

UNIVERSITÀ DEGLI STUDI DI SALERNO



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE  
ED ELETTRICA E MATEMATICA APPLICATA

**TITOLO**

**Relazione Progetto**

**Simulated Car Racing Championship**

*Autori:*

Longobardi Francesco

Montillo Andrea

# Indice

<b>Introduzione</b>	1
<b>Capitolo 2: Scelte Progettuali</b>	3
2.1 - NEAT	3
2.2 - Scelta dei parametri	6
2.3 - Scelta del tracciato	8
<b>Capitolo 3: Prime sperimentazioni</b>	10
3.1 - Tuning dei parametri	12
3.2 - Valutazione performance	18
3.3 - Analisi delle criticità	20
<b>Capitolo 4: Nuove sperimentazioni</b>	22
4.1 - Problema del danno	22
4.1.1 Risultati ottenuti	22
4.1.2 Analisi addestramento	23
4.2 - Addestramenti multipli	24
4.2.1 Risultati ottenuti	26
4.2.2 Analisi prestazioni sui singoli tracciati	29
4.2.3 Considerazioni finali	31
<b>Capitolo 5: Variazione iperparametri</b>	33
<b>Conclusioni</b>	40

# Introduzione

In questo documento si descrive come, attraverso tecniche di Computazione Naturale, è possibile ottimizzare le performance di un controllore artificiale il cui obiettivo è quello di partecipare a gare automobilistiche simulate attraverso TORCS.

*The Open Car Racing Simulator* è un simulatore 3D, open source, che permette facilmente di progettare controllori automobilistici consentendogli di gareggiare su tracciati con diverse caratteristiche.

Si è deciso di ottimizzare un controllore già esistente, basato sulla libreria Snake Oil, il quale risulta essere il più performante rispetto ad altre alternative.

Snake Oil è una libreria Python, utile per interfacciarsi con il simulatore TORCS, la quale è stata aggiornata con le estensioni per i server utilizzati nella competizione Simulated Car Racing, permettendo lo scambio dei dati tra il controllore, ovvero il client, ed il server sul quale viene avviata la gara.

In particolare, sono disponibili i valori di diversi sensori in ogni istante della gara, in base a questi input il controllore dovrà stabilire quali sono i valori degli effettori. Di seguito vengono riportate le tabelle ottenute dal sito ufficiale del framework, le quali illustrano i sensori e gli attuatori disponibili

Name	Range (unit)	Description
angle	$[-\pi, +\pi]$ (rad)	Angle between the car direction and the direction of the track axis.
curLapTime	$[0, +\infty)$ (s)	Time elapsed during current lap.
damage	$[0, +\infty)$ (point)	Current damage of the car (the higher is the value the higher is the damage).
distFromStart	$[0, +\infty)$ (m)	Distance of the car from the start line along the track line.
distRaced	$[0, +\infty)$ (m)	Distance covered by the car from the beginning of the race
fuel	$[0, +\infty)$ (l)	Current fuel level.
gear	$\{-1, 0, 1, \dots, 7\}$	Current gear: -1 is reverse, 0 is neutral and the gear from 1 to 7.
wheelSpinVel	$[0, +1]$ (rad/s)	Vector of 4 sensors representing the rotation speed of wheels.
lastLapTime	$[0, +1)$ (s)	Time to complete the last lap
opponents	$[0, 200]$ (m)	Vector of 36 opponent sensors: each sensor covers a span of 18 (10 degrees) within a range of 200 meters and returns the distance of the closest opponent in the covered area. The 36 sensors cover all the space around the car, spanning clockwise from $+\pi$ up to $-\pi$ w.r.t car ax.

racePos	{1,2, ..., N}	Position in the race with respect to other cars.
rpm	[2000,7000] (rpm)	Number of rotation per minute of the car engine.
focus	[0,200] (m)	Vector of 5 range finder sensors: each sensor returns the distance between the track edge and the car within a range of 200 meters. The sensors sample, with a resolution of one degree, a ve degree space along a specific direction provided by the client (the direction is defined with the focus command and must be in the range $[-\pi/2, +\pi/2]$ w.r.t. the car axis).
speedX	$(-\infty, +\infty)$ (km/h)	Speed of the car along the longitudinal axis of the car.
speedY	$(-\infty, +\infty)$ (km/h)	Speed of the car along the transverse axis of the car.
speedZ	$(-\infty, +\infty)$ (km/h)	Speed of the car along the Z axis of the car.
track	[0,200] (m)	Vector of 19 range finder sensors: each sensors returns the distance between the track edge and the car within a range of 200 meters. By default, the sensors sample the space in front of the car every 10 degrees, spanning clockwise from $+\pi/2$ up to $-\pi/2$ with respect to the car axis. However, the configuration of the range nder sensors (i.e., the angle w.r.t. to the car axis) before the beginning of each race. When the car is outside of the track (i.e., pos is less than -1 or greater than 1), the returned values are not reliable.
trackPos	$[-\infty, +\infty]$ (m)	Distance between the car and the track axis. The value is normalized w.r.t to the track width: it is 0 when car is on the axis, -1 when the car is on the right edge of the track and +1 when it is on the left edge of the car. Values greater than 1 or smaller than -1 mean that the car is outside of the track.
wheelSpinVel	[0, $+\infty$ ] (rad/s)	Vector of 4 sensors representing the rotation speed of wheels.
z	$[-\infty, +\infty]$ (m)	Distance of the car mass center from the surface of the track along the Z axis.

*Tabella 1.1: Descrizione dei sensori disponibili. I range di variazione sono riportati con le rispettive unità di misura (se definite).*

Name	Range	Description
accel	[0,1]	Virtual gas pedal (0 means no gas, 1 full gas).
brake	[0,1]	Virtual brake pedal (0 means no brake, 1 full brake).
clutch	[0,1]	Virtual clutch pedal (0 means no clutch, 1 full clutch).
gear	-1,0,1, ..., 7	Gear value.
steering	[-1,1]	Steering value: -1 and +1 means respectively full right and left, that corresponds to an angle of 0.785398 rad.
focus	[-90,90]	Focus direction (see the focus sensors in Table 1) in degrees.
meta	0,1	This is meta-control command: 0 do nothing, 1 ask competition server to restart the race.

*Tabella 1.2: Descrizione degli effettori disponibili.*

## Capitolo 2: Scelte Progettuali

Per raggiungere l'obiettivo prefissato, è stata applicata la *Neuro Evolution of Augmenting Topologies* (NEAT), ovvero un algoritmo genetico per l'evoluzione di reti neurali artificiali. La motivazione di questa scelta risiede nel modo in cui è espresso il problema. Esso, infatti, consiste nell'assegnare i corretti valori delle uscite, gli effettori, a partire dai valori degli ingressi, i sensori, e quindi modellare il problema attraverso una rete neurale artificiale risulta particolarmente efficace. In particolare, non si vogliono porre vincoli sulla struttura stessa della rete, non è noto a priori la complessità di cui si ha bisogno e dunque è risultato opportuno scegliere la tecnica NEAT.

Per l'utilizzo di questo tipo di algoritmo è stata utilizzata l'omonima [libreria Python](#) che ha permesso, dunque, di sfruttare algoritmi standard per quanto concerne l'evoluzione della rete, ovvero algoritmi di riproduzione, speciazione, mutazione, ecc.

### 2.1 - NEAT

La scelta di utilizzare una tecnica di *Neuro Evolution* è dovuta al fatto che queste risultano particolarmente efficaci per task di *Reinforcement Learning*, dal momento che non necessitano di un apprendimento supervisionato.

La Neuro Evolution è una forma di intelligenza artificiale in cui si utilizzano algoritmi genetici per sviluppare una popolazione formata da reti neurali artificiali. Queste tecniche si basano, quindi, su classici algoritmi di cui si riportano i passaggi base:

- Inizializzazione casuale della popolazione.
- Valutazione della fitness per ogni individuo della popolazione.
- Selezione degli individui migliori in base alla fitness.
- Crossover tra gli individui scelti per generare nuove soluzioni ibride.
- Mutazione casuale, possibile per ogni individuo.
- Rimpiazzo della popolazione con la nuova generazione.
- Si ripetono i passaggi dal secondo punto, finché non si è arrivati ad una condizione di terminazione dell'algoritmo.

Quando vengono applicate queste tecniche alle reti neurali artificiali, i tre problemi principali da considerare, che lo stesso NEAT si trova a dover risolvere, sono:

1. come applicare in maniera significativa l'algoritmo di crossover a diverse topologie di reti neurali artificiali;
2. come proteggere innovazioni topologiche che richiedono qualche generazione per poter essere ottimizzate e, quindi, evitarne la prematura scomparsa dalla popolazione;
3. come minimizzare la complessità topologica degli individui durante l'evoluzione senza utilizzare una funzione di fitness artificiosa.

In NEAT, ogni individuo, o genoma, è formato da due tipi di geni individuati col nome di “*node genes*” e “*connection genes*”, i quali contengono informazioni come il peso della connessione e i nodi coinvolti nella stessa. Esistono, quindi, due possibili mutazioni, una riguardante i pesi delle connessioni, l'altra riguardante la struttura della rete.

Quest'ultima può risultare sia come una aggiunta o rimozione di una connessione, sia come una aggiunta o rimozione di un nodo. Inoltre, ogni gene può essere presente nell'individuo, ma potrebbe essere disabilitato e, quindi, non risultare visibile nel *fenotipo*, ovvero la rete.

Infine, viene assegnato ad ogni gene un “*numero di innovazione*” che viene incrementato per i nuovi geni che compaiono in seguito ad una mutazione strutturale. In questo modo si tiene traccia dell'origine storica di ogni gene e, quindi, del genoma stesso.

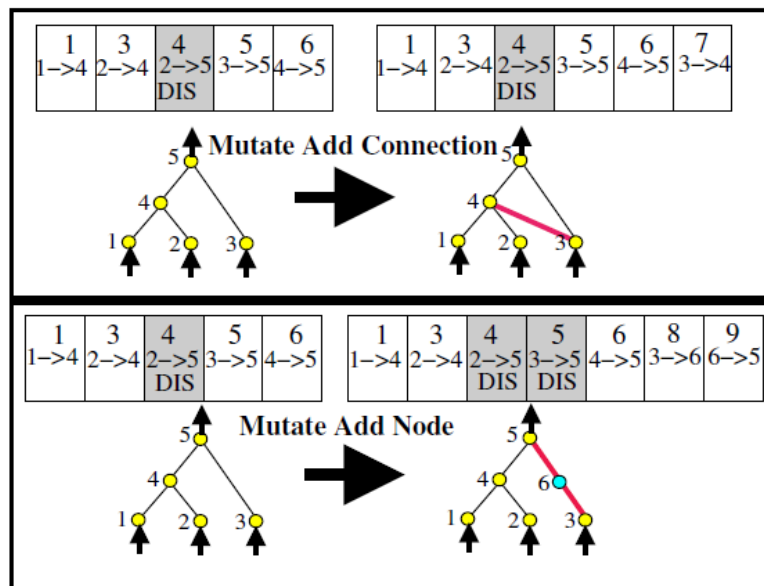


Figura 2.1: Le due tipologie di mutazione strutturale di un genoma. In particolare in entrambi i casi si tratta di un'aggiunta del gene. Si noti, inoltre, come alcuni geni siano disabilitati e si notino i valori numerici assegnati ad ogni gene per il tracciamento storico.

