico. Fonte: Fanny CASTETS - Opera propria, CC BY-SA 3.0 31
Figura 24. Struttura di una rete neurale artificiale..... 32
Figura 25. Schematizzazione matematica di una rete neurale..... 33
Figura 26. Rete neurale feedforward a singolo strato 34
Figura 27. Rete neurale feedforward multistrato..... 35
Figura 28. Rete neurale ricorsiva con neuroni nascosti [18] 36
Figura 29. Neural Network Time Series app su MATLAB 37

Figura 30. Diagramma NARXNET 38
Figura 31. Architettura NARXNET (1/2) 38
Figura 32. Architettura NARXNET (2/2) 39
Figura 33. Neural Network Fitting app su MATLAB 40
Figura 34. Distribuzione popolazione iniziale - GA 42
Figura 35. Distribuzione popolazione nella generazione 60, 80, 95 e 100 - GA 43
Figura 36. Opzioni di default – funzione ga MATLAB 46
Figura 37. Tipi di figli nell'algoritmo genetico su Matlab 47
Figura 38. Schema del caso studio 49
Figura 39. Numero massimo di avviamenti orari per le pompe con installazione a secco e sommerse 50
Figura 40. Schema del fungo nel caso reale 50
Figura 41. Schema del caso reale su EPANET 51
Figura 42. Inserimento curva caratteristica e curva di efficienza pompa su EPANET 51
Figura 43. Inserimento caratteristiche del fungo su EPANET 52
Figura 44. Impostazione Times Options su EPANET 52
Figura 45. Controlli ON-OFF impostati alle pompe 53
Figura 46. Controlli ON-OFF impostati alle pompe su EPANET 53
Figura 47. Output grafico di EPANET: richiesta di acqua da parte dell'utenza 54
Figura 48. Output grafico di EPANET: livello acqua nel fungo 54
Figura 49. Output grafico di EPANET: portata di acqua attraverso le 3 pompe 54
Figura 50. Output grafico di EPANET: dettaglio portata di acqua attraverso la pompa1 55
Figura 51. Output tabellare di EPANET: richiesta di acqua da parte dell'utenza 55
Figura 52. Output tabellare di EPANET: portata di acqua attraverso la pompa1 56
Figura 53. Esempio codice principale GA: CASO 2 - 1 pompa 58
Figura 54. Esempio codice principale GA: CASO 4 - 3 pompe 58
Figura 55. Esempio codice con fitness GA: CASO 2 - 1 pompa 59
Figura 56. Output grafici: caso base - 1 pompa 60
Figura 57. Confronto CASO BASE - CASO 1: 1 pompa 61
Figura 58. GA: caso 1 - 1 pompa 61
Figura 59. Output grafici: caso 1 - 1 pompa 63
Figura 60. Confronto CASO BASE - CASO 2: 1 pompa 63
Figura 61. GA: caso 2 - 1 pompa 64
Figura 62. Output grafici: caso 2 - 1 pompa 65
Figura 63. Riepilogo confronti GA - 1 pompa 65

Figura 64. Output grafici: caso base - 3 pompe 67
Figura 65. Confronto CASO BASE - CASO 1: 3 pompe 68
Figura 66. GA: caso 1 - 3 pompe 68
Figura 67. Output grafici: caso 1 - 3 pompe 70
Figura 68. Confronto CASO BASE - CASO 2: 3 pompe 71
Figura 69. GA: caso 2 - 3 pompe 71
Figura 70. Output grafici: caso 2 - 3 pompe 72
Figura 71. Confronto CASO BASE - CASO 3: 3 pompe 73
Figura 72. GA: caso 3 - 3 pompe 73
Figura 73. Output grafici: caso 3 - 3 pompe 75
Figura 74. Confronto CASO BASE - CASO 4: 3 pompe 76
Figura 75. GA: caso 4 - 3 pompe 76
Figura 76. Output grafici: caso 4 - 3 pompe 78
Figura 77. Riepilogo confronti GA - 3 pompe 79
Figura 78. Flowchart modello NN 80
Figura 79. Parte di script allenamento NN1 82
Figura 80. Parte di script allenamento NN2 83
Figura 81. Modello NN: regolazione ON-OFF 83
Figura 82. Modello NN: regolazione con valvole 84
Figura 83. Regolazione ON-OFF pompe 85
Figura 84. Regressione risultati ON-OFF 1 ora 86
Figura 85. Confronto livello acqua fungo ON-OFF 1 ora 86
Figura 86. Confronto portate pompe ON-OFF 1 ora 87
Figura 87. Regressione risultati ON-OFF 10 minuti 87
Figura 88. Confronto livello acqua fungo ON-OFF 10 minuti 88
Figura 89. Confronto portate pompe ON-OFF 10 minuti 88
Figura 90. Regressione risultati ON-OFF 1 minuto 89
Figura 91. Confronto livello acqua fungo ON-OFF 1 minuto 89
Figura 92. Confronto portate pompe ON-OFF 1 minuto 89
Figura 93. Regolazione Valvole 91
Figura 94. Regressione risultati Valvole 1 ora 92
Figura 95. Confronto livello acqua fungo Valvole 1 ora 92
Figura 96. Confronto portate pompe Valvole 1 ora 93
Figura 97. Regressione risultati Valvole 10 minuti 94
Figura 98. Confronto livello acqua fungo Valvole 10 minuti 94

Figura 99. Confronto portate pompe Valvole 10 minuti 94
Figura 100. Regressione risultati Valvole 1 minuto 95
Figura 101. Confronto livello acqua fungo Valvole 1 minuto 95
Figura 102. Confronto portate pompe Valvole 1 minuto 95
Figura 103. Confronto risultati ottimizzazione ON-OFF 98
Figura 104. Codice: ottimizzazione ON-OFF 99
Figura 105. Ga: ottimizzazione ON-OFF 99
Figura 106. Codice: ottimizzazione valvole 101
Figura 107. Ga: ottimizzazione ON-OFF 102
Figura 108. Grafico risultati: GA - 1 pompa 103
Figura 109. Grafico risultati: GA - 3 pompe 104
Figura 110. Grafico risultati: ottimizzazione ON-OFF 106

Indice delle tabelle

Tabella 1. Riepilogo regolazione GA - 1 pompa..... 57
Tabella 2. Riepilogo regolazione GA - 3 pompe..... 57
Tabella 3. Confronto modello NN ON-OFF 86
Tabella 4. Confronto modello NN Valvole..... 92
Tabella 5. GA: ottimizzazione ON-OFF 97
Tabella 6. Confronto risultati ottimizzazione ON-OFF 98
Tabella 7. GA: ottimizzazione valvole..... 100
Tabella 8. Confronto risultati ottimizzazione valvole..... 100
Tabella 9. Riassunto risultati NN..... 105
Tabella 10. Ipotesi di partenza e t necessario ottimizzazione ON-OFF 105
Tabella 11. Ipotesi di partenza e t necessario: ottimizzazione con valvole..... 106
Tabella 12. Tabella risultati: ottimizzazione con valvole..... 106

INTRODUZIONE

In questo lavoro di tesi è stato utilizzato il Machine Learning per l'ottimizzazione energetica di un impianto idrico asservito a una industria tessile: in particolare sono state utilizzare le reti neurali per creare un modello del sistema, e successivamente gli algoritmi genetici per cercare di ottimizzare il sistema stesso.

Nel primo capitolo viene trattata l'importanza dell'efficientamento nel settore idrico, partendo dai dati sui consumi energetici e sul prelievo dell'acqua.

Nel secondo capitolo viene introdotto il software EPANET, utilizzato per simulare la rete idrica e ricavare i dati necessari ad allenare le reti neurali.

Nel terzo capitolo, dopo una generale panoramica sulle reti neurali, sono state introdotte le reti neurali scelte per questa trattazione tra quelle presenti su MATLAB.

Nel quarto capitolo ci si è focalizzati sull'ottimizzazione e in particolare sulla funzione *algoritmo genetico* presente su MATLAB.

Nel quinto capitolo è stato presentato il caso studio. Dapprima è stata simulata la rete idrica grazie all'utilizzo del software EPANET. Successivamente è stata ottimizzata la rete idrica grazie all'utilizzo di MATLAB e dell'EPANET MATLAB Toolkit: sono stati cambiati i controlli sulle pompe introducendo i valori ottimali trovati dall'algoritmo genetico. Dopodiché è stato sviluppato un modello surrogato tramite l'utilizzo delle reti neurali con l'utilizzo del software MATLAB. Infine è stato ottimizzato con gli algoritmi genetici il modello precedentemente creato con le reti neurali.

La funzione obiettivo da minimizzare scelta per questa trattazione, è il consumo di energia da parte delle pompe. Sono stati analizzati due tipi di regolazione: ON-OFF e valvole. Di conseguenza, le variabili lasciate libere di mutare sono state:

- per la regolazione ON-OFF, il livello minimo e quello massimo raggiunto dall'acqua presente nel serbatoio, in cui le pompe si accenderanno e spegneranno;

- pe la regolazione con valvole, il livello raggiunto dall'acqua presente nel serbatoio in cui varia l'apertura delle valvole.

Capitolo 1

Disponibilità e consumi idrici

L'acqua è una risorsa indispensabile alla vita sul nostro pianeta. In certe zone del mondo essa è limitata e non è garantita la fruibilità da parte di tutti. In altre zone, come la nostra, avvengono invece grandi sprechi di questo prezioso bene e gli scenari futuri diventano sempre più critici e incerti. Per questo motivo ottimizzare il suo uso diventa fondamentale.
Il grafico nella Figura 1 mostra a livello mondiale il prelievo di acqua e la crescita della popolazione nell'ultimo secolo.

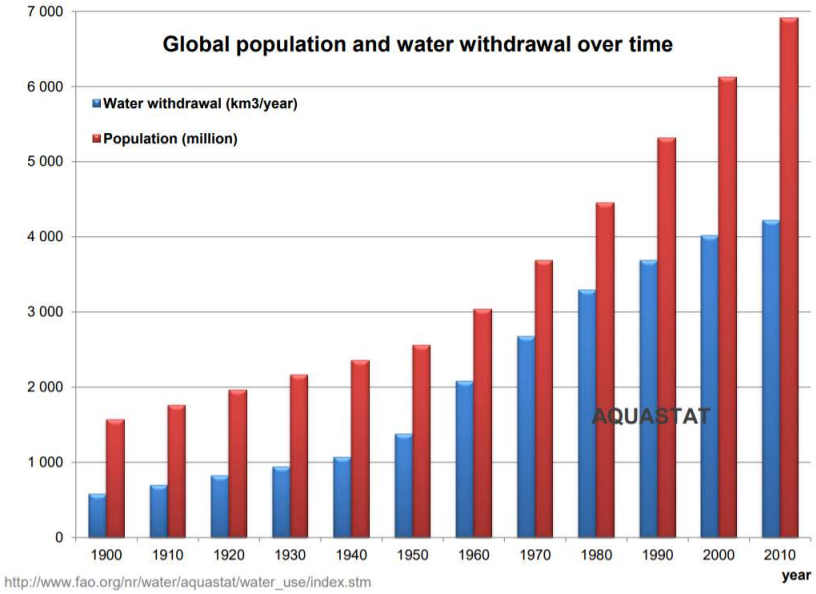


Figura 1. Confronto tra popolazione mondiale e prelievo di acqua dal 1900 al 2010

La popolazione mondiale è cresciuta di 4,4 volte nell'ultimo secolo mentre i prelievi d'acqua sono aumentati di 7,3 volte nello stesso periodo: perciò i prelievo di acqua sono aumentati 1,7 volte più velocemente della popolazione mondiale. Si può però anche notare dal grafico che mentre la popolazione mondiale continua a crescere esponenzialmente, l'aumento di prelievi di acqua ha rallentato negli ultimi decenni. Tra il 1900 e il 2010 la popolazione mondiale è cresciuta del 340% o del 1,3% l'anno in media, variando dal 1% negli anni dal 1900 al 1940, fino al 1,6% negli anni dal 1970 al 2010. Tra il 1900 e il 2010 il prelievo di acqua è cresciuto del 630% o del 1,8% all'anno in media, variando dal 1,5% negli anni dal 1900 al 1940, al 1,1% negli anni dal 1970 al 2010. La

crescita più elevata nel prelievo di acqua è avvenuta tra il 1950 e il 1960, con il 4,2% in un anno, mentre è stata solo dello 0,5% negli anni tra il 2000 e il 2010 [10].

Si distinguono 3 principali tipi di prelievo di acqua a livello mondiale: agricoltura (che comprende irrigazione, allevamento e acquacoltura), acquedotti (che include l'uso domestico) e l'industria.
Nel grafico presente in Figura 2 viene mostrato il prelievo mondiale di acqua nell'ultimo secolo a opera dei 3 settori appena citati (agricoltura, acquedotti e industria) e l'evaporazione dell'acqua dalla superficie dei laghi artificiali che si formano in seguito alla costruzione delle dighe. Quest'ultimo non è propriamente un prelievo di acqua, ma può essere considerato come un consumo di acqua antropogenico, in quanto questa evaporazione non sarebbe avvenuta senza la costruzione della diga da parte dell'umano, per esempio per produrre elettricità tramite l'idroelettrico.

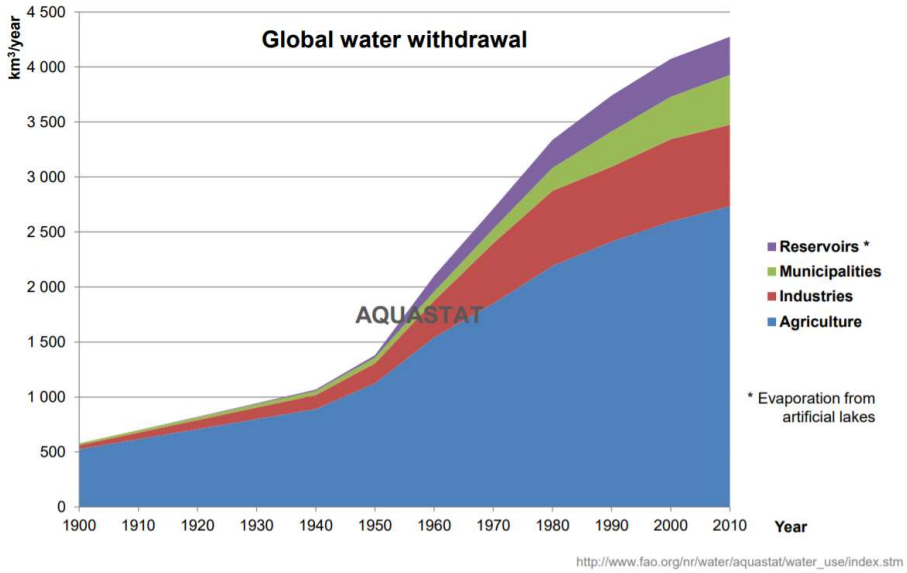


Figura 2. Prelievo mondiale di acqua da parte dei principali settori dal 1900 al 2010

A livello globale il 69% dell'acqua viene prelevata per far fronte alla richiesta per l'agricoltura, il 19% per le industrie e il 12% per gli acquedotti. Queste percentuali variano vistosamente da regione a regione, come si può notare dal grafico presente nella Figura 3.

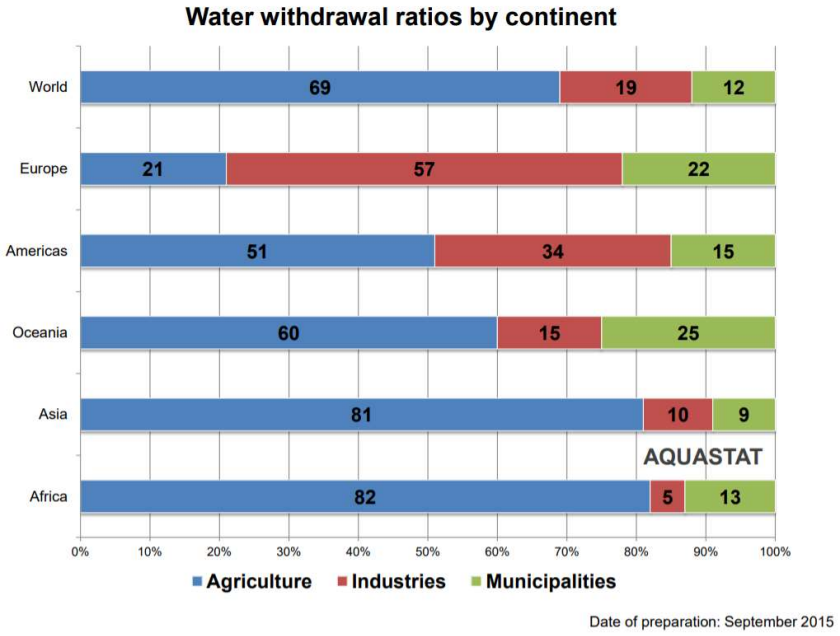


Figura 3. Confronto tra i vari continenti del prelievo di acqua da parte dei principali settori

Il prelievo di acqua destinato all'agricoltura dipende molto sia dal clima dei paesi che dall'importanza dell'agricoltura nell'economia del paese. In alcuni continenti, come Africa e Asia l'acqua prelevata per l'agricoltura arriva addirittura a più dell'80% del totale, mentre in Europa arriva appena al 21% [10].

Si prevede, come mostrato nella Figura 4, che la quota dell'industria e dei settori energetici nella domanda globale di acqua crescerà fino al 24% entro il 2050, con i maggiori aumenti assoluti in Asia ed Africa, mentre l'America settentrionale sarà l'unica regione con dati in diminuzione [12].

	2010 (km³/anno)	Percentuale dell'uso totale dell'acqua del continente	2050 (km³/anno)	Percentuale dell'uso totale dell'acqua del continente	Tasso di cambiamento 2050 (km³/anno) come % del 2010 (km³/anno)
Africa	18	8%	64	18%	353%
Asia	316	10%	760	19%	240%
America settentrionale e centrale	229	35%	182	27%	80%
America meridionale	31	19%	47	21%	153%
Europa	241	54%	325	58%	135%
Oceania	2	5%	3	7%	144%
Mondo	**838**	**18%**	**1.381**	**24%**	**165%**

Figura 4. Domanda di acqua per uso industriale per continente, confronto 2010 e 2050

Nonostante il tasso di aumento dell'uso di acqua dolce si sia ridotto in quasi tutti gli Stati membri dell'Organizzazione per la cooperazione e lo sviluppo economico (OCSE), dove i tassi di utilizzo dell'acqua pro capite tendono a essere tra i più alti al mondo, esso continua ad aumentare nella

maggioranza delle economie emergenti, nonché nei paesi a basso e medio reddito. Gran parte di questa crescita può essere attribuita all'insieme dei seguenti fattori: crescita della popolazione, sviluppo economico e mutevoli modelli di consumo [6].

A causa della sempre più alta attenzione nei confronti dei cambiamenti climatici è indispensabile ottimizzare i sistemi di distribuzione della risorsa idrica dal punto di vista energetico: sia per quanto riguarda le reti di distribuzione già esistenti (tramite interventi di riqualificazione) sia per le reti di distribuzione di nuova costruzione (tramite una progettazione attenta e ottimale).

Negli ultimi decenni l'utilizzo del Machine Learning si è diffuso in modo prorompente nella risoluzione di molteplici problemi ingegneristici. In particolare, l'impiego dell'Intelligenza Artificiale ha trovato applicazione nel campo della visione artificiale, riconoscimento vocale, controlli robotici [9] e, negli ultimi anni, anche in altre branche dell'ingegneria. L'ottimizzazione nella gestione delle reti idriche mediante tali tecniche rispecchia appieno questa tendenza.
I modi più immediati per attuare l'ottimizzazione dei sistemi idrici sono cercare di limitare le perdite lungo la rete e utilizzare pompe per il sollevamento dell'acqua più efficienti in modo da diminuire i consumi energetici.
In Italia il 5% del fabbisogno elettrico nazionale è utilizzato dal settore idrico e i costi di *operation and maintenance* di una stazione di pompaggio di un impianto idrico pesano per il 60% rispetto ai costi totali [2].
Una corretta gestione degli impianti idrici è indispensabile per uno sviluppo sostenibile. Inoltre essa ha un impatto su vari settori:

- Economico, maggiore è la percentuale di risorsa persa, maggiore è la quantità di acqua che bisogna immettere in rete con conseguente aumento dei costi di estrazione, trattamento e trasporto;
- Ambientale, l'aumento della risorsa richiesta dalla rete porta ad un maggiore stress per le fonti di approvvigionamento e a maggiori emissioni in atmosfera a causa della quantità maggiore di energia necessaria per le operazioni di depurazione, potabilizzazione e pompaggio dell'acqua;
- Sociale, le perdite idriche causano disagi ai consumatori a causa della carenza di approvvigionamento.

E' necessario comprendere che il concetto di acqua come risorsa rinnovabile illimitata deve appartenere al passato: l'acqua dolce infatti scarseggia sempre di più e oltre 2 miliardi di persone vivono in aree già soggette a stress idrico. Secondo valutazioni indipendenti, il mondo dovrà

affrontare un deficit idrico globale del 40% entro il 2030. Questa situazione è aggravata da sfide globali come il COVID-19 e i cambiamenti climatici [6].

L'industria tessile in particolare ha bisogno di utilizzare molto acqua per poter mandare avanti le proprie lavorazioni, senza contare l'impiego dei terreni adibiti alla coltivazione di cotone e di altre fibre. Si stima che l'industria tessile e dell'abbigliamento a livello globale abbia utilizzato 79 miliardi di metri cubi di acqua nel 2015. Alcune stime indicano che per la fabbricazione di una maglietta di cotone occorrano 2.700 litri di acqua dolce, un volume pari a quanto una persona dovrebbe bere in 2 anni e mezzo [12].

Figura 5. Impatto ambientale dell'industria tessile nel mondo nel 2015

In Italia quasi il 6,5% del fabbisogno di acqua è richiesto dalle industrie tessili [7].

Nella Figura 6 è mostrato come secondo degli studi il consumo di risorse non rinnovabili da parte dell'industria tessile sia destinato ad aumentare dal 2015 al 2050.

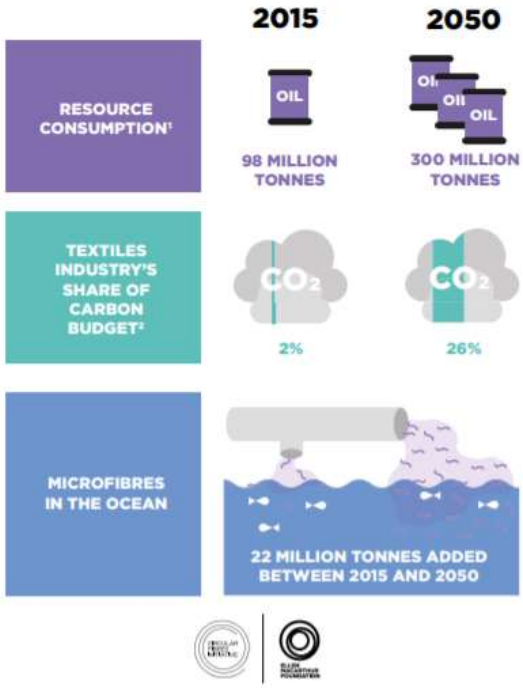


Figura 6. Consumo delle risorse non rinnovabili da parte dell'industria tessile. Fonte: Circular Fibres initiative analysis

Visto il grande dispendio di acqua che caratterizza l'industria tessile, e di energia che caratterizza invece il settore idrico in generale, il presente lavoro di ottimizzazione mira proprio a diminuire i sopracitati sprechi.

Capitolo 2

Simulazione di una rete idrica tramite EPANET

Il modello fisico del sistema idrico preso in esame per il seguente lavoro è stato creato utilizzando il software EPANET, il più usato in questo ambito per la progettazione e verifica delle reti idriche. L'utilizzo di Epanet permette di verificare il comportamento idraulico e la qualità delle acque all'interno di un sistema di distribuzione dell'acqua in pressione, sia in condizioni stazionarie che quasi stazionarie.

2.1 Il software EPANET

EPANET 2.0 (Rossman, 2000) è un software di simulazione idraulica sviluppato dall'EPA (Environmental Protection Agency), ossia dall'Agenzia per la protezione dell'ambiente degli Stati Uniti d'America.

Esso è in grado di fornire numerose informazioni: portata circolante in ogni condotta della rete, pressione ai nodi, livello raggiunto dall'acqua nei serbatoi, concentrazione di una data sostanza nella rete di distribuzione, etc.

L'accuratezza del modello è strettamente correlata alla conoscenza del sistema fisico e dell'utenza da esso servita: il livello di dettaglio perseguibile nell'analisi è legato alla quantità e alla tipologia di dati che si hanno a disposizione.

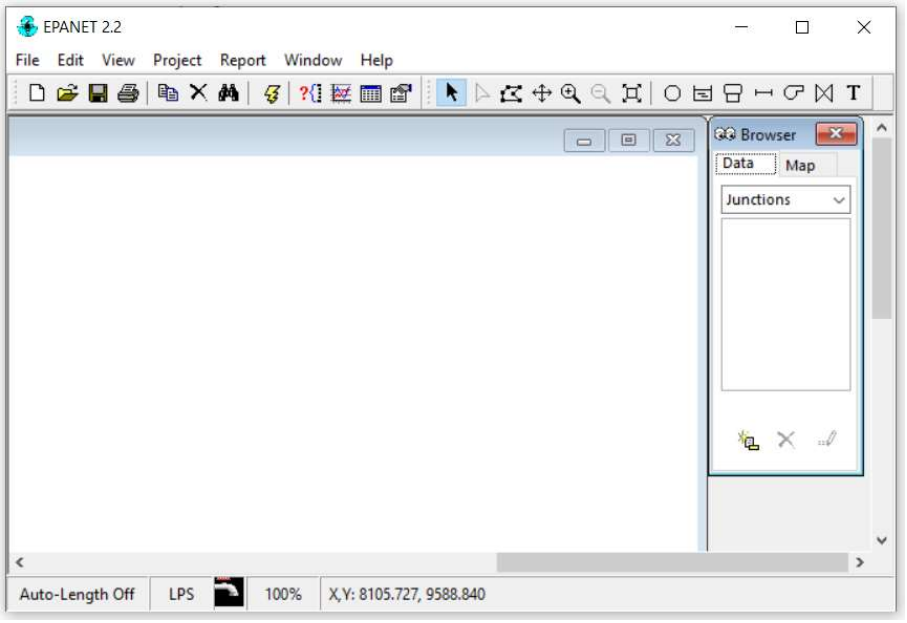


Figura 7. Schermata iniziale del software EPANET

I dati in ingresso riguardano caratteristiche fisiche della rete idrica presa in esame (topologia, diametri, lunghezze, quote degli elementi, etc,), parametri idraulici (coefficiente di perdita di carico delle valvole, scabrezza, etc), condizioni di esercizio (distribuzione delle portate erogate ai nodi) e altre condizioni come livelli nei serbatoi e stato di funzionamento delle pompe. Alcuni dei parametri in ingresso sono di facile determinazione (come diametri e lunghezze delle condotte), mentre altri (come la scabrezza dei tubi) non sono direttamente misurabili. L'operatore deve perciò effettuare una stima, affetta da un certo grado di incertezza, che sarà poi la causa della differenza esistente tra modello di simulazione e realtà.

Il modello può essere definito tramite una serie di equazioni che descrivono il comportamento della rete, attraverso le quali, noti i parametri in ingresso, si ricavano le grandezze incognite. La rete di distribuzione idrica può essere vista come un insieme di elementi di collegamento (condotte, valvole e pompe, ciascuno aventi determinate proprietà) connessi tra loro da punti di estremità, detti nodi, che sono i punti della rete in cui gli elementi si uniscono tra loro. I nodi possono essere: punti di consumo di acqua (nodi di domanda), punti di entrata per l'acqua (nodi di sorgente o pozzi), serbatoi o cisterne (nodi di immagazzinamento). Ad essi si aggiungono elementi detti non fisici come curve, andamenti temporali e controlli.

Il sistema di equazioni che descrive il moto nelle condotte è costituito da:

- Equazioni di continuità nei nodi: esprimono il principio di conservazione della massa, ossia che la portata entrante in un nodo è complessivamente uguale a quella uscente. Sono equazioni lineari e possono essere scritte per ogni nodo.

$$\sum_{i=1}^{n} q_j + Q_i = 0$$

  Dove:

  - q_j è la portata proveniente dal tratto j-esimo convergente nel nodo i-esimo;

  - Q_i è la portata concentrata nel i-esimo nodo.

- Equazioni di continuità dei carichi nei tronchi della rete: legano le perdite di carico alla portata circolante nel tronco. Sono equazioni non lineari, una per ogni tratto "l".

$$\Delta H_j = \beta \cdot \frac{q_j^{\alpha}}{D_j^{\gamma}} \cdot L_j$$

  Dove:

  - ΔH_j indica la perdita di carico;

  - q_j è la portata circolante;

  - D_j rappresenta il diametro;

  - L_j è la lunghezza del tratto;

  - α e γ sono esponenti caratterizzanti della formula;

  - β è il coefficiente di scabrezza della condotta, dipendente dal materiale di cui è costituita la stessa.

Noti i diametri, le lunghezze e le scabrezze delle condotte, le portate erogate lungo i tratti e le quote piezometriche dei serbatoi, le incognite del problema sono le pressioni ai nodi della rete e i valori delle portate circolanti nei tronchi. Il numero delle equazioni è pari al numero delle incognite; il sistema viene risolto con metodo iterativo.

Il software permette di osservare l'evoluzione delle principali variabili del sistema in tutti i nodi o rami della rete ad intervalli di tempo prefissati, e di riassumere i risultati su tabelle e grafici.

2.2 Gli elementi della rete

Verranno analizzati di seguito i singoli elementi della rete:

- Tubi (pipes): trasportano l'acqua da un nodo all'altro. La direzione di scorrimento del flusso va dall'estremità a carico maggiore a quella a carico minore. La perdita di carico dovuta all'attrito che ha luogo nel passaggio nei tubi si esprime attraverso la relazione:

$$h_L = a * q^b$$

dove:

  - h_L è la perdita di carico in piedi
  - q indica la portata in cubic foot for seconds
  - a è un coefficiente di resistenza
  - b è l'esponente della portata

EPANET può utilizzare tre tipi di formule (quindi di coefficienti): la formula di Hazen-Williams, la più utilizzata; quella di Darcy-Weisbach, la più adatta a flussi in moto laminare e a fluidi diversi dall'acqua; quella di Chezy-Manning, usata per canali aperti. Sono riportati nella tabella di seguito, presente nel manuale utente di EPANET, il coefficiente di resistenza(A) e l'esponente di portata(B) relativi a ciascuna legge:

Formula	*Resistance Coefficient (A)*	*Flow Exponent (B)*
Hazen-Williams	$4.727\ C^{-1.852}\ d^{-4.871}\ L$	1.852
Darcy-Weisbach	$0.0252\ f(\varepsilon,d,q)d^{-5}L$	2
Chezy-Manning	$4.66\ n^2\ d^{-5.33}\ L$	2

Notes: C = Hazen-Williams roughness coefficient
ε = Darcy-Weisbach roughness coefficient (ft)
f = friction factor (dependent on ε, d, and q)
n = Manning roughness coefficient
d = pipe diameter (ft)
L = pipe length (ft)
q = flow rate (cfs)

Figura 8. Tabella con coefficienti per calcolo perdita di carico nei tubi su EPANET

Material	*Hazen-Williams C (unitless)*	*Darcy-Weisbach ε (feet x 10^{-3})*	*Manning's n (unitless)*
Cast Iron	130 – 140	0.85	0.012 - 0.015
Concrete or Concrete Lined	120 – 140	1.0 - 10	0.012 - 0.017
Galvanized Iron	120	0.5	0.015 - 0.017
Plastic	140 – 150	0.005	0.011 - 0.015
Steel	140 – 150	0.15	0.015 - 0.017
Vitrified Clay	110		0.013 - 0.015

Figura 9. Tabella dei coefficienti di rugosità (EPANET)

Figura 10. Simbolo dell'elemento tubo nel software EPANET

- Pompe: per pompa si intende un componente atto a sollevare una certa quantità di acqua. La curva caratteristica è la relazione che lega la portata elaborata dalla pompa al carico fornito al fluido: EPANET rappresenta queste curve con una relazione del tipo di

$$h_G = h_O - a * q^b$$

Dove:

  - h_g è il carico fornito dalla pompa al fluido (in piedi)
  - h_o è il carico a portata nulla
  - a è un coefficiente di resistenza
  - b è l'esponente della portata

Fornendo al programma 3 punti della curva caratteristica della pompa, questo è capace di stimare i valori di a e b.

Il programma assume che il fluido scorra all'interno delle pompe in una sola direzione e che queste operino entro i limiti di portata e carico stabiliti dalle loro curve caratteristiche: nel caso in cui il sistema richieda che una pompa fornisca al fluido più del carico a pressione nulla EPANET tenterà di chiuderla ed emetterà un messaggio di avviso.

E' possibile far entrare in funzione le pompe (o bloccarle) in orari prestabiliti o nel caso in cui vengano soddisfatte determinate condizioni (ad esempio l'acqua all'interno del serbatoio scende o sale al di sopra di un valore prestabilito).

Figura 11. Simbolo dell'elemento pompa nel software EPANET

- Valvole: Oltre alle valvole di controllo situate all'interno dei tubi (che possono essere rappresentate come completamente aperte o chiuse) tramite EPANET è possibile rappresentare valvole che controllano la pressione o la portata in specifici punti della rete idrica. Queste sono considerate come archi di lunghezza insignificante dei quali devono essere specificati i nodi a monte e a valle.

  Posso essere rappresentate valvole che:

  - Riducono la pressione della estremità a valle
  - Mantengono un valore minimo di pressione nel nodo a monte quando la pressione del nodo a valle va al di sotto di un certo valore
  - Provocano una prestabilita perdita di pressione (valvole di laminazione)
  - Controllano la portata
  - Simulano una parziale chiusura regolando il coefficiente delle perdite di carico concentrate della valvola stessa (valvole a farfalla)

Figura 12. Simbolo dell'elemento valvola nel software EPANET

- Nodi: è necessario fornire per ciascuno di essi l'altezza sul livello del mare, affinché possa essere calcolato il contributo al valore della pressione dovuto all'altitudine. Devono anche essere fornite al programma tutte le informazioni riguardanti il consumo o l'immissione di nuova acqua per ogni nodo che non sia un serbatoio. Sono classificati come nodi anche i serbatoi (Tanks) e le cisterne (Reservoires) (in entrambi i casi esiste una superficie di pelo libero, tuttavia nelle seconde il livello dell'acqua rimane costante). Epanet rappresenta la variazione nel livello dell'acqua di un serbatoio con la formula:

$$\Delta y = (q/A) * \Delta t$$

  Dove:

  - Δy rappresenta la variazione del livello d'acqua (in metri)
  - q è la portata che entra o esce dal serbatoio
  - A indica l'area della sezione trasversale del serbatoio

- Δt rappresenta l'intervallo di tempo in secondi

E' inoltre necessario fornire al programma il valore massimo e minimo consentito per il livello dell'acqua all'interno dei serbatoi. I reservoir rappresentano fonti esterne di acqua, cioè laghi, fiumi o pozzi, perciò non deve essere associato loro alcun consumo o immissione di acqua .

Figura 13. Simboli degli elementi nodi nel software EPANET

Il primo simbolo rappresenta un nodo di domanda o di immissione, il secondo un reservoir e il terzo un serbatoio.

2.3 Costruzione di un modello

Vediamo ora l'esempio della costruzione di una semplice rete.
Attraverso la barra presente in alto nell'interfaccia di EPANET si inizia a costruire la rete che si vuole riprodurre.

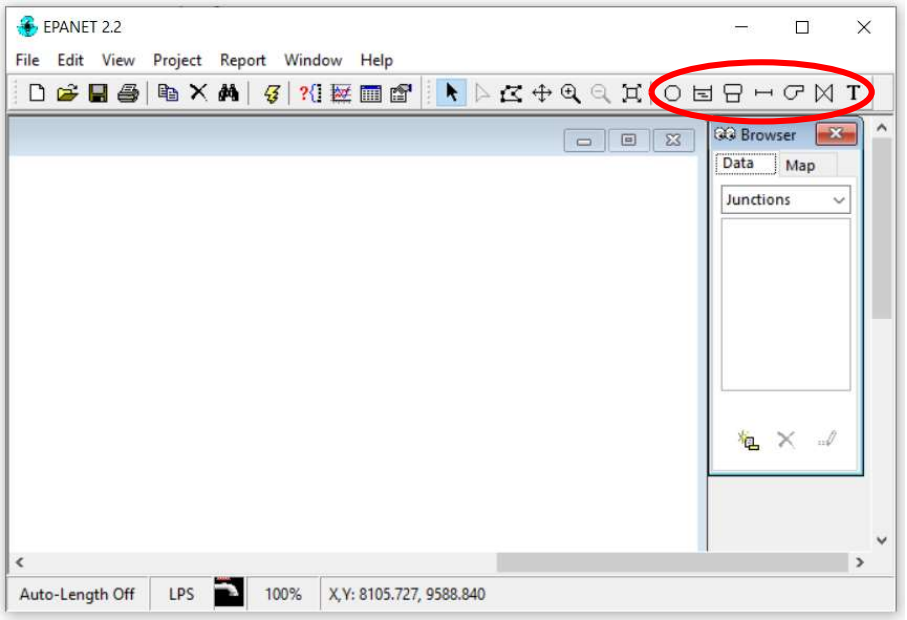


Figura 14. Interfaccia grafica di EPANET per iniziare

I primi elementi da inserire sono i reservoir, seguiti dai nodi di giunzione e dai serbatoi. Successivamente si collocano i tubi, necessari a unire i vari nodi. Per ultimo si inseriscono le pompe e le valvole se presenti.

Il tasto T è necessario per assegnare i nomi ai vari elementi.

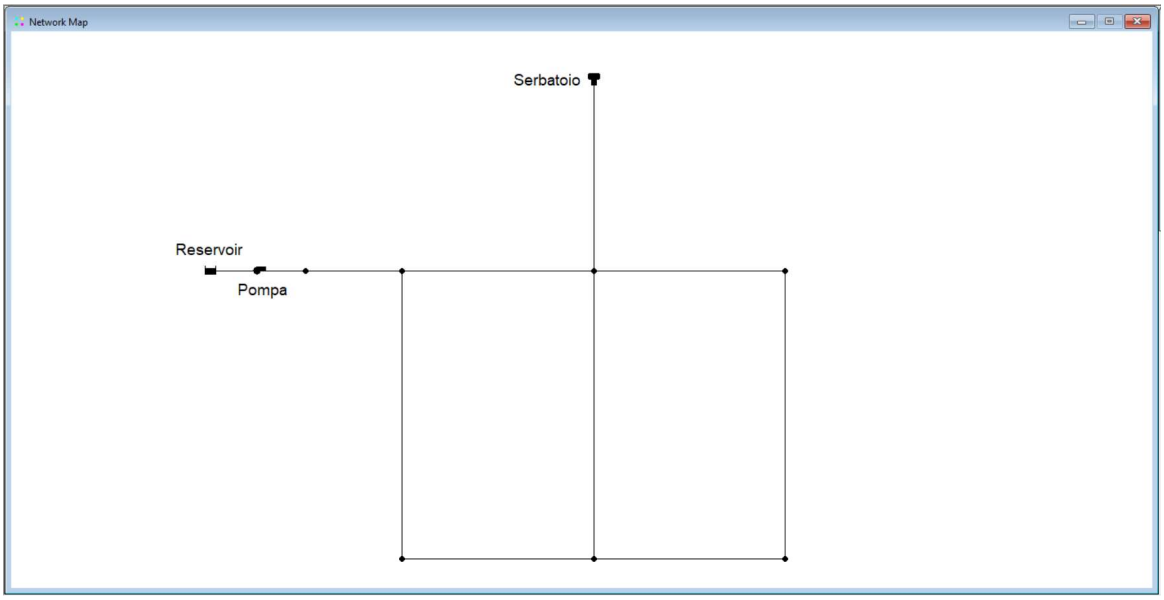


Figura 15. Esempio costruzione rete

Man mano che gli elementi vengono aggiunti al progetto EPANET assegna delle proprietà di default. Per cambiare i valori delle diverse proprietà si clicca sull'oggetto: si apre così il "Property Editor" attraverso il quale si possono cambiare le proprietà che si desidera.

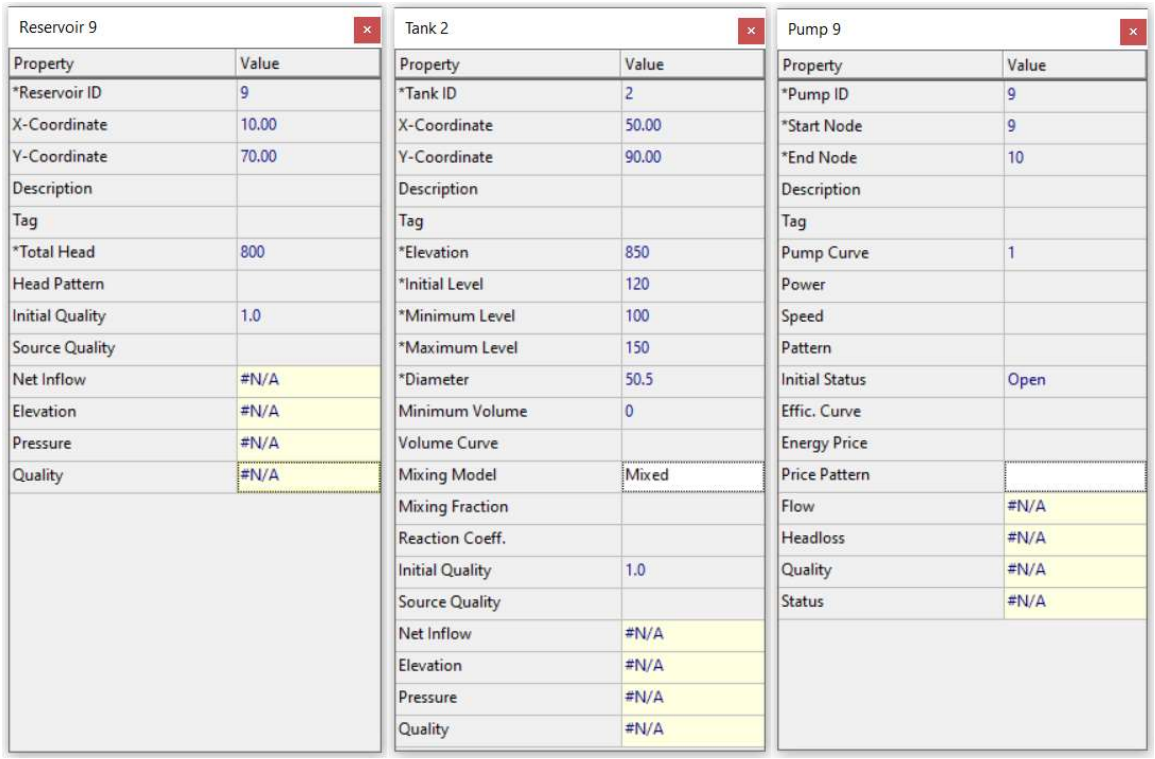


Figura 16. Property Editor dei vari elementi su EPANET

Le proprietà precedute dagli asterischi vanno immesse obbligatoriamente per poter portare avanti l'analisi.

Alla pompa è stato necessario assegnare una curva, operazione che si attua tramite la mascherina presente sulla destra dello schermo.

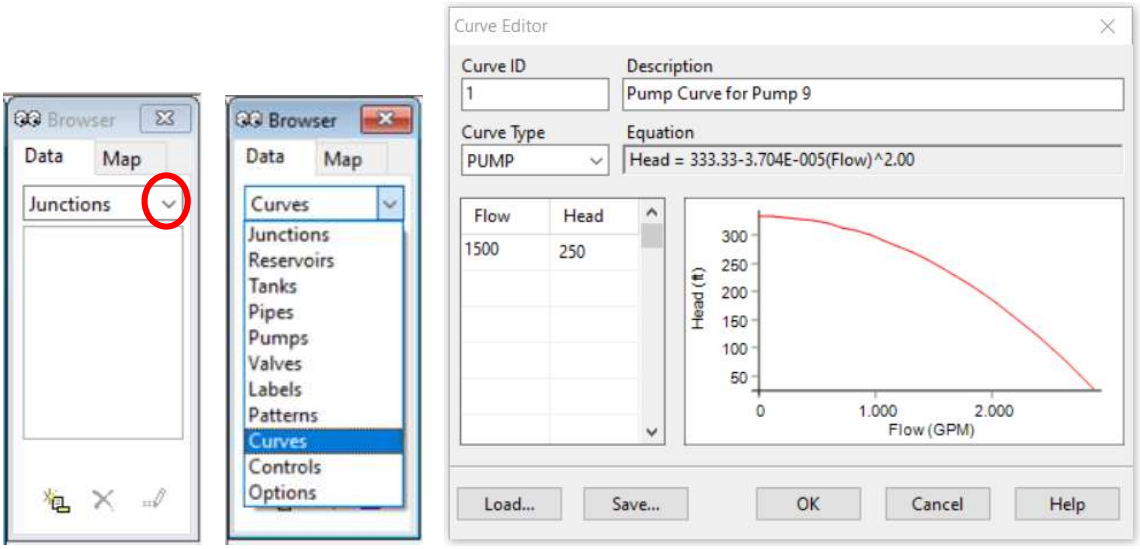


Figura 17. Inserimento della curva caratteristica della pompa su EPANET

Tramite la stessa mascherina è anche possibile entrare nelle opzioni:

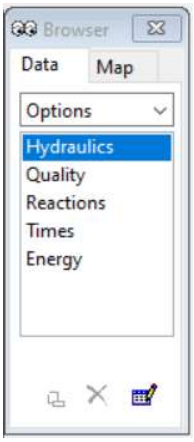


Figura 18. Options su EPANET

Tra queste figurano le "opzioni idrauliche" in cui tramite la prima opzione è possibile cambiare le unità di misura passando da quelle americane (ad esempio GMP - gallions/min) a quelle del sistema metrico internazionale (ad esempio LPS – liters/sec)

Hydraulics Options

Property	Value
Flow Units	GPM
Headloss Formula	H-W
Specific Gravity	1.0
Relative Viscosity	1.0
Maximum Trials	40
Accuracy	0.001
If Unbalanced	Continue
Default Pattern	1
Demand Multiplier	1.0
Emitter Exponent	0.5
Status Report	Yes
CHECKFREQ	2
MAXCHECK	10
DAMPLIMIT	0

Figura 19. Proprietà idrauliche su EPANET

Impostando come unità di misura LPS le lunghezze saranno in metri anziché in piedi e la potenza in chilowatt anziché in cavallo-vapore.

Attraverso la seconda opzione è possibile impostare la formula con cui si calcolano le perdite di carico nei tubi dovute all'attrito. Le tre opzioni sono: la formula di Hazen-Williams, la più utilizzata;

quella di Daircy-Weisbach, la più adatta a flussi in moto laminare e a fluidi diversi dall'acqua; quella di Chezy-Manning, usata per canali aperti.

Attraverso l'opzione tempo è possibile impostare la durata della simulazione e il time-step con cui si vogliono visualizzare i risultati

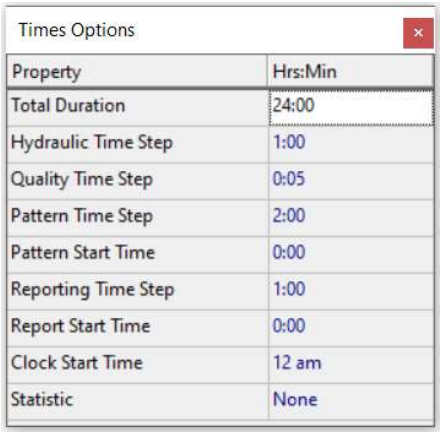


Figura 20. Time Options nelle Options di EPANET

Una volta impostate tutte le proprietà necessarie alla simulazione si clicca il simbolo che fa partire la simulazione. Una volta terminata sarà possibile visualizzare gli output sia in forma tabellare che in forma grafica , per ogni singolo nodo o giunzione [8].

Capitolo 3

Reti neurali

Il Machine Learning rappresenta un cambiamento senza precedenti nell'ambito informatico. Grazie a questa nuova metodologia, la conoscenza viene estratta dagli esempi e dall'esperienza, attraverso l'analisi di dati tangibili. Esso grazie all'unione di statistica e informatica, rende le macchine in grado di apprendere, proprio come un uomo nella sua fase di crescita e apprendimento. Ad esempio, per un essere umano, saper distinguere un gatto da un cane deriva dalla consapevolezza di aver acquisito le diverse caratteristiche delle due entità e dal memorizzarne i tratti distintivi sapendoli riconoscere.

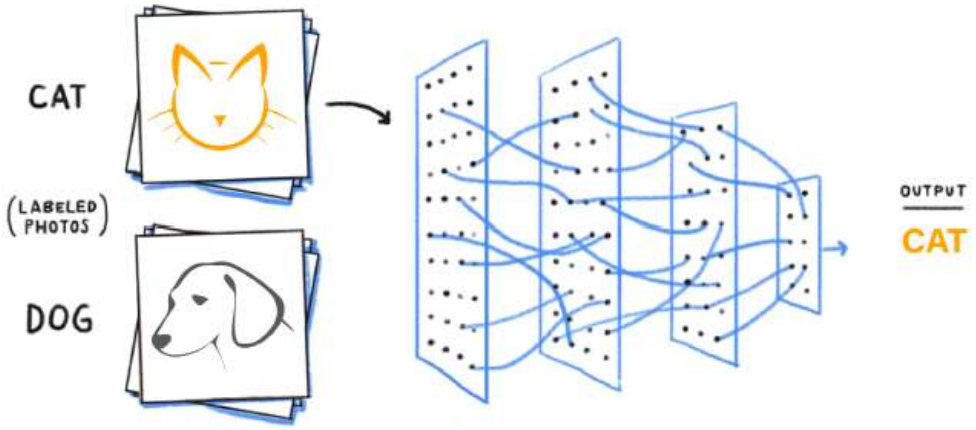


Figura 21. Esempio di Machine Learning

Un computer programmato per imparare acquisirà trend statistici, grazie ai dati che gli vengono sottoposti, che gli permetteranno di riconoscere e distinguere i due animali. Imparando dai dati, potrebbe capire da solo le varie caratteristiche, e rappresentarle in un dataset, cioè un insieme di dati, organizzati in elementi matematici di base come vettori o matrici.

Nella fase di apprendimento, è importante che sia il computer ad identificare i trend che collegano i dati e stabilire una funzione tramite cui gli input futuri verranno riconosciuti; mentre il programmatore dovrà solamente indirizzarlo su quale dataset scegliere o come applicare ciò che ha imparato, al fine di prendere la decisione migliore. In questa fase ci saranno degli errori, ma più

dati l'algoritmo di apprendimento riceve, più efficaci possono diventare le sue predizioni o classificazioni: perciò è molto importante che la mole di dati da cui il computer trae conoscenza sia quanto più grande possibile per aumentare le capacità di generalizzazione, ma non troppo grande da non rendere l'algoritmo capace di applicare le leggi che lo governano.
In generale, quindi, si collezionano numerosi esempi che portano al corretto output, dato un certo input.

Esistono, in generale, tre diverse tipologie di apprendimento:

- SUPERVISIONATO (supervised learning): esiste una conoscenza pregressa dei dati. L'obiettivo è quello di approssimare la funzione che lega input ed output, così che, nel caso in cui si abbiano nuovi dati di input, sia possibile prevedere le variabili di output per quei dati. L'apprendimento supervisionato tratta principalmente la regressione e la classificazione [14].
- NON SUPERVISIONATO (unsupervised learning): si usano dati senza nessuna conoscenza pregressa, ma si ricercano pattern all'interno di essi. Le tecniche più utilizzate sono il clustering o la dimensionality reduction [15].
- RINFORZATO (reinforcement learning): questo tipo di apprendimento interagisce con l'ambiente per imparare la serie di combinazioni che portano ai migliori risultati, memorizzando le precedenti esperienze. In base alle azioni considerate virtuose o dannose, l'algoritmo prenderà le decisioni future in funzione dei feedback che ha ricevuto durante la fase di apprendimento [16].

Le reti neurali offrono quindi, in generale, un insieme di strumenti che permette di risolvere problemi nell'ambito della classificazione, della regressione e del controllo non-lineare. Oltre ad avere un'elevata velocità di elaborazione hanno la capacità di imparare la soluzione da un determinato insieme di esempi. In molte applicazioni questo permette di raggirare il bisogno di sviluppare un modello dei processi fisici alla base del problema, che spesso può essere difficile, e talvolta impossibile, da trovare. Tuttavia i principali svantaggi nascono dal bisogno di scegliere un adeguato insieme di esempi e i problemi principali si hanno quando la rete deve rispondere ad ingressi sostanzialmente diversi da quelli dell'insieme degli esempi [14].

Lo sviluppo delle reti neurali artificiali nasce dall'osservazione delle reti neurali biologiche.
Le reti neurali rappresentano un punto di contatto tra la neurologia, la psicologia e l'intelligenza artificiale. Attraverso di esse si cerca di emulare, con dispositivi software o hardware, i

comportamenti delle trasmissioni sinaptiche che avvengono nel cervello durante la fase di apprendimento e trasmissione dell'informazione.
Come si evince dal nome stesso, si ispirano all'elemento base tramite cui il cervello elabora le informazioni: il neurone.

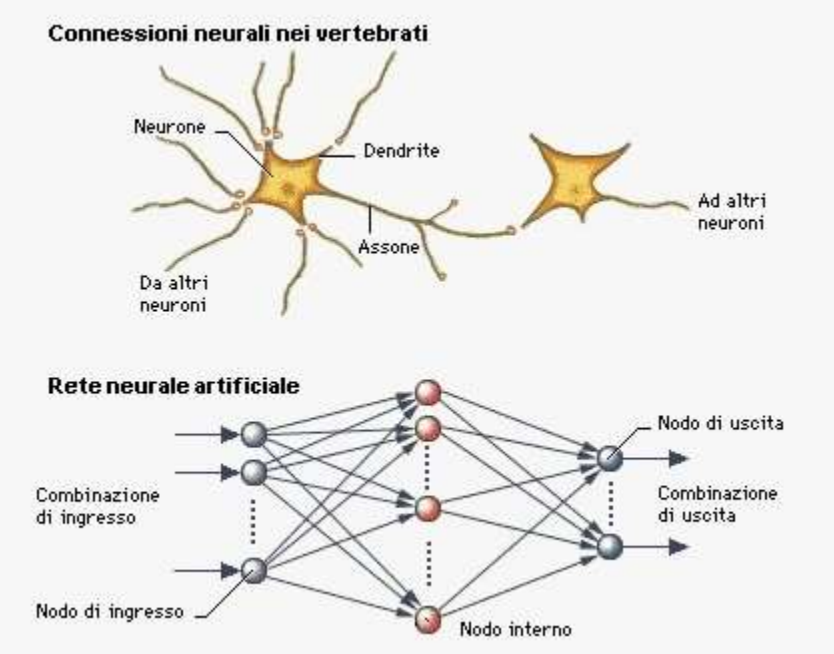


Figura 22. Rete neurali biologiche e reti neurali artificiali

Un neurone biologico è caratterizzato da una semplice struttura costituita da 3 elementi essenziali:

- corpo cellulare, in cui è presente il nucleo che dirige tutte le attività del neurone;
- assone, singola fibra lunga che trasmette messaggi dal corpo cellulare ai dendriti di altri neuroni;
- dendriti, fibre che ricevono messaggi da altri neuroni e inoltrano tali messaggi al corpo cellulare.

I neuroni ricevono input elettrici dai dendriti, assorbono energia e la rilasciano tramite gli assoni, i canali di output, ad altri neuroni vicini, trasmettendo l'informazione sotto forma di impulso elettrico attraverso altri elementi chiamate sinapsi. Nel caso in cui il segnale favorisca o meno l'attivazione del neurone ricevente, la sinapsi può essere eccitatoria o inibitoria e solo dopo un certo periodo di tempo, chiamato tempo refrattario, il neurone può generare un ulteriore impulso.
Perciò i dendriti rappresentano gli ingressi del neurone mentre l'assone ne rappresenta l'uscita. La comunicazione tra i neuroni avviene invece grazie alle sinapsi.

Figura 23. Neurone al microscopio ottico. Fonte: Fanny CASTETS - Opera propria, CC BY-SA 3.0

I neuroni sono delle celle elettricamente attive ed il cervello umano ne contiene circa 10^{11}.
Ogni neurone è tipicamente connesso ad un migliaio di altri neuroni e, di conseguenza, il numero di sinapsi nel cervello supera 10^{14} [17].
Il neurone si può trovare in 2 stati: attivo o a riposo. Quando il neurone si attiva, produce un impulso elettrico che viene trasportato lungo l'assone. Una volta che il segnale raggiunge la sinapsi esso provoca il rilascio di sostanze chimiche (neurotrasmettitori) che attraversano la giunzione ed entrano nel corpo di altri neuroni. In base al tipo di sinapsi, che possono essere eccitatori o inibitori, queste sostanze aumentano o diminuiscono rispettivamente la probabilità che il successivo neurone si attivi. Ad ogni sinapsi è associato un peso che ne determina il tipo e l'ampiezza dell'effetto eccitatore o inibitore. Quindi ogni neurone effettua una somma pesata degli ingressi provenienti dagli altri neuroni e, se questa somma supera una certa soglia, il neurone si attiva.
Ogni neurone, che opera ad un ordine temporale del millisecondo, rappresenta un sistema di elaborazione relativamente lento; tuttavia, l'intera rete ha un numero molto elevato di neuroni e sinapsi che possono operare in modo parallelo e simultaneo, rendendo l'effettiva potenza di elaborazione molto elevata.
Inoltre la rete neurale biologica ha un'alta tolleranza ad informazioni poco precise e sbagliate: ha infatti la facoltà di apprendimento e generalizzazione.

Analogamente, una rete neurale artificiale è costituita dall'interconnessione di semplici unità di elaborazione, i cosiddetti neuroni artificiali, che hanno la propensione ad estrarre la conoscenza dai dati e a memorizzarla tramite dei pesi, chiamati pesi sinaptici. Le reti neurali artificiali si distinguono per due importanti caratteristiche:

- sono state concepite con l'abilità di poter approssimare qualsiasi funzione, sia essa lineare o non lineare;
- aggiornano i pesi sinaptici durante l'allenamento. I campioni vengono somministrati alla rete e i pesi vengono modificati in modo da ridurre la distanza minima tra il target, l'obiettivo della rete, e l'uscita trovata.

3.1 Introduzione alle reti neurali artificiali

Le reti neurali artificiali sono dei sistemi di elaborazione dell'informazione il cui comportamento trae ispirazione dai sistemi nervosi biologici. Anche la rete neurale artificiale possiede molte semplici unità di elaborazione variamente connesse tra di loro: i neuroni. Alcune di queste unità ricevono informazioni dall'ambiente, altre emettono risposte nell'ambiente e altre ancora, se presenti, comunicano solamente con le unità all'interno della rete: esse sono definite rispettivamente unità di ingresso (input), unità di uscita (output) e unità nascoste (hidden).

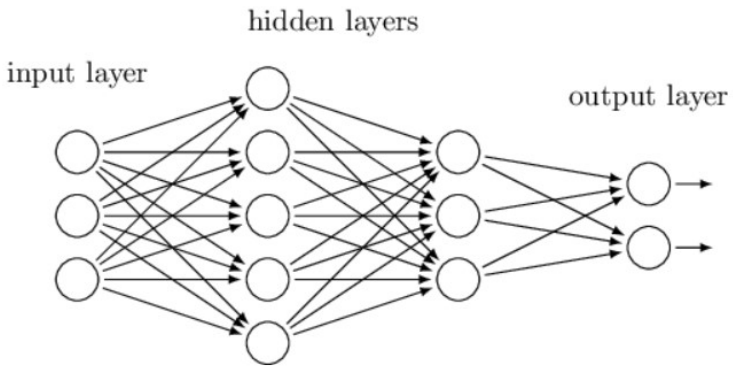


Figura 24. Struttura di una rete neurale artificiale

Le reti neurali artificiali simulano il cervello umano simulando l'apprendimento secondo connessioni tra diversi strati, detti layer, con un apprendimento sempre più profondo che porta ad estrarre dai dati delle caratteristiche (features) di livello via via superiore. I suddetti layer sono costituiti da unità o nodi, ciascuno dei quali simula il funzionamento di un neurone, ricevendo uno o più input e, a seguito di un'elaborazione, fornendo un output.

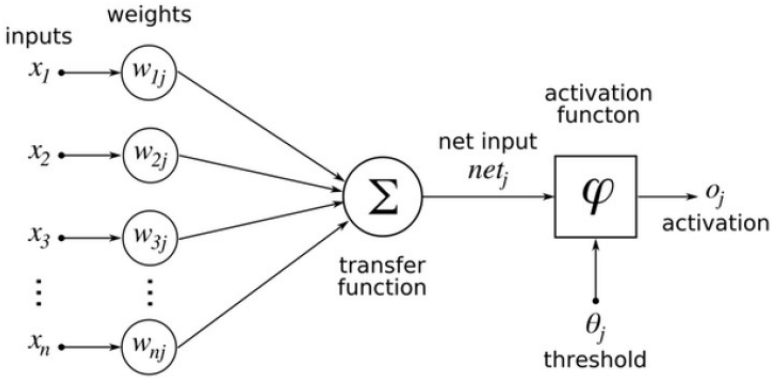


Figura 25. Schematizzazione matematica di una rete neurale

Facendo riferimento alla precedente Figura 25, gli input (le x) vengono moltiplicate per i pesi, (indicati con le w) e, una volta sommati tutti, si applica la funzione di attivazione per ottenere l'output: fase nota come "Forward Propagation". La fase di apprendimento vero e proprio avviene grazie alla "Backward Propagation": i parametri e quindi la struttura della rete sono in grado di aggiornarsi attraverso quello che viene definito allenamento. Durante l'allenamento i dati vengono suddivisi in 3 dataset in quanto oltre all'allenamento sono fondamentali altri due step, la validazione e il testing su esempi che la rete non ha mai visto prima. Il numero di dati da destinare a ciascun dataset deve essere scelto in maniera ponderata.

Dal punto di vista matematico un neurone k può essere rappresentato attraverso l'equazione seguente:

$$u_k = \sum_{j=1}^{m} w_{kj} \cdot x_j$$

$$y_k = \varphi(u_k + b_k)$$

Dove:

- x_i sono i pesi sinaptici del neurone k;
- u_k è la combinazione lineare degli input del neurone k;
- b_k è il valore di soglia del neurone k;
- φ (x) è la funzione di attivazione;
- y_k è l'output elaborato dal neurone k.

Il modo in cui i neuroni di una rete sono collegati e strutturati ne definisce l'architettura ed è strettamente connessa all'algoritmo di training utilizzato per allenare la rete.

Nel presente lavoro, per la creazione del modello utilizzato, sono state utilizzate due reti: una di tipi feedback e una di tipo feedforward.

RETE NEURALE FEEDFORWARD

Una rete neurale feedforward (Feedforward Neural Network, FFNN) è la prima rete neurale messa a punto e in essa le informazioni si muovono in una sola direzione. Nella sua forma più semplice, illustrata nella Figura 26, c'è solo uno strato d'ingresso che dirige l'informazione allo strato dei neuroni d'uscita, dove essa viene processata.

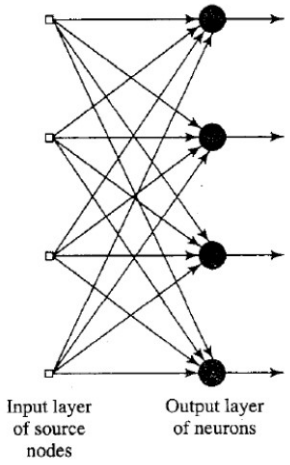


Figura 26. Rete neurale feedforward a singolo strato

Questa rete è definita a singolo strato (single-layer network) poiché lo strato d'ingresso non viene contato in quanto in esso non avviene nessuna operazione sui dati. In questa rete gli ingressi, opportunamente pesati, vengono elaborati direttamente dai neuroni dello strato d'uscita.
Una rete neurale multistrato (multilayer network) è caratterizzata da uno o più strati di neuroni nascosti, il cui scopo è far da tramite tra ingresso esterno e uscita della rete, elaborando le informazioni fornite alla rete ed estraendo statistiche di ordine superiore rispetto ad una rete monostrato. Di solito la funzione d'attivazione è di tipo sigmoidale o tangente iperbolica per i neuroni degli strati nascosti, mentre per i neuroni dell'uscita può essere sia di forma sigmoidale che lineare, in base al compito che la rete neurale dovrà risolvere.

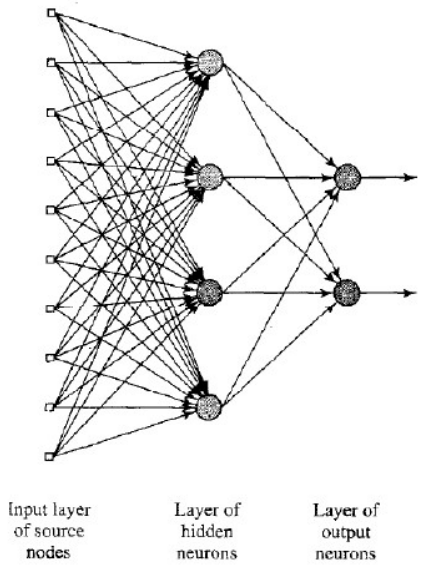


Figura 27. Rete neurale feedforward multistrato

I nodi nel primo strato ricevono ognuno un elemento dal vettore d'ingresso e lo trasferiscono ai neuroni del secondo strato (ovvero il primo strato nascosto), i quali elaborano le informazioni ricevute e le passano ai neuroni dello strato immediatamente successivo, proseguendo così fino ai neuroni dell'ultimo strato della rete, la cui uscita corrisponde alla risposta della rete al vettore d'ingresso. Tipicamente i neuroni di ogni strato della rete hanno in ingresso solamente i segnali d'uscita dello strato precedente e dirigono la loro uscita esclusivamente ai neuroni dello strato successivo.

Le reti neurali feedforward sono le più utilizzate, sia grazie alla loro semplicità che alla loro versatilità. In particolare sono spesso usate per il riconoscimento vocale ed ottico, per la classificazione, per l'identificazione e il controllo di sistemi, per la predizione e per l'approssimazione di funzioni [18].

RETE NEURALE FEEDBACK

Una feedback neural network o rete neurale ricorsiva (recurrent neural network, RNN), a differenza di una rete feedforward, ha almeno una connessione che va ad uno strato precedente.

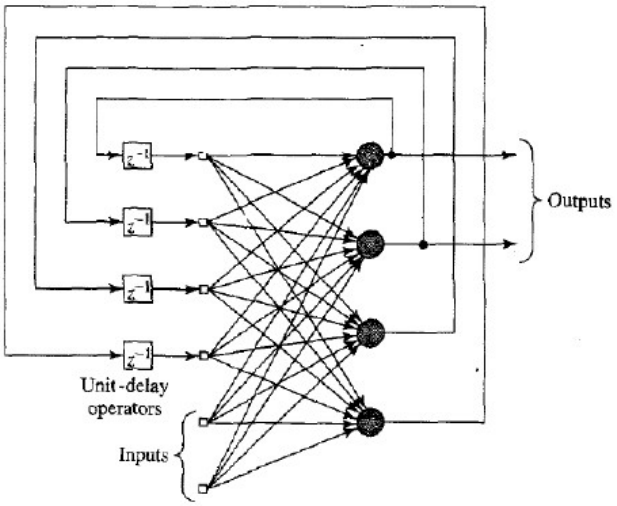


Figura 28. Rete neurale ricorsiva con neuroni nascosti [18]

Quella mostrata nella precedente Figura 28, è costituita da uno strato di ingressi, uno strato di neuroni nascosti e due neuroni d'uscita. La ricorsione avviene sia dai neuroni nascosti, sia dai neuroni d'uscita. La presenza di feedback ha un grandissimo impatto sulle performance e sulle capacità di apprendimento della rete.

La presenza di ricorsioni richiede l'introduzione di elementi di ritardo. Proprio la presenza di ritardi differenzia profondamente il comportamento di una rete ricorsiva da quello di una rete feedforward: nonostante una rete neurale feedforward possa modellare una grande varietà di relazioni ingresso-uscita, essa ha il difetto di mancare totalmente di memoria dinamica. L'uscita di una FFNN sarà sempre la stessa se in ingresso le viene presentato lo stesso vettore. Questo comportamento non rispecchia affatto quello biologico: la memoria è una caratteristica essenziale degli animali e permette loro di rispondere in modo diverso quando si ripresenta un evento a cui avevano precedentemente risposto in maniera inappropriata. Una rete ricorsiva, invece, ha una memoria teoricamente infinita, proprio grazie alla presenza di feedback e ritardi, e la sua uscita non dipende solamente dal vettore in ingresso, ma anche dallo stato interno della rete, permettendole di rispondere in maniera diversa ad uno stesso ingresso.

3.2 Reti neurali su MATLAB

Per il presente lavoro di tesi sono state utilizzate due reti neurali presenti su MATLAB nel gruppo di applicativi sul *machine learning and deep learning*: NARXNET e FITNET.

NARXNET

La rete neurale NARXNET è presente nel tool chiamato *Neural Network Time Series app* ed è una rete neurale del tipo feedback (vedi paragrafo precedente).

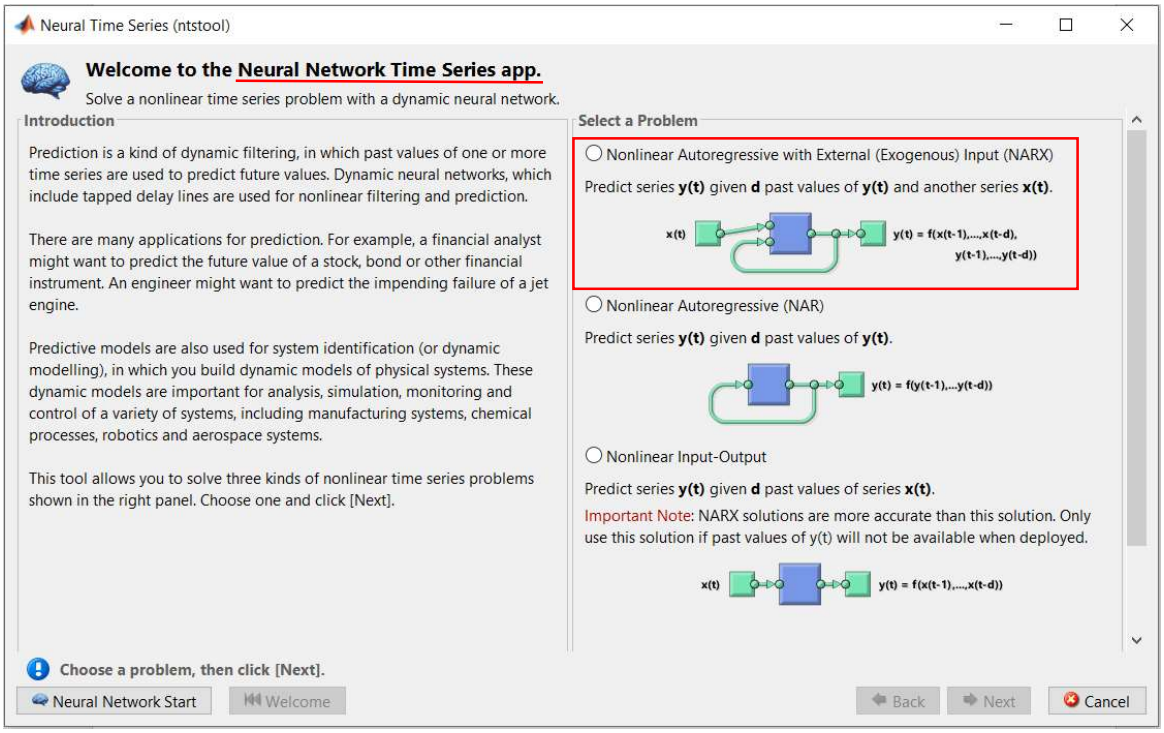


Figura 29. Neural Network Time Series app su MATLAB

NARXNET, acronimo di Nonlinear Autoregressive with External (Exogenous) Input Network, è una rete neurale autoregressiva non lineare con input esterni, che permette di calcolare gli output della rete istante per istante, tenendo anche conto degli output calcolati negli istanti precedenti.
La NARXNET è una rete neurale dinamica ricorrente, che include connessioni di feedback in molti strati della rete. L'equazione che definisce il suo modello è la seguente:

$$y(t) = f\left(y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u)\right)$$

dove il valore successivo del segnale di output y(t) è dato dai valori precedenti dello stesso segnale di output e dai valori precedenti della variabile indipendente di input, cioè gli input esogeni [20].
La funzione f può essere approssimata usando una rete neurale feedforward, come mostrato nella Figura 30. La rete di feedforward composta da due strati, permette di avere input e output anche multidimensionali.

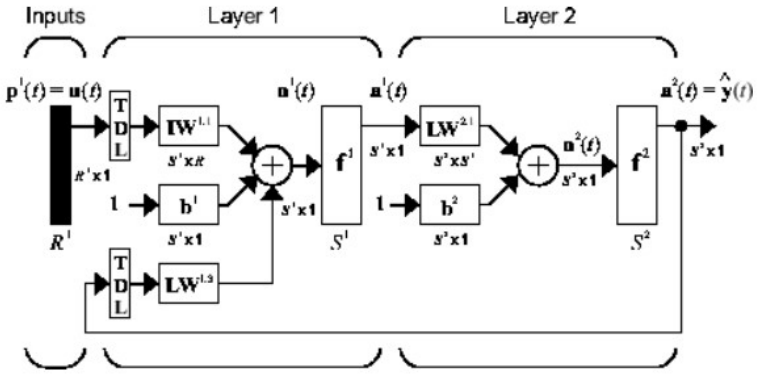


Figura 30. Diagramma NARXNET

Sono molti i campi in cui può essere usata questa rete neurale: oltre che per predire, come in questo lavoro, può essere usata in generale nella modellazione di sistemi dinamici non lineari.
Si può considerare l'output della rete NARXNET come la stima del sistema dinamico non lineare che si sta cercando di modellizzare. L'output rientra negli input della rete feedforward, come mostra l'architettura della rete in Figura 31.a.

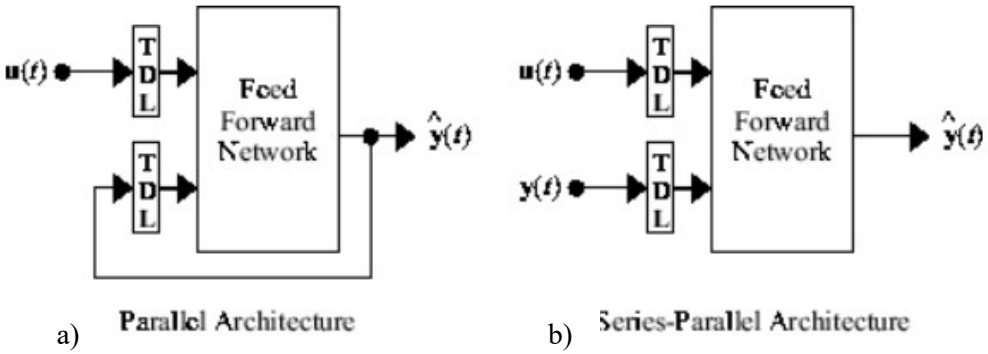


Figura 31. Architettura NARXNET (1/2)

Affinché il vero output sia disponibile durante l'allenamento della rete, è necessario creare una architettura come quella mostrata in Figura 31.b, in cui viene utilizzato il vero output, invece di quello stimato frutto della retroazione. Questo porta a due vantaggi: il primo è che l'input della rete feedforward è più accurato; il secondo è che la rete risultante è una pura architettura feedforward e

la backpropagation può essere utilizzata per l'allenamento [20]. L'allenamento della rete avviene quindi utilizzando l'architettura mostrata in Figura 31.b.

La configurazione mostrata nella Figura 31.a viene, chiamata anche *ciclo chiuso* è utile per le previsioni multi-step in avanti, utilizzata in questo lavoro.

Tipicamente si procede quindi in questo modo: la rete viene creata nella forma a *ciclo aperto* (Figura 31.a), e solo una volta che è stata allenata (includendo anche la validazione e il testing), viene trasformata nella forma a *ciclo chiuso* che permette le previsioni.