A questo punto è possibile illustrare come viene risolto il primo problema dell'elenco di cui sopra. In particolare, dati due genomi di cui bisogna effettuare il crossover, vengono innanzitutto presi i geni comuni, ovvero quelli con le stesse caratteristiche compreso il numero di

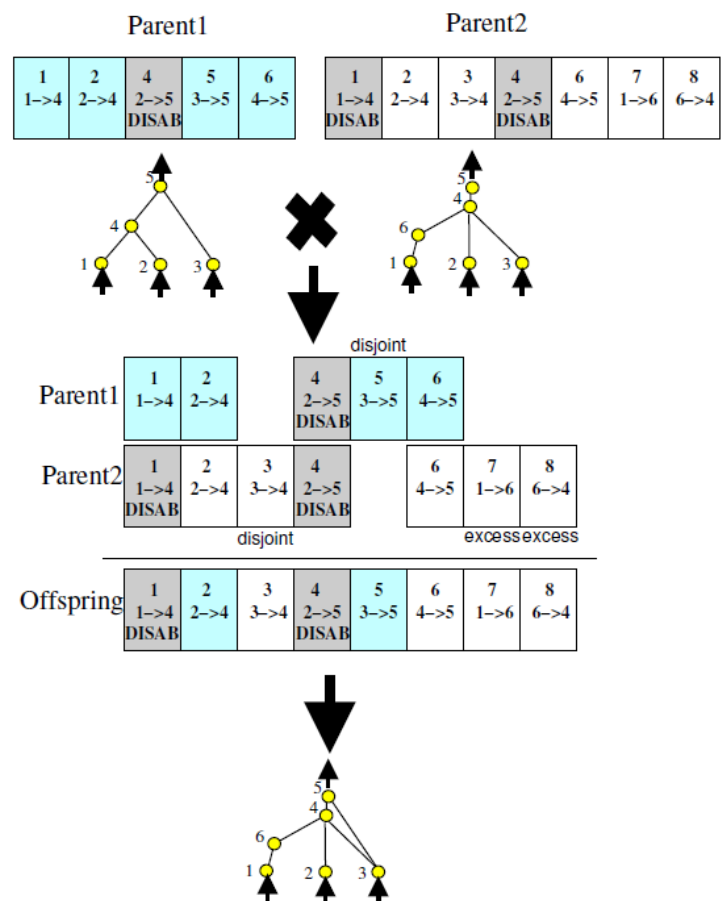
innovazione, mentre i geni in conflitto vengono ereditati dal genitore con fitness maggiore.

Il secondo problema viene, invece, risolto tramite i numeri di innovazione che permettono di differenziare i genomi con diverse strutture topologiche, formando quindi diverse specie di individui. Dunque, la competizione avviene solo tra genomi della stessa specie, impedendo quindi la scomparsa di innovazioni topologiche che portano ad una iniziale perdita di fitness.

Infine, il terzo ed ultimo problema viene risolto impostando una condizione iniziale più semplice possibile.

In particolare, inizialmente si considerano semplicemente lo strato di input e quello di output della rete e, quindi, eventuali nodi hidden verranno aggiunti solamente attraverso mutazioni strutturali.

In questo modo, vengono evolute solo quelle innovazioni topologiche che portano ad un effettivo miglioramento in termini di fitness. Quindi, NEAT ricerca la soluzione in un insieme minimo di geni riuscendo a convergere verso una soluzione ottimale in poco tempo rispetto alle altre tecniche di evoluzione di reti neurali artificiali.



## 2.2 - Scelta dei parametri

Con lo scopo di evitare tempi di addestramento della rete insostenibili, si è scelto un numero di input e di output minore rispetto a quelli disponibili. In particolare, sono stati scelti quelli reputati maggiormente significativi, ovvero, per quanto riguarda gli input:

- *Angle*: angolo tra la direzione del veicolo e l'asse centrale della strada.
- La velocità della vettura, espressa sulle tre coordinate X, Y e Z.
- *TrackPos*: posizione della vettura rispetto l'asse centrale della strada.
- *Track*: distanza tra il bordo della pista e l'auto. Dei 19 sensori disponibili, ne sono stati scelti soltanto 5, in particolare lungo le direzioni  $\{-45, -19, 0, 19, 45\}$ .

Invece, per quanto riguarda gli output si sottolinea che questi vengono considerati come variazioni da applicare al comportamento del controllore base. Si è scelto di utilizzare:

- Acceleratore
- Freno
- Sterzo

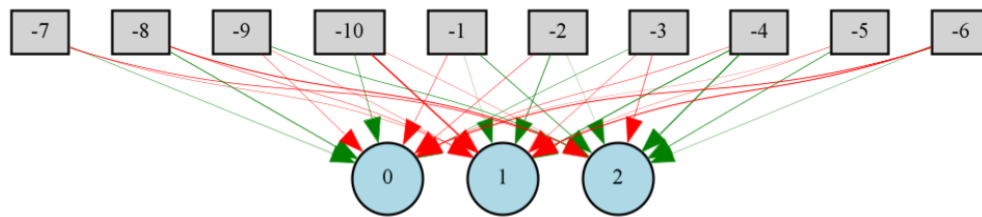
In particolare, il sensore *Track* lungo la direzione 0 è stato considerato come il massimo tra i sensori lungo le direzioni  $\{-7, 0, 7\}$ . La scelta è stata fatta nell'ottica di individuare delle situazioni particolari del tracciato, come ad esempio la parte finale di una curva.

Inoltre, la configurazione base dei sensori *Track* è configurabile nell'intervallo che va da -90 gradi a +90 gradi. La scelta fatta si è basata su alcune brevi sperimentazioni le quali hanno evidenziato che, considerando un numero di generazioni ed un numero di individui nella popolazione fissato, i parametri scelti sono quelli di maggior impatto e permettono un apprendimento migliore con il conseguente aumento delle performance rispetto ad altre varianti considerate, ovvero:

- ❖  $\{-90, -60, 30, 0, 30, 60, 90\}$
- ❖  $\{-45, -12, -4, 0, 4, 12, 45\}$



La rete iniziale considerata è composta da 10 neuroni nello strato di input e 3 in quello di output senza strati hidden ed, inoltre, è una “*fully connected*”, ovvero tutti gli input sono collegati a tutti gli output della rete.



*Figura 2.2: Immagine di una rete allo stato iniziale. Il colore rosso dei collegamenti indica un peso negativo dello stesso.*

Oltre ai parametri che caratterizzano la gara, divisi, come si è visto, in input dei sensori e output sugli effettori, il controllore base presenta molteplici “meta-parametri” per regolare il proprio funzionamento. Si è, però, ritenuto opportuno mantenere questi parametri fissati essendo il numero troppo elevato per un’analisi specifica e, soprattutto, avendo considerato le prestazioni del controllore base come un buon punto di partenza, senza necessità di raffinamenti per lo sviluppo del controllore in esame.

Passando, invece, all’analisi dei parametri che caratterizzano l’algoritmo utilizzato, in generale sono molteplici, ma in questo elaborato si è deciso di svolgere un’analisi dettagliata solo di alcuni di essi, ovvero:

- *population size*: dimensione della popolazione di genomi per ogni generazione.
- *add node rate*: tasso di aggiunta di un nodo nella rete.
- *del node rate*: tasso di rimozione di un nodo dalla rete.

La dimensione della popolazione risulta essere il parametro che maggiormente influenza l’andamento dell’addestramento, mentre gli altri due iperparametri scelti sono quelli che aiutano maggiormente a mantenere una differenziazione elevata della topologia degli individui all’interno della popolazione.

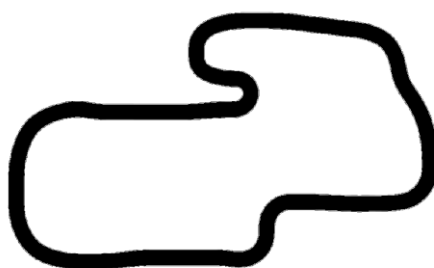
Il numero di generazioni è stato fissato a 20 in quanto, dopo alcune sperimentazioni, si è notata una convergenza dei risultati già dopo le prime 10-15 generazioni. Per quanto riguarda, invece, la funzione di attivazione dei neuroni, è stata scelta la sigmoide ed, inoltre, il tasso di aggiunta o rimozione dei collegamenti tra i neuroni è stato scelto pari a 0.5. Infine, si è scelto un elitismo pari ad 1 essendo di interesse esclusivamente il miglior genoma all’interno della popolazione ed essendo, soprattutto, le dimensioni della popolazione utilizzate limitate.

Per quanto riguarda gli altri parametri si è fatto affidamento ai valori di default proposti dalla libreria utilizzata.

## 2.3 - Scelta del tracciato

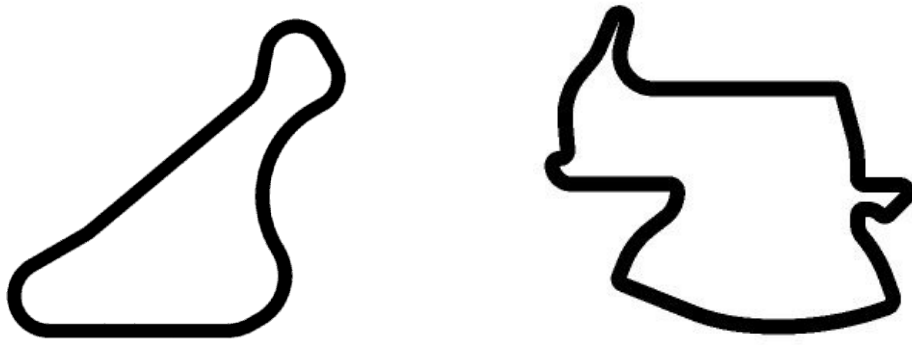
Per le prime sperimentazioni, l'addestramento del controllore si baserà sulla guida senza avversari su un tracciato scelto accuratamente per un totale di due giri, fissando un tempo massimo di percorrenza pari a 200 secondi. La scelta del tracciato si è basata su alcune caratteristiche chiave. In particolare, si è ritenuto necessario considerare un tracciato che avesse curve strette, curve larghe ma anche rettilinei, ed inoltre è risultato opportuno avere un numero di curve verso destra e verso sinistra circa pari in modo da non viziare l'addestramento della rete. La scelta è, dunque, ricaduta sul tracciato "Ruudskogen" il cui contorno è mostrato in figura 2.2.

L'obiettivo del controllore è, quindi, percorrere il tracciato nel minor tempo possibile, difatti per far evolvere la rete è stata stabilita come funzione di fitness la velocità media mantenuta dall'individuo durante la gara. Non si è, dunque, tenuto conto di altre variabili come il danno subito dalla macchina o la benzina consumata.



*Figura 2.2: Outline del tracciato Ruudskogen*

Una volta addestrato il controllore, questo verrà testato su altri due circuiti. In particolare, uno risulta essere molto semplice, utile semplicemente ad individuare miglioramenti o peggioramenti del controllore sviluppato rispetto a quello base, mentre il secondo risulta essere molto complicato in quanto caratterizzato da diverse curve molto insidiose (molto strette, molto spigolose, ad angolo). Un ulteriore fattore tenuto in considerazione per la scelta è stato la lunghezza del tracciato. Questa risulta essere non particolarmente elevata per tutti i circuiti scelti, consentendo tempi di addestramento meno onerosi. I due tracciati sono rispettivamente "CG Speedway number 1" e "Aalborg", di cui si riporta il contorno in figura 2.3.



*Figura 2.3: a sinistra: outline del tracciato CG Speedway number 1  
a destra: outline del tracciato Aalborg*

Infine, è bene precisare che gli esperimenti prevedono cinque diversi addestramenti per ogni configurazione degli iperparametri, con configurazioni iniziali casuali e diverse. Dunque, se ne analizzerà il comportamento medio durante l'analisi degli addestramenti. Nel confronto con il controllore base verranno, quindi, utilizzate le cinque migliori soluzioni trovate, ovvero i migliori individui delle popolazioni, durante gli addestramenti. Un numero di inizializzazioni casuali maggiore avrebbe portato indubbiamente ad un'analisi più precisa dei risultati, ma non è stato possibile a causa delle risorse a disposizione.