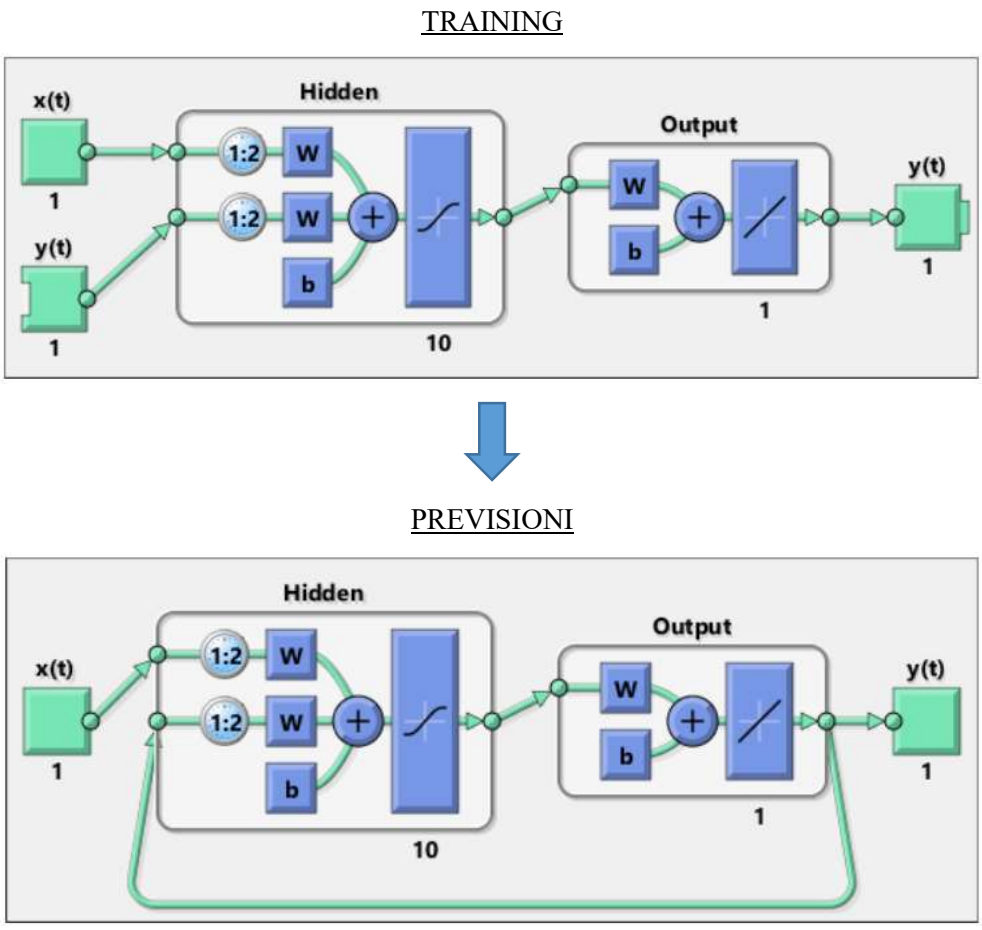


Figura 32. Architettura NARXNET (2/2)

FITNET

La rete neurale FITNET è presente nel tool chiamato *Network Fitting app* ed è una rete neurale del tipo feedforward (vedi paragrafo precedente).

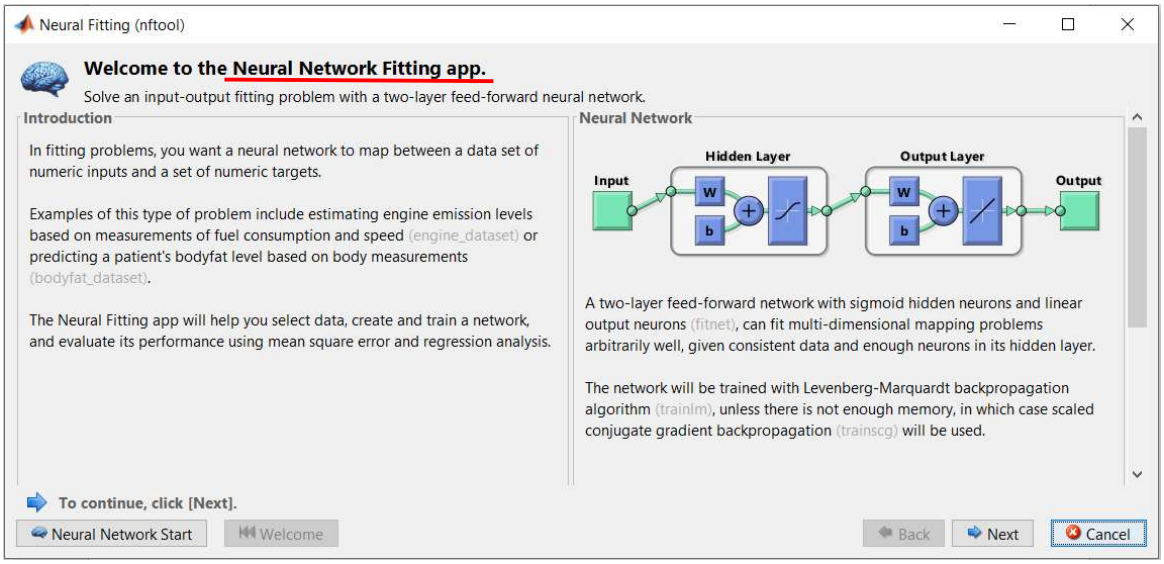


Figura 33. Neural Network Fitting app su MATLAB

La FITNET è la più semplice rete neurale che si possa usare per i problemi di fitting e di regressione. Allenando questa rete neurale, le si danno in set di input, in modo che trovi delle associazioni con un set di output di riferimento.

Si costruisce dapprima la rete con il numero desiderato di strati nascosti e si sceglie l'algoritmo di training, infine si allena la rete con un set di dati. Una volta che la rete neurale è stata allenata e ha trovato un collegamento generale tra i dati di input e quelli di output, essa può essere usata per trovare nuovi output a partire da input con cui la rete neurale non è stata allenata.

Capitolo 4

Algoritmi genetici

Gli algoritmi genetici (o GA, Genetic Algorithms), proposti per la prima volta nel 1975 da J.H. Holland, sono delle procedure finalizzate alla risoluzione di problemi complessi di ottimizzazione. Essi si basano sul concetto di selezione naturale ed evoluzione della specie: selezionano le soluzioni migliori di un problema ricombinandole tra loro, in modo tale che esse convergano verso un punto di ottimo.

4.1 Introduzione agli algoritmi genetici

Sin dalle origini le tecniche dell'informatica hanno tratto ispirazione dalla biologia cercando di emulare il funzionamento del cervello umano (reti neurali), il suo apprendimento (machine learning) e l'evoluzione biologica (algoritmi genetici). Come brevemente accennato sopra gli algoritmi genetici hanno come base l'idea di far evolvere una popolazione di soluzioni candidate per un problema mediante l'uso di operatori "suggeriti" dalle leggi della selezione naturale della specie [4].

Il primo step in un problema di ottimizzazione è quello di individuare una funzione obiettivo (o funzione di fitness) F, da minimizzare (o più raramente massimizzare), attraverso il cambiamento di una o più variabili indipendenti (x). Dal punto di vista matematico la funzione obiettivo può essere rappresentata come segue:

$$min\ F(x)$$

L'algoritmo genetico dovrà trovare il set di variabili che portano al valore minimo di questa funzione.
Ispirandosi all'evoluzione naturale della specie, questa metodologia eredita da essa anche la nomenclatura utilizzata:

- individuo; è l'insieme degli n valori x_1, x_2, x_3,…,x_n da cui dipende la funzione obiettivo;
- popolazione; è l'insieme degli individui;

- cromosoma; ciascun individuo viene identificato da un cromosoma, cioè una stringa di lunghezza costante a sua volta formata da geni. Di solito i geni sono binari, rappresentati con valori 0 o 1;
- generazione; indica la popolazione in un dato istante temporale;
- genitori e figli; ad ogni iterazione l'algoritmo considera gli individui migliori di quella generazione come genitori, e da questi crea i figli, che saranno gli individui della generazione successiva.

L'algoritmo crea una generazione iniziale randomica; nell'esempio della Figura 34, la popolazione iniziale è formata da 20 individui ed è stato imposto come limite il fatto che gli individui siano in un range compreso tra 0 e 1.

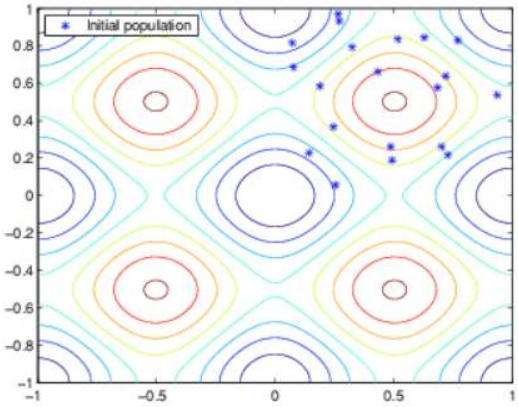


Figura 34. Distribuzione popolazione iniziale - GA

Inizialmente la distanza tra gli individui che fanno parte della stessa generazione è più ampia (si veda Figura 34) in quanto è maggiore lo spazio da essi esplorato: questa distanza è chiamata diversità. Man mano che vengono create nuove generazioni si può notare come la diversità tra gli individui diminuisce sempre di più [21].

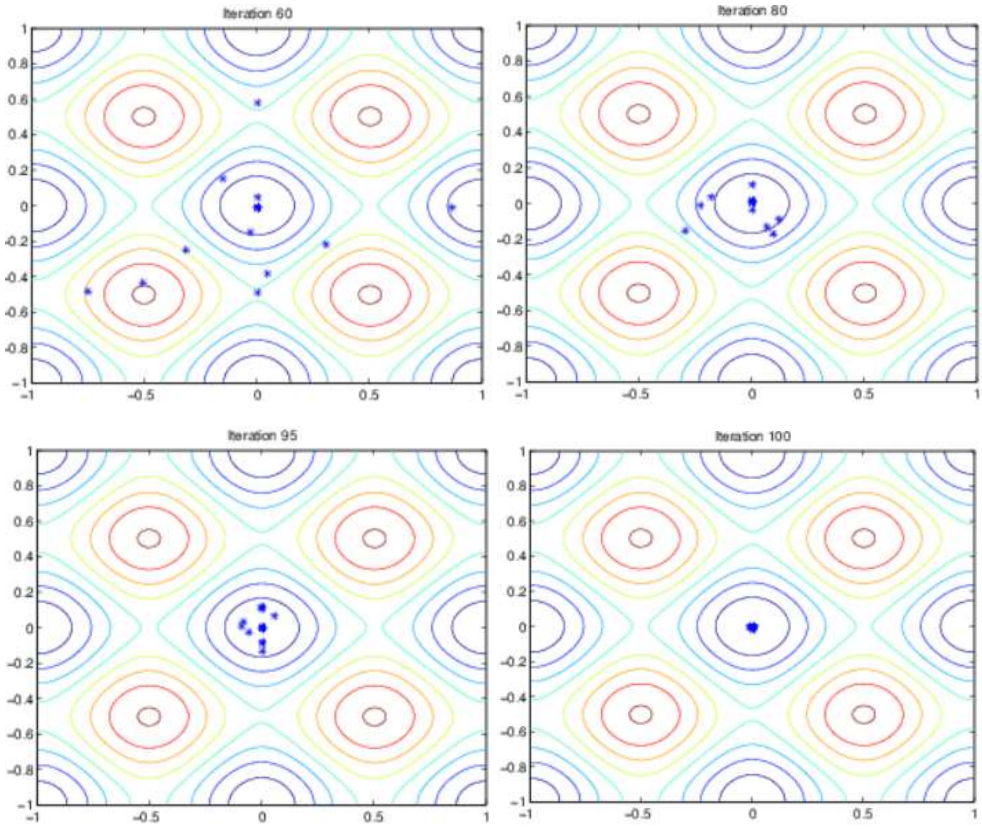


Figura 35. Distribuzione popolazione nella generazione 60, 80, 95 e 100 - GA

Nella Figura 35 si nota come la distribuzione della popolazione nella generazione 60, 80, 95 e 100 via via si avvicini sempre di più al punto di ottimo.

In natura la specie si evolve unendo il proprio patrimonio genetico: i figli avranno un patrimonio genetico derivante in parte da un genitore (padre) e in parte dall'altro (madre). La selezione naturale fa si che riescano a sopravvivere solamente gli individui più forti, quelli che hanno il valore della funzione obiettivo più vicina all'ottimo.

In generale, un algoritmo genetico è così strutturato:

1) viene definita a caso una popolazione iniziale di possibili soluzioni al problema preso in esame;
2) per ogni individuo della popolazione viene calcolata la funzione di fitness, associando a ciascuno di essi un punteggio per valutarne le prestazioni;
3) gli individui migliori vengono scelti come genitori della generazione successiva;
4) i genitori vengono opportunamente modificati per creare degli individui figli potenzialmente migliori, che faranno parte della generazione successiva;

5) viene ricalcolata la fitness per gli individui della nuova generazione creata a partire dai genitori scelti dalla popolazione precedente;
6) viene controllato se il criterio di STOP impostato è stato soddisfatto: per esempio quando si raggiunge un numero massimo di iterazioni oppure se la migliore fitness della popolazione attuale non si discosta di molto dalla migliore fitness delle x generazioni precedenti. Il numero delle generazioni x da prendere in considerazione può essere impostato. Se il criterio di STOP non è stato soddisfatto si riprende dal punto 3).

Come già accennato, gli individui considerati genitori di una popolazione vengono modificati per produrre i figli, che saranno gli individui della popolazione successiva. Queste modifiche sono chiamate *operazioni genetiche*, e le più importanti sono crossover e mutazione.
Il crossover coinvolge due individui genitori:

1 0 0 | 0 0 0
1 1 1 | 0 0 1

Viene scelta una posizione k ($1 \leq k \leq 6$) in modo del tutto causale e si effettua uno scambio di geni. K viene chiamato anche punto di crossover.

1 0 0 | 0 0 1
1 1 1 | 0 0 0

Una variante di questa oprazione consiste nella scelta casuale di due punti di crossover k e l (con $l \leq k$):

1 0 | 0 0 | 0 0
1 1 | 1 0 | 0 1

e si effettua

1 0 | 1 0 | 0 0
1 1 | 0 0 | 0 1

La mutazione riguarda invce un singolo gene, che data una certa probabilità, viene cambiato nel valore opposto.

1 0 0 | 0 | 0 0

diventa:

1 0 0 | 1 | 0 0

4.2 Algoritmi genetici su MATLAB

Per questo lavoro di tesi si è lavorato in ambiente MATLAB, software che presenta tra gli altri un tool di ottimizzazione. Questo tool permette, tramite la funzione denominata *ga*, di impostare l'algoritmo genetico e tutte le opzioni desiderate. Si può usare l'interfaccia grafica digitando *optimtool ('ga')* oppure, come si è fatto per questo lavoro, si richiama la funzione nella linea di comando nel seguente modo:

$$[x, fval, exitflag] = ga\,(@\text{fitness}, \text{nvars}, \text{A}, \text{b}, \ldots, \text{Aeq}, \text{beq}, \text{lb}, \text{ub}, \text{options})$$

dove gli input alla funzione sono:

- @fitness = richiamo alla funzione di fitness;
- nvars = numero di variabili indipendenti della funzione di fitness;
- A = matrice per i vincoli lineari di disuguaglianza;
- b = vettore per i vincoli lineari di disuguaglianza;
- Aeq = matrice per i vincoli lineari di uguaglianza;
- beq = vettore per i vincoli lineari di uguaglianza;
- lb = vettore che indica i limiti inferiori delle x;
- ub = vettore che indica i limiti superiori delle x;
- options = struttura necessari a impostare le opzioni desiderate per l'algoritmo genetico, se non viene impostato si utilizzano le opzioni di default mostrate nella Figura 36;

e gli output della funzione sono:

- x = miglior soluzione trovata dall'algoritmo, restituita sotto forma di vettore;
- fval =valore numerico della funzione di fitness migliore trovata dall'algoritmo, fval = f (x);
- exitflag = numero intero che da indicazioni sul quale sia il motivo per cui l'algoritmo si è fermato [22].

```
    PopulationType: {'doubleVector'}
      PopInitRange: {[0;1]}
    PopulationSize: {20}
        EliteCount: {2}
 CrossoverFraction: {0.8}
    ParetoFraction: {0.35}
MigrationDirection: {'forward'}
 MigrationInterval: {20}
 MigrationFraction: {0.2}
       Generations: {100}
         TimeLimit: {Inf}
      FitnessLimit: {-Inf}
     StallGenLimit: {50}
    StallTimeLimit: {Inf}
           TolFun: {1e-6}
           TolCon: {1e-6}
 InitialPopulation: {[]}
     InitialScores: {[]}
    InitialPenalty: {10}
     PenaltyFactor: {100}
       CreationFcn: {Use constraint dependant}
 FitnessScalingFcn: {@fitscalingrank}
      SelectionFcn: {Stochastic uniform}
      CrossoverFcn: {scattered}
       MutationFcn: {Use constraint dependant}
DistanceMeasureFcn: {@distancecrowding}
         HybridFcn: {[]}
           Display: {'final'}
         OutputFcns: {[]}
          PlotFcns: {[]}
      PlotInterval: {1}
        Vectorized: {'off'}
       UseParallel: {'never'}
```

Figura 36. Opzioni di default – funzione ga MATLAB

L'algoritmo genetico su MATLAB, presenta la seguente struttura:

1) viene creata una popolazione iniziale randomica;
2) l'algoritmo crea una sequenza di nuove generazioni. Per creare ciascuna generazione successiva esso utilizza gli individui della generazione precedente, attraverso i seguenti passaggi:
   - a ciascun individui della popolazione viene assegnato un punteggio, collegato al suo valore di fitness (più il valore di fitness è basso più il punteggio è alto);
   - vengono selezionati i due individui con il punteggio più alto, chiamati genitori;
   - alcuni degli individui col punteggio più alto passano automaticamente alla generazione successiva senza subire cambiamenti;
   - vengono creati dei figli a partire dai genitori selezionati precedentemente;
   - i figli vengono creati operando variazioni randomiche ai singoli genitori (mutazione) o combinando tra loro i due genitori (crossover);
   - per ultimo viene sostituita la popolazione corrente con quella successiva, frutto delle predette variazioni;
3) l'algoritmo si ferma quando viene soddisfatto un criterio di STOP precedentemente impostato. Alcuni criteri di stop possono essere: raggiungimento del numero massimo di

generazioni, raggiungimento del tempo impostato, raggiungimento di un determinato valore di fitness, raggiungimento del numero massimo di generazioni consecutive in cui la fitness non cambia [21].

L'algoritmo genetico crea quindi 3 tipi di figli per la generazione successiva (Figura 37):

- gli elitari, che sono passati alla generazione successiva direttamente dalla precedente perché avevano il miglior valore di fitness;
- i crossover, che vengono creati combinando i genitori;
- i mutati, che sono creati introducendo variazioni randomiche ai singoli genitori [21].

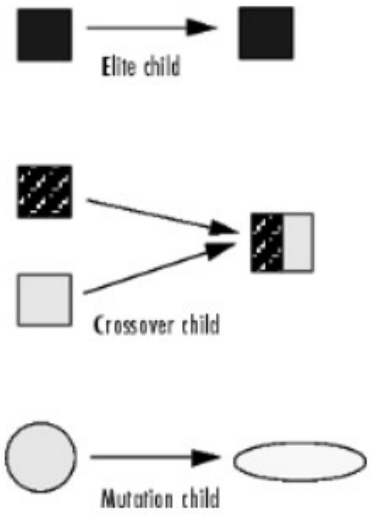


Figura 37. Tipi di figli nell'algoritmo genetico su Matlab

Sia i figli creati tramite crossover che quelli creati tramite mutazione sono essenziali per l'algoritmo; quelli creati tramite crossover permettono all'algoritmo di estrarre i geni migliori da individui diversi e ricombinarli per creare un figlio potenzialmente superiore; mentre le mutazioni aggiungono diversità alla popolazione e aumentano la possibilità di ricerca in spazi più ampi.
Si può decidere quanti figli di ciascun tipo l'algoritmo deve creare specificando il numero di figli elitari (*EliteCount*) da far passare direttamente alla generazione successiva e la frazione di popolazione (*CrossoverFraction*), oltre i figli elitari, che sarà creata tramite crossover [21].

Si presenta adesso un esempio, prendendo come riferimento la Figura 36. La popolazione è formata da 20 individui. 2 di essi passano direttamente alla generazione successiva perché sono figli elitari.

La frazione di crossover è di default 0.8. Per cui dei 18 individui rimanente 18*0.8=14 saranno figli frutto del crossover; i rimanenti 20-2-14 = 4 saranno i figli frutto della mutazione.

La funzione di default per la mutazione, chiamata *mutationgaussian*, somma all'individuo di partenza un numero randomico preso da una distribuzione gaussiana con media 0.
La funzione di default per il crossover, chiamata *crossoverscattered* crea un vettore random binario e seleziona i geni dal primo genitore quando il vettore è 1 e dal secondo genitore quando il vettore è 0. Combina così i geni per creare il figlio.

Capitolo 5

Analisi del caso studio

Il caso studio preso in esame è rappresentato nella figura successiva:

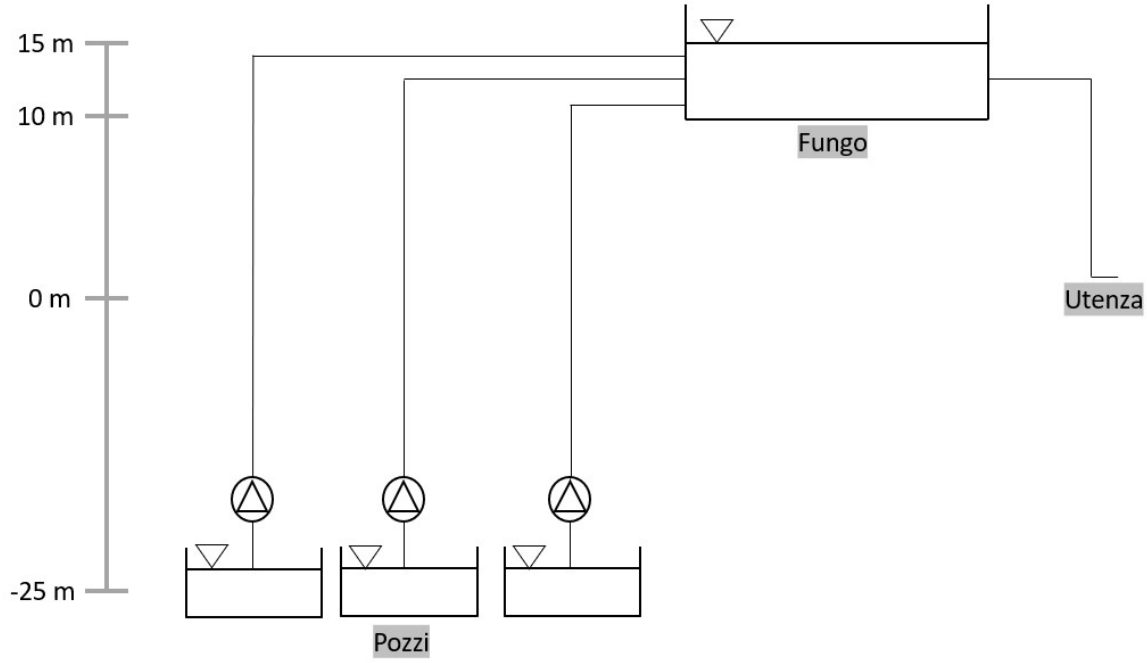


Figura 38. Schema del caso studio

Si tratta di un impianto di sollevamento, composto da 3 pompe sommerse che pompano l'acqua dalla falda sotterranea (posta a -25m rispetto al suolo), fino a un fungo posto a 10 m di altezza dal suolo. L'impianto tessile in cui viene utilizzata l'acqua pompata si trova a livello del suolo.

In generale gli impianti di sollevamento si compongono di serbatoio di aspirazione, condotta di aspirazione, stazione di pompaggio, condotta di mandata e serbatoio di mandata. La correttezza del dimensionamento, insieme all'accuratezza della realizzazione, sono necessari presupposti per il buon funzionamento sia dal punto di vista dell'efficienza tecnica che del contenimento dei costi di esercizio. Ciò richiede però adeguati investimenti inziali. Per il mantenimento in perfetta efficienza dell'impianto nel tempo è necessario che il numero di avviamenti orari non ecceda i limiti indicati nella seguente tabella.

Potenza del motore	Numero di avviamenti orari	
	Installazione a secco	Installazione sommersa
fino a 7,5 kW	15	30
fino a 30 kW	12	24
oltre 30 kW	10	20

Figura 39. Numero massimo di avviamenti orari per le pompe con installazione a secco e sommerse

Lo schema del fungo considerato nel caso studio viene mostrato nella figura successiva.

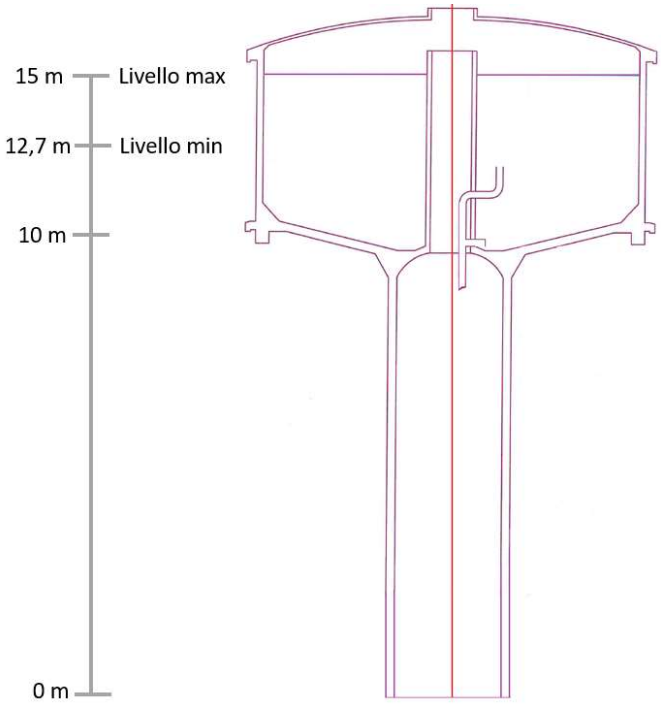


Figura 40. Schema del fungo nel caso reale

5.1 Simulazione della rete tramite EPANET

Il sistema idrico è stato modellizzato grazie all'utilizzo del software EPANET.

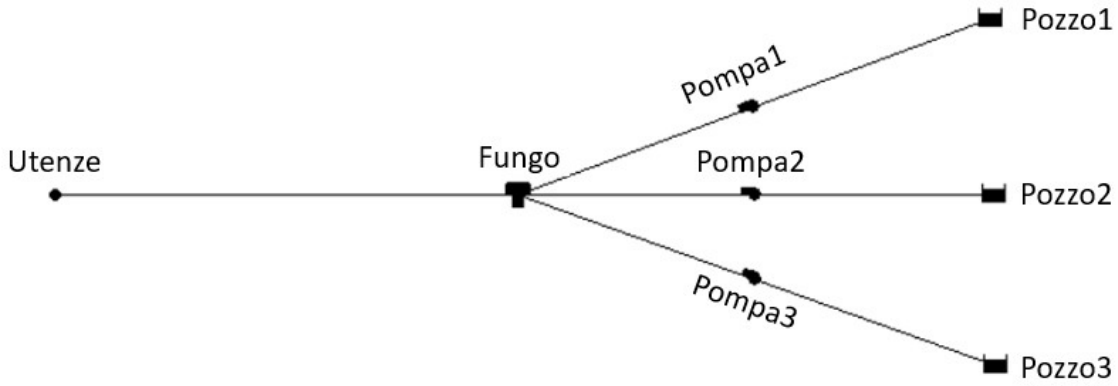


Figura 41. Schema del caso reale su EPANET

I dati di input richiesti dal software sono:

- Le altitudini (altezze slm) in m dei vari nodi;
- La richiesta oraria di acqua da parte dell'utenza in l/s;
- Le caratteristiche di tutte le condotte presenti (lunghezza, diametro e rugosità);
- Le caratteristiche del fungo (altezza dal suolo, livello iniziale, livello massimo, livello minimo e diametro);
- Le curve caratteristiche e le curve di efficienza delle pompe.

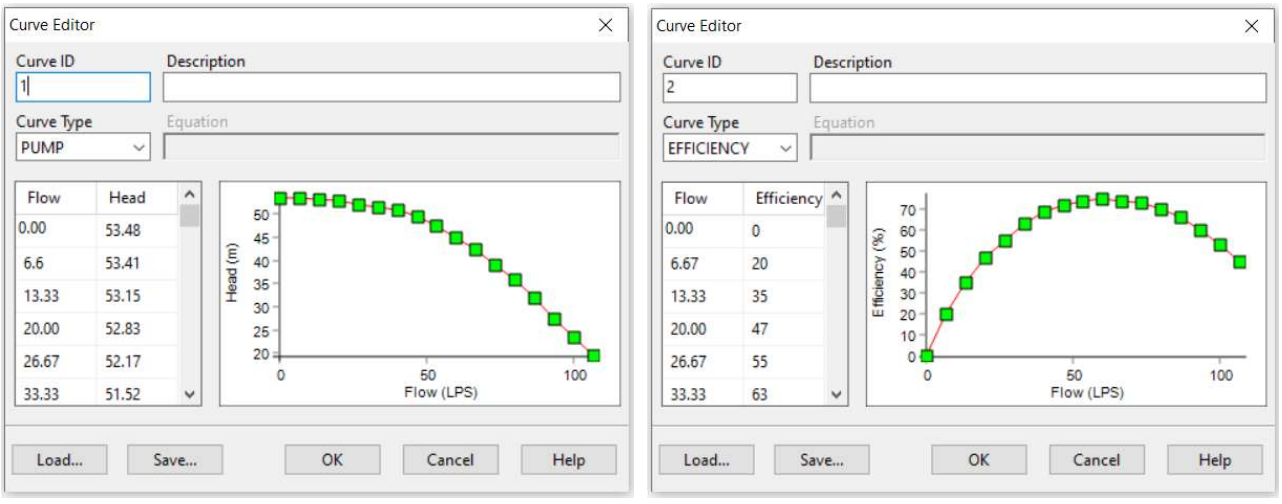


Figura 42. Inserimento curva caratteristica e curva di efficienza pompa su EPANET

Figura 43. Inserimento caratteristiche del fungo su EPANET

Nelle *times options* del software è necessario inserire le ore totali su cui fare la simulazione e il time step con cui si vogliono analizzare e visualizzare gli output, per esempio 1h o 10 minuti. In questo caso sono stati impostati 10 minuti.

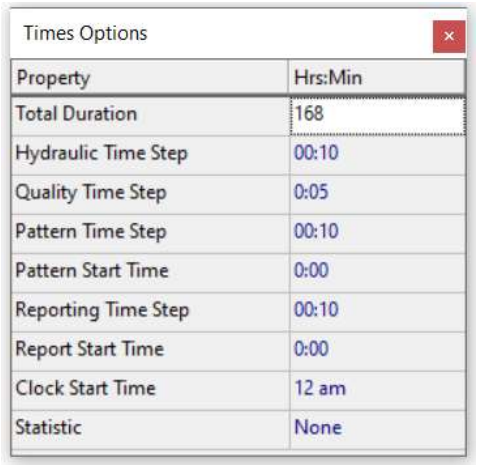


Figura 44. Impostazione Times Options su EPANET

Sono stati inseriti anche i controlli presenti per le pompe: l'ON e l'OFF delle 3 pompe sono stati impostati in base al livello dell'acqua raggiunto nel fungo. Per esempio la Pompa1 si accende quando il livello dell'acqua nel serbatoio si abbassa fino a 3,4m e si spegne quando raggiunge i 5m.

	POMPA1	POMPA2	POMPA3
ON	3,4m	3,2m	3m
OFF	5m	4,8m	4,6m

Figura 45. Controlli ON-OFF impostati alle pompe

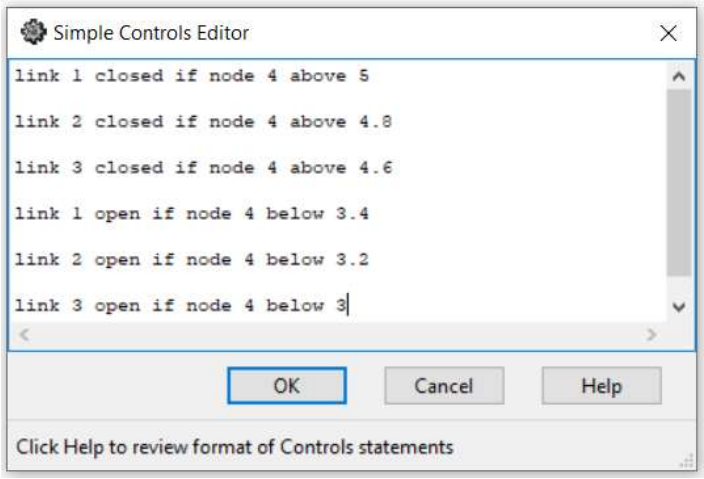


Figura 46. Controlli ON-OFF impostati alle pompe su EPANET

Gli output di Epanet posso essere visualizzati sia sotto forma grafica che tabellare.
In questo caso specifico è stata simulata la prima settimana di febbraio 2020, che andava da sabato 1 febbraio 2020 a venerdì 7 febbraio 2020.
Alcuni output utili per lo studio del modello sono la richiesta di acqua da parte dell'utenza, il livello dell'acqua nel fungo e la portata di acqua che passa attraverso ciascuna pompa. Nelle successive figure si mostrano gli output per una settimana di simulazione.

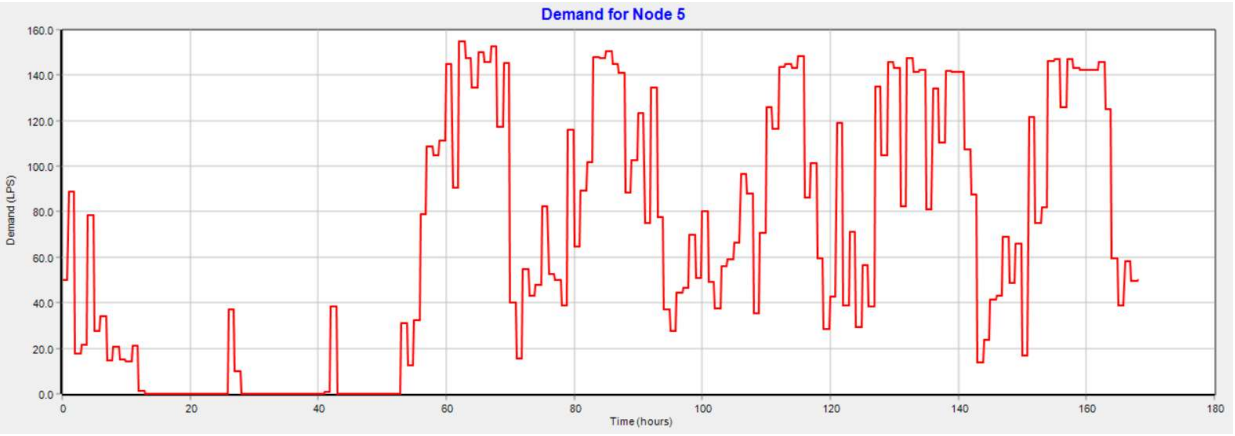


Figura 47. Output grafico di EPANET: richiesta di acqua da parte dell'utenza

Si può notare come nelle prime 48 ore di simulazione la richiesta di acqua da parte dell'utenza sia particolarmente bassa se non addirittura pari a 0: questo è facilmente spiegato dal fatto che i primi due giorni di febbraio nell'anno 2020 corrispondevano al sabato e alla domenica.

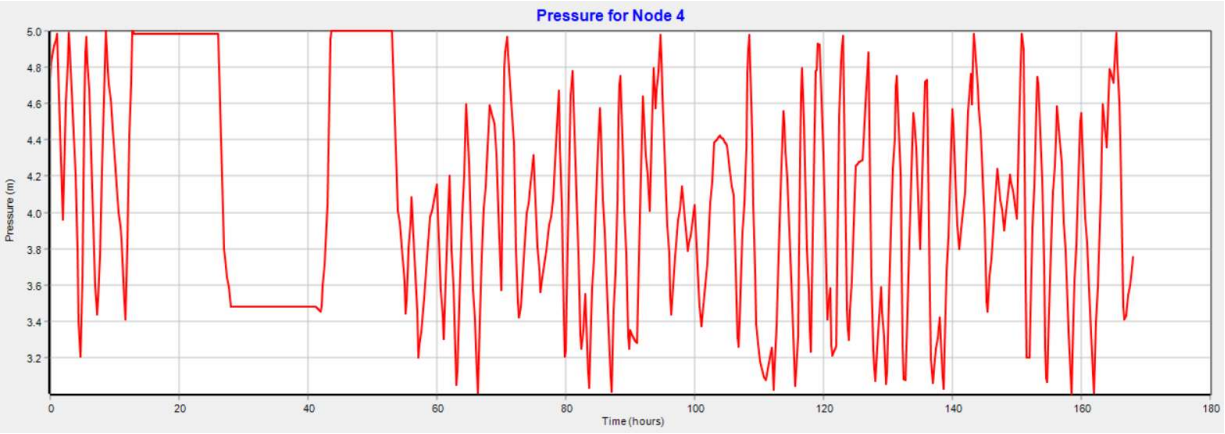


Figura 48. Output grafico di EPANET: livello acqua nel fungo

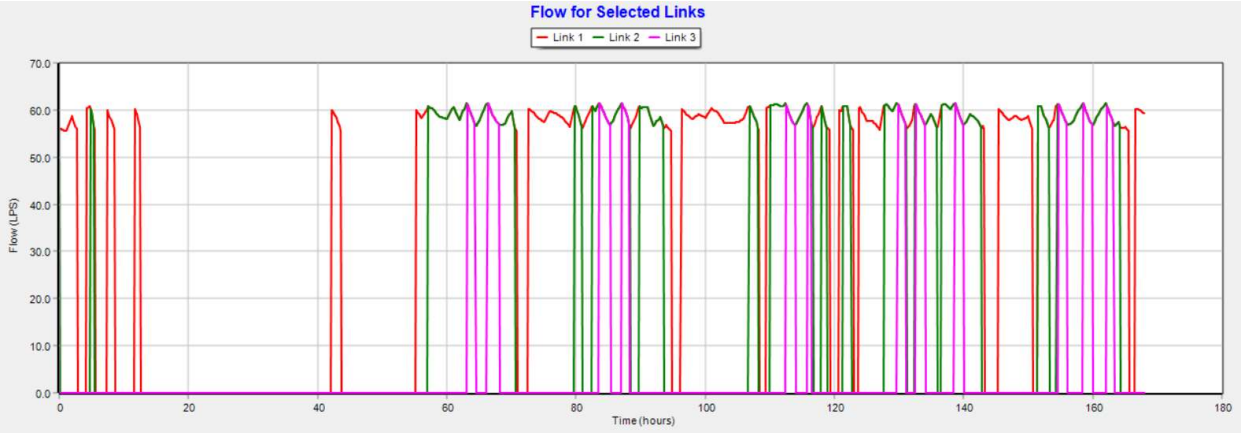


Figura 49. Output grafico di EPANET: portata di acqua attraverso le 3 pompe

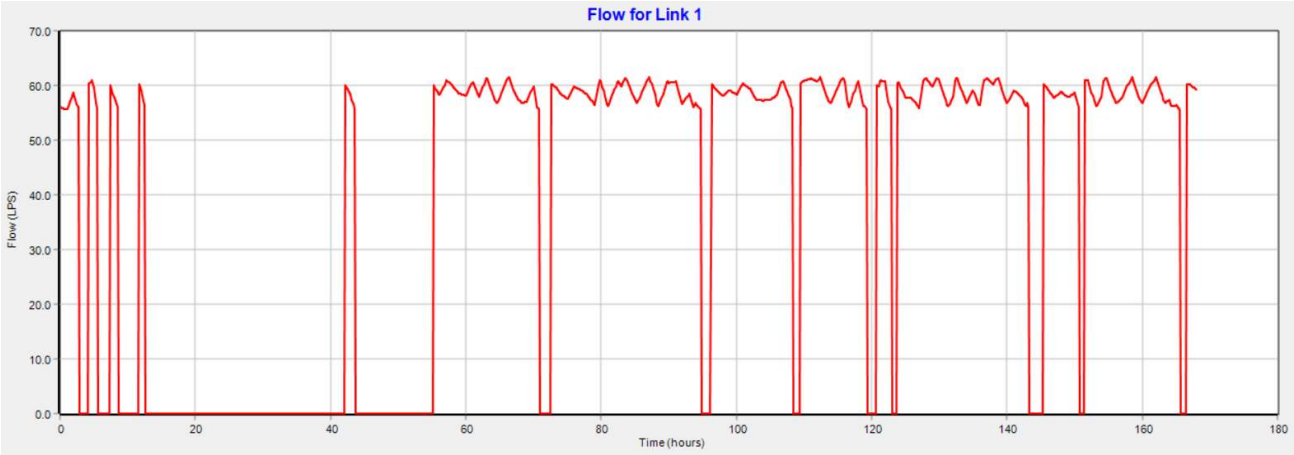


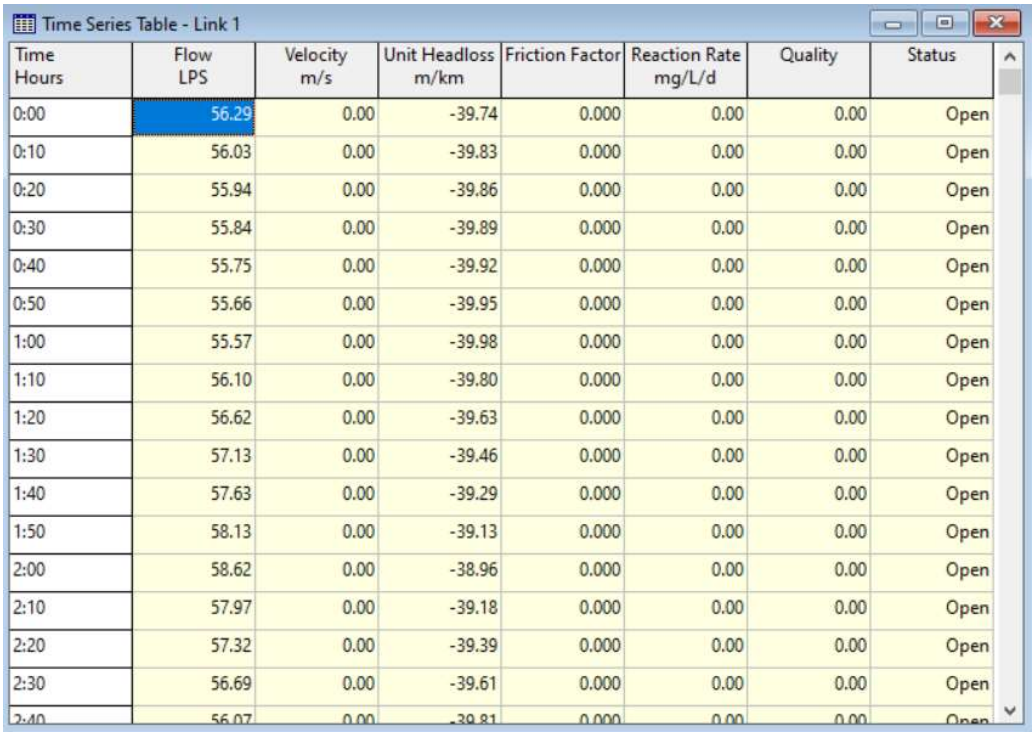
Figura 50. Output grafico di EPANET: dettaglio portata di acqua attraverso la pompa1

Vengono mostrati di seguito un paio di output di EPANET di tipo tabellare.

Time Series Table - Node 5

Time Hours	Demand LPS	Head m	Pressure m	Quality
0:00	50.00	14.74	14.74	0.00
0:10	50.00	14.83	14.83	0.00
0:20	50.00	14.86	14.86	0.00
0:30	50.00	14.89	14.89	0.00
0:40	50.00	14.92	14.92	0.00
0:50	50.00	14.95	14.95	0.00
1:00	88.89	14.98	14.98	0.00
1:10	88.89	14.80	14.80	0.00
1:20	88.89	14.63	14.63	0.00
1:30	88.89	14.46	14.46	0.00
1:40	88.89	14.29	14.29	0.00
1:50	88.89	14.12	14.12	0.00
2:00	17.50	13.96	13.96	0.00
2:10	17.50	14.18	14.18	0.00
2:20	17.50	14.39	14.39	0.00
2:30	17.50	14.61	14.61	0.00
2:40	17.50	14.81	14.81	0.00
2:50	17.50	14.99	14.99	0.00
3:00	21.67	14.90	14.90	0.00

Figura 51. Output tabellare di EPANET: richiesta di acqua da parte dell'utenza



Time Series Table - Link 1

Time Hours	Flow LPS	Velocity m/s	Unit Headloss m/km	Friction Factor	Reaction Rate mg/L/d	Quality	Status
0:00	56.29	0.00	-39.74	0.000	0.00	0.00	Open
0:10	56.03	0.00	-39.83	0.000	0.00	0.00	Open
0:20	55.94	0.00	-39.86	0.000	0.00	0.00	Open
0:30	55.84	0.00	-39.89	0.000	0.00	0.00	Open
0:40	55.75	0.00	-39.92	0.000	0.00	0.00	Open
0:50	55.66	0.00	-39.95	0.000	0.00	0.00	Open
1:00	55.57	0.00	-39.98	0.000	0.00	0.00	Open
1:10	56.10	0.00	-39.80	0.000	0.00	0.00	Open
1:20	56.62	0.00	-39.63	0.000	0.00	0.00	Open
1:30	57.13	0.00	-39.46	0.000	0.00	0.00	Open
1:40	57.63	0.00	-39.29	0.000	0.00	0.00	Open
1:50	58.13	0.00	-39.13	0.000	0.00	0.00	Open
2:00	58.62	0.00	-38.96	0.000	0.00	0.00	Open
2:10	57.97	0.00	-39.18	0.000	0.00	0.00	Open
2:20	57.32	0.00	-39.39	0.000	0.00	0.00	Open
2:30	56.69	0.00	-39.61	0.000	0.00	0.00	Open
2:40	56.07	0.00	-39.81	0.000	0.00	0.00	Open

Figura 52. Output tabellare di EPANET: portata di acqua attraverso la pompa1

Per rendere gli output del modello di EPANET più facilmente fruibili, analizzabili e utilizzabili per la successiva fase di Machine Learning, è stato utilizzato un tool che lega il software EPANET con l'ambiente Matlab.

Il tool utilizzato si chiama Toolkit EPANET-Matlab ed è un open-source software sviluppato da KOIS Research Center for Intelligent System and Networks of the University of Cyprus. Esso opera in ambiente Matlab per fornire un'interfaccia di programmazione al software EPANET. Al suo interno sono presenti comandi e funzioni in grado di visualizzare i risultati prodotti dalle librerie EPANET direttamente su Matlab, in modo che siamo più facilmente fruibili, per esempio per lo sviluppo di un algoritmo genetico o di rete neurale [13].

5.2 Ottimizzazione della regolazione del sistema tramite algoritmi genetici

Gli algoritmi genetici sono stati utilizzati per ottimizzare l'utilizzo delle pompe nel sistema idrico. La funzione obiettivo che si è deciso di minimizzare sono i consumi delle pompe. Le variabili ottime da trovare sono invece il livello dell'acqua nel serbatoio in cui le pompe si accendono o spengono. Si è partiti dal caso base con una sola pompa presente per poi passare a quello con tutte e 3 le pompe. Il tempo di simulazione è 1 settimana e il Δt temporale è 1 ora.
Di seguito sono presenti le tabelle con i risultati ottenuti: vengono evidenziati con sfondo rosso i valori che sono stati lasciati liberi di variare. Successivamente vengono mostrati nel dettaglio gli output grafici delle singole simulazioni.

	REGOLAZIONE		Consumi [kWh]	t
	ON	OFF		
Caso base - 1 pompa	ON fisso = 3m	OFF fisso = 5m	5723,6	-
Caso 1 - 1 pompa	ON fisso = 3m	OFF variabile = 3,26m	5651,86	1,5 h
Caso 2 - 1 pompa	ON variabile = 3m	OFF variabile = 3,25m	5649,45	1,5 h

Tabella 1. Riepilogo regolazione GA - 1 pompa

	REGOLAZIONE						Consumi [kWh]	t
	ON Pompa1	OFF Pompa1	ON Pompa2	OFF Pompa2	ON Pompa3	OFF Pompa3		
Caso base - 3 pompe	ON fisso = 3m	OFF fisso = 5m	ON fisso = 3m	OFF fisso = 5m	ON fisso = 3m	OFF fisso = 5m	5723,7	-
Caso 1 - 3 pompe	ON fisso = 3m	OFF variabile = 3,2m	ON fisso = 3m	OFF fisso = 5m	ON fisso = 3m	OFF fisso = 5m	5705,25	1 h
Caso 2 - 3 pompe	ON variabile = 3m	OFF variabile = 3,21m	ON fisso = 3m	OFF fisso = 5m	ON fisso = 3m	OFF fisso = 5m	5702,91	1 h
Caso 3 - 3 pompe	ON fisso = 3m	OFF variabile = 3,2m	ON fisso = 3m	OFF variabile = 3,21m	ON fisso = 3m	OFF variabile = 3,26m	5664,68	1,5 h
Caso 4 - 3 pompe	ON variabile = 3m	OFF variabile = 3,4m	ON variabile = 3m	OFF variabile = 3,5m	ON variabile = 3m	OFF variabile = 3,2m	5670,85	1 h

Tabella 2. Riepilogo regolazione GA - 3 pompe

Il codice completo è formato da due script:

- il codice principale, nel quale si impostano le opzioni dell'algoritmo genetico.
    1) Viene dichiarato il numero di variabili che l'algoritmo deve trovare, i limiti inferiori e superiori di ognuna e quali vincoli ciascuna di esse deve rispettare;

2) vengono impostate le opzioni dell'algoritmo: grandezza della popolazione per ogni generazione *PopulationSize* (per questa trattazione si è scelta pari a 100 individui), il numero massimo di generazioni *MaxGeneration* (in questo lavoro pari a 10), e il numero di generazioni consecutive *MaxStallGeneration* oltre le quali fermare l'algoritmo nel caso in cui i consumi ottimali non varino da una generazione all'altra di più di 1 kWh (*FunctionTolerance*);
3) viene chiamata la funzione di MATLAB algoritmo genetico *ga*;
4) viene chiamata la funzione "fitness" (spiegata nel punto successivo) la quale restituisce il valore del consumo corrispondente alla migliore regolazione trovata dall'algoritmo genetico.

Nella Figura 53 e Figura 54 seguenti sono riportati due esempi di codice principale, sia del caso con 1 pompa che di quello con 3 pompe, con evidenziate tutte le varie fasi.

GA_1pompa.m

```
1 -  clc
2 -  close all
3 -  clear all
4
5 -  tic
6
7 -  ObjectiveFunction = @fitness;
8 -  nvars = 2;    % Numero delle variabili = 2 ---> min1 e max1
9 -  LB = [3, 3.2];   % Limiti inferiori
10 - UB = [4.8, 5];  % Limiti superiori
11 - A = [1,-1];
12 - b = [-0.2];
13
14 - options = optimoptions('ga','PopulationSize',100,'MaxGenerations',10,'MaxStallGenerations',8,'FunctionTolerance', 3.5e-5,'PlotFcn',@gaplotbestf);
15
16 - [x,fval,exitflag,output] = ga(ObjectiveFunction,nvars,A,b,[],[],LB,UB,[],options);
17
18 - [Consumo_tot,C3,C2] = fitness(x);
```

1) 2) 3) 4)

Figura 53. Esempio codice principale GA: CASO 2 - 1 pompa

GA_3pompe.m

```
1 -  clc
2 -  close all
3 -  clear all
4
5 -  tic
6
7 -  ObjectiveFunction = @fitness;
8 -  nvars = 6;    % Numero delle variabili = 6 ---> min1 e max1, min2 e max2, min3 e max3
9 -  LB = [3, 3.2, 3, 3.2, 3, 3.2];   % Limiti inferiori
10 - UB = [4.8, 5, 4.8, 5, 4.8, 5];  % Limiti superiori
11 - A = [1 -1 0 0 0 0;0 0 1 -1 0 0;0 0 0 0 1 -1];
12 - b = [-0.2;-0.2;-0.2];
13
14 - options = optimoptions('ga','PopulationSize',100,'MaxGenerations',10,'MaxStallGenerations',8,'FunctionTolerance', 3.5e-5,'PlotFcn',@gaplotbestf);
15
16 - [x,fval,exitflag,output] = ga(ObjectiveFunction,nvars,A,b,[],[],LB,UB,[],options);
17
18 - [Consumo_tot,C3,C2] = fitness(x);
```

1) 2) 3) 4)

Figura 54. Esempio codice principale GA: CASO 4 - 3 pompe

- il codice con la funzione fitness, che permette il calcolo effettivo dei consumi una volta cambiati gli intervalli di accensione/spegnimento delle pompe con i valori trovati dall'algoritmo.
    1) Vengono salvati in delle variabili i valori trovati dall'algoritmo genetico;
    2) tramite il tool EPANET-MATLAB Toolkit viene caricato su MATLAB il file .inp del modello (precedentemente simulato su EPANET);
    3) vengono cancellati i controlli sulle pompe;
    4) gli stessi vengono riscritti utilizzando come valori di ON e OFF i valori trovati dall'algoritmo;
    5) viene rilanciata la simulazione tramite EPANET MATLAB Toolkit;
    6) vengono calcolati i consumi.

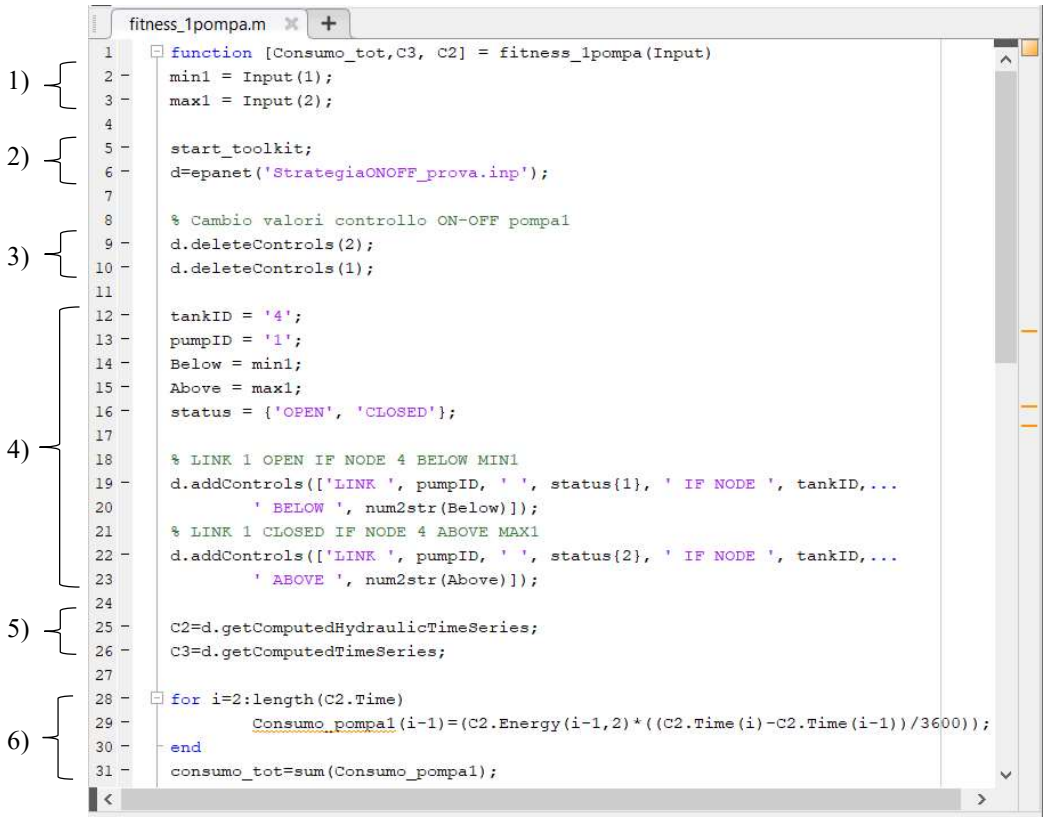


Figura 55. Esempio codice con fitness GA: CASO 2 - 1 pompa

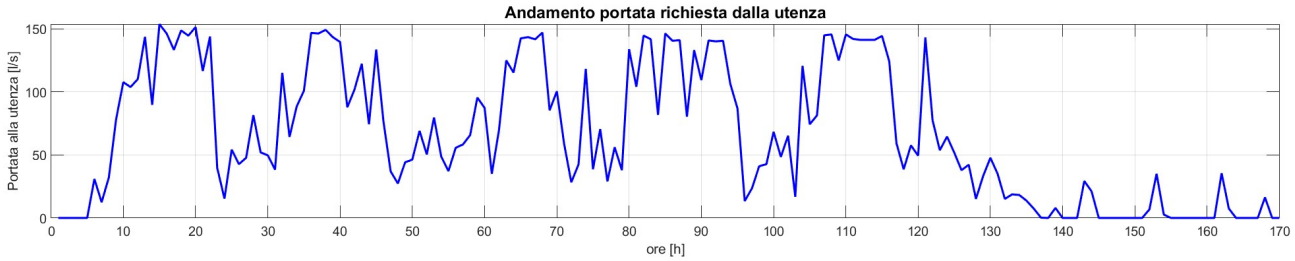
Per il calcolo dei consumi è stata utilizzata una funzione presente nel toolkit EPANET MATLAB Toolkit che restituisce il valore di potenza istantaneo: una volta moltiplicato per il tempo di accensione della pompa si è arrivati al valore di energia.

CASO BASE - 1 POMPA

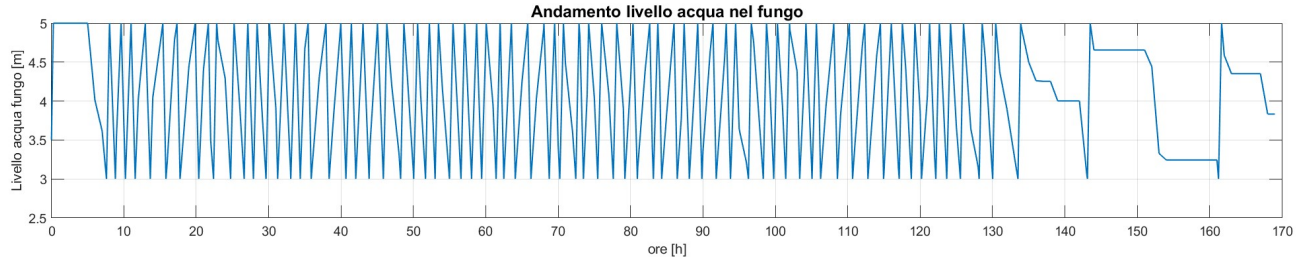
Nel caso base la pompa si accende quando il livello dell'acqua nel serbatoio raggiunge 3 m e si spegne quando raggiunge i 5 m.

Gli output grafici del modello sono i seguenti:

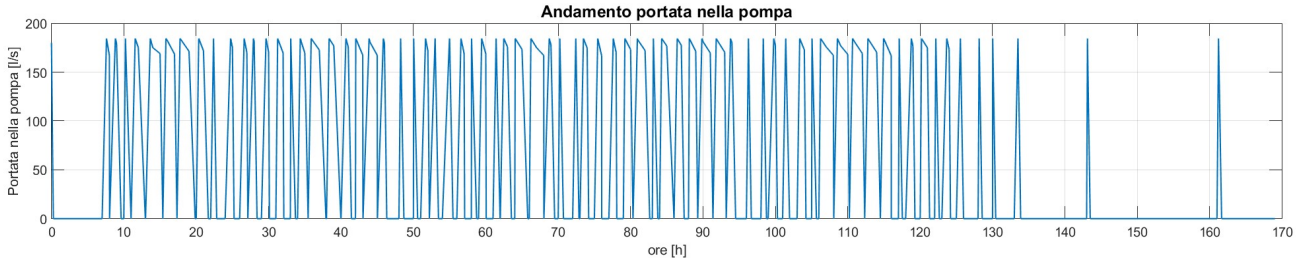
a)



b)



c)



d)

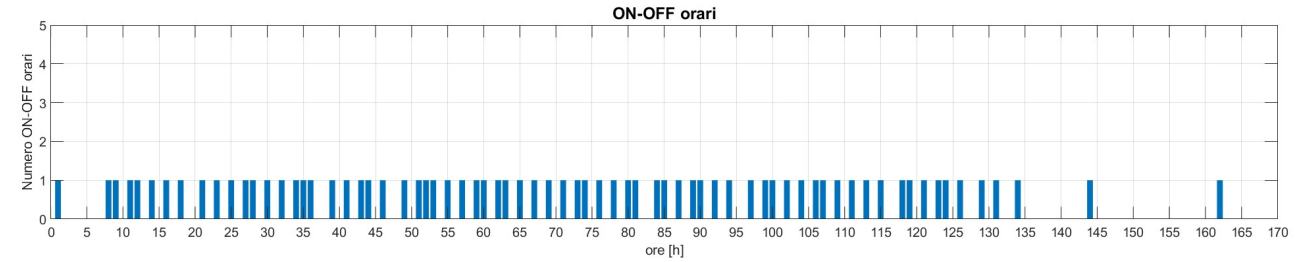


Figura 56. Output grafici: caso base - 1 pompa

Si può notare dai grafici precedenti come l'andamento dell'acqua nel fungo (Figura 56.b) vari regolarmente tra il minimo di 3 m e il massimo di 5, mentre la pompa si accende e spegne regolarmente nei giorni infrasettimanali e di meno nel week end (Figura 56.c) , in quanto come si

può notare dalla Figura 56.a, la richiesta da parte dell'utenza è sensibilmente minore nel fine settimana.

CASO 1 - 1 POMPA

L'ON della pompa è fissato a 3 m. L'OFF viene lasciato libero di variare tra 3,2 e 5m. L'algoritmo trova il valore ottimale di spegnimento della pompa quando il livello dell'acqua nel serbatoio raggiunge 3,26m. Nella figura seguente è evidenziato in rosso il parametro lasciato libero di variare.

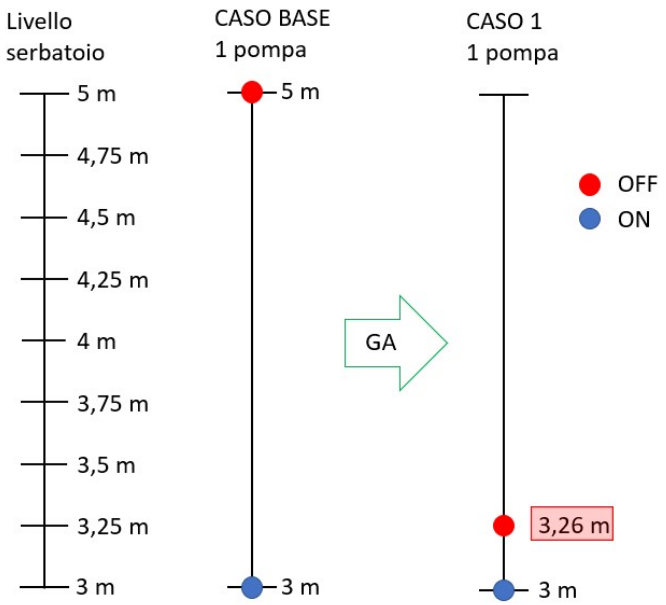


Figura 57. Confronto CASO BASE - CASO 1: 1 pompa

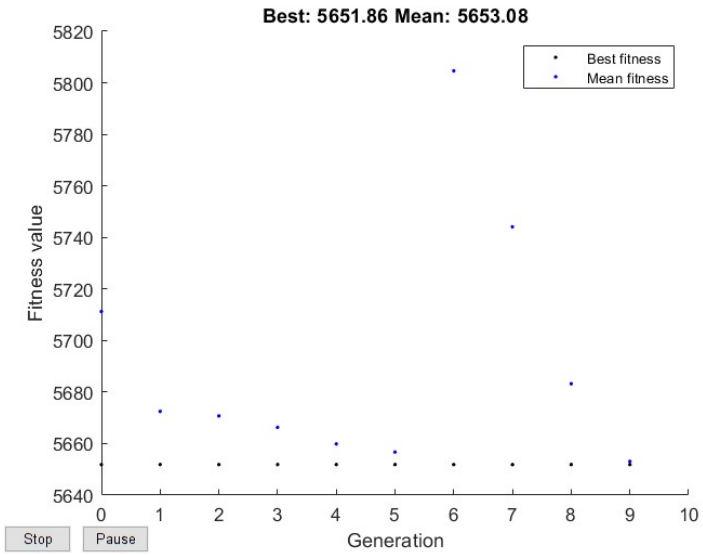
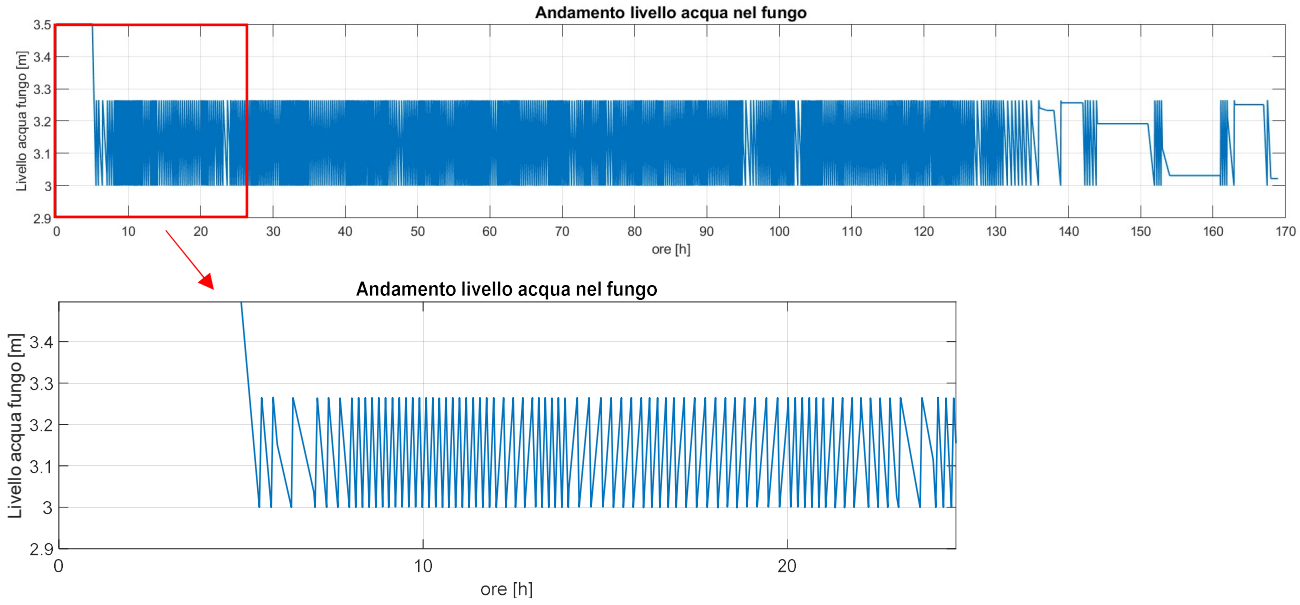


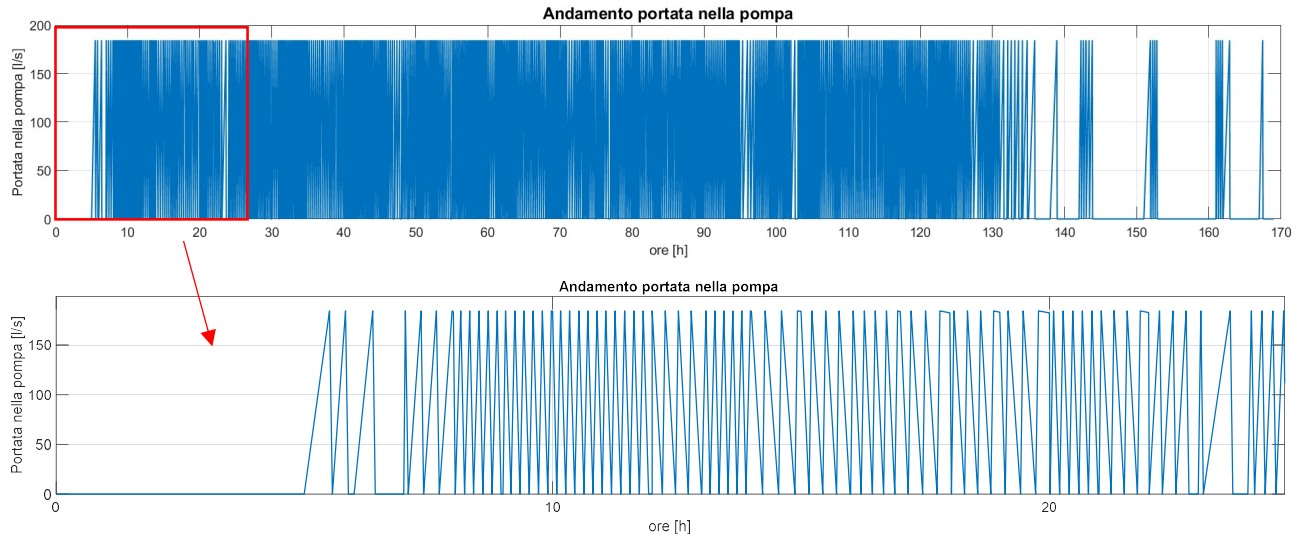
Figura 58. GA: caso 1 - 1 pompa

Nella Figura 58 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

a)



b)



c)

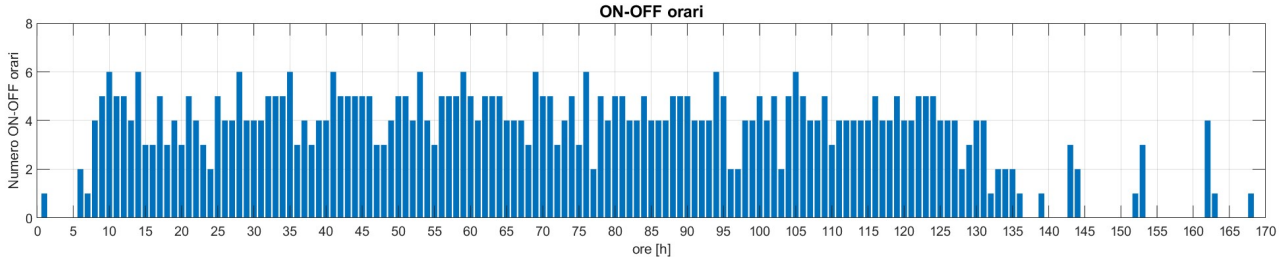


Figura 59. Output grafici: caso 1 - 1 pompa

Nelle precedenti Figura 59.a e Figura 59.b si possono notare i dettagli dell'andamento dell'acqua nel serbatoio e della portata di acqua nella pompa: gli andamenti sono così fitti per via dei numerosissimi ON-OFF orari, che si notano anche nell'ultimo grafico (Figura 59.c). Al valore ottimale dell' OFF pari a 3,26m si è arrivati impostando un controllo nel numero di ON-OFF massimi orari, che deve essere massimo pari a 6.

CASO 2 - 1 POMPA

In questo secondo caso l'ON e l'OFF vengono lasciati entrambi liberi di variare alla ricerca dell'ottimo: l'ON tra 3 e 4,8m e l'OFF tra 3,2 e 5m. Tra l'ON e l'OFF è impostata una differenza minima di 0,2m. L'algoritmo trova i valori ottimali pari a 3m per l'ON e 3,25m per l'OFF.

Nella figura seguente sono evidenziati in rosso i parametri lasciati liberi di variare.

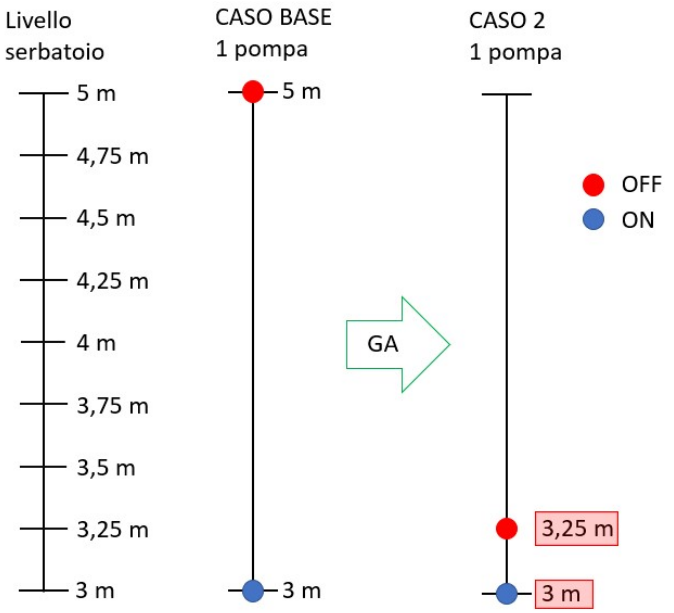


Figura 60. Confronto CASO BASE - CASO 2: 1 pompa

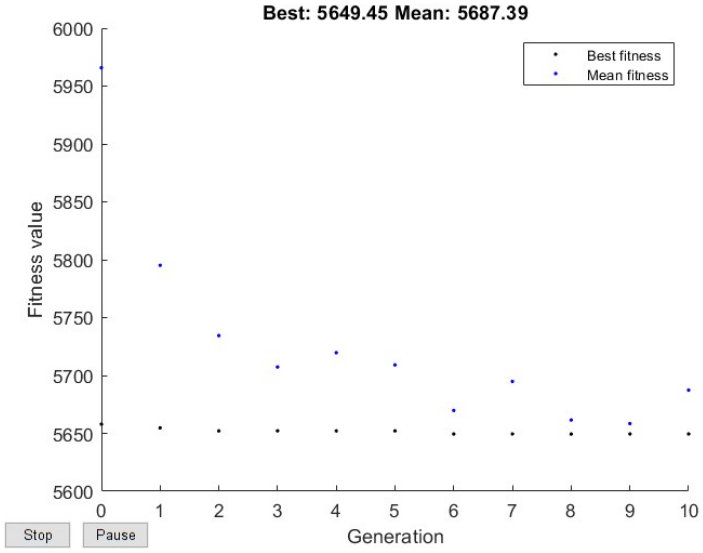
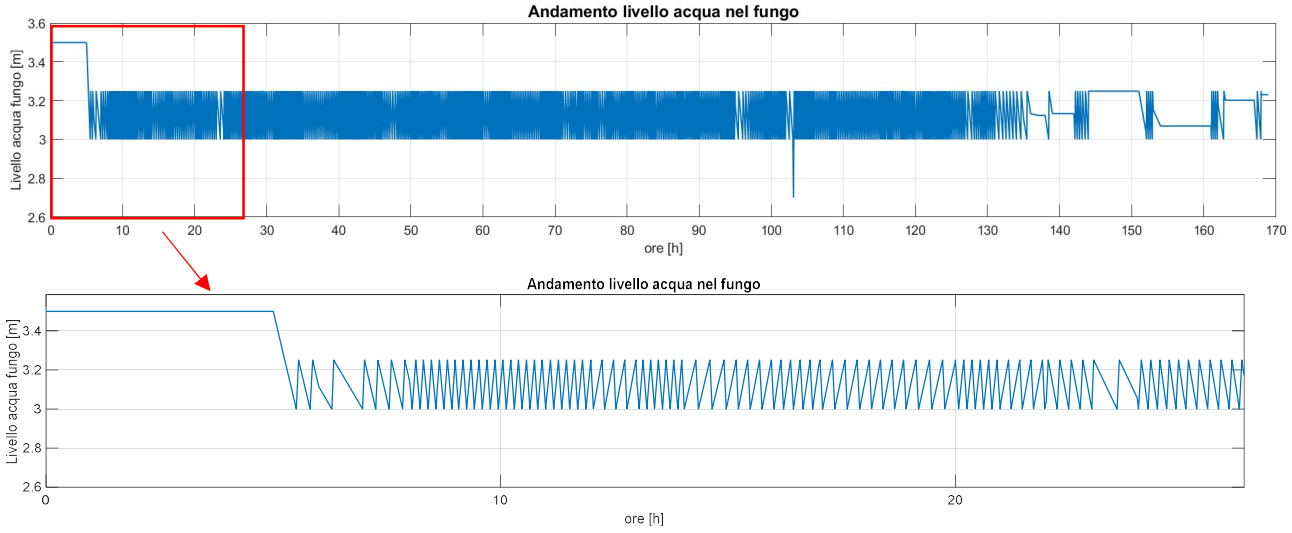


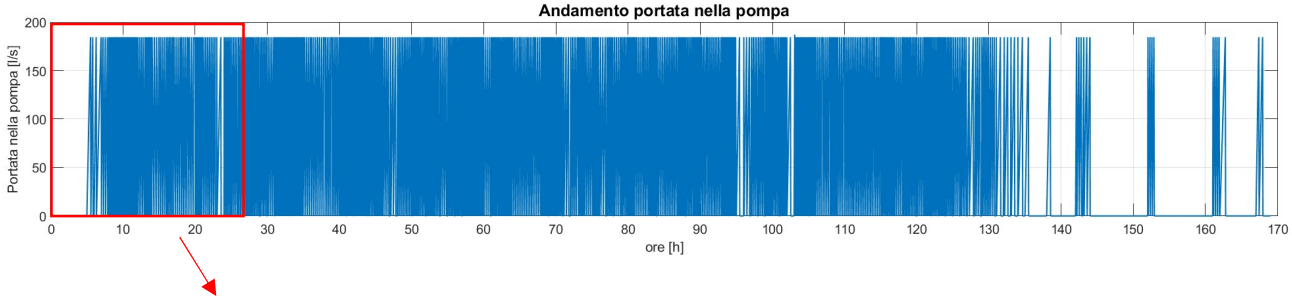
Figura 61. GA: caso 2 - 1 pompa

Nella Figura 61 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

a)



b)



c)

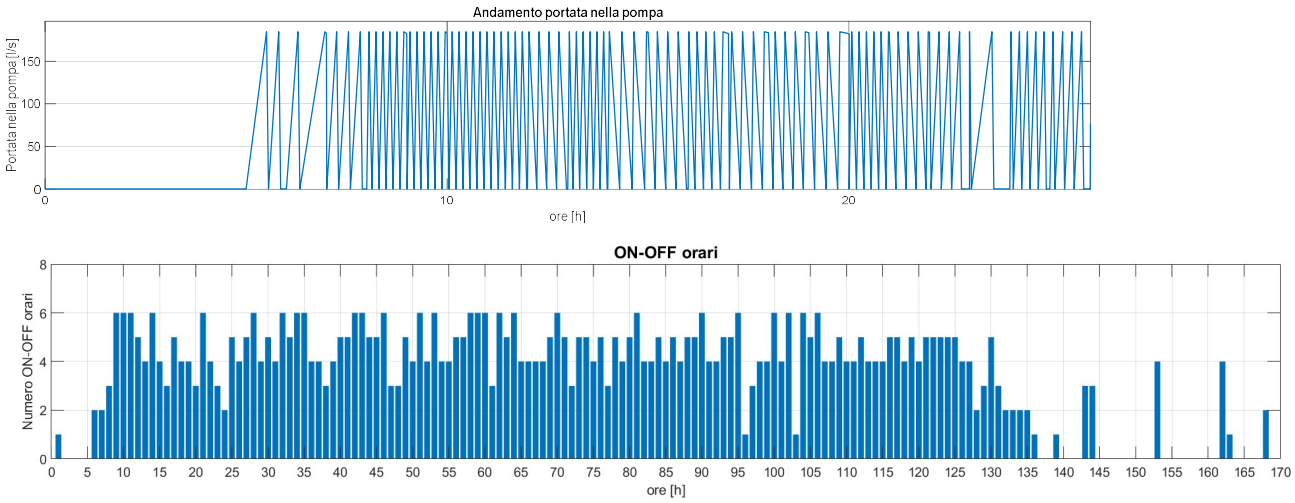


Figura 62. Output grafici: caso 2 - 1 pompa

Si può notare come nel CASO 2 – 1 POMPA, nonostante l'ON-OFF sia lasciato libero di variare in tutto il range del serbatoio, esso tende a trovare l'accensione e lo spegnimento dei valori più bassi possibili: l'accensione può avvenire tra 3 e 4,8m e viene trovato il valore ottimale pari a 3m, lo spegnimento può variare tra 3,2 e 5m e viene trovato il valore ottimale pari a 3,25m.