## Capitolo 3: Prime sperimentazioni

La prima sperimentazione effettuata prevede, quindi, un'unica fase di addestramento sul tracciato Ruudskogen. L'obiettivo è quello di determinare la configurazione migliore degli iperparametri considerati, data un'evoluzione basata su 20 generazioni. La fitness di ogni genoma viene valutata in base alla velocità media mantenuta durante la gara, ponendo, quindi, come obiettivo massimo 333  $km/h$  che è la velocità massima della vettura utilizzata e che quindi rappresenta, idealmente, la massima velocità media ottenibile.

Di seguito si presentano i dettagli dell'evoluzione con una dimensione della popolazione pari a 20, un *add node rate* ed un *del node rate* entrambi pari a 0.5.

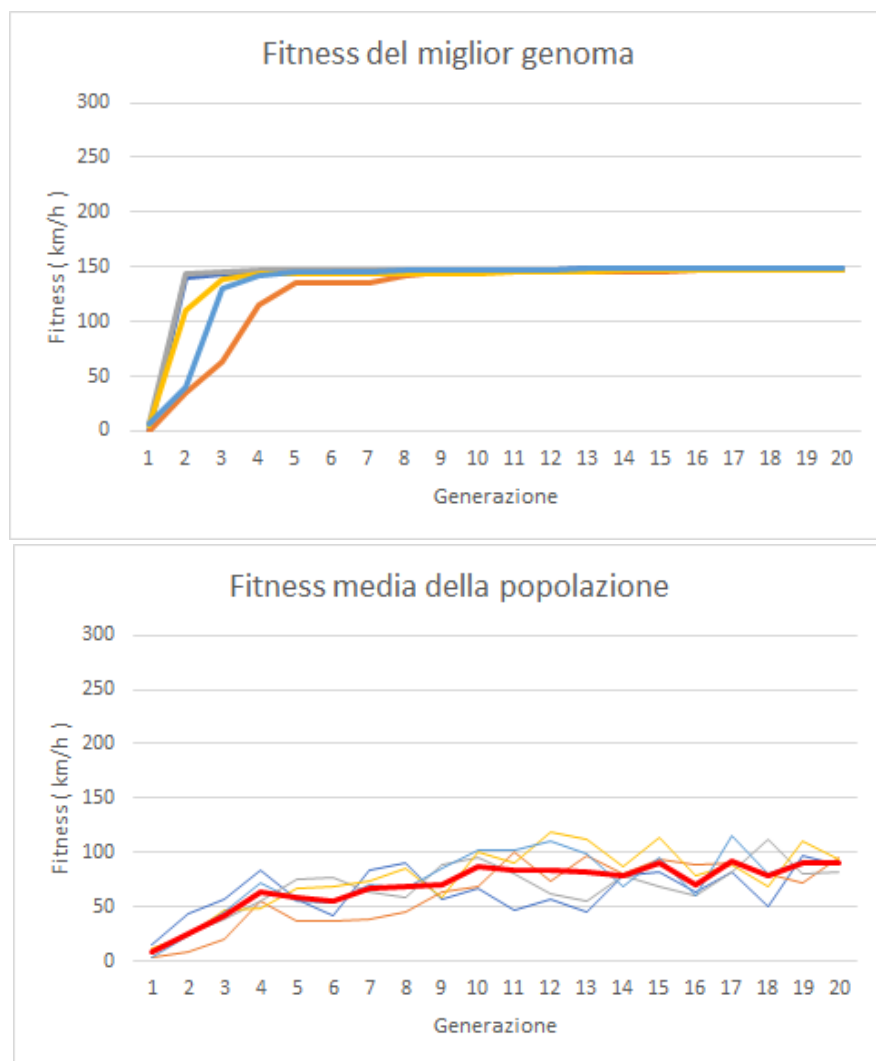
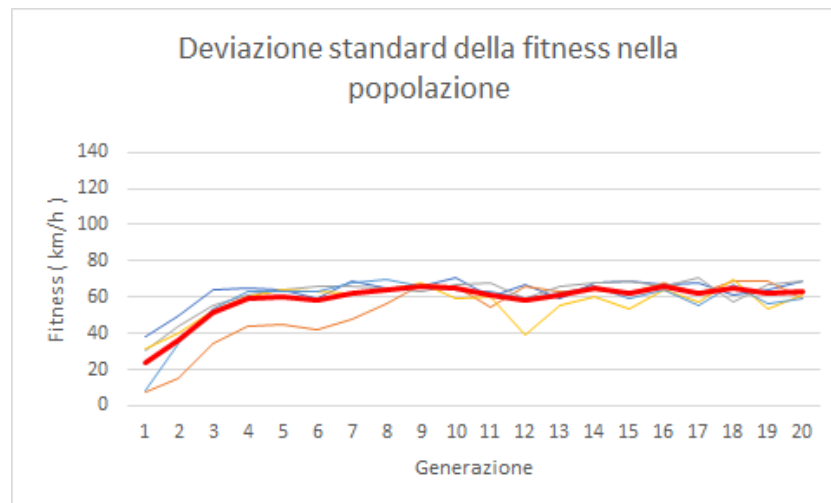


Figura 3.1: in alto: andamento della fitness del miglior genoma all'avanzare delle generazioni nelle cinque sperimentazioni fatte. In basso: andamento della fitness media della popolazione. La linea di maggiore spessore corrisponde all'andamento medio delle altre cinque linee.

Dai grafici risulta evidente che già dopo le prime dieci generazioni si giunge ad una convergenza della fitness del miglior individuo ad un valore intorno ai 150  $km/h$ . Questo valore non riesce ad essere migliorato nel tempo a disposizione, suggerendo

una stagnazione della popolazione. Analizzando, allo stesso modo, la fitness media della popolazione, mostrata nel secondo grafico, si nota una prima fase di crescita più o meno lineare seguita da una in cui il valore medio non cresce, ma sono comunque presenti forti oscillazioni e, dunque, non si ha una convergenza verso l'individuo migliore.

Per comprendere il livello di stagnazione è, quindi, necessario passare all'analisi di un parametro statistico del secondo ordine, in particolare l'andamento della deviazione standard della fitness all'interno della popolazione.



*Figura 3.2: Andamento della deviazione standard della fitness all'interno della popolazione per i cinque addestramenti effettuati. In rosso, l'andamento medio dei cinque addestramenti.*

Il grafico mostra una prima fase di aumento lineare, indice della comparsa nella popolazione di individui con fitness superiore alla media, ed in seguito una fase pressoché stabile, che suggerisce appunto una stagnazione della popolazione. In ogni caso non si ha una convergenza verso un'unica soluzione, ovvero un unico individuo. Questa convergenza, infatti, sarebbe stata individuata da una decrescita della deviazione standard nelle generazioni finali.

Un ulteriore fattore da analizzare per comprendere meglio l'andamento dell'addestramento ed una sua eventuale stagnazione è il fattore di speciazione, ovvero quante specie diverse sono presenti nella popolazione durante le varie generazioni. In accordo alla tecnica utilizzata (NEAT), la formazione di una nuova specie avviene quando un individuo nella popolazione risulta particolarmente differente dall'individuo che rappresenta maggiormente le caratteristiche della propria specie, in particolare la differenza genetica deve superare il valore di default pari a 3. Nella sperimentazione effettuata, comunque, questo parametro non può essere tenuto in considerazione, in quanto il tempo dedicato all'evoluzione della popolazione e la dimensione della stessa non hanno portato a cambiamenti topologici così sostanziosi da determinare la creazione di nuove specie. Si è, infatti, ottenuta una distanza

genetica media circa pari ad 1. Una considerazione ulteriore su questo parametro verrà fatta nel paragrafo successivo.

### 3.1 - Tuning dei parametri

In questo paragrafo, si considera il confronto tra le diverse configurazioni dei parametri nell'ottica di trovare un set di parametri migliore al fine di confrontarlo con il controllore base in termini di performance.

Come detto in fase di progettazione, i parametri su cui si è effettuato uno studio sono quelli che dovrebbero influenzare maggiormente l'andamento dell'addestramento. Come, però, risulterà evidente dai dati raccolti, non si è riusciti ad evidenziare una differenza effettiva.

Si analizzano dapprima i risultati ottenuti al variare della *population size*, di cui si riporta la media e la deviazione standard sui 5 addestramenti effettuati della miglior fitness raggiunta da un genoma in quanto, ai fini della gara automobilistica e degli obiettivi del progetto, è questo il valore da ottimizzare.

	MEDIA	STD DEV
Pop size = 10	146.96	0.897
Pop size = 20	147.63	1.348
Pop size = 30	148.28	1.634

*Tabella 3.1: Valori della media e deviazione standard della fitness del miglior genoma per i tre valori di Population size utilizzati.*

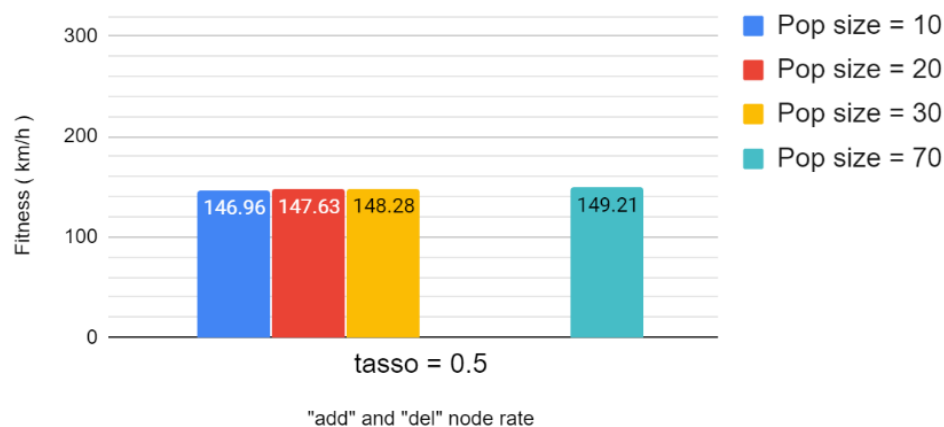
Applicando il “Friedman aligned ranks test”, con analisi post-hoc di Holm, i risultati non risultano essere statisticamente differenti nemmeno con un livello di significatività dell'analisi basso, pari al 10%. Si è, infatti, ottenuto:

Comparison	Statistic	Adjusted p-value	Result
pop30 vs pop10	1.90919	0.16871	H0 is accepted
pop20 vs pop10	1.27279	0.40618	H0 is accepted
pop30 vs pop20	0.63640	0.52452	H0 is accepted

Questo risultato viene confermato anche con l'utilizzo degli altri algoritmi di analisi post-hoc utilizzati, ovvero quelli di Nemenyi, Finner, Hockberg e Shaffer.

Questo aspetto da un lato suggerisce la necessità di considerare un aumento delle variazioni del parametro in questione, dall'altro evidenzia come, essendo il controllore sviluppato una semplice aggiunta a quello base, si raggiungerà una fitness buona dopo poche generazioni a prescindere dalla dimensione della popolazione.

Si è, dunque, eseguita un'indagine più approfondita, andando da un lato ad aumentare in maniera sostanziale il numero di generazioni per l'apprendimento mantenendo fissata la population size, dall'altro ad aumentare quest'ultima con a disposizione sempre lo stesso numero di generazioni. Di seguito si riporta il grafico ottenuto con la variazione dei valori della population size.