CONFRONTO CASI: 1 POMPA

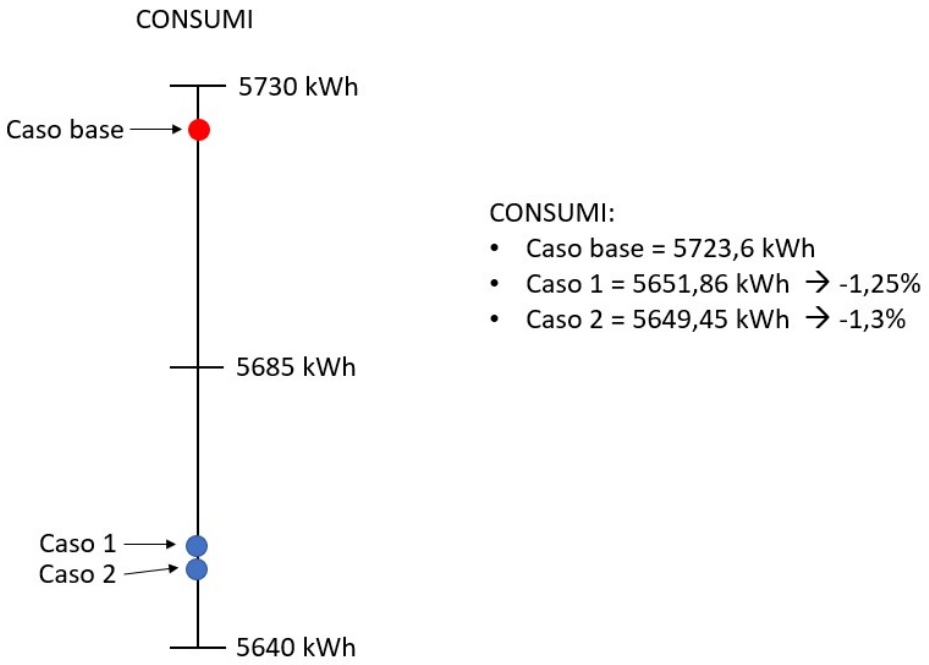


Figura 63. Riepilogo confronti GA - 1 pompa

Dai risultati si evince che nei CASO 1 e 2 lo spegnimento della pompa scende fino quasi al minimo possibile, soluzione che permette al sistema di consumare di meno tenendo la pompa accesa meno a lungo. Nel CASO 2, con entrambe le variabili libere di cambiare, la soluzione tende a raggiungere il minimo possibile sia nell'ON che nell'OFF.

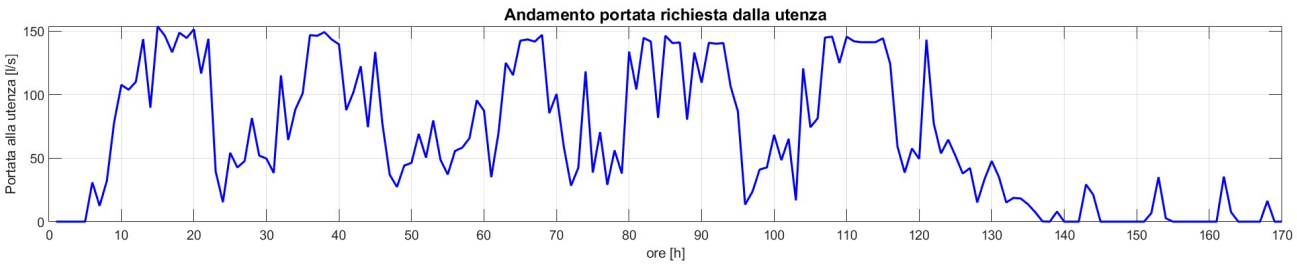
Si rimanda alla Tabella 1 per il riepilogo dei risultati trovati.

CASO BASE: 3 POMPE

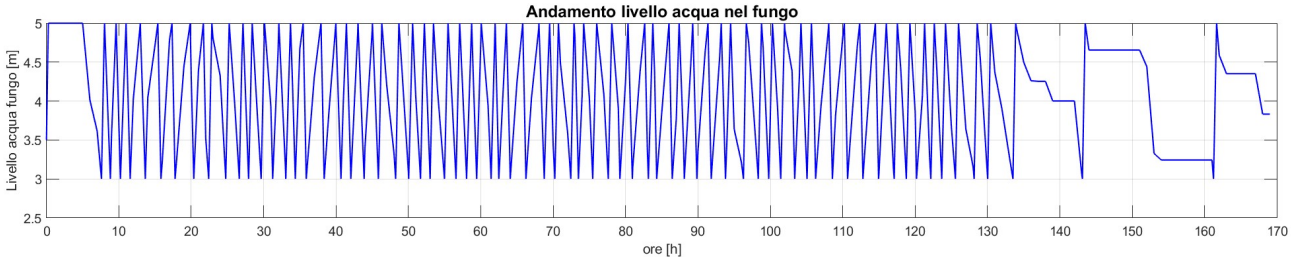
Nel caso base le tre pompe si accendono tutte insieme quando il livello dell'acqua nel serbatoio è pari a 3m e si spengono quando raggiunge i 5m.

Gli output grafici del modello sono i seguenti:

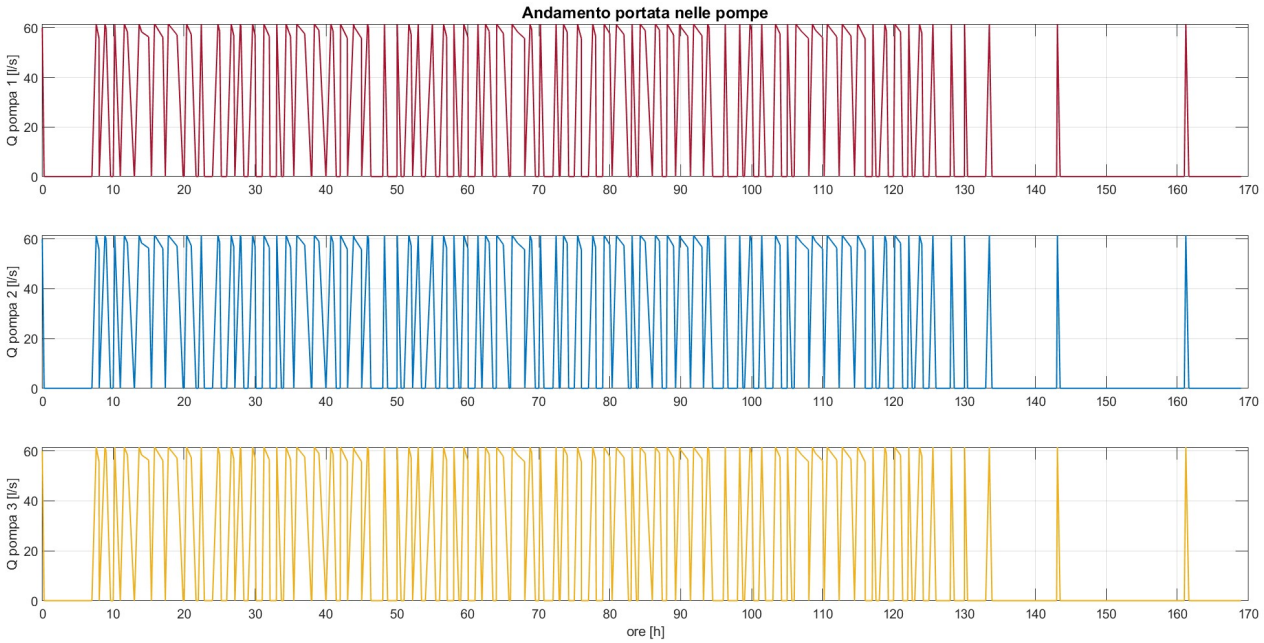
a)



b)



c)



d)

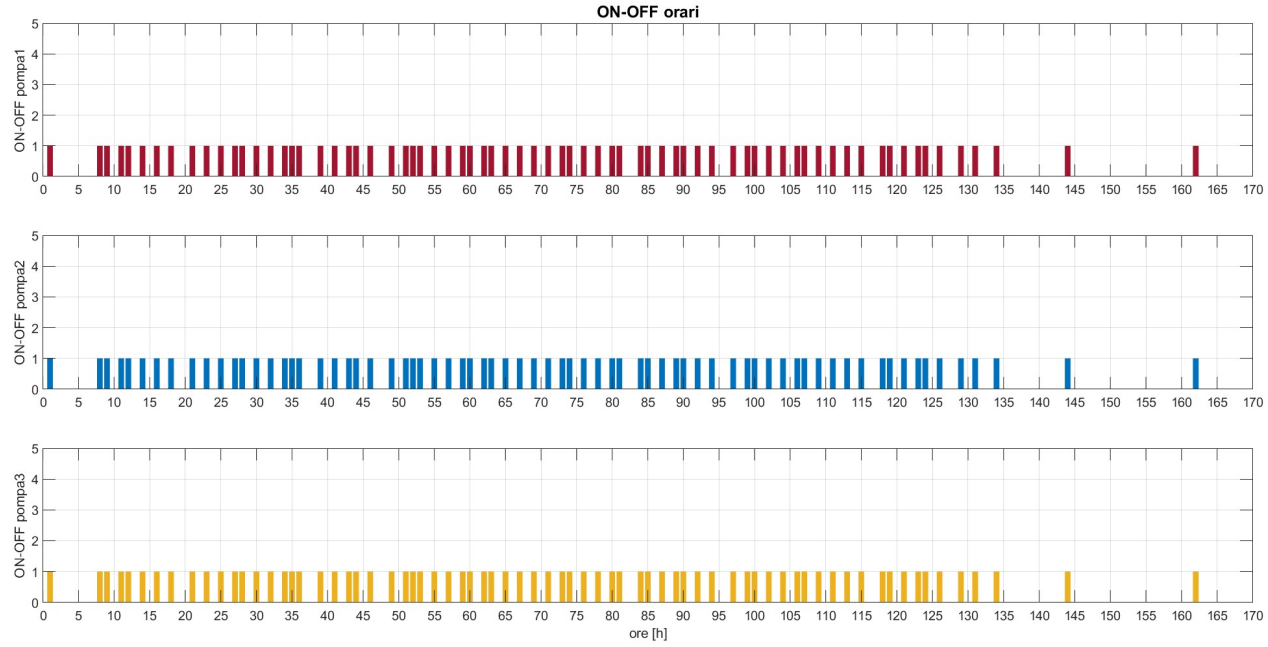


Figura 64. Output grafici: caso base - 3 pompe

Si può notare dai grafici precedenti come l'andamento dell'acqua nel fungo (Figura 64.b) vari regolarmente tra il minimo di 3 m e il massimo di 5, mentre la pompa si accende e spegne regolarmente nei giorni infrasettimanali e di meno nel week end (Figura 64.c) , in quanto come si

può notare dalla Figura 64.a, la richiesta da parte dell'utenza è sensibilmente minore nel fine settimana.

CASO 1: 3 POMPE

L'ON della pompa 1, 2 e 3 è fissato a 3 m. L'OFF della pompa 2 e 3 è fissato a 5m, mentre quello della pompa 1 viene lasciato libero di variare tra 3,2 e 5m. L'algoritmo trova il valore ottimale di spegnimento della pompa1 quando il livello dell'acqua nel serbatoio raggiunge 3,2m. Nella figura seguente sono evidenziati in rosso i parametri lasciati liberi di variare.

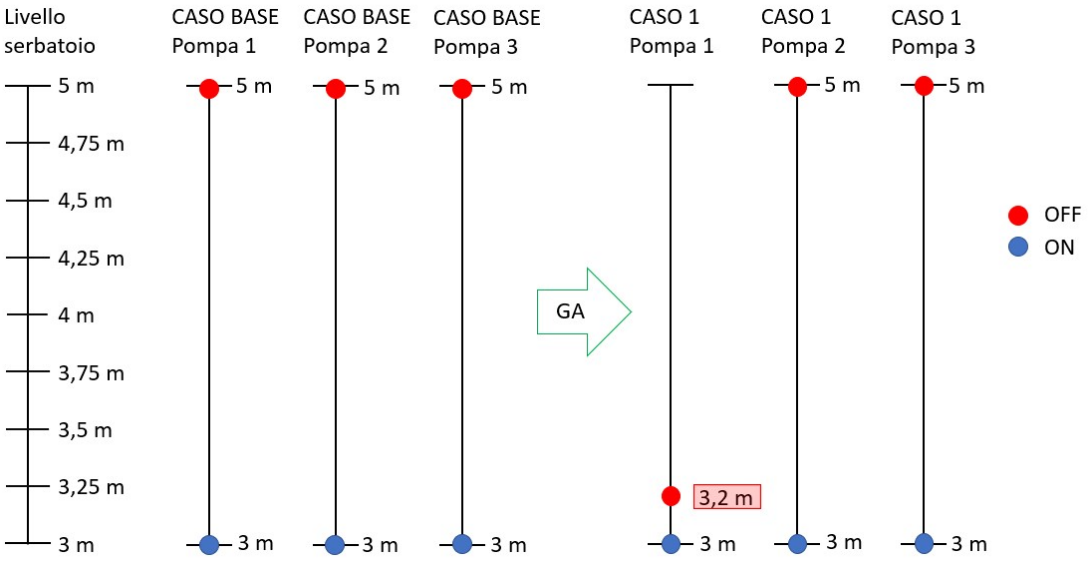


Figura 65. Confronto CASO BASE - CASO 1: 3 pompe

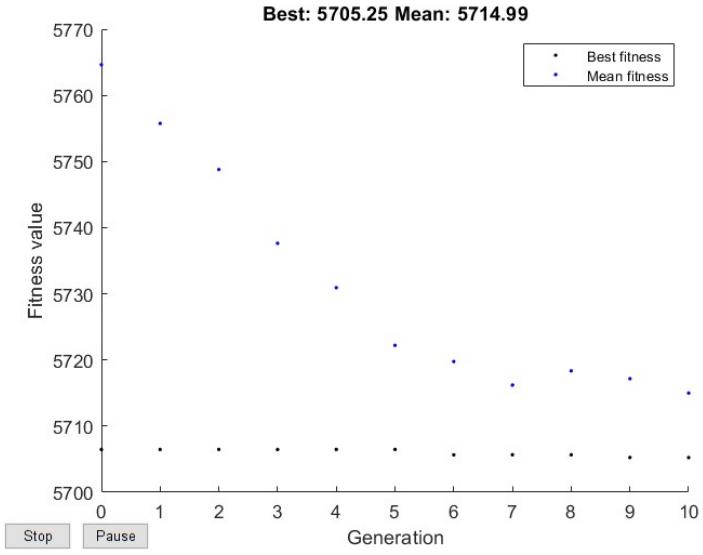
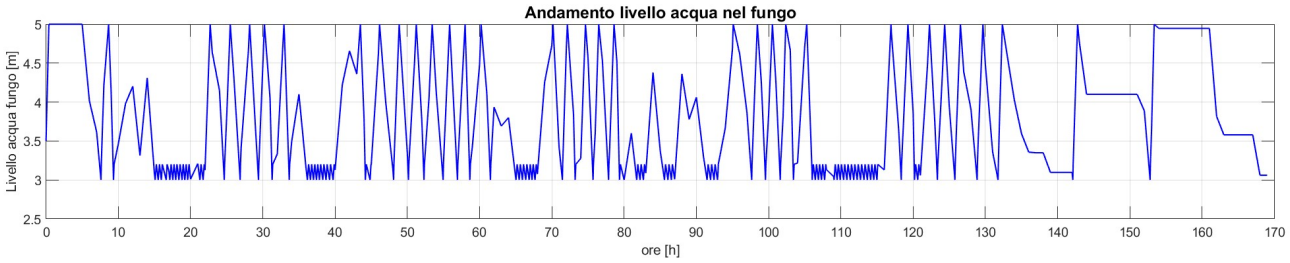


Figura 66. GA: caso 1 - 3 pompe

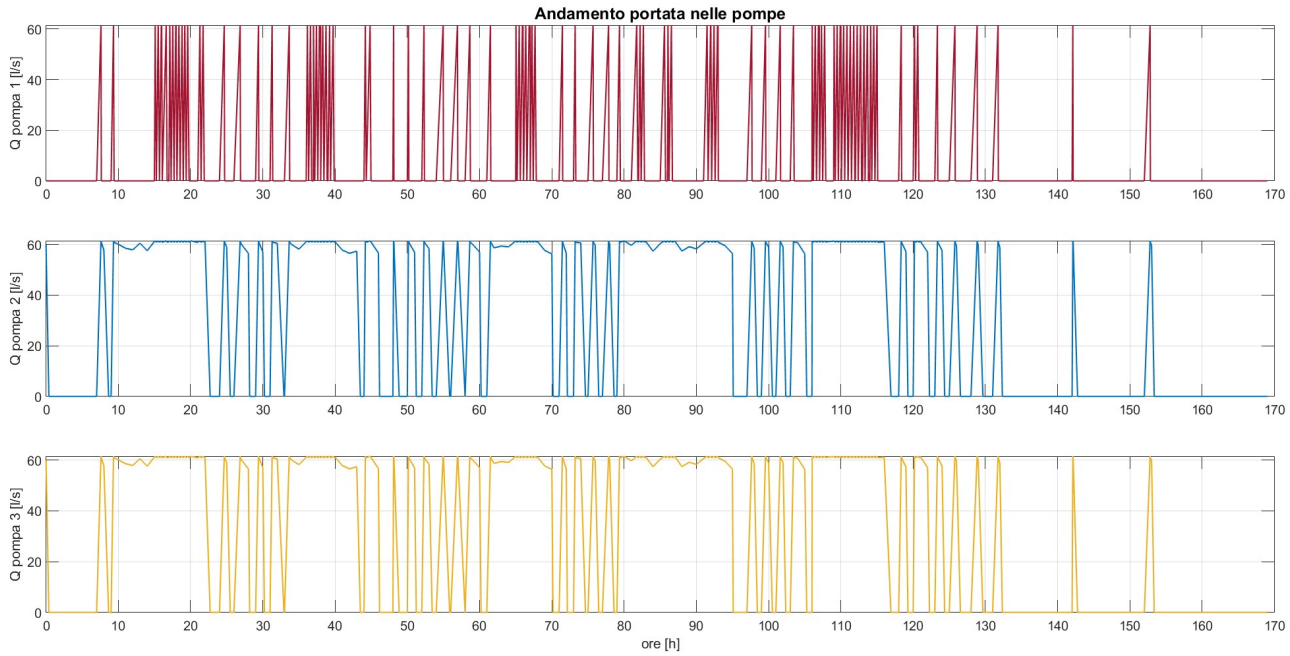
Nella Figura 66 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

Di seguito vengono mostrati gli output grafici della simulazione:

a)



b)



c)

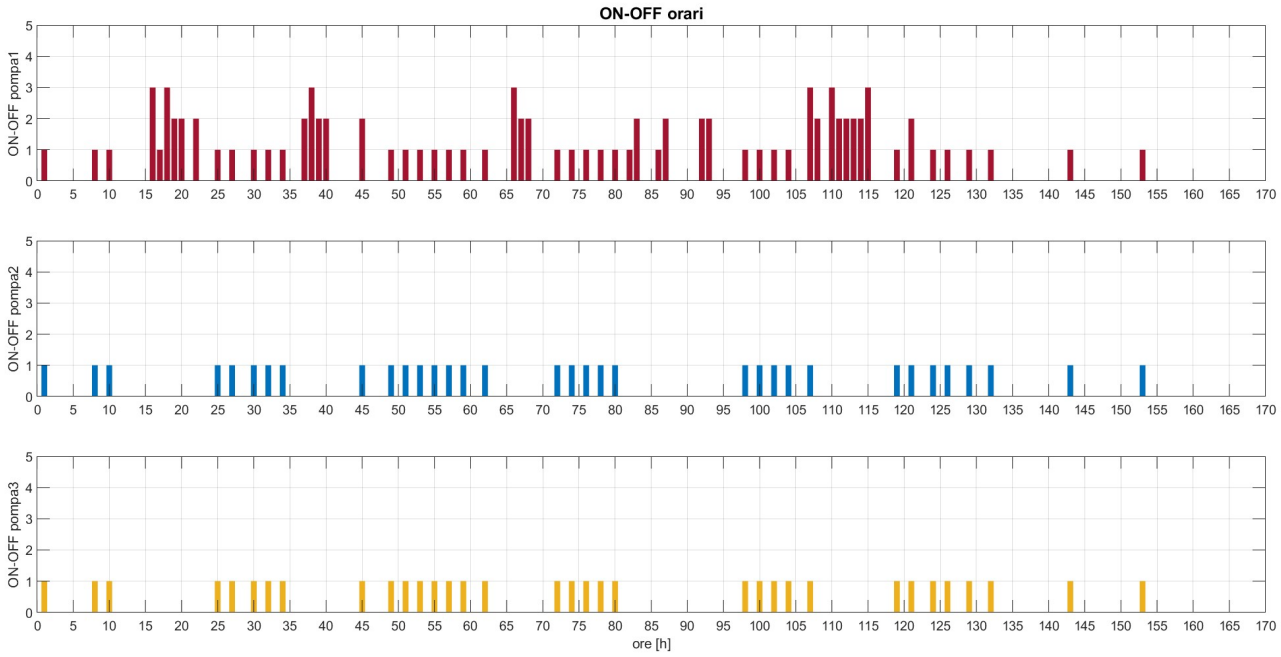


Figura 67. Output grafici: caso 1 - 3 pompe

Analizzando i grafici precedenti e soffermandosi sulla pompa1 che è la sola con una regolazione differente, si può notare come l'OFF ottimale trovato pari a 3,2m, porta il livello nell'acqua nel serbatoio a non variare più regolarmente tra 3 e 5m come nel caso base, ma in alcune ore esso oscilla solo tra 3 e 3,2 (Figura 67.a) portando la pompa1 a subire numerosi ON-OFF orari (Figura 67.c). Nelle ore in cui la pompa1 si accende e spegne con molta frequenza, le pompe 2 e 3 rimangono accese (Figura 67.b)

CASO 2: 3 POMPE

L'ON della pompa 1 viene lasciato libero di variare tra 3 e 4,8m, mentre quello della pompa 2 e 3 è fissato a 3 m. L'OFF della pompa 2 e 3 è fissato a 5m, mentre quello della pompa 1 viene lasciato libero di variare tra 3,2 e 5m. L'algoritmo trova il valore ottimale di accensione della pompa1 quando il livello dell'acqua nel serbatoio raggiunge 3m, e quello di spegnimento quando raggiunge 3,21m. Nella figura seguente sono evidenziati in rosso i parametri lasciati liberi di variare.

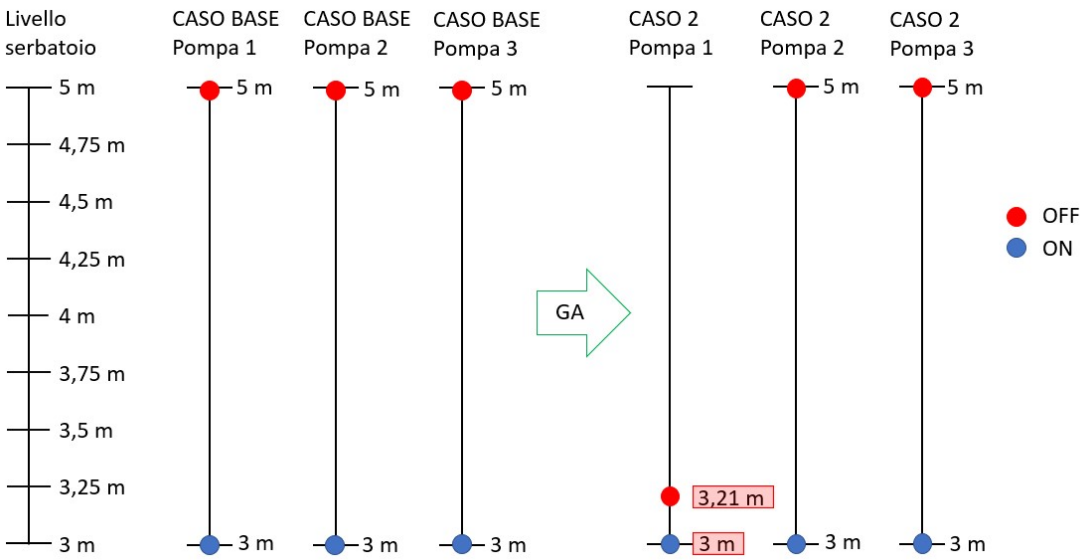


Figura 68. Confronto CASO BASE - CASO 2: 3 pompe

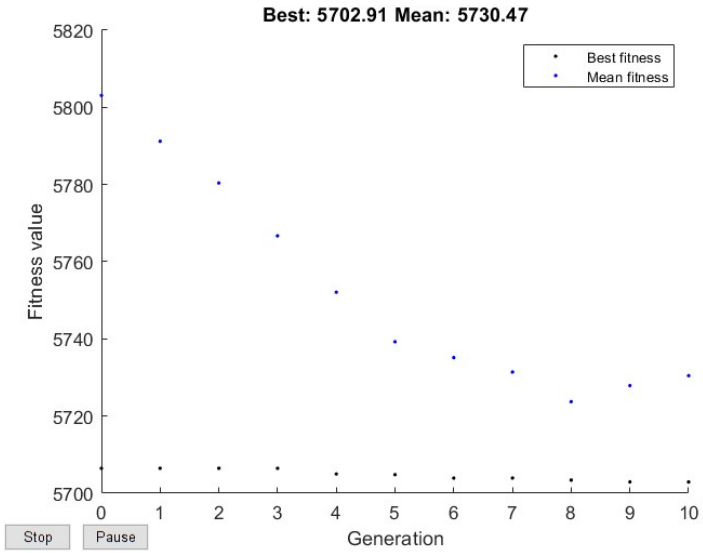
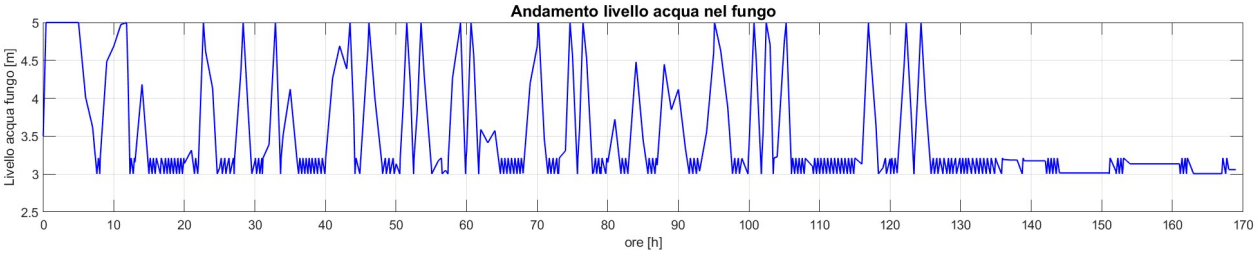


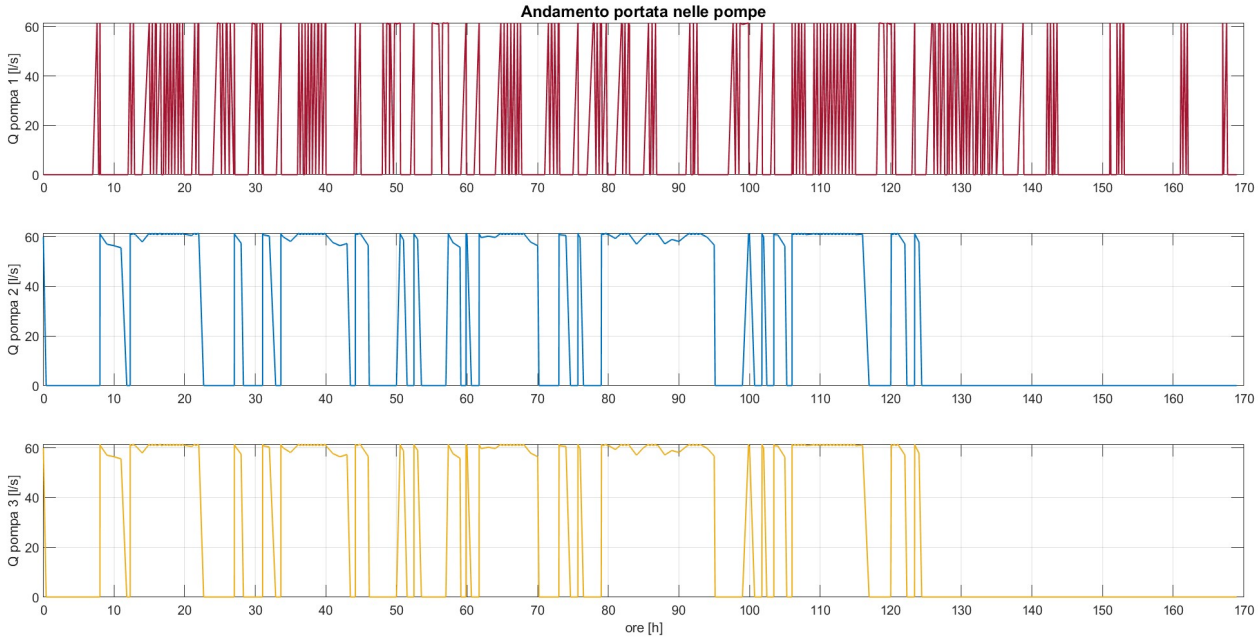
Figura 69. GA: caso 2 - 3 pompe

Nella Figura 69 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

a)



b)



c)

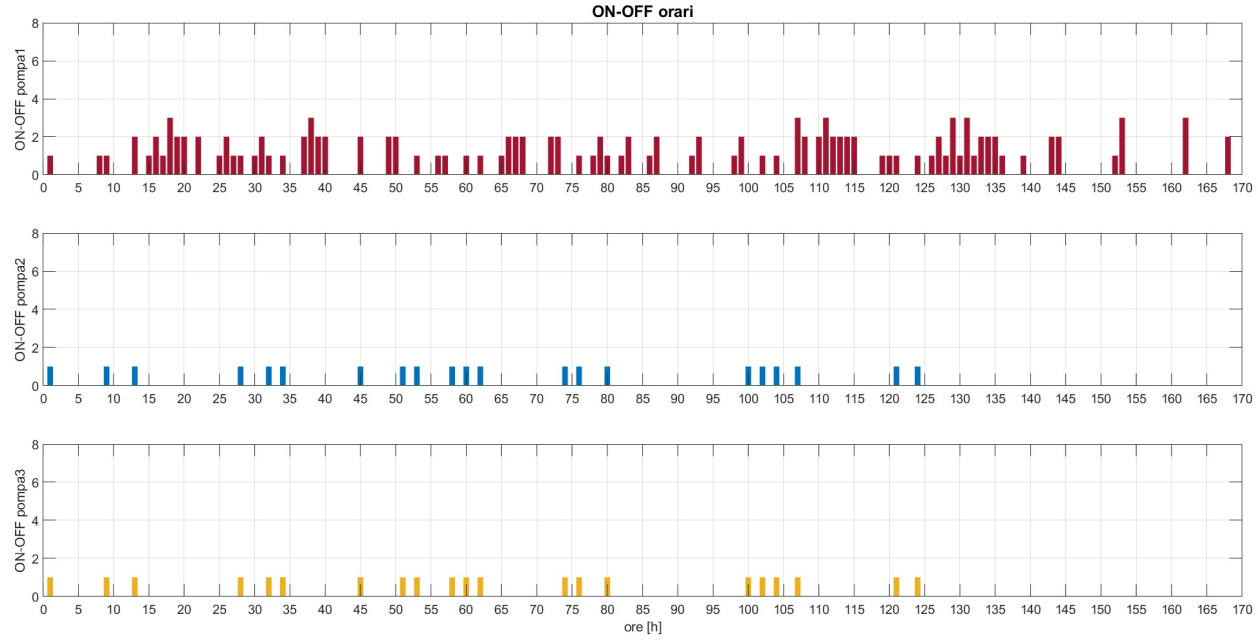


Figura 70. Output grafici: caso 2 - 3 pompe

I grafici del CASO1 e del CASO2 sono molto simili, avendo l'algoritmo trovato degli ON-OFF per la pompa1 quasi uguali: valgono le stesse considerazioni già fatte per il CASO1.

CASO 3: 3 POMPE

L'ON della pompa 1, 2 e 3 è fissato a 3 m. L'OFF della pompa 1, 2 e 3 è lasciato libero di variare tra 3,2 e 5m. L'algoritmo trova il valore ottimale di spegnimento delle 3 pompe rispettivamente a 3,2m, 3,21m e 3,26m. Nella figura seguente sono evidenziati in rosso i parametri lasciati liberi di variare.

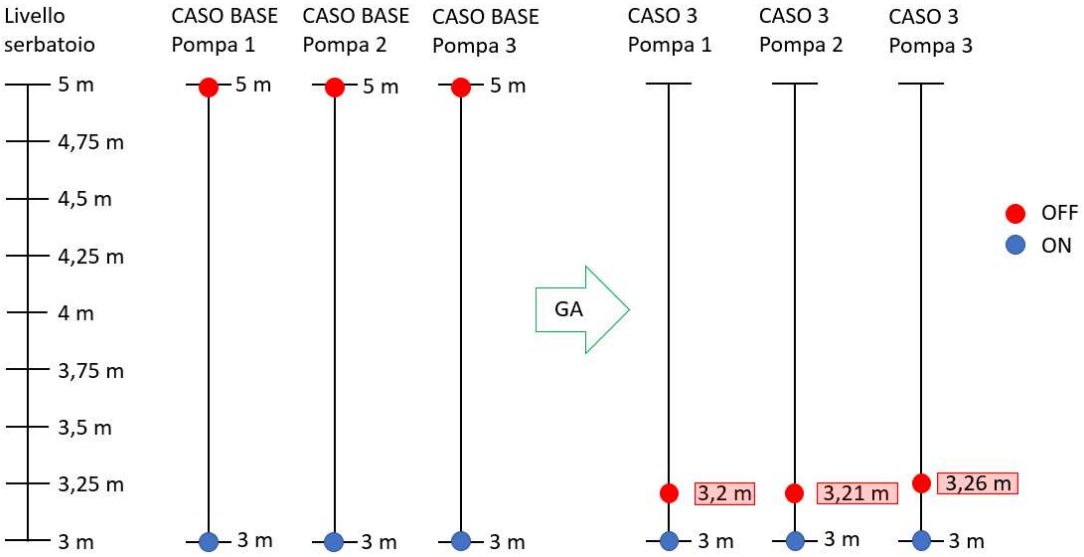


Figura 71. Confronto CASO BASE - CASO 3: 3 pompe

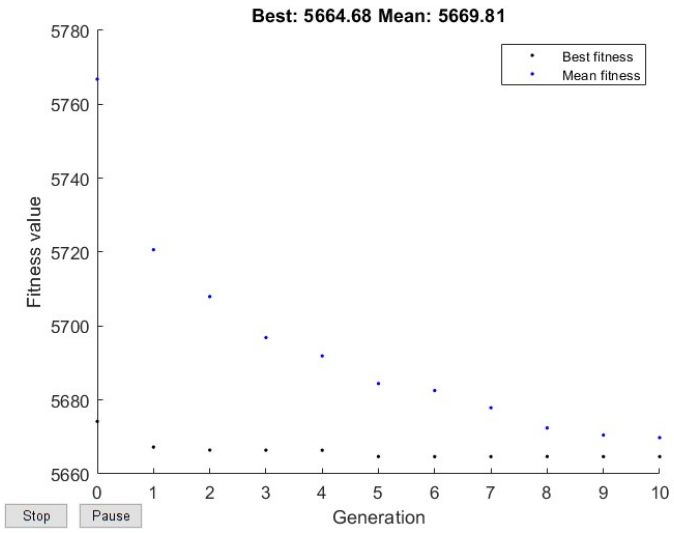
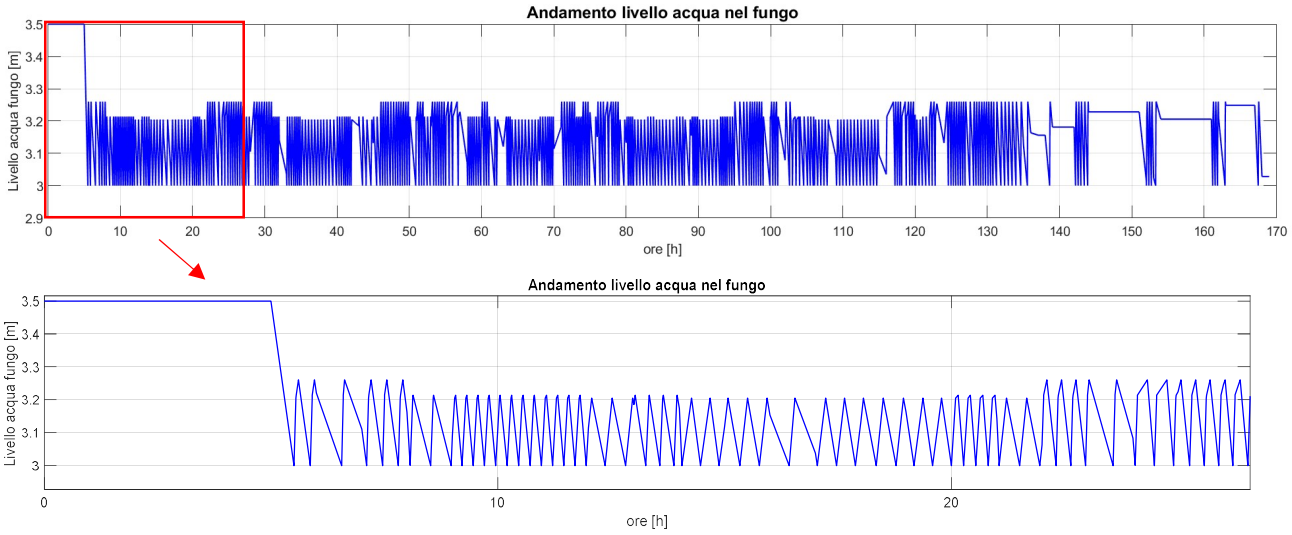


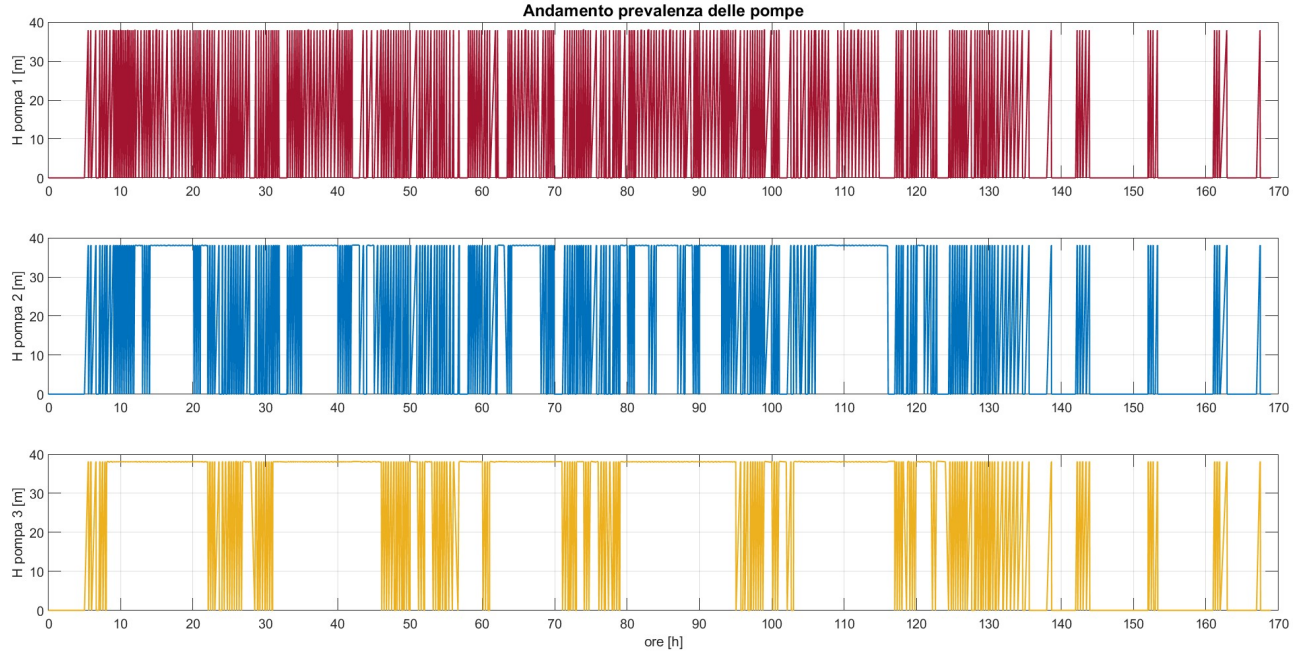
Figura 72. GA: caso 3 - 3 pompe

Nella Figura 72 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

a)



b)



c)

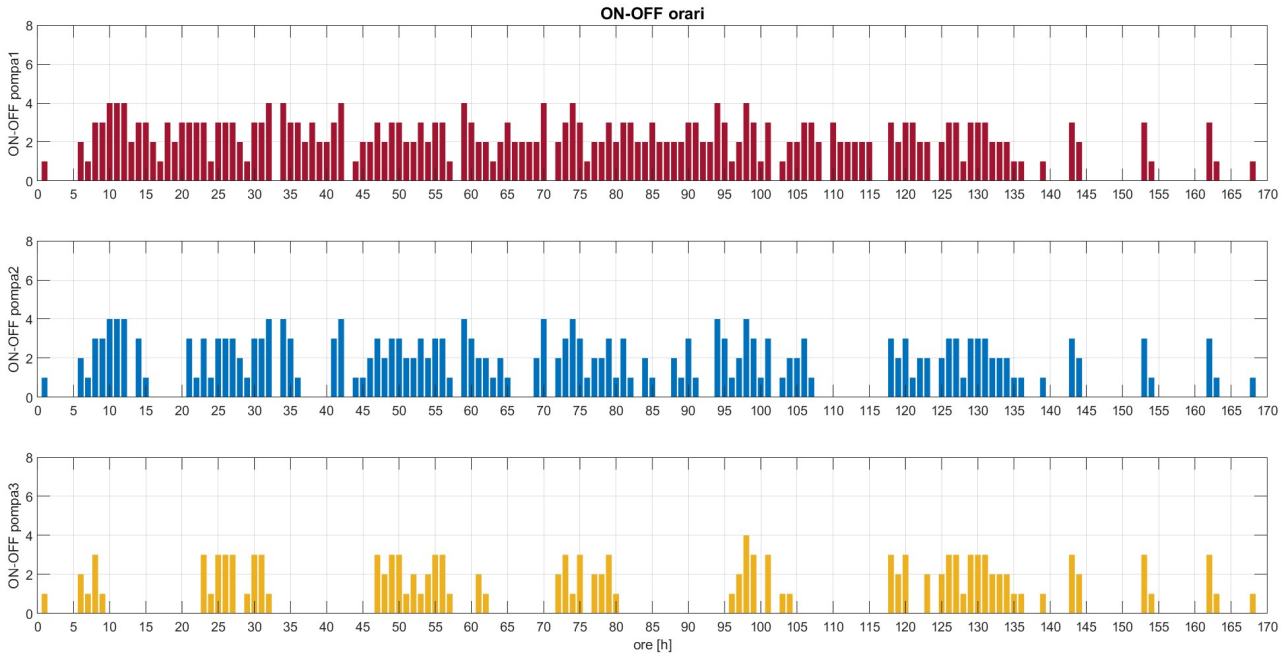


Figura 73. Output grafici: caso 3 - 3 pompe

I valori ottimali di spegnimento trovati per le 3 pompe portano il valore dell'acqua nel serbatoio a oscillare frequentemente tra 3 e 3,26m (Figura 73.a). Come si nota dalle Figura 73.b, la portata varia frequentemente tra 0 e il massimo, mentre la pompa1, che presenta il minimo valore di spegnimento trovato (3,2m), è quella che subisce il numero più alto di ON-OFF orari (Figura 73.c).

CASO 4: 3 POMPE

L'ON della pompa 1, 2 e 3 è lasciato libero di variare tra 3 e 4,8m. L'OFF della pompa 1, 2 e 3 è lasciato libero di variare tra 3,2 e 5m. L'algoritmo trova il valore ottimale di accensione delle 3 pompe rispettivamente a 3m, 3m e 3m, e valori di spegnimento pari rispettivamente a 3,4m 3,5m e 3,2m. Nella figura seguente sono evidenziati in rosso i parametri lasciati liberi di variare.

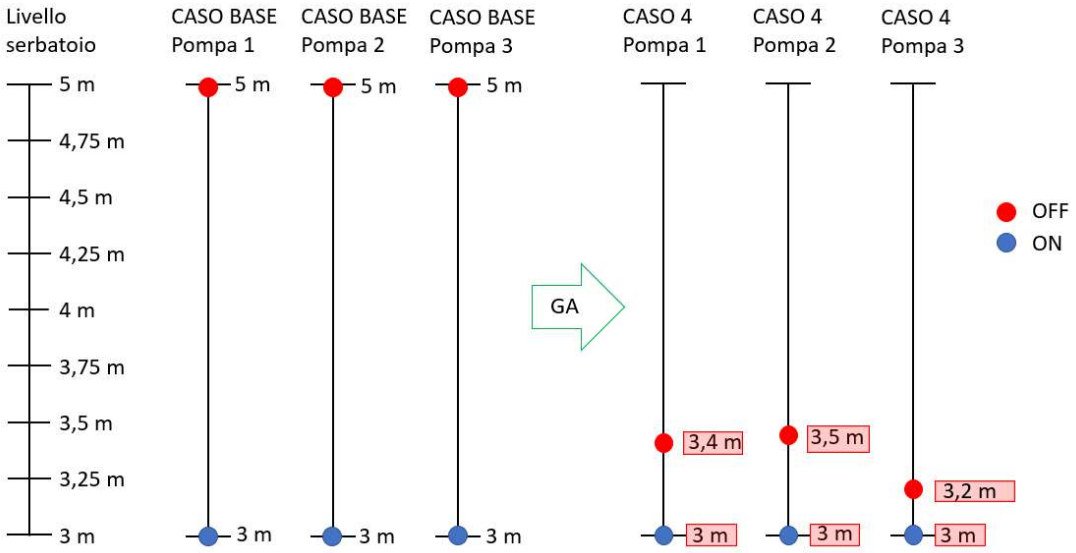


Figura 74. Confronto CASO BASE - CASO 4: 3 pompe

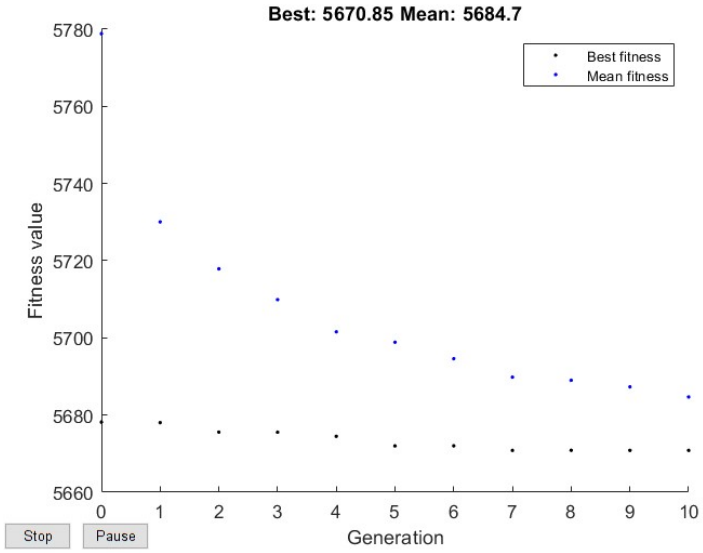
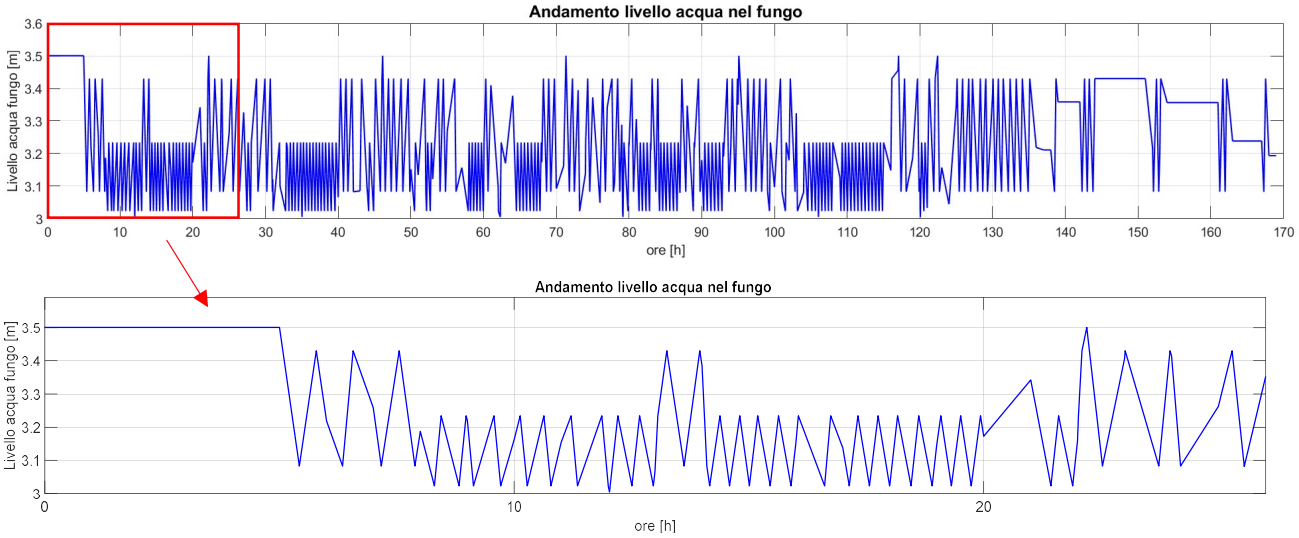


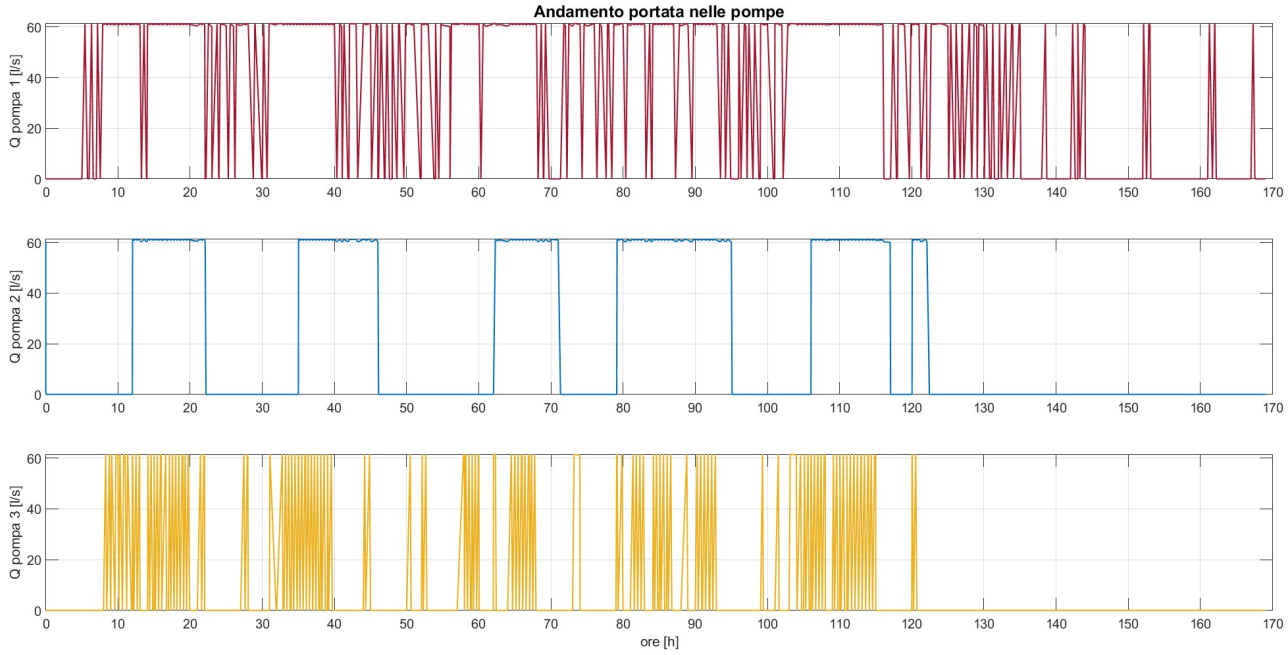
Figura 75. GA: caso 4 - 3 pompe

Nella Figura 75 è mostrato il grafico con l'andamento della fitness per ogni generazione: il pallino nero rappresenta il miglior valore trovato per quella generazione, mentre quello azzurro rappresenta la media dei valori trovati per quella generazione.

a)


Andamento livello acqua nel fungo
Livello acqua fungo [m]
ore [h]
Andamento livello acqua nel fungo
Livello acqua fungo [m]
ore [h]

b)


Andamento portata nelle pompe
Q pompa 1 [l/s]
Q pompa 2 [l/s]
Q pompa 3 [l/s]
ore [h]

c)

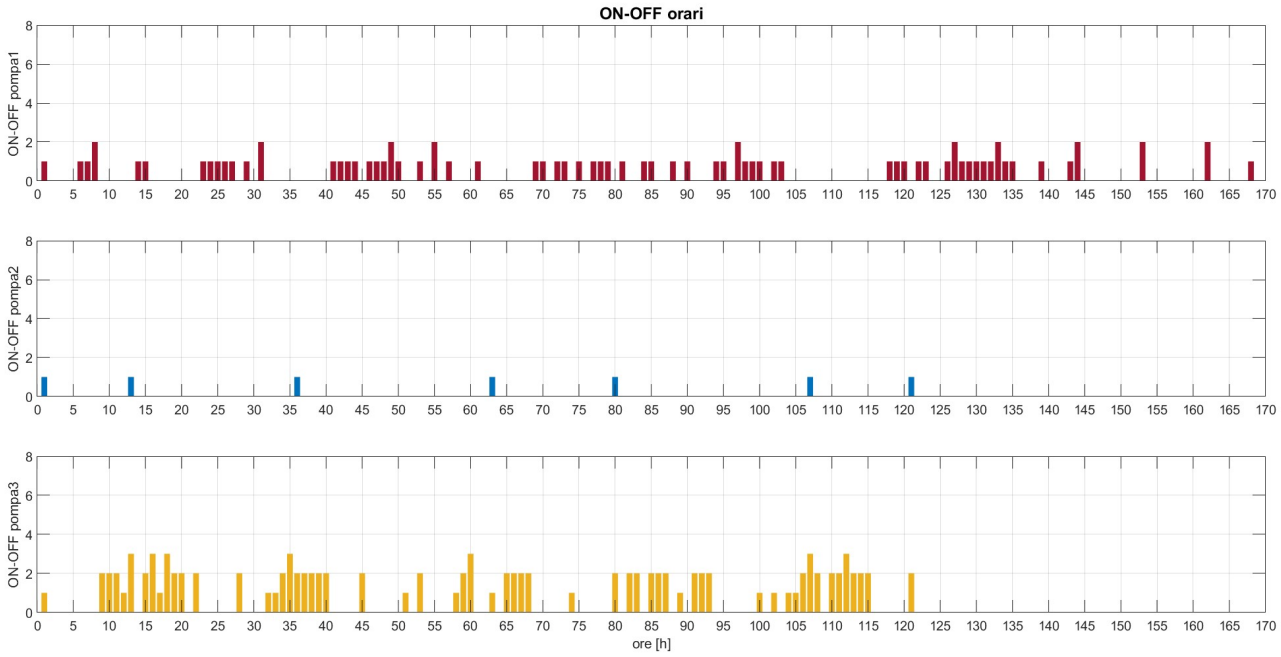


Figura 76. Output grafici: caso 4 - 3 pompe

In questo 4' e ultimo caso con 3 pompe, esattamente come è avvenuto nel CASO 2 – 1 POMPA, nonostante l'ON-OFF per tutte le pompe sia stato lasciato libero di variare in tutto il range del serbatoio, esso tende a trovare l'accensione e lo spegnimento in valori molto bassi: l'accensione, che può avvenire tra 3 e 4,8m, viene trovata per tutte le pompe pari a 3m; mentre lo spegnimento, che può variare tra 3,2 e 5m, viene trovato rispettivamente pari a 3,4, 3,5 e 3.2m. Questo consente al livello dell'acqua nel serbatoio di variare solo tra 3 e 3,6m (Figura 76.a). La pompa che subisce più ON-OFF orari è invece la 3, poiché presenta anche il minimo valore di OFF pari a 3,2m.

CONFRONTO CASI: 3 POMPE

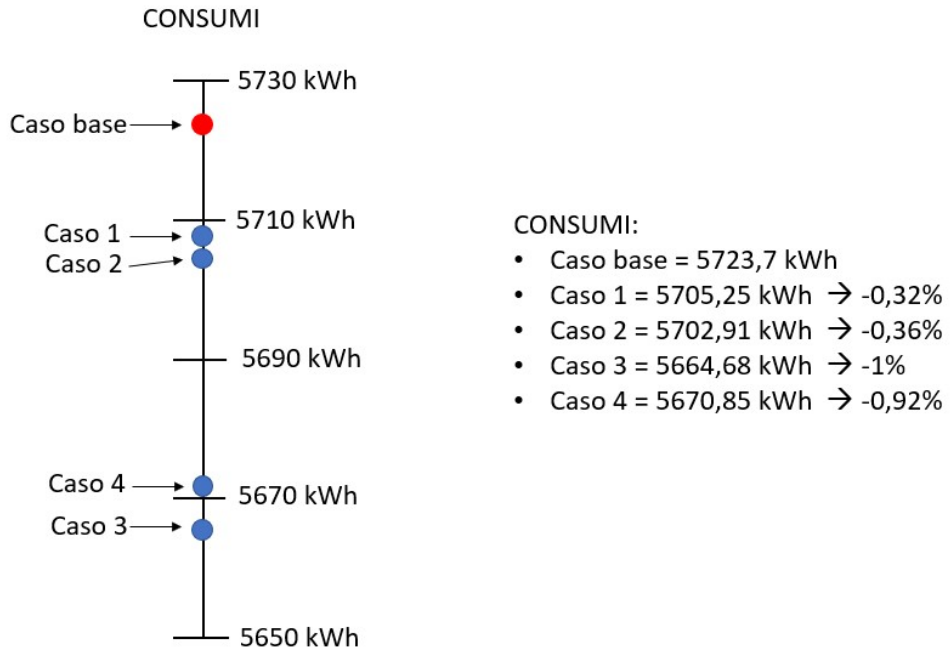


Figura 77. Riepilogo confronti GA - 3 pompe

Dall'analisi dei risultati si evince che gli ON e OFF ottimi trovati dall'algoritmo scendono sempre fino quasi al minimo possibile, soluzione che permette al sistema di consumare di meno tenendo le pompe accese meno a lungo. I consumi minori si hanno nel CASO 3 e nel CASO4, che sono anche quelli in cui più variabili sono libere di cambiare.

Si rimanda alla Tabella 2 per il riepilogo dei risultati trovati.

5.3 Sviluppo del modello surrogato tramite reti neurali

È stato creato un modello di reti neurali per la previsione dei consumi di una settimana tipo per due tipi di regolazione: ON-OFF e regolazione con valvole.
Per la realizzazione del modello si è proceduto per step:

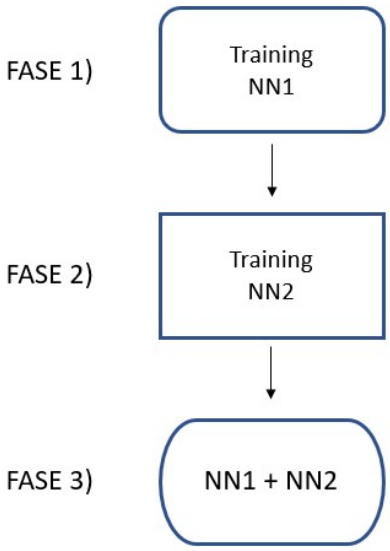


Figura 78. Flowchart modello NN

FASE 1) Allenamento della NN1

Tipologia rete: narxnet (MATLAB)

Input del training:

- Portata richiesta dall'utenza Q_u [l/s]
- Portata attraverso le pompe Q_1, Q_2, Q_3 [l/s]

Output del training:

- Livello acqua nel fungo z [m]

FASE 2) Allenamento della NN2

Tipologia rete: fitnet (MATLAB)

Input del training:

- Livello acqua nel fungo z [m]
- Setting delle pompe (ON-OFF: 0 se spenta,1 se accesa; valvole: k corrispondente)

Output del training:

- Portata attraverso le pompe Q_1, Q_2, Q_3 [l/s]

FASE 3) Unione di NN1 e NN2

Istante per istante viene eseguito il modello.

Tutti i dati utilizzati per l'allenamento delle due reti, sono risultati di simulazioni eseguite precedentemente tramite EPANET MATLAB Toolkit, che permette una manipolazione più facile dei dati, in quanto gli stessi risultano direttamente disponibili su MATLAB senza bisogno di passaggi intermedi dal software EPANET a MATLAB. Le reti sono state allenate con dati riferiti a 5 mesi, da settembre 2019 a gennaio 2020. Mentre la previsione è stata eseguita sulla prima settimana di febbraio.

ALLENAMENTO NN1

Per l'allenamento della NN1 ci si è serviti della rete neurale NARXNET presente su MATLAB.
I vari passaggi per l'allenamento della NN1 (Figura 79) sono spiegati di seguito:

1) vengono caricati su MATLAB i dati di input-output su cui la rete si deve allenare;
2) vengono convertiti i dati in una forma più facilmente utilizzabile dalla rete neurale;
3) viene scelta la funzione di training, in questo caso la funzione di Levenberg-Marquardt;
4) vengono impostati gli 'InputDelays' e i 'FeedbackDelays', in questa trattazione 5, e i numero di strati nascosti con i neuroni presenti in ogni strato. In questa trattazione sono stati impostati 3 strati nascosti da 10 neuroni ognuno;
5) viene impostata la rete neurale e i parametri vengono divisi tra la fase di training (70%, la fase di valutazione (15%) e la fase di testing (15%);
6) la rete viene allenata;
7) vengono caricati dei dati in input per fare una prova sulla bontà del modello;
8) la rete viene salvata;
9) viene usata la rete neurale appena allenata per avere degli output a partire da degli input che la rete non ha mai visto.
10) vengono stampati a video e graficati i dati con le evidenze della previsione.

Per ulteriori dettagli su tutti i passaggi riguardanti la rete NARXNET si rimanda al capitolo 3.2.

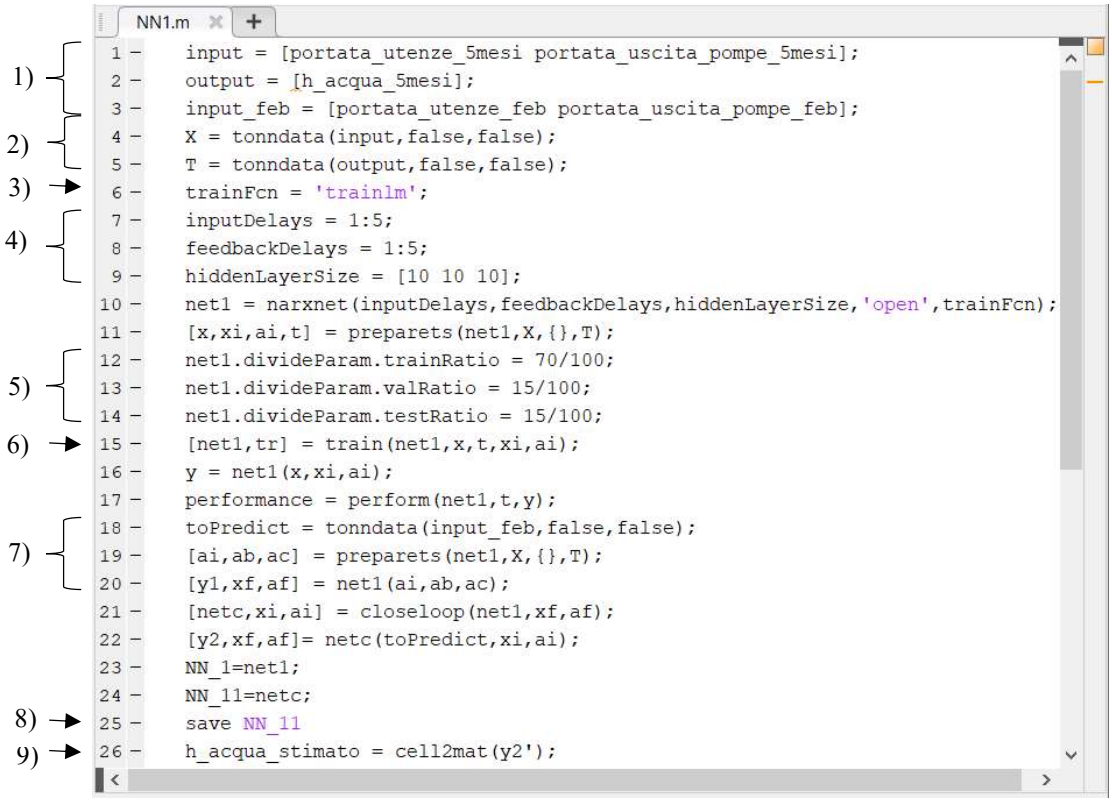


Figura 79. Parte di script allenamento NN1

ALLENAMENTO NN2

Per l'allenamento della NN2 ci si è serviti della rete neurale FITNET presente su MATLAB.

I vari passaggi per l'allenamento della NN2 (Figura 80) sono spiegati di seguito:

1) vengono caricati su MATLAB i dati di input-output su cui la rete si deve allenare;
2) viene scelta la funzione di training, in questo caso la funzione di Levenberg-Marquardt;
3) viene impostato i numero di strati nascosti con i neuroni presenti in ogni strato. In questa trattazione sono stati impostati 3 strati nascosti da 10 neuroni ognuno;
4) viene impostata la rete neurale e i parametri vengono divisi tra la fase di training (70%, la fase di valutazione (15%) e la fase di testing (15%);
5) la rete viene allenata;
6) la rete viene salvata;
7) viene usata la rete neurale appena allenata per avere degli output a partire da degli input che la rete non ha mai visto.
8) vengono stampati a video e graficati i dati con le evidenze della previsione.

Per ulteriori dettagli su tutti i passaggi riguardanti la rete NARXNET si rimanda al capitolo 3.2.

NN2.m

```
1)  input = [h_acqua_5mesi setting_5mesi];
    output = [portata_uscita_pompe_5mesi];
    input_feb = [h_acqua_feb setting_feb];
    x = input';
2)  t = output';
3)  trainFcn = 'trainlm';
    hiddenLayerSize = [10 10 10];
    net2 = fitnet(hiddenLayerSize,trainFcn);
4)  net2.divideParam.trainRatio = 70/100;
    net2.divideParam.valRatio = 15/100;
    net2.divideParam.testRatio = 15/100;
5)  [net2,tr] = train(net2,x,t);
6)  NN_2=net2;
7)  save NN_2
    output_stimato = net2(input_feb');
    output_stimato=output_stimato';
    portate_stimate=output_stimato(:,1:3);
    Cyt1=corrcoef(portate_stimate,portata_uscita_pompe_feb);
    R1=Cyt1(2,1);
8)  R1_2=R1^2;
    figure(1)
    plotregression(portata_uscita_pompe_feb,portate_stimate,'portate attraverso le pompe');
```

Figura 80. Parte di script allenamento NN2

UNIONE di NN1 e NN2

Il modello di reti neurali utilizzato per la previsione può essere schematizzato come segue, con:

- Q_u = portata richiesta dall'utenza [l/s]
- Q_1, Q_2, Q_3 = portata attraverso le pompe 1, 2 e 3 [l/s]
- z = livello acqua nel fungo [m]
- [0, 0, 0] = modellino setting pompe f(z)

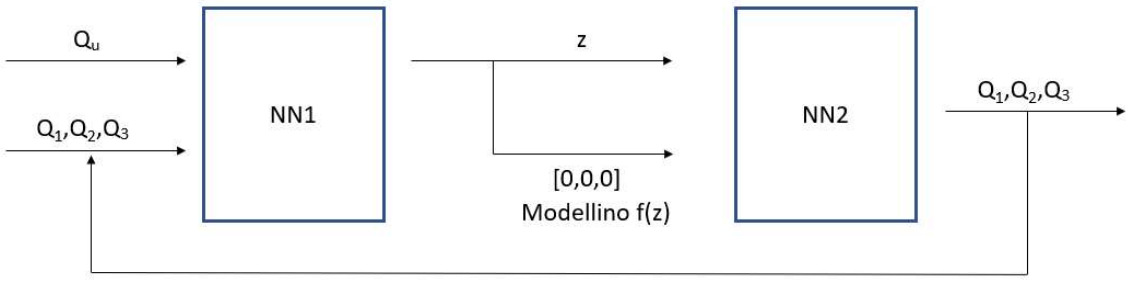


Figura 81. Modello NN: regolazione ON-OFF

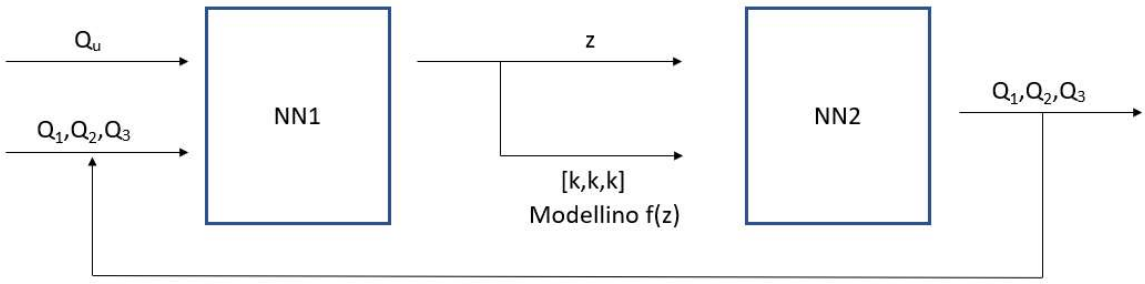


Figura 82. Modello NN: regolazione con valvole

Come si nota nelle figure precedenti, l'output della NN2 nell'istante t fa da input alla NN1 nell'istante successivo t+1.
Nell'unione delle due reti neurali, l'unico input esterno al modello risulta essere la richiesta dell'acqua da parte dell'utenza, mentre tutti i restanti dati vengono previsti istante per istante dal modello stesso.

Il modellino centrale f(z) permette:

- Nella regolazione ON-OFF di determinare se le pompe sono accese (1) o spente (0) in base al livello dell'acqua presente nel serbatoio;
- Nella regolazione con valvole di determinare il coefficiente di perdita k della valvola corrispondente al livello dell'acqua presente nel serbatoio.

Per verificare l'accuratezza di previsione del modello si è deciso di utilizzare il coefficiente di determinazione R^2, un indicatore statistico che indica quanto il modello «fitta» i dati. Esso misura il legame tra la variabilità dei dati e la correttezza del modello statistico utilizzato. La definizione di coefficiente di determinazione è la seguente:

$$R^2 = 1 - \frac{RSS}{TSS}$$

dove:

- $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ è la devianza totale;
- $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ è la devianza residua;
- y_i sono i dati osservati;
- $\bar{y}$ è la loro media;
- $\hat{y}_i$ sono i dati stimati dal modello.

R2 varia tra - e 1: quando è 0 il modello utilizzato offre una spiegazione dei dati non migliore del valore medio (RSS=TSS); quando è 1 il modello spiega perfettamente i dati. Un modello peggiore della media (RSS>TSS) ha coefficiente $R^2 < 0$ [19].
L'R^2 rappresenta un indicatore, che partendo dalla retta di regressione, sintetizza in un unico valore di quanto la grandezza analizzata si discosta da tale retta, indica quanto la regressione sia un buon modello per analizzare i dati.

Sia per la regolazione ON-OFF che per quella con valvole sono state allenate le reti considerando diversi intervalli temporali: 1 ora, 10 minuti e 1 minuto. Questo perché utilizzando l'intervallo temporale di 1 ora ci si è resi conto che la simulazione non era abbastanza precisa; si è quindi proceduto diminuendo progressivamente gli intervalli stessi.

REGOLAZIONE ON-OFF

La regolazione degli ON-OFF impostata per le 3 pompe è quella mostrata nella figura successiva.