*Figura 3.3: Valor medio sui cinque addestramenti effettuati della fitness del miglior genoma, ottenuta al variare della population size. Il tasso indica il valore di "add" e "del" rate.*

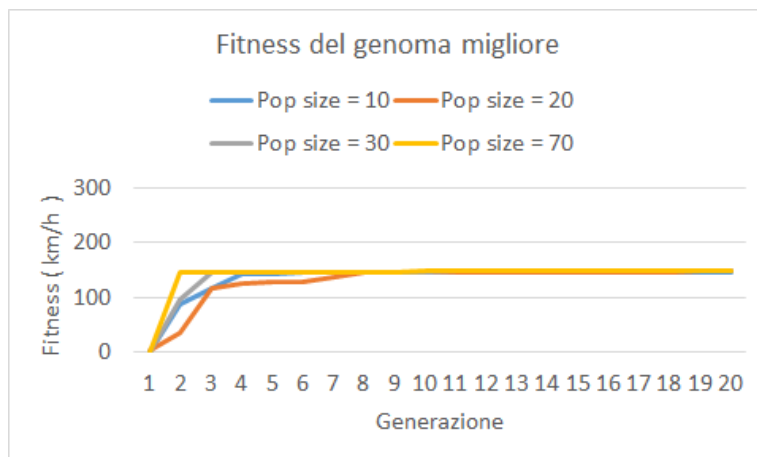
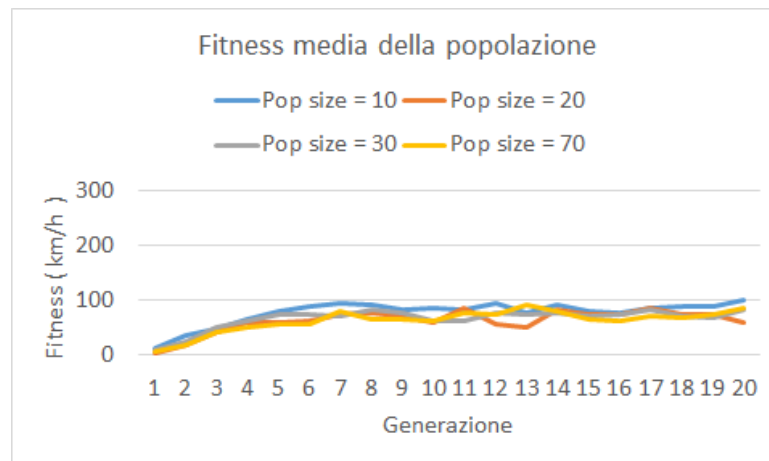
Invece, i risultati ottenuti aumentando il numero di generazioni possibili e considerando la Pop size pari a 20 e il tasso pari 0.7 sono:

	MEDIA	DEV STD
70 generazioni	149.654	2.126429872
20 generazioni	147.946	0.742266667

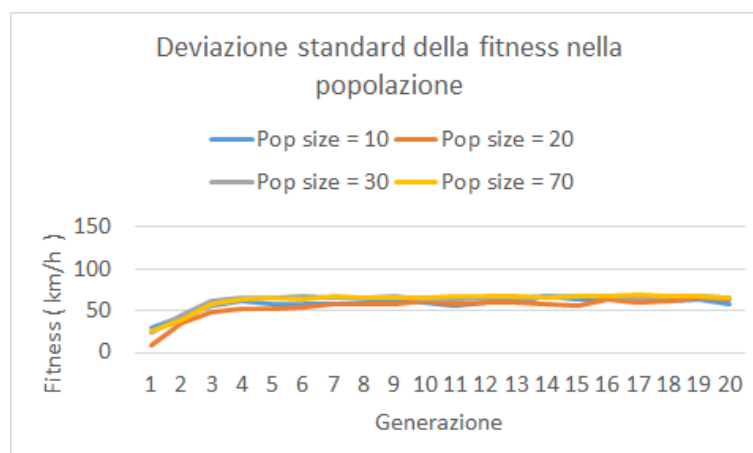
*Tabella 3.2: Valori della media e deviazione standard della fitness del miglior genoma per i due addestramenti con numero di generazioni disponibili diversi.*

Come si può notare, la variazione dei parametri a valori di gran lunga maggiori non ha portato a miglioramenti significativi dal punto di vista statistico, nonostante si sia verificato un aumento esponenziale dei tempi di addestramento. Per ottenere, quindi, una configurazione migliore dei parametri bisognerebbe, probabilmente, aumentarli ulteriormente, ma non si è ritenuto opportuno effettuare questa ulteriore sperimentazione considerando le risorse a disposizione e l'obiettivo del progetto. Avendo verificato come le varie configurazioni degli iperparametri non abbiano portato a miglorie dal punto di vista dell'individuo migliore della popolazione al termine del

training, a questo punto si passa all'analisi dell'andamento dei valori della fitness lungo le generazioni cercando di individuare quale configurazione dei parametri comporti un addestramento più rapido e, in generale, migliore.



*Figura 3.4: andamento della fitness media, della deviazione standard e del miglior genoma all'avanzare delle generazioni e a seconda del valore della dimensione della popolazione.*



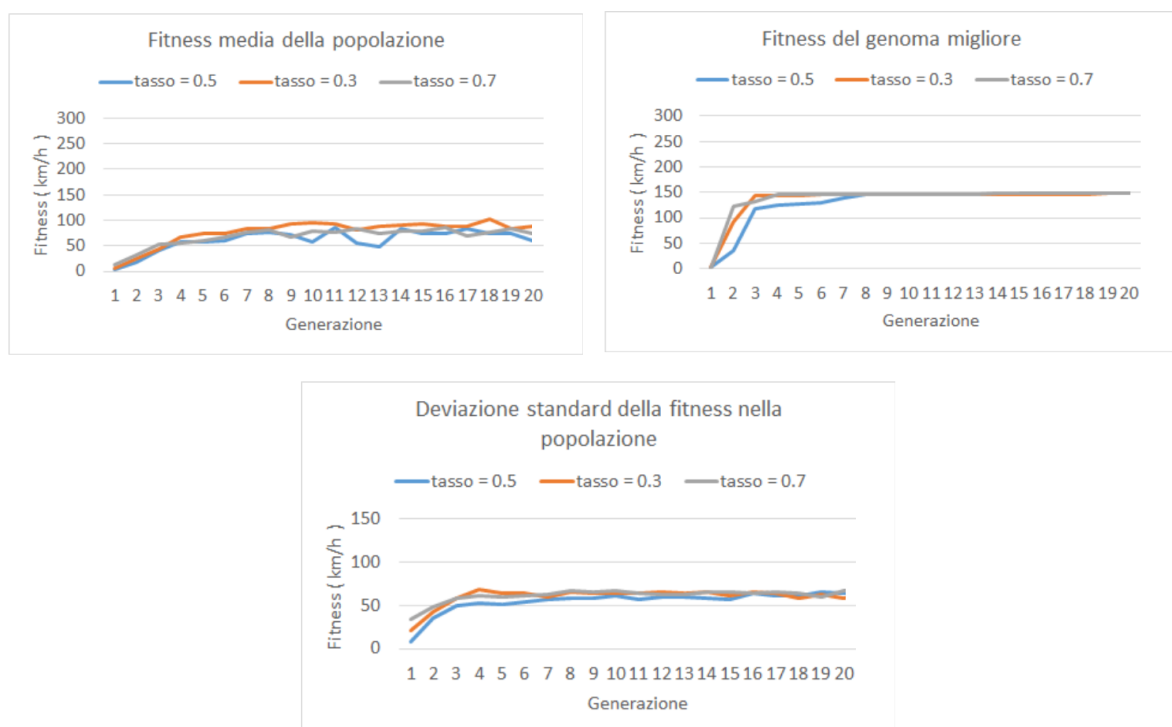
Quindi, analizzando i quattro valori della population size considerati, l'andamento medio risulta pressoché identico, ovvero si ha una fase iniziale di incremento lineare e poi un andamento stabile.

Per quanto riguarda l'andamento della fitness del genoma migliore, tutte e quattro le configurazioni convergono ben prima del numero di generazioni a disposizione. La



configurazione con pop size pari a 20 risulta andare peggio ma è un comportamento anomalo provocato da una configurazione iniziale random che si discosta particolarmente dalle altre come è visibile dalla [figura 3.1](#) in cui erano riportati tutti i cinque addestramenti. La configurazione migliore risulta essere quella con dimensione della popolazione pari a 70, che riesce immediatamente a convergere verso il valore massimo.

Discorsi analoghi si possono fare considerando la variazione dell'altro iperparametro, ovvero il tasso di aggiunta e rimozione dal nodo. Inizialmente si sono considerati valori uguali per la rimozione e l'aggiunta del nodo nella rete, ottenendo quindi delle reti bilanciate nel corso delle generazioni. Si riportano di seguito i relativi grafici.



*Figura 3.5: andamento della fitness media, della deviazione standard e del miglior genoma all'avanzare delle generazioni e a seconda del valore del tasso di aggiunta o rimozione di un nodo.*

Nel seguito, invece, verranno utilizzate combinazioni con un tasso di aggiunta del nodo più elevato rispetto a quello di rimozione. L'obiettivo nella variazione di questi parametri non è tanto ottimizzare la fitness di un singolo genoma, quanto differenziare maggiormente gli individui all'interno della popolazione ricercando l'attivazione del meccanismo di speciazione che porterebbe, poi, a miglioramenti in termini di fitness, ma solo in caso di addestramenti ben più lunghi rispetto a quelli possibili per questo progetto.

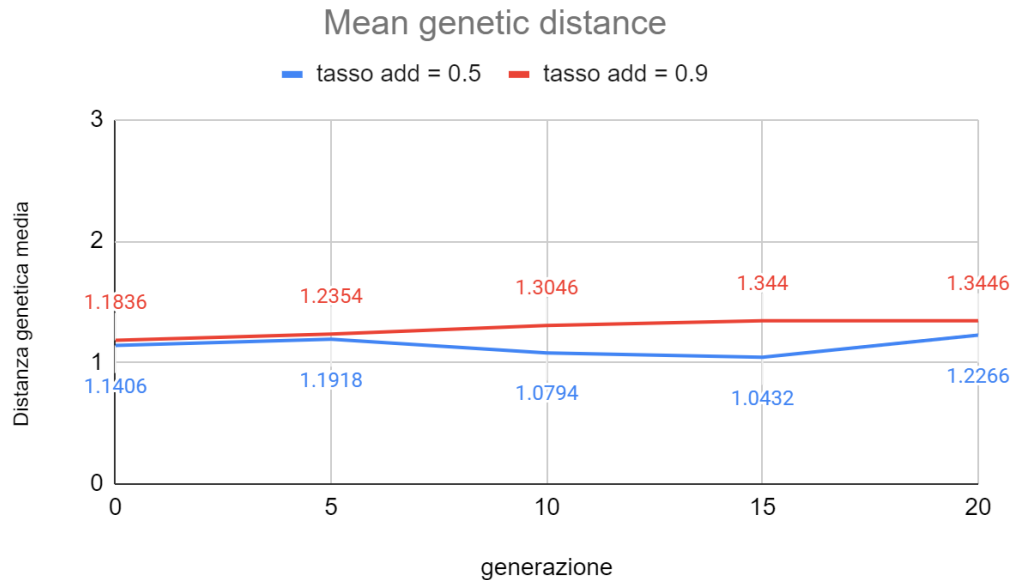
Quindi, con valori del tasso pari per l'aggiunta e la rimozione del nodo, non si hanno differenze in termini di fitness del miglior genoma (di seguito è riportata la tabella), né di specie nella popolazione. Infatti, la popolazione ha mantenuto una distanza genetica media di circa 1, con deviazione standard pari a 0.5 circa, mentre il valore di soglia per la creazione della nuova specie si ricorda essere pari a 3. Non si è, dunque, venuta a formare una nuova specie, probabilmente a causa delle poche iterazioni disponibili per l'addestramento.