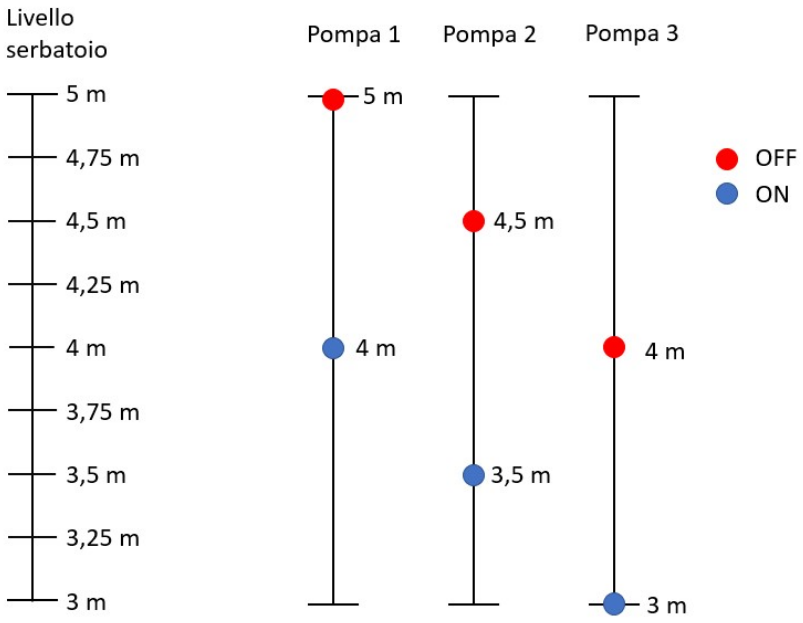


Figura 83. Regolazione ON-OFF pompe

Le tempistiche per l'esecuzione del modello e la precisione con cui si riescono a prevedere le portate attraverso le pompe in una settimana tipo sono illustrate nella Tabella 3.

	t esecuzione	R^2 portate pompe
1 h	10 secondi	0,75
10 minuti	29 secondi	0,92
1 minuto	12 minuti	0,86

Tabella 3. Confronto modello NN ON-OFF

CASO ON-OFF: 1 ORA

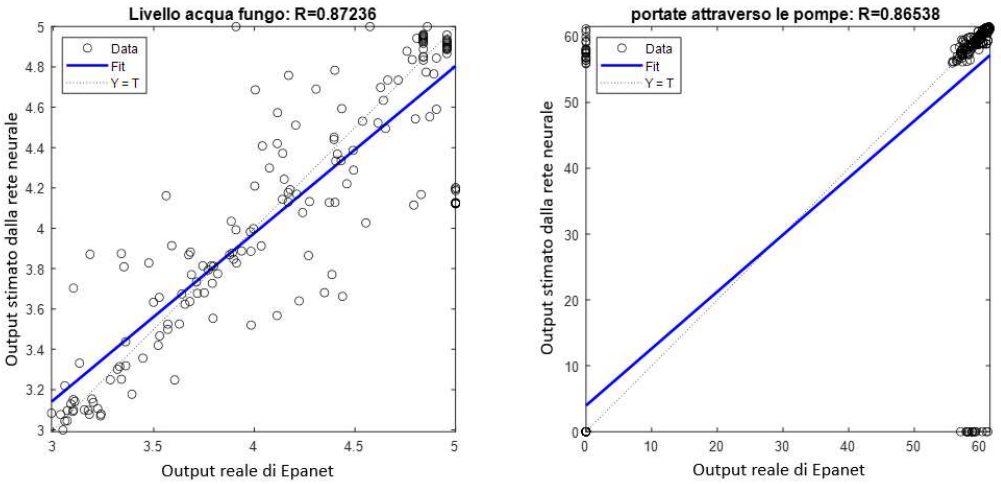


Figura 84. Regressione risultati ON-OFF 1 ora

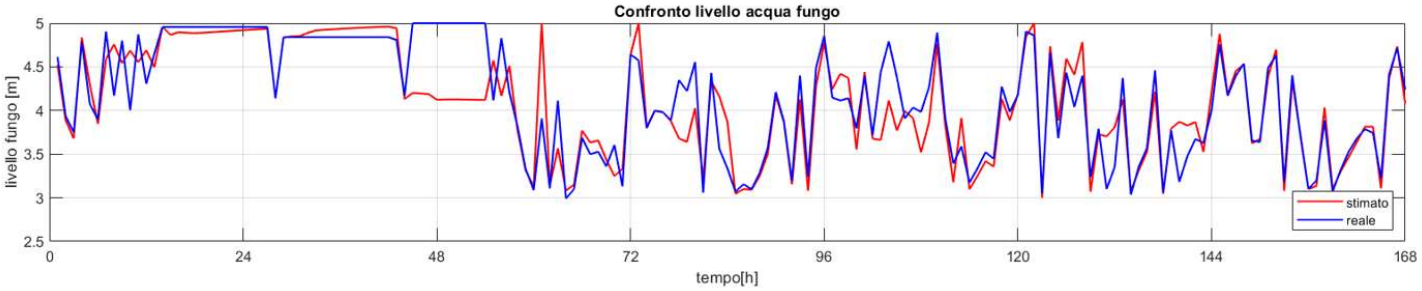


Figura 85. Confronto livello acqua fungo ON-OFF 1 ora

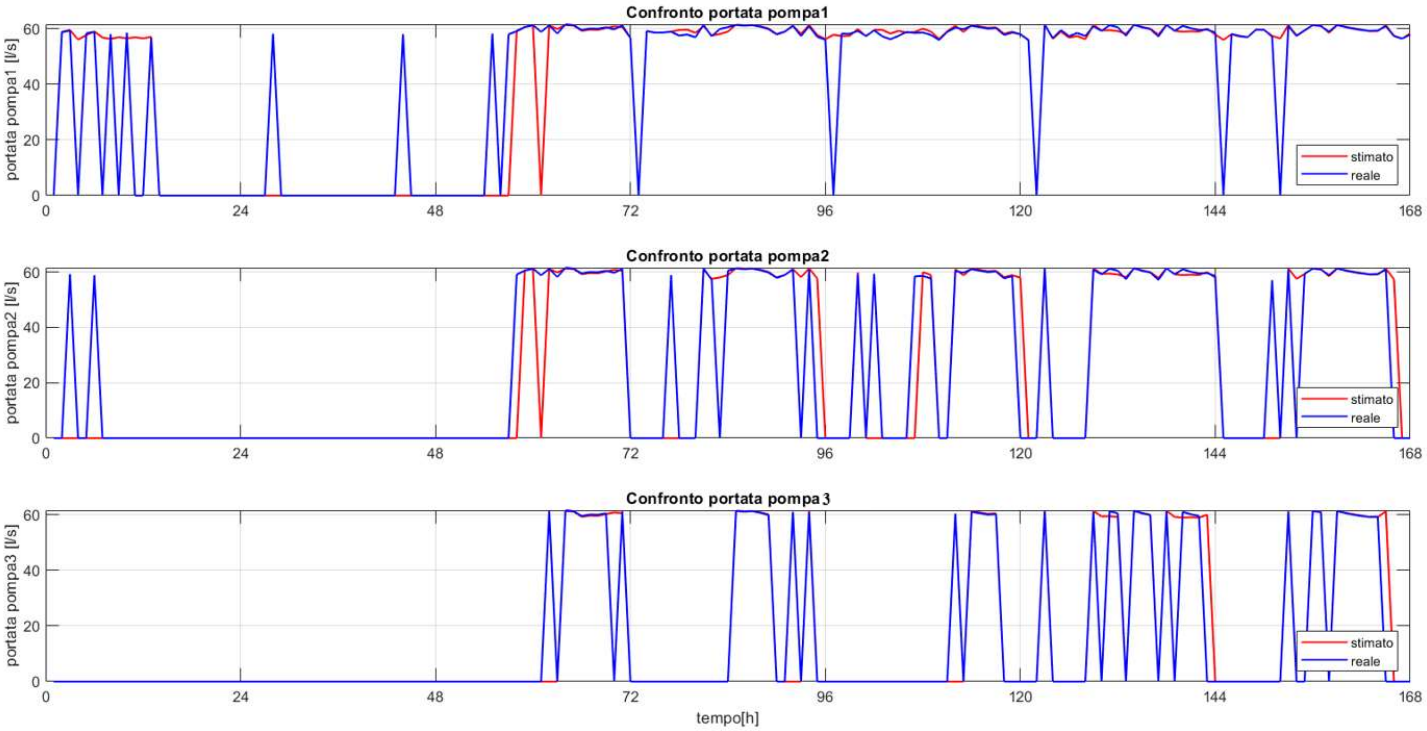


Figura 86. Confronto portate pompe ON-OFF 1 ora

Come si nota dalle figure precedenti i risultati ottenuti non sono precisi; si è quindi proceduto cambiando l'intervallo temporale di riferimento a 10 minuti.

CASO ON-OFF: 10 MINUTI

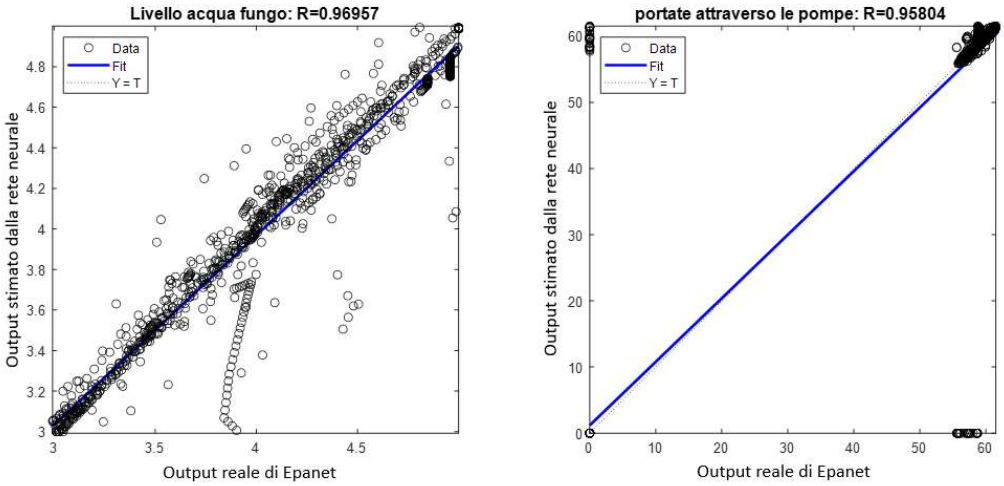


Figura 87. Regressione risultati ON-OFF 10 minuti

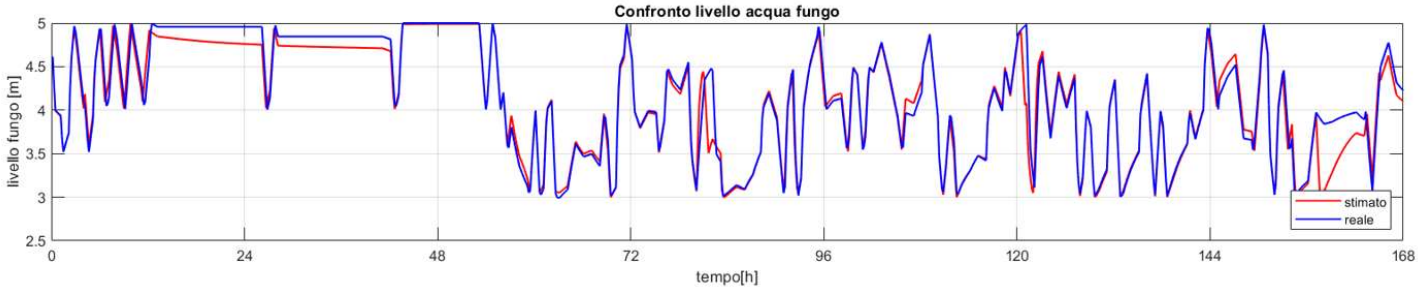


Figura 88. Confronto livello acqua fungo ON-OFF 10 minuti

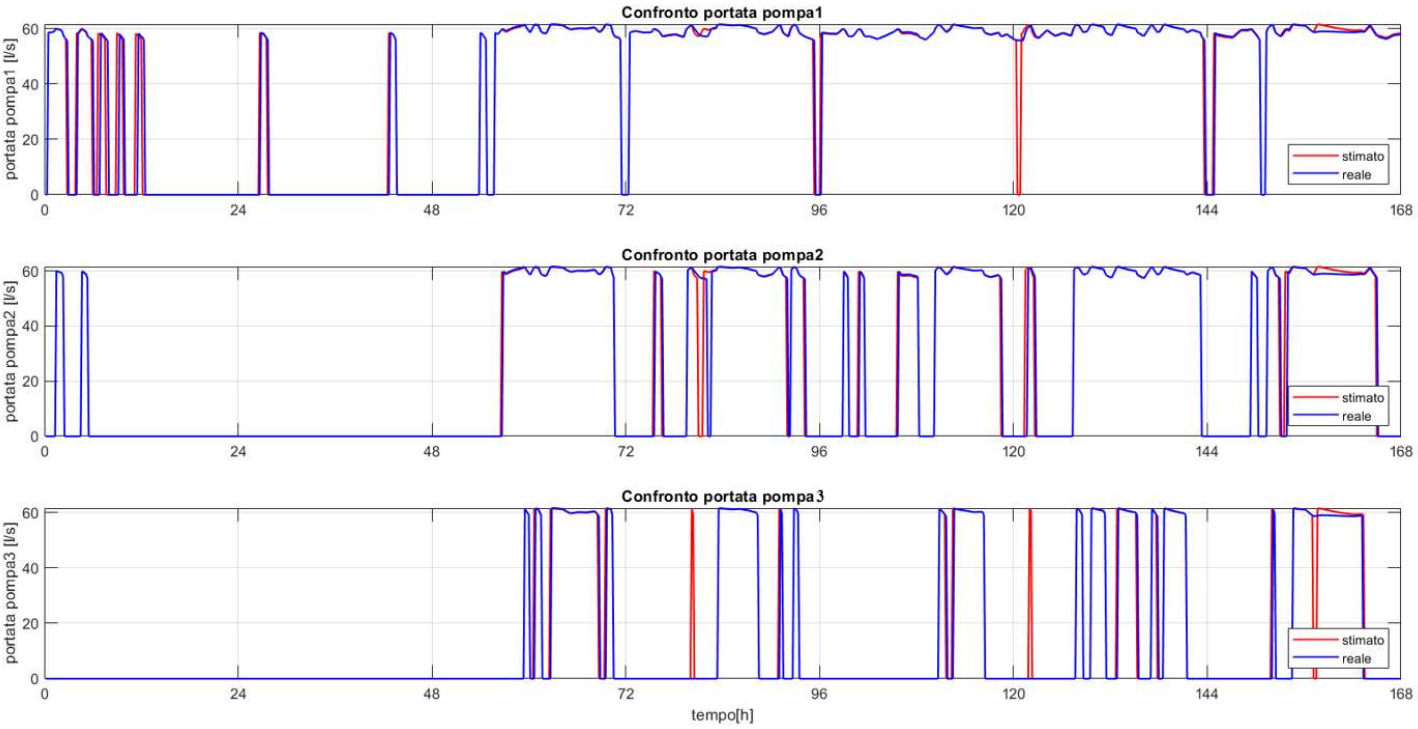


Figura 89. Confronto portate pompe ON-OFF 10 minuti

In questo caso sia il livello dell'acqua nel fungo che le portate d'acqua nelle pompe riescono ad essere previsti in maniera abbastanza precisa dal modello, come si vede dalle figure precedenti.

CASO ON-OFF: 1 MINUTO

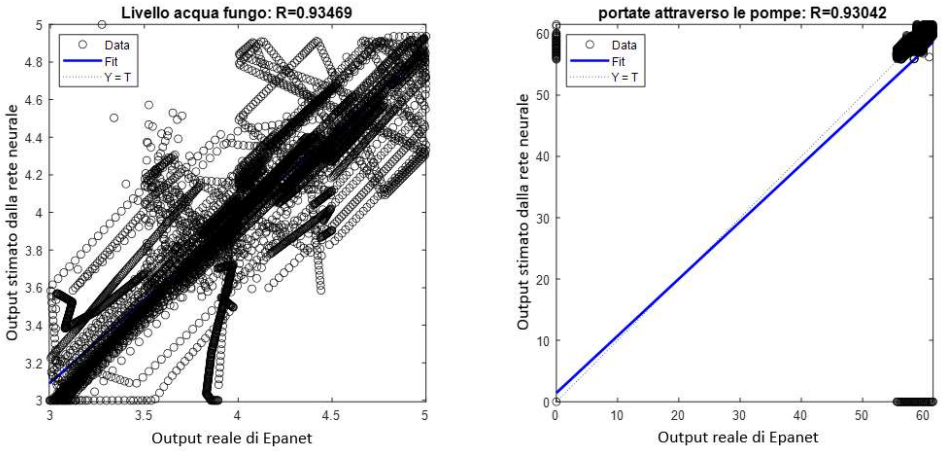


Figura 90. Regressione risultati ON-OFF 1 minuto

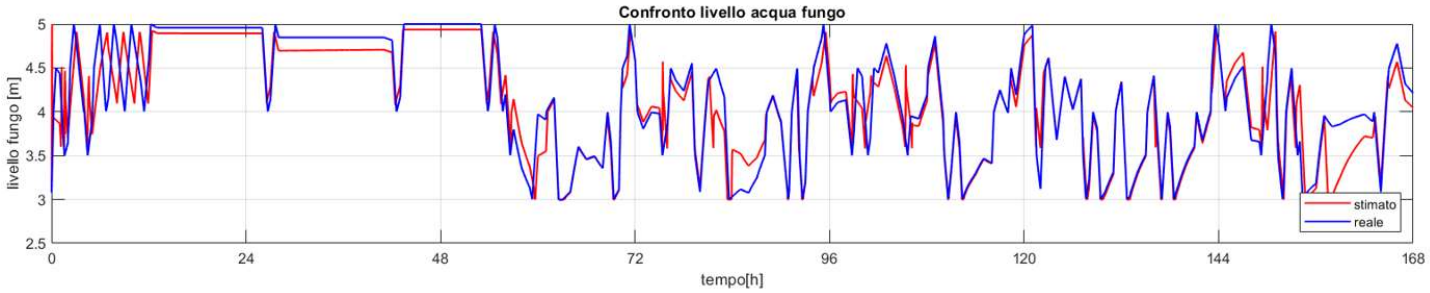


Figura 91. Confronto livello acqua fungo ON-OFF 1 minuto

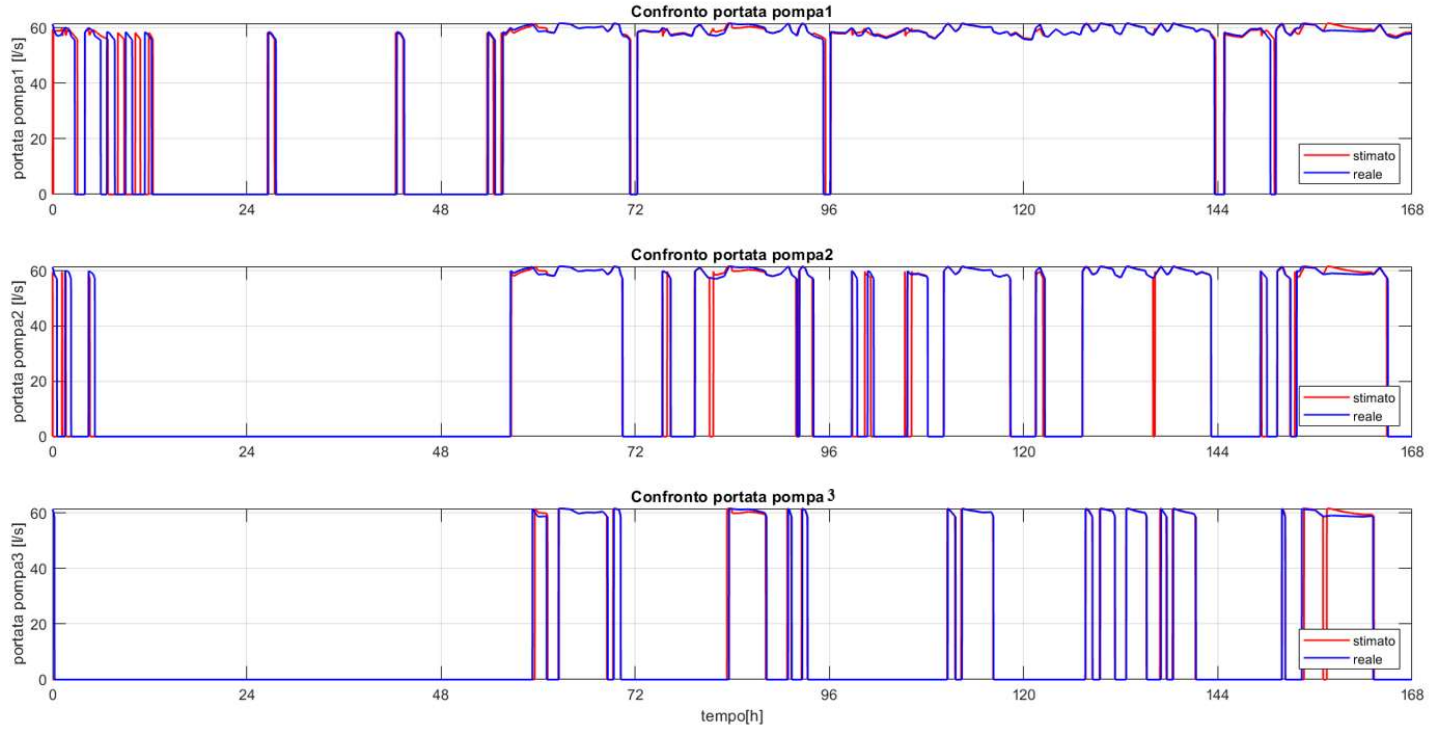


Figura 92. Confronto portate pompe ON-OFF 1 minuto

Discretizzando ulteriormente il tempo, fino ad arrivare ad 1 minuto, si nota che la previsione è migliore rispetto al modello orario, ma peggiora leggermente rispetto al modello con dati ogni 10 minuti. Questo perché peggiora sensibilmente la previsione del livello dell'acqua nel serbatoio; perciò l'errore della prima rete neurale si trascina anche nella seconda, essendo gli output della prima, gli input della seconda.

REGOLAZIONE VALVOLE

La regolazione delle valvole impostata per le 3 pompe è quella mostrata nella figura successiva.

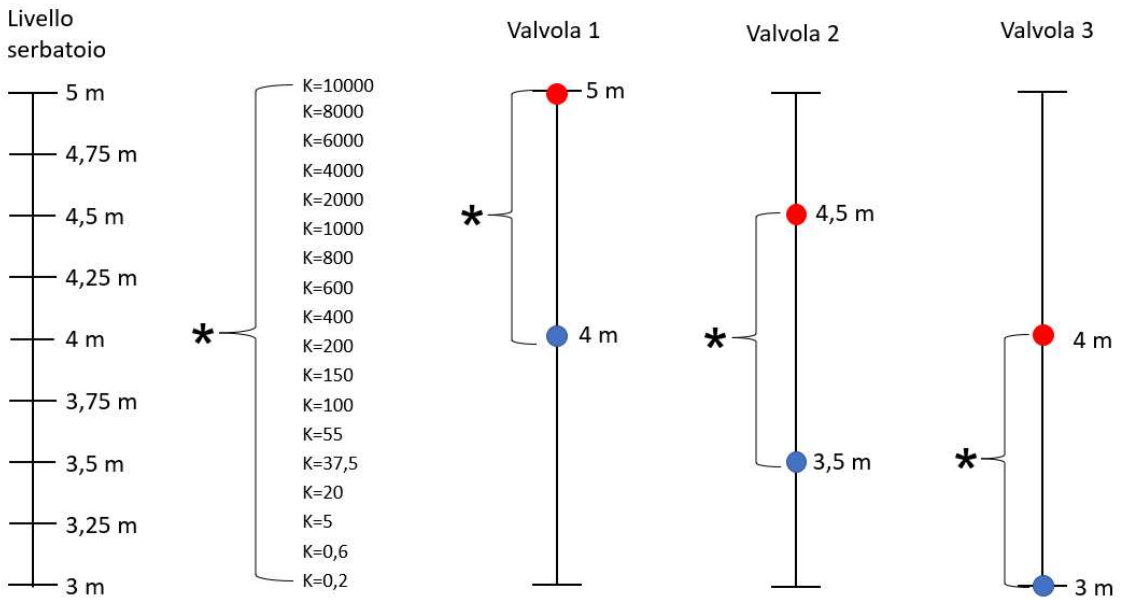


Figura 93. Regolazione Valvole

Il k rappresentato in figura rappresenta il coefficiente di perdita della valvola, necessario per il calcolo della perdita di carico H_k attraverso la valvola stessa, calcolato come segue:

$$H_k = k \cdot \frac{v^2}{2 \cdot g}$$

dove:

- H_k = perdita di carico in m
- v = velocità in m/s
- g = 9,81 m/s^2

Ognuna delle 3 valvole presenti nel sistema, è caratterizzata da intervalli di regolazione diversi, come mostrato in figura 79. All'interno di ogni intervallo i k sono stati distribuiti in modo uniforme e lineare.

Le tempistiche per l'esecuzione del modello sono illustrate nella tabella successiva.

	t esecuzione	R^2 portate pompe
1 h	8 secondi	0,60
10 minuti	27 secondi	0,86
1 minuto	12 minuti	0,99

Tabella 4. Confronto modello NN Valvole

CASO VALVOLE: 1 ORA

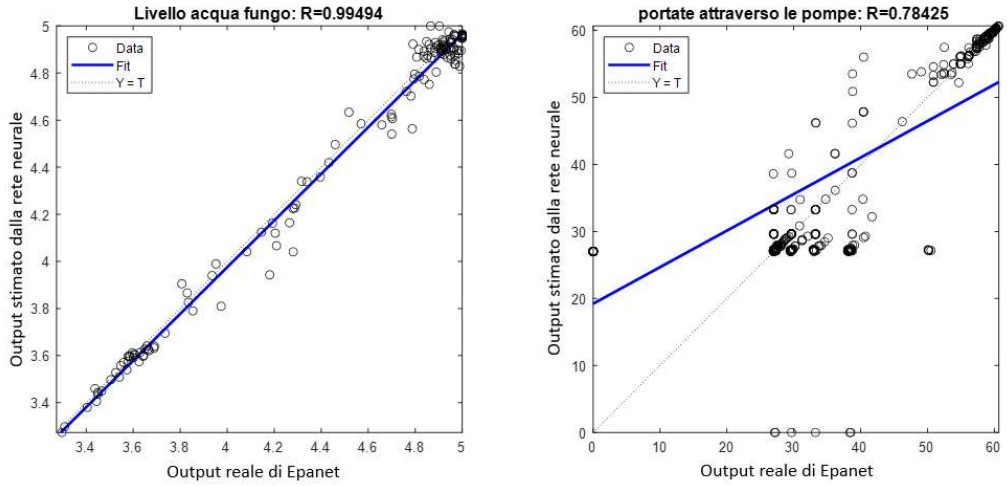


Figura 94. Regressione risultati Valvole 1 ora

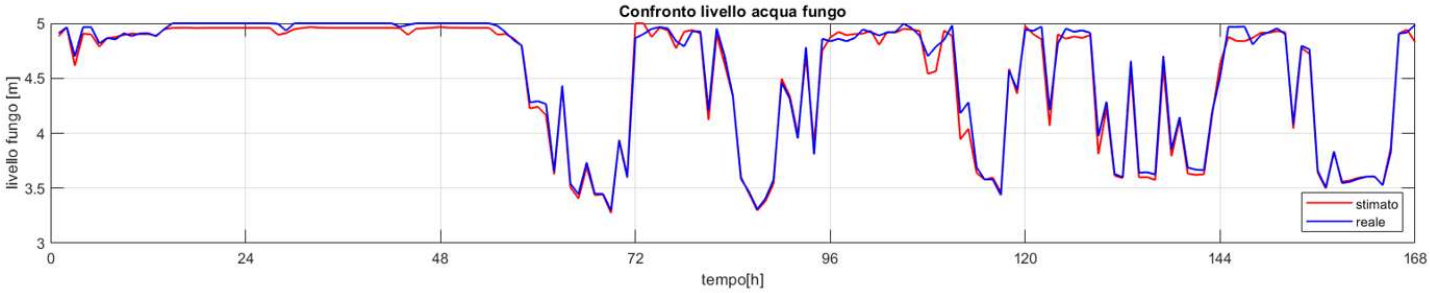


Figura 95. Confronto livello acqua fungo Valvole 1 ora

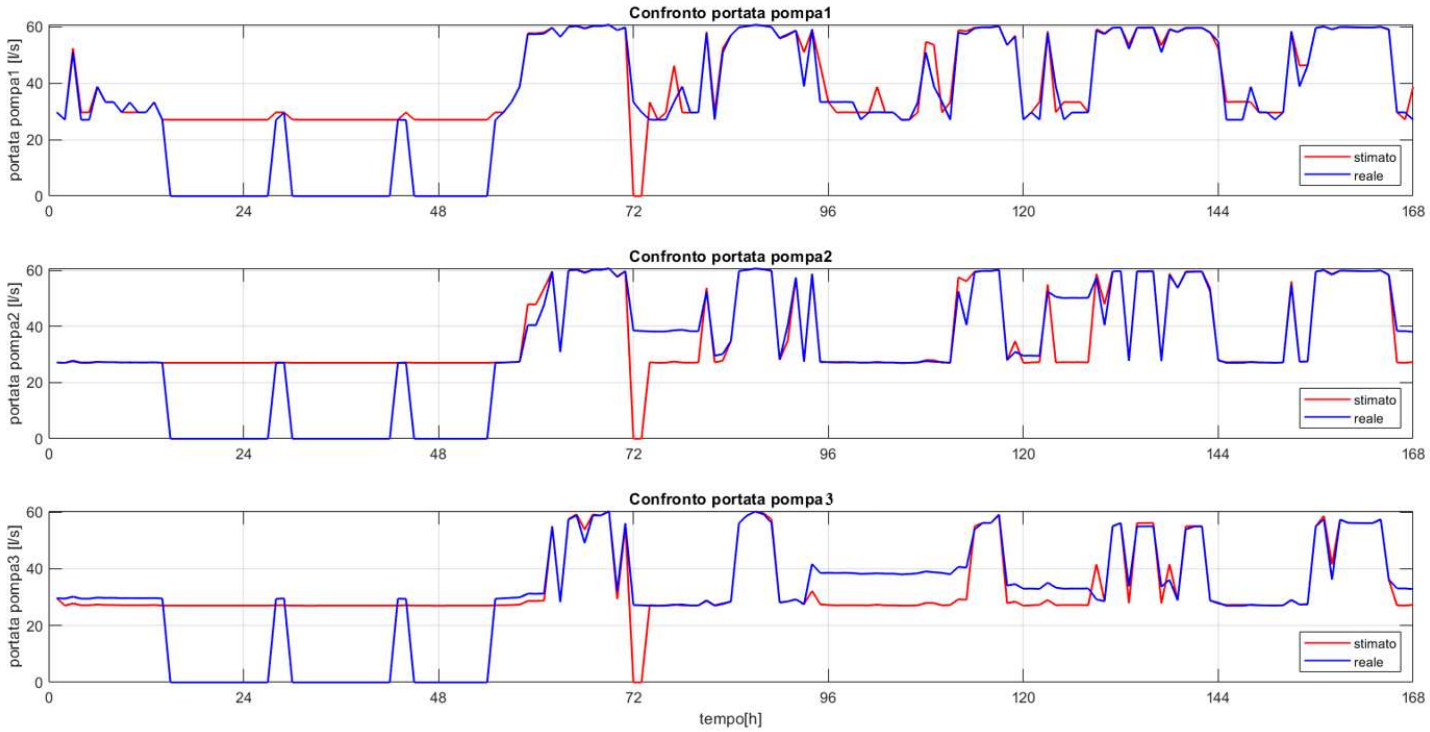


Figura 96. Confronto portate pompe Valvole 1 ora

Nonostante la previsione del livello dell'acqua nel serbatoio sia abbastanza precisa, non basta affinché anche la portata di acqua attraverso le pompe lo sia. Nel caso delle valvole infatti gli intervalli del k sono talmente ristretti, circa uno ogni 5 cm, che basta che l'acqua nel serbatoio sia prevista non perfettamente che il k della valvola viene settato sbagliato nel modellino intermedio tra le due reti neurali e di conseguenza anche l'acqua attraverso le pompe non viene prevista bene. Per cercare di migliorare le performance si è diminuito l'intervallo temporale passando da 1 ora a 10 minuti.