	MEDIA	STD DEV
tasso = 0.3	147.85	1.535
tasso = 0.5	147.63	1.348
tasso = 0.7	147.95	0.742

*Tabella 3.3: Valori della media e deviazione standard della fitness del miglior genoma per i tre valori del tasso di aggiunta e rimozione nodo utilizzati.*

L'ultima sperimentazione effettuata ha, dunque, previsto un tasso di aggiunta nodo pari a 0.9 ed un tasso di rimozione del nodo pari a 0.5. La dimensione della popolazione scelta è pari a 30, in modo da avere una possibilità di formazione di nuove specie maggiore ed, in generale, di aumentare la distanza genetica all'interno della popolazione in modo da evidenziarne gli effetti sui risultati dell'addestramento.

Il risultato non ha, però, portato a differenze sul piano della fitness, caratterizzata dal solito andamento discusso precedentemente. Per quanto riguarda la differenziazione all'interno della popolazione, non ci sono state insorgenze di nuove specie e, quindi, si è deciso di analizzare l'andamento della distanza genetica, riportato di seguito in figura 3.6.



*Figura 3.6: andamento della media della distanza genetica all'interno della popolazione. Il grafico mostra la media sui 5 addestramenti effettuati con valori del tasso di aggiunta nodo diversi.*

Dal grafico si può osservare come un valore del tasso di aggiunta nodo maggiore rispetto a quello di rimozione porti, con l'avanzare delle generazioni, ad una crescita della distanza genetica. Con un valore uguale dei due tassi, invece, la distanza genetica tende ad oscillare sempre in uno stesso intorno, come era immaginabile. Comunque, nonostante la crescita della distanza media, il valore di soglia pari a 3 non viene superato da alcun individuo e non si ha, quindi, la formazione di nuove specie.

Nel continuo di questo documento si è ritenuto opportuno proseguire con le sperimentazioni, con l'obiettivo di confrontare le prestazioni del controllore sviluppato con quelle del controllore base, ritenendo soddisfacente l'analisi effettuata fino a questo punto sull'incidenza di ogni parametro, la quale verrà ripresa nel capitolo 5. La configurazione che si utilizzerà per il confronto appena descritto prevede una population size pari a 20 ed un tasso pari a 0.5. E' stata effettuata questa scelta dal momento che risulta essere un buon compromesso considerando i tempi di addestramento e le prestazioni ottenute.

## 3.2 - Valutazione performance

In questo paragrafo si andranno a confrontare le performance del controllore di base rispetto al controllore sviluppato. Entrambi i controllori verranno testati sui tre tracciati presentati nel secondo capitolo, e si terrà conto dei valori di maggior interesse per ogni tracciato, ovvero:

- Il tempo impiegato per effettuare due giri.
- La velocità media mantenuta durante la gara.
- La velocità massima raggiunta.
- Eventuali fuoripista con relativi danni alla vettura.

Mentre per valutare le prestazioni del controllore base è necessaria una sola simulazione in quanto il suo comportamento sul tracciato è, chiaramente, deterministico, per il controllore sviluppato è necessario considerare la media delle prestazioni del miglior genoma (ovvero il genoma con la fitness più elevata) di ognuno dei 5 diversi addestramenti effettuati.

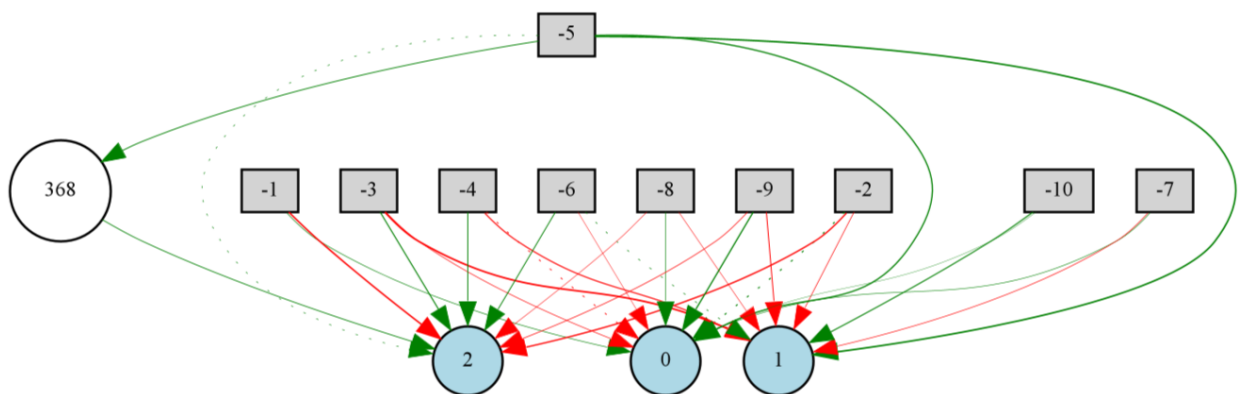
Controllore	Tracciato	Tempo impiegato (s)	Vel. media (km/h)	Vel. max (km/h)	num. fuoripista
<b>Base</b>	Ruudskogen	165	142.86	229	2
	Speedway	98	151.12	226	0
	Aalborg	176	105.83	212	20
<b>Implementato</b>	Ruudskogen	159.6	147.8482	232.8	2
	Speedway	96.66666667	153.4033333	229.3333333	0.3333333333
	Aalborg	185.5	100.38	216.5	20.25

*Tabella 3.4: Confronto dei valori ottenuti simulando una gara sui tre circuiti in analisi. Tra parentesi sono indicate le unità di misura se presenti. I risultati del controllore sviluppato sono le medie dei risultati ottenuti con gli individui migliori dei 5 addestramenti fatti.*

Risulta necessario specificare come, per quanto riguarda il controllore sviluppato, alcuni addestramenti abbiano portato ad avere una vettura che non è riuscita a completare la gara sui circuiti diversi da quello per cui è stato addestrato, a causa di eventuali fuoripista. In particolare è successo due volte su cinque per quanto riguarda il tracciato Aalborg, il che aggrava ulteriormente le prestazioni già pessime, ed una volta sul tracciato Speedway per il quale sembravano esserci prestazioni migliori. I risultati di questi tre addestramenti non sono stati considerati per il calcolo dei valori riportati nella tabella appena presentata.

Dunque, Il controllore sviluppato ottiene un netto miglioramento sul circuito Ruudskogen, per il quale è stato addestrato. Da notare, inoltre, come per il conseguimento di questo risultato non è stato necessario un aumento della complessità della rete neurale artificiale, ma anzi molti collegamenti sono stati rimossi e in generale è bastato un affinamento dei pesi.

Di seguito si mostra una delle cinque reti ottenute dai cinque addestramenti. Le altre non vengono riportate in quanto molto simili. E' stato, infatti, al massimo aggiunto un nodo, quindi i cambiamenti strutturali rispetto la rete base si traducono praticamente in una semplificazione attraverso la rimozione di alcuni collegamenti.



*Figura 3.7: Immagine della rete ottenuta. In rosso i collegamenti con peso negativo. I collegamenti tratteggiati sono geni disabilitati nell' individuo.*

Dalla figura si nota la comparsa di un nuovo nodo, identificato con chiave 368, che però è stato aggiunto da poche generazioni e, infatti, si limita a collegare il nodo di input -5 con quello di output 2.

Infine, bisogna sottolineare come il controllore sviluppato risulta peggiore sui due circuiti su cui non è stato addestrato, ovvero Aalborg e Speedway, come era ampiamente prevedibile.

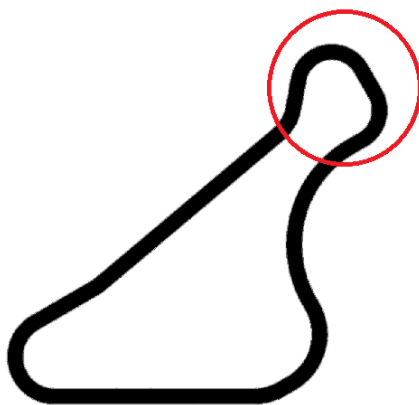
Un altro problema riscontrato è il danno subito. Entrambi i controllori hanno riportato, in seguito a diversi fuoripista, diversi danni alla vettura soprattutto sui circuiti più complessi, Aalborg e Ruudskogen.

### 3.3 - Analisi delle criticità

Come mostrato nella sezione precedente, i problemi principali del controllore sviluppato sono due:

1. scarsa capacità di gareggiare su altri circuiti oltre quello per cui è stato addestrato, ovvero si riscontrano difficoltà nel terminare i due giri sul circuito;
2. durante la gara, avvengono diversi fuoripista con conseguente danneggiamento e/o rallentamento del veicolo.

Per quanto riguarda il primo problema, si è visto come il peggioramento sul tracciato più semplice, Speedway, non è motivato dai tempi di percorrenza o in generale dalla prestazione del controllore, quanto dal fatto che uno degli addestramenti ha portato ad una soluzione (il genoma migliore) che non riesce a portare a termine la gara, ma si blocca alla prima curva leggermente più ostica del tracciato.

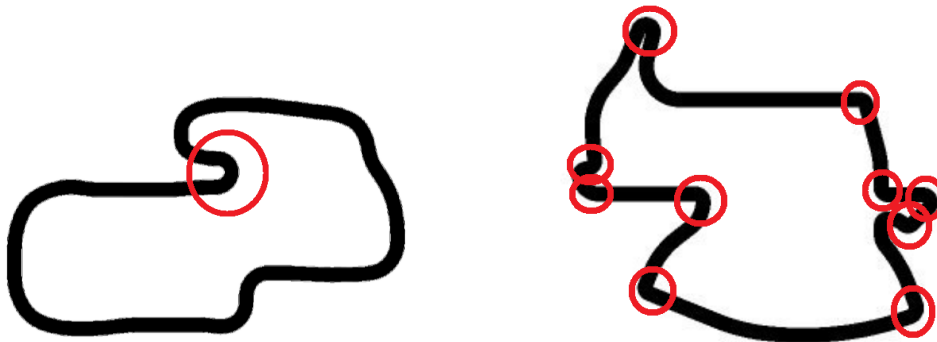


*Figura 3.8: Tracciato Speedway. In rosso cerchiata la curva del tracciato più ostica, che viene fallita dal controllore una volta su cinque.*

Come è possibile notare dall'immagine, la curva in questione è particolare perché, percorsa in senso antiorario, è composta da due curve in rapida successione che vengono subito dopo una curva larga. Il controllore può, quindi, essere portato ad accelerare molto durante la curva larga e non accorgersi subito del pericolo di avere di fronte una curva a gomito, mantenendo quindi velocità elevate come se fosse una curva normale e, quindi, tendendo ad effettuare un fuoripista nella seconda parte della curva.

Invece, sul tracciato Aalborg ci sono problemi gravi relativi principalmente alle curve particolari ed insidiose, in quanto molto spigolose, che caratterizzano il circuito. Per questo motivo, il controllore tende a bloccarsi in una di queste e a non portare nemmeno a termine la gara, oppure ad avere tempi di percorrenza peggiori e, contemporaneamente, un numero elevato di fuoripista, che si traduce in un quantitativo di danni inaccettabile.

Per meglio comprendere il secondo problema, di seguito si mostrano tutti i punti dei tracciati Aalborg e Ruudskogen in cui sia il controllore in analisi che il controllore base sono usciti fuori pista.



*Figura 3.9: A sinistra il tracciato Ruudskogen. A destra il tracciato Aalborg. I punti in cui si ha avuto un fuoripista sono cerchiati in rosso*

Si evidenzia che il controllore sviluppato commette un fuoripista anche nel tracciato Ruudskogen, per il quale è stato addestrato, in corrispondenza di una curva particolarmente difficile essendo l'unica caratterizzata da un raggio interno minore di novanta gradi.

Entrambi i problemi comportano gravi conseguenze e, dunque, nel prossimo capitolo di questo documento si proverà a porre rimedio, cercando di sviluppare un controllore migliore.

## Capitolo 4: Nuove sperimentazioni

Lo scopo di questo capitolo è quello di migliorare le performance del controllore progettato. Come si è visto nel capitolo precedente, il controllore base raggiunge risultati migliori quando si utilizza come test un circuito diverso da quello utilizzato in fase di addestramento. Ulteriore problema è che il controllore sviluppato risulta subire danni in seguito a dei fuoripista, la qual cosa va evitata se si vogliono ottenere performance migliori.

### 4.1 - Problema del danno

Per quanto riguarda la problematica del danno, e, più in generale, dei fuoripista, si è deciso di cambiare la funzione di valutazione dei genomi, ovvero la funzione di fitness. In particolare, si simula la gara per il genoma considerando la velocità media esattamente come prima, ma, in aggiunta, si considera il danno subito durante tutta la gara. In particolare, viene assegnata una fitness negativa all'individuo che ha subito danni in modo da far sì che venga scartato nelle future generazioni.

#### 4.1.1 Risultati ottenuti

Dunque, sono stati ripetuti gli addestramenti ed i relativi test e di seguito si presentano i risultati ottenuti.

Controllore	Tracciato	Tempo impiegato (s)	Vel. media (km/h)	Vel. max (km/h)	num. fuoripista	Danni subiti
<b>Base</b>	Ruudskogen	169	139.5	228	2	235
	Speedway	98	151.16	226	0	0
	Aalborg	202	92.23	206	20	9325
<b>Implementato</b>	Ruudskogen	159	148.58	230	0	0
	Speedway	//	//	//	//	//
	Aalborg	//	//	//	//	//

*Tabella 4.1: Confronto dei valori ottenuti simulando una gara sui tre circuiti in analisi. Tra parentesi sono indicate le unità di misura se presenti. I risultati del controllore sviluppato sono le medie dei risultati ottenuti con gli individui migliori dei 5 addestramenti fatti. Le righe con valori "//" indicano un non completamento del rispettivo circuito.*



Considerando il circuito su cui è stato addestrato il controllore, ovvero Ruudskogen, si noti come, oltre al fatto che i danni subiti dal nuovo controllore sono pari a zero, questo abbia mantenuto le performance dell'addestramento precedente, mostrando anzi un leggero miglioramento grazie al fatto che la macchina non perde tempo prezioso uscendo fuori pista.

Quindi, andando a scartare quegli individui con un comportamento da evitare, non solo si è ottenuto un controllore che mantiene la pista, ma inoltre si sono migliorati gli altri indici prestazionali.

Dalla tabella mostrata è possibile osservare anche come le performance del controllore base siano peggiorate considerando i danni. Infatti, avviando il server in modo che si tenga conto anche di questi ultimi, il controllore base tende a limitare la propria velocità nelle situazioni critiche, provocando, dunque, un peggioramento delle prestazioni su ogni tracciato.

Un'ultima considerazione va fatta riguardo i tracciati Speedway ed Aalborg.

Dato che il controllore sviluppato è stato addestrato unicamente su Ruudskogen, non riesce quasi mai a completare la gara sugli altri tracciati (solo un addestramento su 5 ha terminato la gara). Rimane, quindi, il problema principale di affrontare circuiti diversi da quello utilizzato in fase di addestramento.

#### 4.1.2 Analisi addestramento

Si riporta il grafico relativo all'addestramento con la funzione di fitness illustrata nel paragrafo precedente.

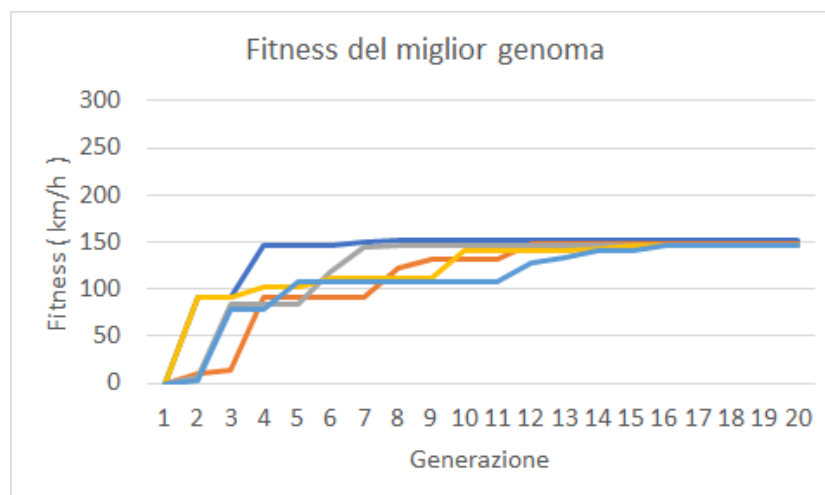


Figura 4.1: Andamento della fitness del miglior genoma all'avanzare delle generazioni nelle cinque sperimentazioni fatte

Con riferimento al grafico presente in [Figura 3.1](#), si può notare come la nuova funzione di fitness abbia portato un netto rallentamento nella convergenza verso una soluzione

stabile. Questa avviene comunque prima del tempo limite fissato di 20 generazioni e, quindi, non è risultato necessario allungare i tempi di addestramento.

Questo rallentamento potrebbe essere dovuto alla maggiore complessità necessaria per la risoluzione del problema. Complessità che, però, non si traduce necessariamente nell'esigenza di una rete più complessa, ma in un raffinamento maggiore dei collegamenti della stessa. Di seguito si riporta l'immagine di una delle cinque reti ottenute dai vari addestramenti.

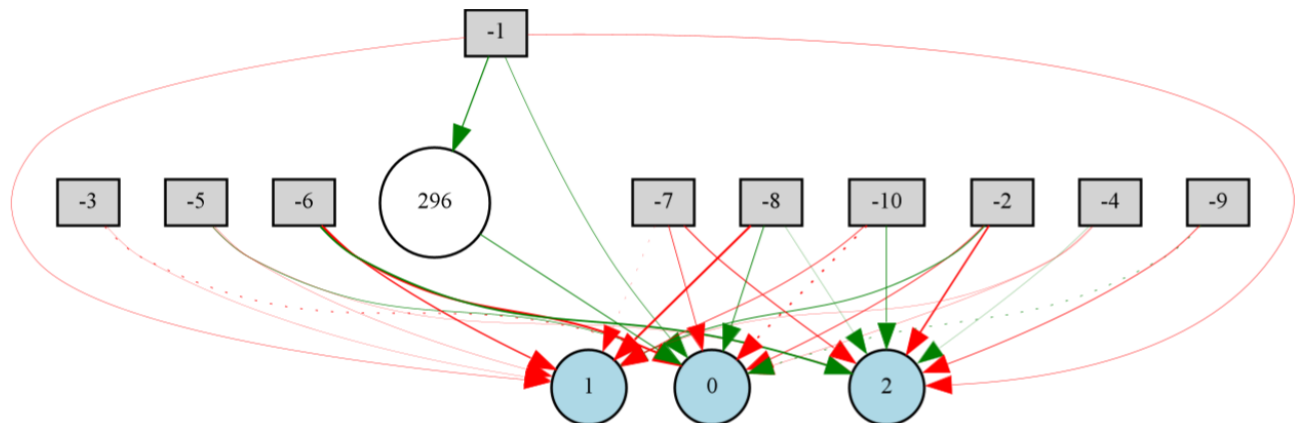


Figura 4.2: Immagine della rete ottenuta. In rosso i collegamenti con peso negativo. I collegamenti tratteggiati sono geni disabilitati nell'individuo.

Un risultato del genere era comunque prevedibile in quanto sono bastate le 20 generazioni a disposizione per ottenere la convergenza del miglior genoma. Nel prossimo paragrafo si vedrà come, addestrando il controllore su più circuiti contemporaneamente, saranno necessarie reti di maggiore complessità.

Infine, essendo questa una soluzione intermedia, che presenta ancora un problema importante da risolvere, si è deciso di non effettuare uno studio sulle variazioni degli iperparametri. Consci che probabilmente per questa tipologia di addestramento una diversa configurazione degli iperparametri avrebbe potuto migliorare la velocità di convergenza, o addirittura le prestazioni finali, si è comunque deciso di andare avanti nello sviluppo e nella progettazione provando, dunque, a migliorare le prestazioni sui circuiti Aalborg e Speedway, come si vedrà nel prossimo paragrafo.

## 4.2 - Addestramenti multipli

Per risolvere il problema rimanente del controllore sviluppato, si è pensato di addestrare il controllore contemporaneamente sui due circuiti di maggiore complessità considerati fino a questo momento, ovvero Ruudskogen ed Aalborg.

In particolare, si è adottata una funzione per la valutazione degli individui nella popolazione che simulasse la gara sia su un tracciato che sull'altro ottenendo le relative performance. Nello sviluppo di tale funzione sono state diverse scelte implementative:

- Il danno subito dal controllore sul tracciato Ruudskogen è completamente vincolante alla valutazione del genoma. Se questo danno è superiore ad una soglia minima (inizialmente posta a 100 ma poi portata a 0 in seguito ad alcune sperimentazioni), allora la fitness del genoma su questo tracciato viene azzerata.
- Il danno subito dal controllore sul tracciato Aalborg ha un peso minore in base all'istante di tempo in cui è avvenuto il primo fuoripista che ha arrecato danni alla vettura. L'idea di base è che, essendo un tracciato molto complesso, può capitare che un genoma effettui un fuoripista alla prima curva, mentre un secondo genoma all'ultima curva. Nel caso di azzeramento diretto della fitness si avrebbe una pari valutazione dei due genomi e ciò provocherebbe un azzeramento della fitness di praticamente tutti gli individui della popolazione che nelle prime generazioni subiscono inevitabilmente danni durante la gara essendo appunto il tracciato pieno di curve difficili. Dunque, la fitness non viene azzerata ma scalata di un fattore, che diminuisce all'aumentare del tempo in cui è avvenuto il primo fuoripista. In particolare, sono stati definiti 8 istanti di tempo come valori soglia, in corrispondenza delle curve particolari del circuito.

```
if danno > 0:
    if istante_primo_danno < T1:
        fitness = fitness / 10
    elif istante_primo_danno < T2:
        fitness = fitness / 9
    ...
    elif istante_primo_danno < T8:
        fitness = fitness / 2
    else:
        fitness = fitness / 1.5
```

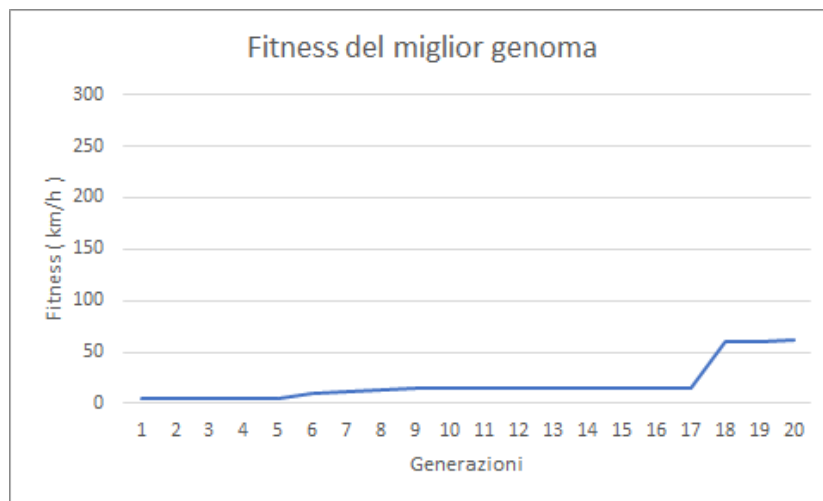
- Una volta valutate le due velocità medie sui due tracciati, si è deciso di aggiungere una ulteriore penalizzazione scalando la fitness totale, ovvero la somma delle fitness sui due tracciati, di un fattore pari a dieci qualora il controllore non superi un livello minimo di fitness su entrambi i circuiti. In questo modo si evita che la popolazione tenda a specializzarsi su un circuito trascurando completamente l'altro. In particolare il valore minimo di fitness per il tracciato Aalborg è 10, mentre per Ruudskogen è 30.