CASO VALVOLE: 10 MINUTI

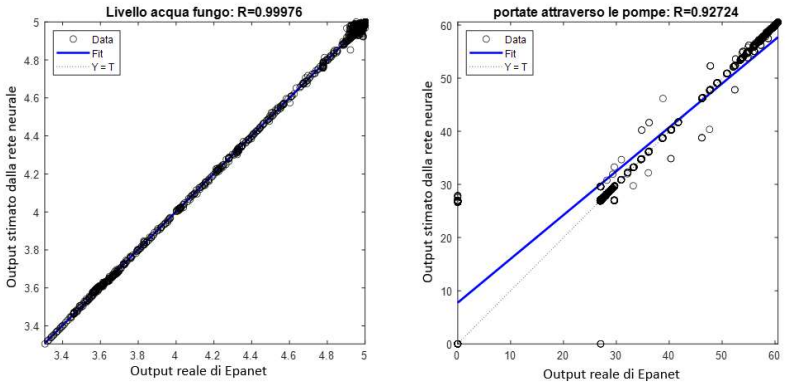


Figura 97. Regressione risultati Valvole 10 minuti

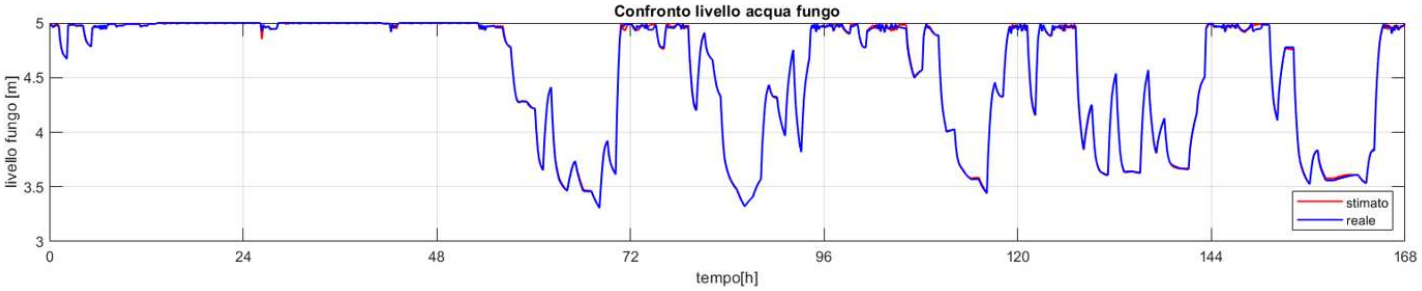


Figura 98. Confronto livello acqua fungo Valvole 10 minuti

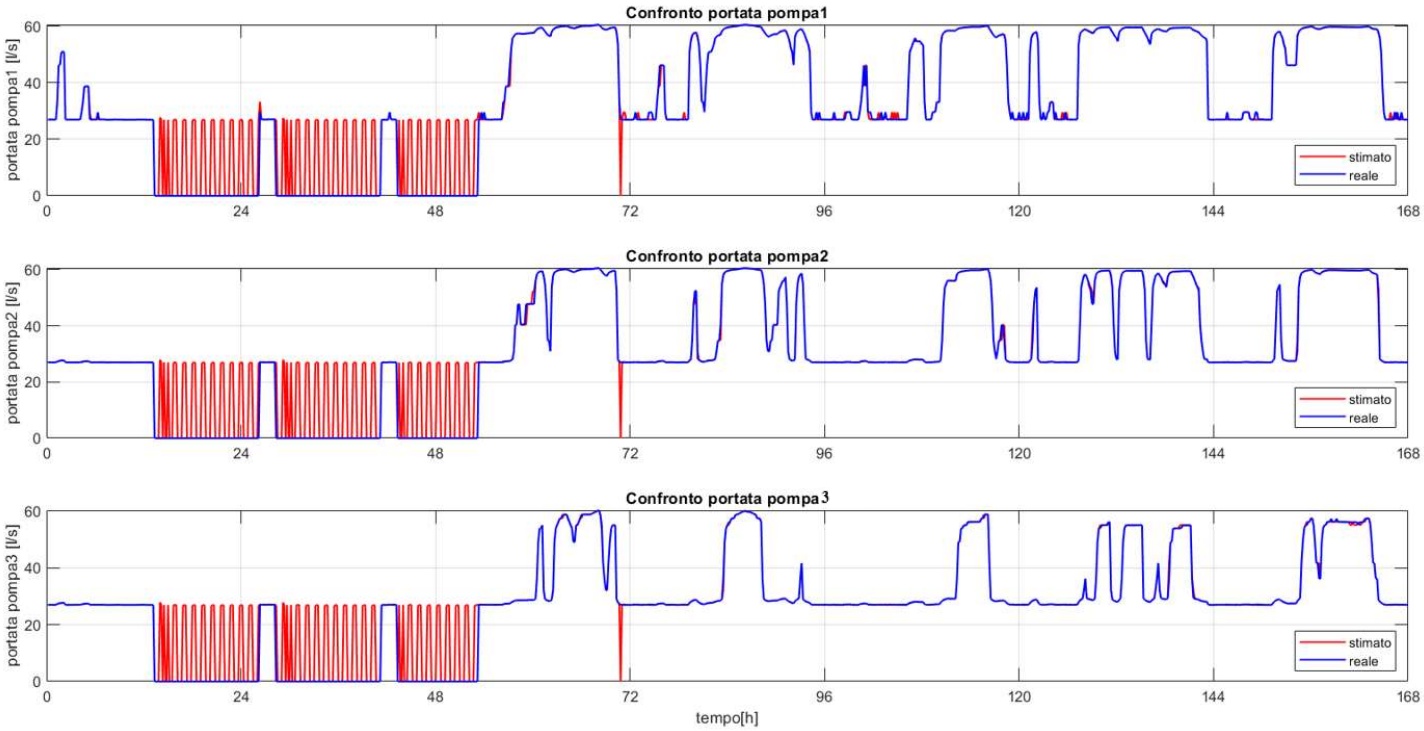


Figura 99. Confronto portate pompe Valvole 10 minuti

CASO VALVOLE: 1 MINUTO

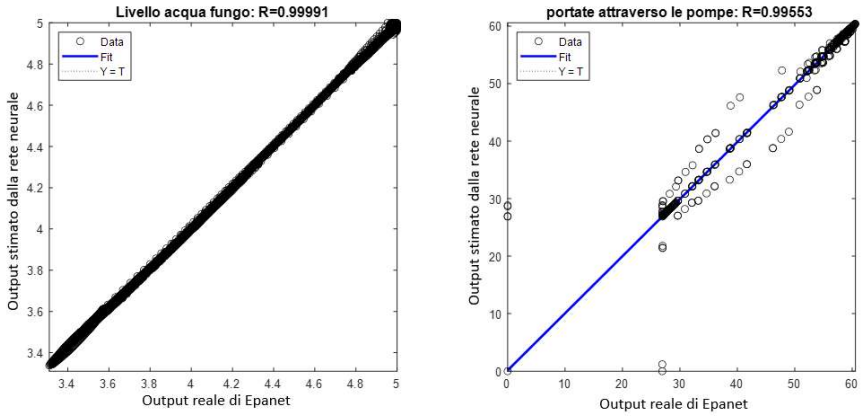


Figura 100. Regressione risultati Valvole 1 minuto

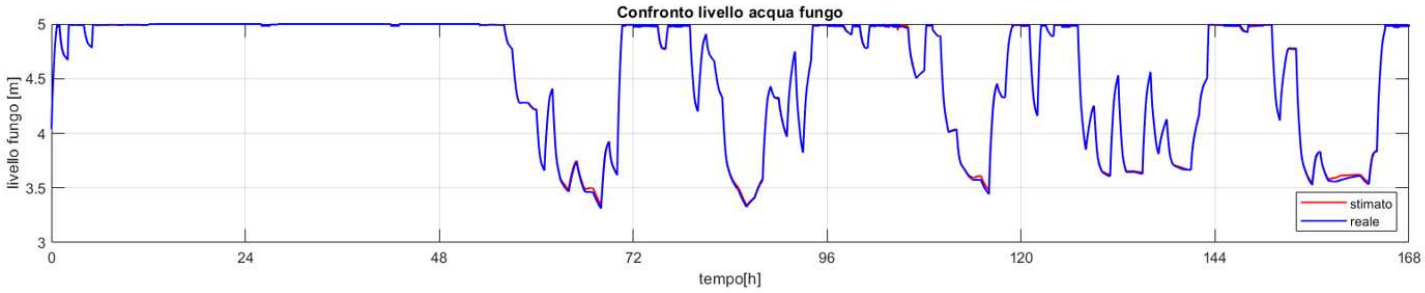


Figura 101. Confronto livello acqua fungo Valvole 1 minuto

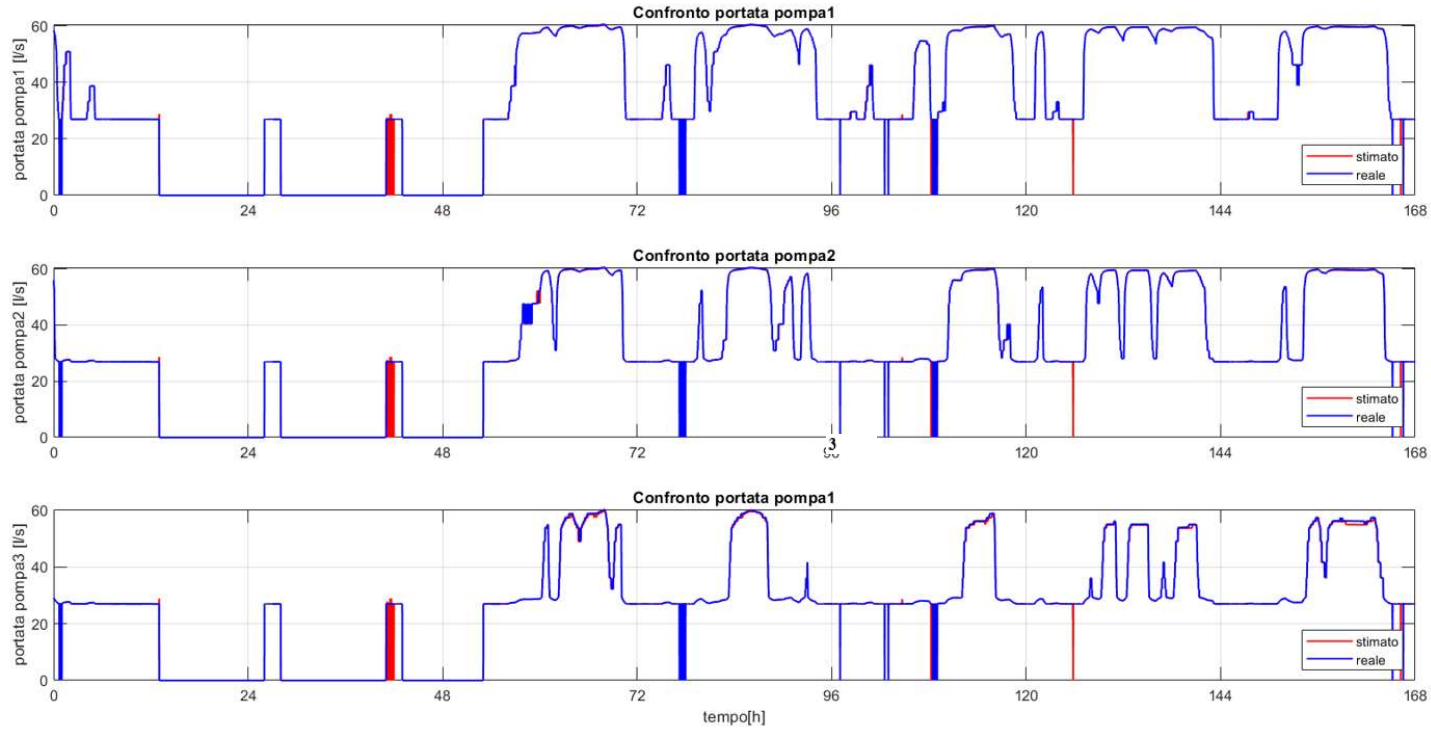


Figura 102. Confronto portate pompe Valvole 1 minuto

In questo ultimo caso il livello dell'acqua nel serbatoio viene previsto molto bene, e di conseguenza anche la portata di acqua attraverso le pompe segue quasi perfettamente l'andamento reale, come si può notare dalla Figura 101 e Figura 102.

5.4 Ottimizzazione tramite algoritmi genetici del modello surrogato

Il fine ultimo del lavoro, tentare di ottimizzare il modello creato con le reti neurali tramite l'utilizzo degli algoritmi genetici, è stato perseguito lasciando inalterata la struttura dell'algoritmo genetico. Esso sarà sempre costituito da due script: quello principale che rimane del tutto inalterato, e quello formato dalla funzione di fitness che invece subirà delle modifiche.

Nel paragrafo 4.2 si è visto come per trovare la regolazione ottima che permetta di minimizzare i consumi, si è proceduto nello script della funzione di fitness, all'utilizzo dell'EPANET MATLAB Toolkit. Grazie a questo strumento è stato possibile cambiare la regolazione delle pompe, impostando come limiti del livello dell'acqua nel serbatoio per l'accensione e lo spegnimento delle pompe, quelli trovati dall'algoritmo genetico, e calcolare infine così i consumi.
Per procedere con l'utilizzo dell'algoritmo genetico anche con le reti neurali si è proceduto con la sostituzione della parte appena descritta con il modello NN1+NN2 descritto nel paragrafo 4.3.
Nell'algoritmo genetico è stata impostata anche come opzione il numero di generazioni consecutive 'MaxStallGeneration', 2, oltre le quali fermare l'algoritmo nel caso in cui i consumi ottimali non varino da una generazione all'altra di più di 1 kWh (*FunctionTolerance*).

OTTIMIZZAZIONE ON-OFF

Per l'ottimizzazione della regolazione ON-OFF si è scelto di utilizzare l'intervallo temporale pari a 10 minuti, che tra i 3 simulati nel Paragrafo 4.3 (1 ora, 10 minuti, 1 minuto) è stato quello che ha dato risultati migliori (si rimanda alla Tabella 3).
Visto i tempi necessari a compiere la simulazione di una settimana, si è proceduto con la simulazione di un solo giorno, cosa che ha permesso di avere tempi non troppo lunghi impostando comunque un numero elevato di individui per ogni popolazione. Nella tabella di seguito vengono mostrate le opzioni impostate nell'algoritmo genetico di ottimizzazione nel caso ON-OFF, il tempo di simulazione necessario e il Δt di riferimento.

	Individui per popolazione	Generazioni	t necessario	Δt
Post - opt	100	5	1 h	10 minuti

Tabella 5. GA: ottimizzazione ON-OFF

Si mostra di seguito una tabella riassuntiva con i risultati ottenuti.

	REGOLAZIONE						Consumi totali [kWh]
	ON Pompa1	OFF Pompa1	ON Pompa2	OFF Pompa2	ON Pompa3	OFF Pompa3	
Pre - opt	ON fisso = 4m	OFF fisso = 5m	ON fisso = 3,5m	OFF fisso = 4,5m	ON fisso = 3m	OFF fisso = 4m	1318,87
Post - opt	ON variabile = 4m	OFF variabile = 4,3m	ON variabile = 4m	OFF variabile = 4,3m	ON variabile = 3m	OFF variabile = 3,2m	1099,5

Tabella 6. Confronto risultati ottimizzazione ON-OFF

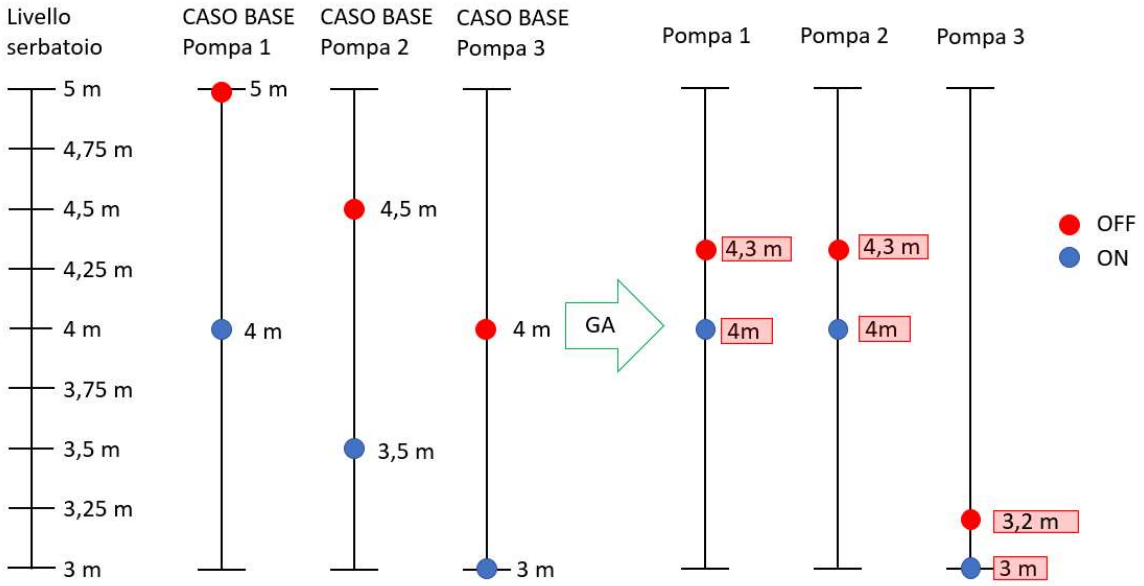


Figura 103. Confronto risultati ottimizzazione ON-OFF

In questa simulazione gli ON-OFF delle 3 pompe sono stati lasciati liberi di variare come indicato in seguito:

- ON pompa1 tra 4 e 4,8m;
- OFF pompa1 tra 4,2 e 5m;
- ON pompa2 tra 3,5 e 4,3;
- OFF pompa 2 tra 3,7 e 4,5;
- ON pompa3 tra 3 e 3,8m;
- OFF pompa 3 tra 3,2 e 4m.

È stato impostato il vincolo di differenza di almeno 0,2m tra l'ON e l'OFF.

Una volta che il codice principale trova i valori la popolazione di ON-OFF essi vengono sostituiti nella funziona fitness all'interno della rete neurale. Nella successiva Figura 104 è illustrato il pezzo

di codice con il modellino che definisce il setting delle pompe a seconda del livello dell'acqua nel serbatoio. L'ON e OFF sono chiamati nel codice rispettivamente min e max.

fitness_ON-OFF.m

```
36 -      if h_acqua_stimata(i)>=max1
37 -          setting_feb_stimato(i,1)=0;
38 -      elseif h_acqua_stimata(i)<=min1
39 -          setting_feb_stimato(i,1)=1;
40 -      elseif min1<h_acqua_stimata(i) && h_acqua_stimata(i)<max1 && setting_feb_stimato(i-1,1)==0
41 -          setting_feb_stimato(i,1)=0;
42 -      elseif min1<h_acqua_stimata(i) && h_acqua_stimata(i)<max1 && setting_feb_stimato(i-1,1)==1
43 -          setting_feb_stimato(i,1)=1;
44 -      end
45 -      if h_acqua_stimata(i)>=max2
46 -          setting_feb_stimato(i,2)=0;
47 -      elseif h_acqua_stimata(i)<=min2
48 -          setting_feb_stimato(i,2)=1;
49 -      elseif min2<h_acqua_stimata(i) && h_acqua_stimata(i)<max2 && setting_feb_stimato(i-1,2)==0
50 -          setting_feb_stimato(i,2)=0;
51 -      elseif min2<h_acqua_stimata(i) && h_acqua_stimata(i)<max2 && setting_feb_stimato(i-1,2)==1
52 -          setting_feb_stimato(i,2)=1;
53 -      end
54 -      if h_acqua_stimata(i)>=max3
55 -          setting_feb_stimato(i,3)=0;
56 -      elseif h_acqua_stimata(i)<=min3
57 -          setting_feb_stimato(i,3)=1;
58 -      elseif min3<h_acqua_stimata(i) && h_acqua_stimata(i)<max3 && setting_feb_stimato(i-1,3)==0
59 -          setting_feb_stimato(i,3)=0;
60 -      elseif min3<h_acqua_stimata(i) && h_acqua_stimata(i)<max3 && setting_feb_stimato(i-1,3)==1
61 -          setting_feb_stimato(i,3)=1;
62 -      end
```

Figura 104. Codice: ottimizzazione ON-OFF

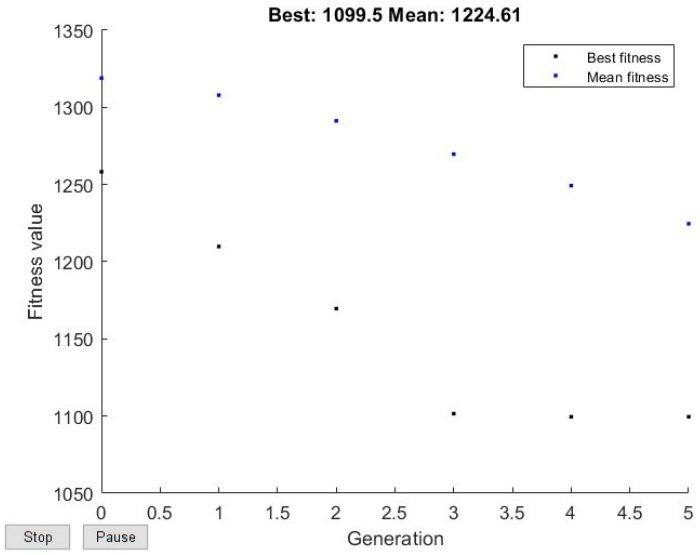


Figura 105. Ga: ottimizzazione ON-OFF

Nella Figura 105 è mostrato l'andamento della fitness nel corso delle 5 generazioni impostate. I pallini neri in basso mostrano il valore della fitness del migliore individuo della popolazione, mentre quelli blu in alto mostrano la media delle fitness di tutti gli individui della popolazione.

OTTIMIZZAZIONE VALVOLE

Per l'ottimizzazione della regolazione con valvole si è scelto di utilizzare l'intervallo temporale pari a 1 minuto, che tra i 3 simulati nel Paragrafo 4.3 (1 ora, 10 minuti, 1 minuto) è stato quello che ha dato risultati migliori (si rimanda alla Tabella 4).

Visto i tempi necessari a compiere la simulazione di una settimana, si è proceduto con la simulazione di un solo giorno, cosa che ha permesso di avere tempi non troppo lunghi impostando un numero non troppo basso di individui per ogni popolazione. Nella tabella di seguito vengono mostrate le opzioni impostate nell'algoritmo genetico di ottimizzazione nel caso con valvole, il tempo di simulazione e il Δt di riferimento

	Individui per popolazione	Generazioni	t necessario	Δt
Post - opt	40	4	2,5 h	1 minuto

Tabella 7. GA: ottimizzazione valvole

La regolazione delle valvole nel caso base è quella illustrata nella Figura 93, in cui il k, coefficiente di perdita della valvola, è distribuito in modo uniforme, per ciascuna valvola, nell'intervallo di riferimento mostrato in figura. Poiché ogni valvola presenta 17 variabili, si è deciso di applicare l'ottimizzazione a una sola valvola, la prima; volendo ottimizzare tutte le valvole le variabili sarebbero state 51.

Il limite che è stato imposto per la ricerca delle variabili è che tra un livello e l'altro ci siano almeno 0,03m di differenza.
Si mostra di seguito una figura riassuntiva di confronto dei risultati ottenuti.

	REGOLAZIONE valvola 1																	Consumi totali [kWh]
Pre – opt	4,06	4,11	4,17	4,22	4,28	4,33	4,39	4,44	4,5	4,56	4,61	4,67	4,72	4,78	4,83	4,89	4,94	1448,75
Post - opt	4,007	4,037	4,067	4,097	4,127	4,157	4,187	4,217	4,247	4,277	4,307	4,337	4,367	4,397	4,427	4,457	5	1345,20

Tabella 8. Confronto risultati ottimizzazione valvole

Come si può notare dalla tabella precedente, mentre nel caso base gli intervalli dei k delle valvole sono equamente distribuiti lungo l'intervallo di riferimento, nel caso ottimizzato i k tendono a distribuirsi nella parte bassa dell'intervallo, rispettando giusto il limite imposto di 0,03m.

Una volta che il codice principale trova gli individui di ogni generazione, essi vengono sostituiti nella funziona fitness all'interno della rete neurale. Nella successiva è illustrato il pezzo di codice con il modellino che definisce il setting delle valvole a seconda del livello dell'acqua nel serbatoio. Il livello dell'acqua nel serbatoio viene chiamato nel codice rispettivamente Vx.

fitness_valvole.m

```
49 -        if h_acqua_stimata(i)>=Vx17
50 -            setting_feb_stimato(i,1)=10000;
51 -        elseif Vx16<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx17
52 -            setting_feb_stimato(i,1)=8000;
53 -        elseif Vx15<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx16
54 -            setting_feb_stimato(i,1)=6000;
55 -        elseif Vx14<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx15
56 -            setting_feb_stimato(i,1)=4000;
57 -        elseif Vx13<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx14
58 -            setting_feb_stimato(i,1)=2000;
59 -        elseif Vx12<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx13
60 -            setting_feb_stimato(i,1)=1000;
61 -        elseif Vx11<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx12
62 -            setting_feb_stimato(i,1)=800;
63 -        elseif Vx10<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx11
64 -            setting_feb_stimato(i,1)=600;
65 -        elseif Vx9<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx10
66 -            setting_feb_stimato(i,1)=400;
67 -        elseif Vx8<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx9
68 -            setting_feb_stimato(i,1)=200;
69 -        elseif Vx7<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx8
70 -            setting_feb_stimato(i,1)=150;
71 -        elseif Vx6<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx7
72 -            setting_feb_stimato(i,1)=100;
73 -        elseif Vx5<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx6
74 -            setting_feb_stimato(i,1)=55;
75 -        elseif Vx4<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx5
76 -            setting_feb_stimato(i,1)=37.5;
77 -        elseif Vx3<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx4
78 -            setting_feb_stimato(i,1)=20;
79 -        elseif Vx2<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx3
80 -            setting_feb_stimato(i,1)=5;
81 -        elseif Vx1<h_acqua_stimata(i) && h_acqua_stimata(i)<=Vx2
82 -            setting_feb_stimato(i,1)=0.6;
83 -        elseif h_acqua_stimata(i)<=Vx1
84 -            setting_feb_stimato(i,1)=0.2;
```

Figura 106. Codice: ottimizzazione valvole

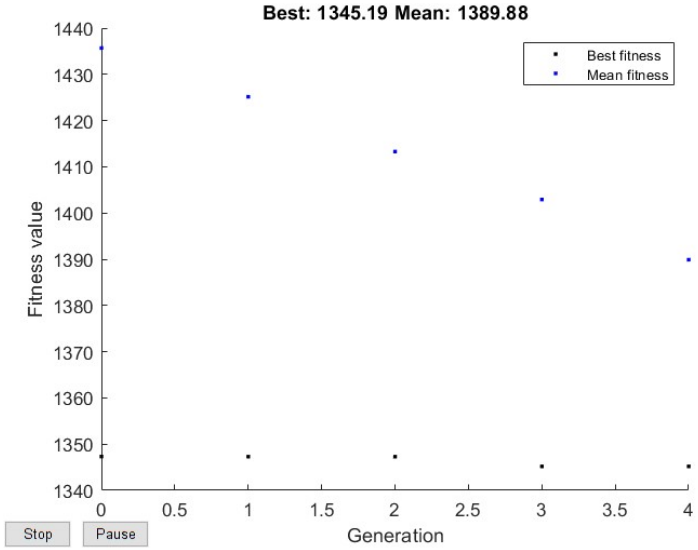


Figura 107. Ga: ottimizzazione ON-OFF

Nella Figura 107 è mostrato l'andamento della fitness nel corso delle 4 generazioni impostate. I pallini neri in basso mostrano il valore della fitness del migliore individuo della popolazione, mentre quelli blu in alto mostrano la media delle fitness di tutti gli individui della popolazione.

Capitolo 6

Conclusioni

In questo lavoro di tesi si è voluta verificare la fattibilità dell'applicazione degli algoritmi genetici a un modello creato tramite reti neurali di un sistema idrico asservito a una industria tessile.

Si è partiti dal software EPANET, tramite il quale è stato creato il modello fisico del sistema, conoscendo la richiesta oraria di acqua da parte dell'utenza, le caratteristiche e lunghezze delle tubazioni presenti, la curva caratteristica e la curva di efficienza delle pompe, i coefficienti di perdita delle valvole e l'altitudine di tutti i nodi presenti nella rete.

Una volta creato il modello fisico è stato possibile, utilizzando l'EPANET MATLAB Toolkit e gli algoritmi genetici, ottimizzare il modello stesso grazie alla modifica sui controlli sulle pompe. Si è partiti dalle prove di ottimizzazione su una sola pompa, per poi passare a quelle sulle 3 pompe che costituiscono il caso studio. Si è cercata la regolazione ottima delle pompe che minimizzi i consumi energetici delle pompe durante una settimana tipo. L'analisi è stata fatta su un Δt di 1 ora. Si riportano nella Figura 108 e Figura 109 i risultati ottenuti.

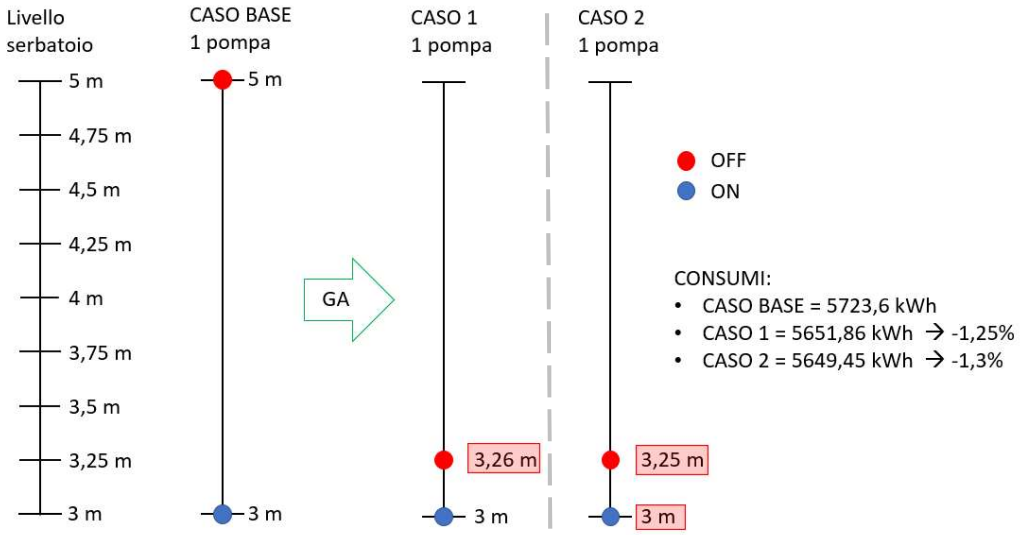


Figura 108. Grafico risultati: GA - 1 pompa

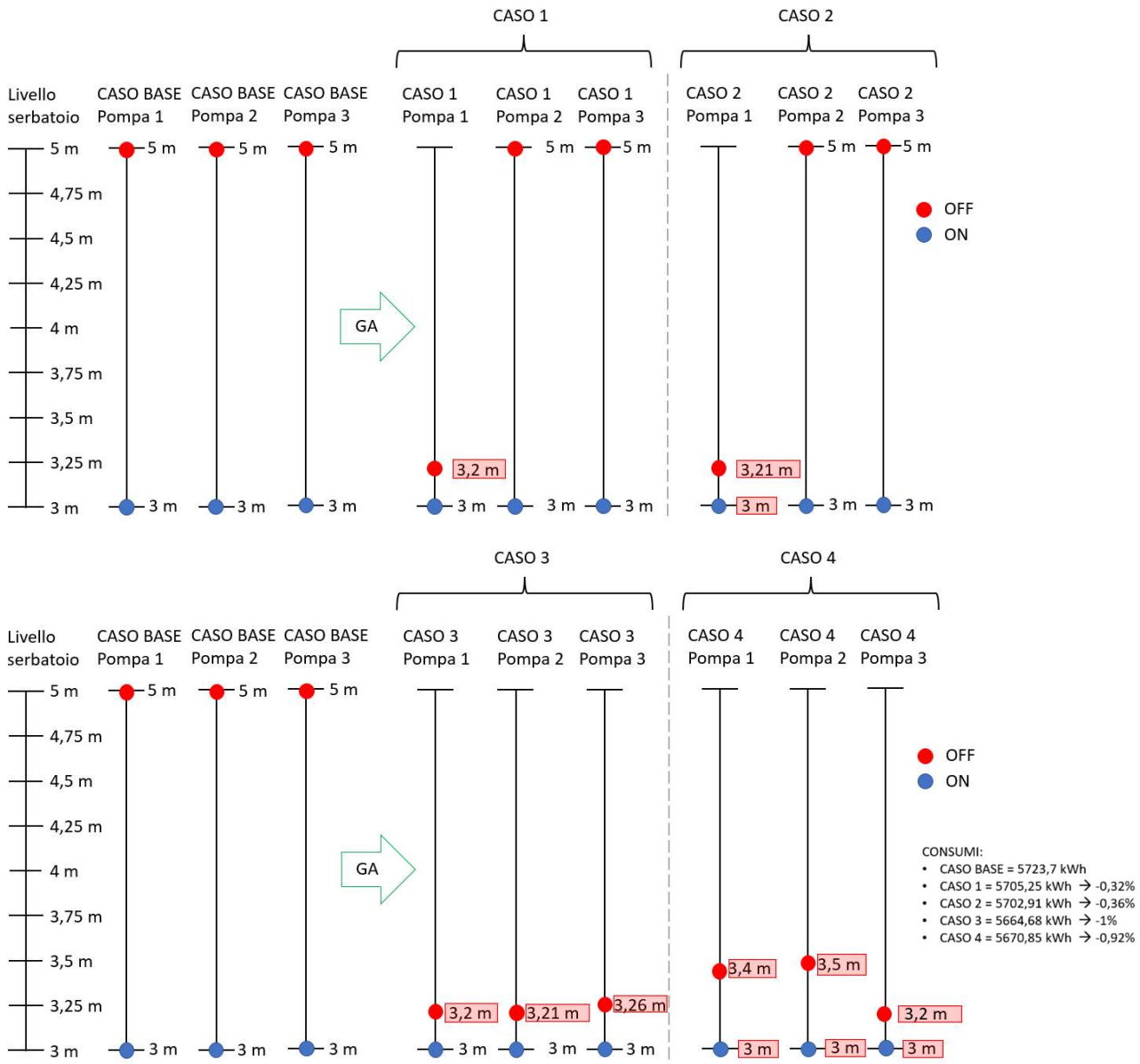


Figura 109. Grafico risultati: GA - 3 pompe

Come si nota nella Figura 108 e Figura 109, in entrambi i casi i valori ottimi della simulazione tendono ad essere i più bassi possibili rispettando i vincoli impostati: avendo lasciato le variabili libere di mutare in un range compreso tra 3 e 5, imposto il vincolo che tra ON e OFF ci siano almeno 0,2m di differenza, si può notare che l'ON viene trovato sempre uguale a 3m (il minimo possibile), e l'OFF viene trovato quasi sempre intorno a 3,2m (anche in questo caso il minimo possibile).

Successivamente, grazie ai dati ricavati a partire dal modello fisico sviluppato con EPANET, si è cercato di creare un modello di reti neurali in grado di simulare sufficientemente bene il sistema. Sono stati analizzati due tipi di regolazione: ON-OFF e con le valvole. Una volta allenate le reti e

con un solo input (richiesta di acqua da parte della utenza), esse sono in grado di prevedere la portata di acqua attraverso le 3 pompe per una settimana, a partire dalla quale è possibile ricavare i consumi. L'analisi è stata fatta a partire da un Δt di 1 ora, ma non avendo ottenuto risultati soddisfacenti, si è deciso di diminuire l'intervallo temporale prima a 10 minuti, poi a 1 minuto. Si riportano nella Tabella 9 i risultato ottenuti.

	REGOLAZIONE ON-OFF		REGOLAZIONE CON VALVOLE	
Δt	t esecuzione	R^2 portate pompe	t esecuzione	R^2 portate pompe
1 h	10 secondi	0,75	8 secondi	0,60
10 minuti	29 secondi	0,92	27 secondi	0,86
1 minuto	12 minuti	0,86	12 minuti	0,99

Tabella 9. Riassunto risultati NN

Come si può notare dalla Tabella 9, diminuendo l'intervallo temporale la precisione di previsione aumenta per la regolazione con valvole, mentre per la regolazione ON-OFF la previsione migliore si ha per il Δt pari a 10 minuti. Da notare anche come con il diminuire dell'intervallo temporale di analisi le tempistiche di esecuzione del modello aumentino sensibilmente.

Infine si è cercato di ottimizzare con gli algoritmi genetici il modello della rete idrica creato precedentemente con le reti neurali. Questa ottimizzazione è stata sviluppata introducendo il modello di reti neurali all'interno della funzione fitness dell'algoritmo genetico. L'ottimizzazione è stata tentata sia per quanto riguarda la regolazione ON-OFF che per quanto riguarda la regolazione con le valvole. Visti i tempi necessari al modello per girare, si è pensato di provare l'ottimizzazione su un giorno invece che su una settimana, e di diminuire (soprattutto per la regolazione con valvole), il numero di individui per popolazione nell'algoritmo genetico.
Per il caso ON-OFF le ipotesi di partenza e il tempo necessario sono illustrate nella Tabella 10 e i risultati ottenuti nella Figura 110.

	REGOLAZIONE ON-OFF				
	Δt	previsione	Individui per popolazione	Generazioni	t esecuzione
post - opt	10 minuti	1 giorno	100	5	1 h

Tabella 10. Ipotesi di partenza e t necessario ottimizzazione ON-OFF

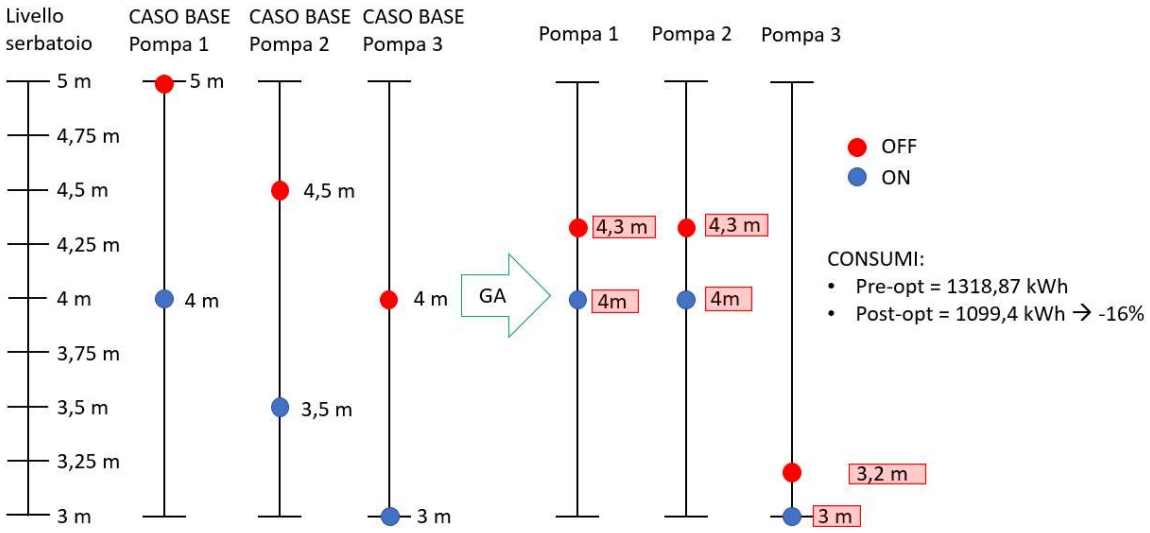


Figura 110. Grafico risultati: ottimizzazione ON-OFF

Anche in questo caso, come visto precedentemente in Figura 108 e Figura 109, i risultati ottimali sono tendenzialmente i minimi possibili, a parte che per la pompa 2.

Per il caso con valvole le ipotesi di partenza e il tempo necessario al modello per girare sono illustrati nella Tabella 11 e i risultati ottenuti nella Tabella 12. In questa trattazione si è deciso, per una questione di tempo, di ottimizzare la regolazione della sola valvola 1, mentre per quanto riguarda la 2 e la 3 si è lasciata la regolazione impostata per il caso base (vedi Figura 93).

	REGOLAZIONE CON VALVOLE				
	Δt	previsione	Individui per popolazione	Generazioni	t esecuzione
Post -opt	1 minuto	1 giorno	40	4	2,5 h

Tabella 11. Ipotesi di partenza e t necessario: ottimizzazione con valvole

	REGOLAZIONE valvola 1																	Consumi totali [kWh]
Pre - opt	4,06	4,11	4,17	4,22	4,28	4,33	4,39	4,44	4,5	4,56	4,61	4,67	4,72	4,78	4,83	4,89	4,94	1448,75
Post - opt	4,007	4,037	4,067	4,097	4,127	4,157	4,187	4,217	4,247	4,277	4,307	4,337	4,367	4,397	4,427	4,457	5	1345,20

Tabella 12. Tabella risultati: ottimizzazione con valvole

Come si vede nella Tabella 12, ottimizzando la regolazione anche solo di una valvola i consumi passano da 1448,75 kWh al giorno a 1345,2 kWh al giorno, con un risparmio di circa il 7,1%.

Il limite principale dell'ottimizzazione messa in atto sono le tempistiche troppo lunghe. Per ottimizzare un solo giorno, il tempo necessario va da 1 ora del caso della regolazione ON-OFF a 2,5 ore del caso della regolazione con valvole. Esplorando l'utilizzo di reti neurali diverse da quelle qui utilizzate o provando a cambiare l'architettura del modello di reti neurali sviluppato, si potrebbe arrivare a diminuire le tempistiche di esecuzione. Una volta abbattuto il tempo di esecuzione del modello, si potrebbe anche aumentare il numero di individui per popolazione e il numero di generazioni impostate, così da dare la possibilità all'algoritmo di trovare soluzioni ancora migliori di quelle trovate in questo lavoro di tesi e di diminuire ulteriormente i consumi.

BIBLIOGRAFIA e SITOGRAFIA

[1] https://indigo.ai/it/machine-learning-definizione-e-campi-applicazione/

[2] Trasporti e infrastrutture: efficienza energetica del settore idrico, Confindustria

[3] ISPRA - Il sistema degli acquedotti in Italia

[4] Gli algoritmi genetici, T. Donateo, Università degli studi di Lecce

[5] Introduzione agli algoritmi genetici. D. Nova

[6] Rapporto mondiale delle Nazioni Unite sullo sviluppo delle risorse idriche 2021

[7] Osservatorio "Water Management" – Politecnico di Milano

[8] EPANET examples

[9] Machine learning: trends, perspectives, and prospects. M. I. Jordan, T. M. Mitchell

[10] http://www.fao.org/aquastat/en/overview/methodology/water-use

[11]https://www.europarl.europa.eu/news/it/headlines/society/20201208STO93327/l-impatto-della-produzione-e-dei-rifiuti-tessili-sull-ambiente-infografica

[12] Burek et al., 2016

[13] D.G. Eliades, M. Kyriakou, S. Vrachimis and M.M. Polycarpou, "EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB", in *Proc. 14th International Conference on Computing and Control for the Water Industry (CCWI)*, The Netherlands, Nov 2016, p.8. (doi:10.5281/zenodo.831493)

[14] INTRODUCTION TO MACHINE LEARNING , by Ethem Alpaydin, MIT Press, 2004

[15] Ghahramani Z. (2004) Unsupervised Learning. In: Bousquet O., von Luxburg U., Rätsch G. (eds) Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science, vol 3176. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28650-9_5

[16] Reinforcement Learning: An Introduction, by Richard S.Sutton and Andrew G. Barto, 2015

[17] https://www.focus.it/scienza/salute/come-funziona-il-cervello

[18] S. Haykin. Neural Networks: a Comprehensive Foundation. Macmillan College Puglishing Company, 866 Third Avenue, New York, 10022, 1st edition, 1994

[19] https://it.wikipedia.org/wiki/Coefficiente_di_determinazione

[20]https://it.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html

[21] https://it.mathworks.com/help/gads/how-the-genetic-algorithm-works.html

[22] https://it.mathworks.com/help/gads/ga.html#mw_46d5ecc6-7ae1-4400-821b-3a29c14be603