### 4.2.1 Risultati ottenuti

La configurazione dei parametri scelta è la solita, quindi:

- ☐ Population size = 20
- ☐ Add node ratio = 0.5
- ☐ Del node ratio = 0.5

Il numero di iterazioni massimo è ancora una volta fissato a 20 nonostante il fatto che in questo caso ogni generazione richiede il doppio del tempo in quanto i genomi vanno valutati su due circuiti. Di seguito si riportano i risultati ottenuti.



*Figura 4.3: Andamento della fitness del miglior genoma trovato lungo le generazioni.*

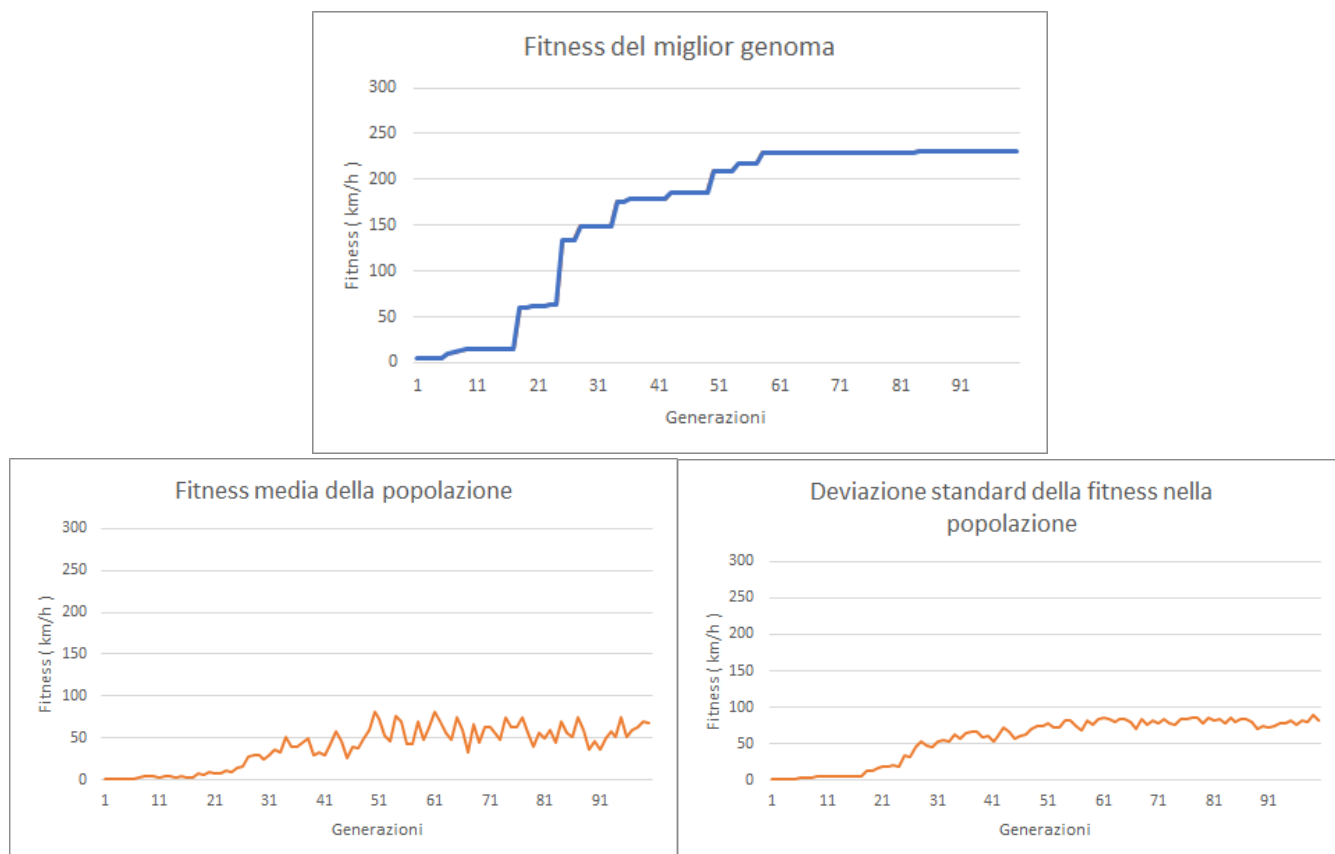
Risulta evidente lo scarso risultato raggiunto, difatti l'obiettivo è quello di ottenere risultati simili se non migliori del controllore base, il quale mantiene una velocità media di 139,5 su Ruudskogen e 92,23 su Aalborg, per un totale di 231,73 km/h. In più, si vuole una gara priva di fuoripista e conseguenti danni alla vettura.

Si nota, però, come questo risultato sia dovuto ad un numero di generazioni troppo basso. Infatti, per risolvere il problema, sarà probabilmente necessario avere genomi di maggiore complessità all'interno della popolazione, e per ottenere ciò bisogna aumentare il numero di generazioni.

Per questo motivo si è deciso di aumentare il numero di generazioni a 100. Ciò ha comportato un aumento dei tempi di addestramento che non ha permesso, purtroppo, di effettuare tutti i cinque addestramenti. Si è comunque deciso di perseguire questa strada dato che:

- 1) Come visto nel capitolo 3, i cinque addestramenti diversi tendono a differenziarsi principalmente nella fase iniziale dell'addestramento, mentre in questo contesto si è interessati quasi esclusivamente al risultato finale.
- 2) Si è preferito, più che addestrare diverse volte con la stessa configurazione dei parametri, cambiare quest'ultima in modo da apprezzare maggiormente gli effetti dei vari iperparametri, la qual cosa non è stata invece possibile nel capitolo 3.

La prima sperimentazione, con la stessa configurazione di parametri descritta in precedenza ma considerando un numero di generazioni pari a 100, ha portato ai seguenti risultati:



*Figura 4.4: in alto: andamento del miglior genoma trovato. In basso: media e deviazione standard della fitness all'interno della popolazione lungo le generazioni.*

Dai grafici, si osserva come, fissando a 100 il numero di generazioni, si riesce ad ottenere una fitness del miglior individuo buona. Il valore raggiunto è, infatti, pari a 231.2 km/h, con in particolare una fitness su Aalborg pari a 87.46 km/h e su Ruudskogen pari a 143.74 km/h.

Dal primo grafico si può notare come il valore di fitness si stabilizzi dopo circa 60 iterazioni, ed allo stesso modo la media e la varianza della popolazione si stabilizzano e tendono ad oscillare nell'intorno di un valore. Si noti inoltre che, essendo i valori della media e della deviazione standard praticamente uguali anche nelle ultime generazioni, la popolazione è formata da individui con una fitness vicina a quella massima, che vengono protetti dal meccanismo di elitismo, e da individui con fitness pressoché nulle, che probabilmente non rispettano i requisiti imposti dalla funzione di fitness utilizzata.

A questo punto vengono mostrati i risultati ottenuti sui tre circuiti.

Controllore	Tracciato	Tempo impiegato (s)	Vel. media (km/h)	Vel. max (km/h)	num. fuoripista	Danni subiti
<b>Base</b>	Ruudskogen	169	139.50	228	2	235
	Speedway	98	151.16	226	0	0
	Aalborg	202	92.23	206	20	9325
<b>Implementato</b>	Ruudskogen	164	143.74	229	0	0
	Speedway	104	142.44	226	0	0
	Aalborg	213	87.46	222	0	0

*Tabella 4.2: Confronto dei valori ottenuti simulando una gara sui tre circuiti in analisi. Tra parentesi sono indicate le unità di misura se presenti. I risultati del controllore sviluppato*

Dalla tabella si può notare come il controllore sviluppato abbia raggiunto il principale scopo di gareggiare senza uscire fuori pista su tutti e tre i tracciati proposti riuscendo, comunque, a far registrare delle prestazioni complessivamente buone. In particolare, sul tracciato Ruudskogen si hanno performance migliori, mentre su Aalborg si ottengono tempi di percorrenza maggiori rispetto al controllore base il quale, però, danneggia gravemente la vettura. Infine, sul tracciato Speedway si ha un peggioramento netto dovuto al fatto che è un circuito completamente nuovo per il controllore implementato.

#### 4.2.2 Analisi prestazioni sui singoli tracciati

A questo punto, si vuole indagare sulle motivazioni che hanno portato ad un comportamento peggiore del controllore sviluppato sul circuito Aalborg. Difatti, come si è visto precedentemente, sul circuito Ruudskogen si hanno netti miglioramenti, mentre il circuito Speedway risulta essere più ostico per il controllore sviluppato semplicemente perché quest'ultimo non è stato addestrato sul tracciato in questione. Tuttavia l'obiettivo di portare a termine la gara viene comunque soddisfatto.

Si è dunque svolta un'analisi dell'andamento della gara. In particolare, si riporta il grafico di confronto tra l'andamento della velocità dei controllori e l'andamento del valore di danno subito all'avanzare del tempo.

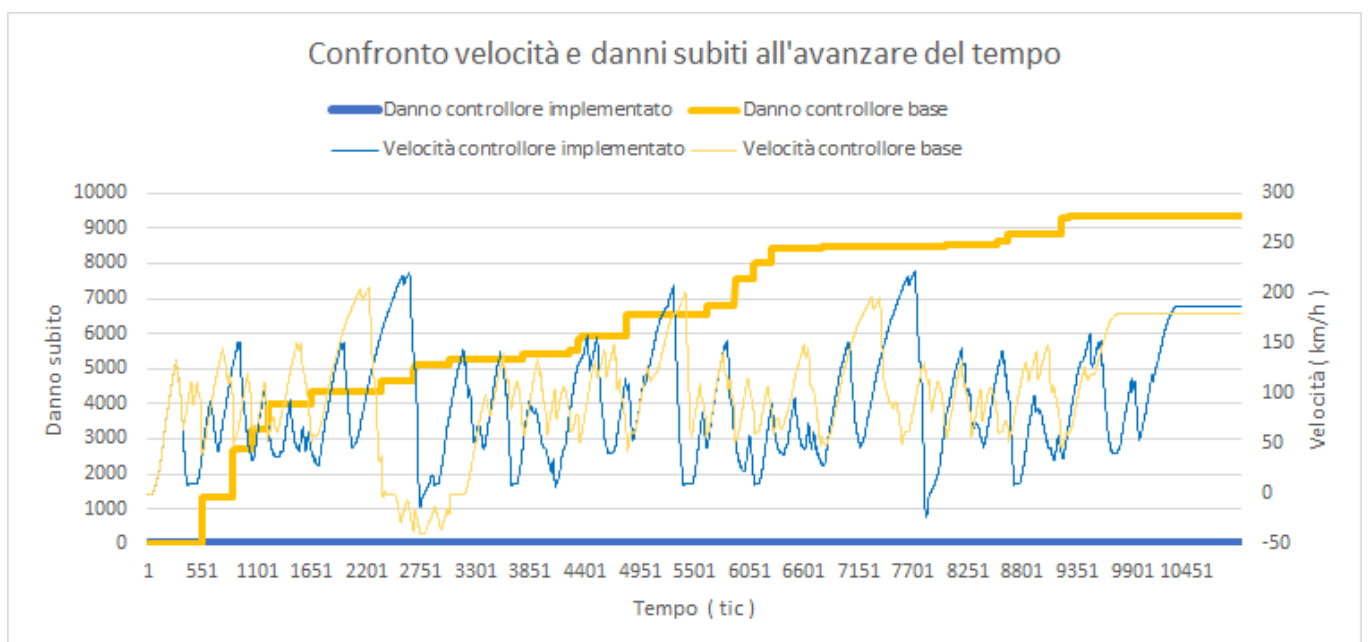


Figura 4.5: Confronto andamento velocità e danni subiti tra i due controllori. Le linee sottili rappresentano le velocità mentre quelle spesse rappresentano i danni.  
Un tic corrisponde ad un cinquantesimo di secondo.

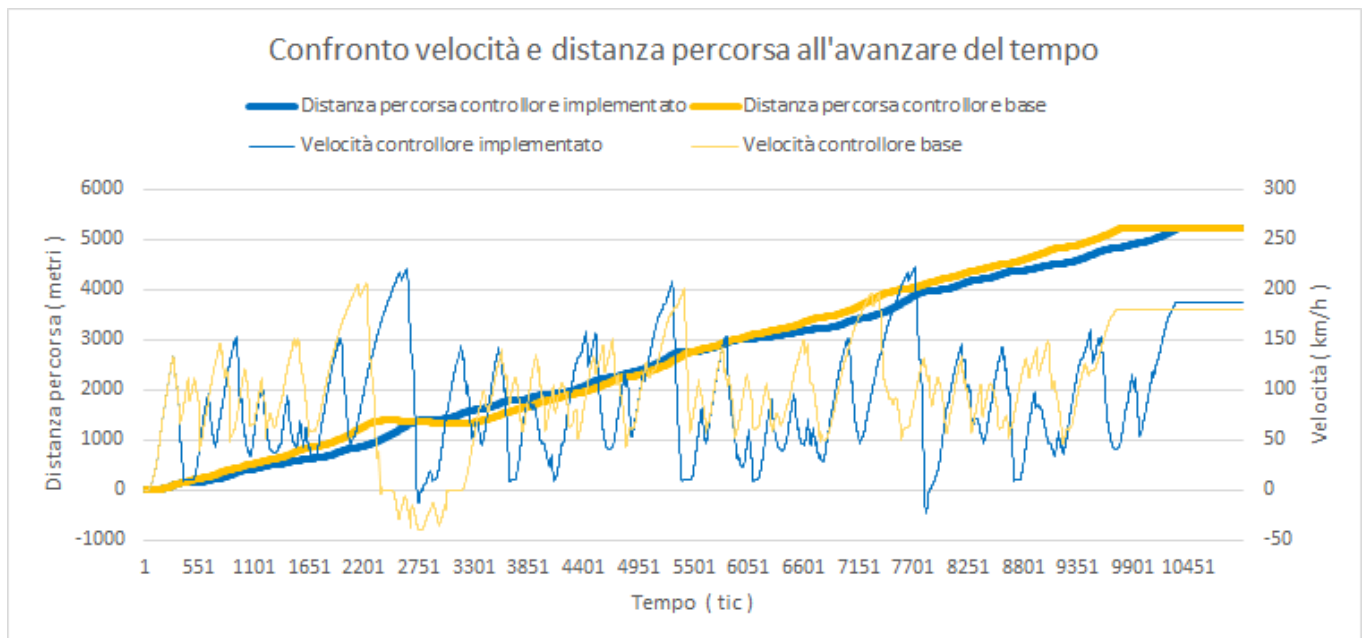
Dal grafico riportato, si nota che:

- Il danno subito dal controllore sviluppato rimane nullo per tutto il tempo della gara, a dispetto di quello subito dal controllore base
- Inizialmente i due controllori si comportano allo stesso modo. Successivamente, intorno al tic 500, si sottolinea come il controllore sviluppato rallenti maggiormente portandosi ad una velocità quasi nulla, mentre il controllore base mantiene una velocità elevata pari a circa 70 km/h. Così facendo il controllore base ha commesso un fuoripista subendo più di mille danni mentre il controllore sviluppato ha mantenuto correttamente la strada ma con una velocità media minore.
- Dopo la parte iniziale, gli andamenti delle velocità risultano più o meno sfalsati dal momento che il controllore implementato raggiunge un punto del tracciato

in tempi diversi. In particolare, il controllore sviluppato risulterà essere in ritardo fino al tic 2750 circa, per poi recuperare ed essere in anticipo fino al tic 5500 circa, dopo il quale tornerà ad essere in ritardo fino a fine gara.

- La differenza tra le velocità medie dei due controllori non è conseguenza di una velocità massima raggiunta dal controllore sviluppato minore, ma di un diverso modo di approcciare le numerose curve del tracciato, che mira a preservare l'integrità della vettura.

Come ulteriore analisi si riporta il grafico di confronto tra l'andamento della velocità dei controllori e l'andamento del valore della distanza percorsa all'avanzare del tempo.



*Figura 4.6: Confronto andamento velocità e distanza percorsa tra i due controllori. Le linee sottili rappresentano le velocità mentre quelle spesse rappresentano la distanza percorsa. Un tic corrisponde ad un cinquantesimo di secondo.*

Innanzitutto, si conferma quanto detto nel terzo punto dell'elenco precedente. In più, si nota come intorno al tic 5000 il primo giro termina con un vantaggio del controllore sviluppato, ma nel secondo giro il controllore base non commette lo stesso fuoripista avvenuto intorno al tic 2200 perché, a causa dei tanti danni subiti, si attiva un meccanismo di prevenzione dai danni, il quale permette di percorrere meglio il tracciato recuperando rispetto al controllore sviluppato. Questo meccanismo comunque non azzerà i danni subiti dalla vettura come è visibile dalla figura 4.5.

A questo punto però, è stata doverosa un'ulteriore analisi. E', infatti, noto che i danni massimi sopportabili dalla vettura utilizzata sono pari a 10000 e, dato che al termine dei due giri testati il controllore base ha raggiunto un valore di danni subiti superiore a 9000, si è pensato di simulare la gara con un ulteriore giro e, quindi, un totale di 3 giri sul circuito Aalborg. Il risultato si riporta nella seguente tabella.



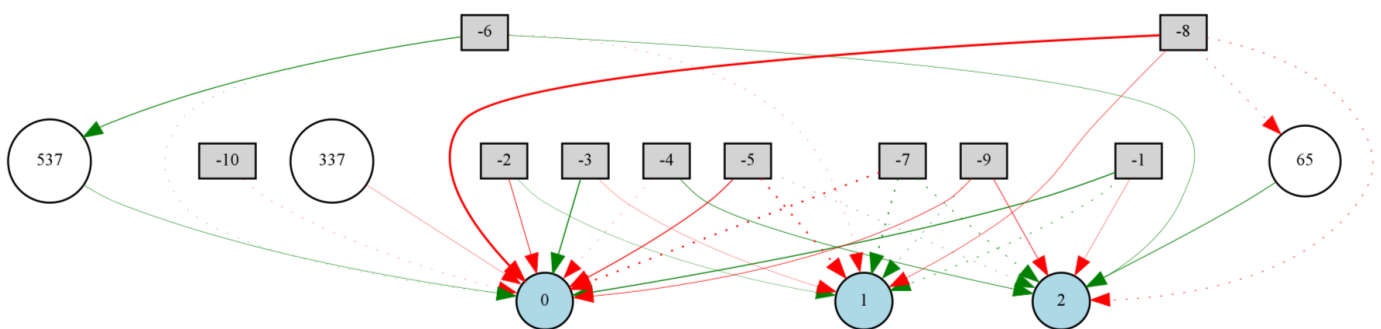
Controllore	Tracciato	Tempo impiegato (s)	Vel. media (km/h)	Vel. max (km/h)	num. fuoripista	Danni subiti
<b>Base</b>	Aalborg	//	//	//	//	10221
<b>Implementato</b>	Aalborg	316	88.42	222	0	0

*Tabella 4.3: Confronto dei valori ottenuti simulando una gara di tre giri sul circuito Aalborg. Tra parentesi sono indicate le unità di misura se presenti. Le righe con valori “//” indicano un non completamento del rispettivo circuito.*

Si evidenzia come il controllore base non riesca a portare a termine la gara dato che supera la soglia massima di danno subito. Al contrario, il controllore implementato conclude correttamente la gara senza uscire fuori pista e mantenendo una velocità media leggermente più alta rispetto a quando sono stati effettuati due giri. Concludendo, si ritiene raggiunto l’obiettivo di implementare un controllore con performance migliori di quello base anche sul tracciato Aalborg.

#### 4.2.3 Considerazioni finali

Si presenta di seguito la rete neurale artificiale rappresentante il miglior genoma di questo addestramento. Si noti come sono stati aggiunti ben tre nodi nello strato hidden, ma la struttura generale rimane molto semplice, ovvero non sono presenti molti collegamenti.



*Figura 4.5: Immagine della rete ottenuta. In rosso i collegamenti con peso negativo. I collegamenti tratteggiati sono geni disabilitati nell'individuo.*

Come si vedrà nel prossimo capitolo, scegliendo iperparametri diversi si sarebbero potute avere reti ben più complesse per ottenere le stesse performance, se non peggiori. Per questo motivo in NEAT si tende a mantenere un valore uguale per i tassi di aggiunta e rimozione di un nodo o di un collegamento ed, inoltre, per questo motivo si parte da una rete base molto piccola e minimale anche quando il task è complesso.

In conclusione, in questo capitolo si è riusciti a sviluppare un controllore che soddisfacesse i requisiti di avere performance migliori del controllore base su più tracciati, considerando complessivamente sia il danno subito che la velocità media. Ovviamente, qualora venissero presi in considerazione circuiti più complicati o molto diversi da quelli utilizzati in fase di addestramento, come nel caso di Speedway, probabilmente il controllore base otterrebbe performance migliori. In ogni caso, il prossimo capitolo non si incentrerà sul migliorare ulteriormente le prestazioni, quanto sul variare alcuni parametri chiave per valutare poi il cambiamento effettivo sul comportamento del controllore al termine dell'addestramento.

## Capitolo 5: Variazione iperparametri

Partendo dal controllore sviluppato nel capitolo precedente, in questo capitolo si vogliono variare alcuni iperparametri in modo da apprezzare gli effetti degli stessi.

Si riporta la configurazione dei parametri utilizzata:

- ☐ Population size = 20
- ☐ Add node ratio = 0.5
- ☐ Del node ratio = 0.5
- ☐ Numero generazioni = 100

Dal capitolo precedente, si può notare che le variazioni sostanziali si hanno nelle prime 50/60 generazioni e, dunque, al fine di evitare incrementi dei tempi di esecuzione, si è considerato un numero di generazioni massimo pari a 60. Inoltre, si sono fatti variare i tassi relativi all'aggiunta e alla rimozione del nodo, la dimensione della popolazione ed altri parametri non citati, che nella configurazione precedente erano:

- ☐ *Elitismo* = 2: numero di individui, migliori in termini di fitness, da mantenere invariati nel corso delle generazioni.
- ☐ *Compatibility threshold* = 3: valore di soglia della distanza genetica tra due individui per determinare se appartengono a specie diverse.

Essendo che ogni addestramento con questi parametri ha richiesto circa 7 ore come tempo di esecuzione, si è deciso di considerare solo tre diverse inizializzazioni per ogni configurazione dei parametri. Ciò non permette di effettuare uno studio adeguatamente approfondito, però consente di avere un'idea base degli effetti di ogni iperparametro.

Di seguito, in figura 5.1, si riporta il grafico dell'andamento della fitness del miglior genoma visto nel capitolo precedente, troncato alla sessantesima generazione, utile per confrontare al meglio gli andamenti dei grafici successivi.

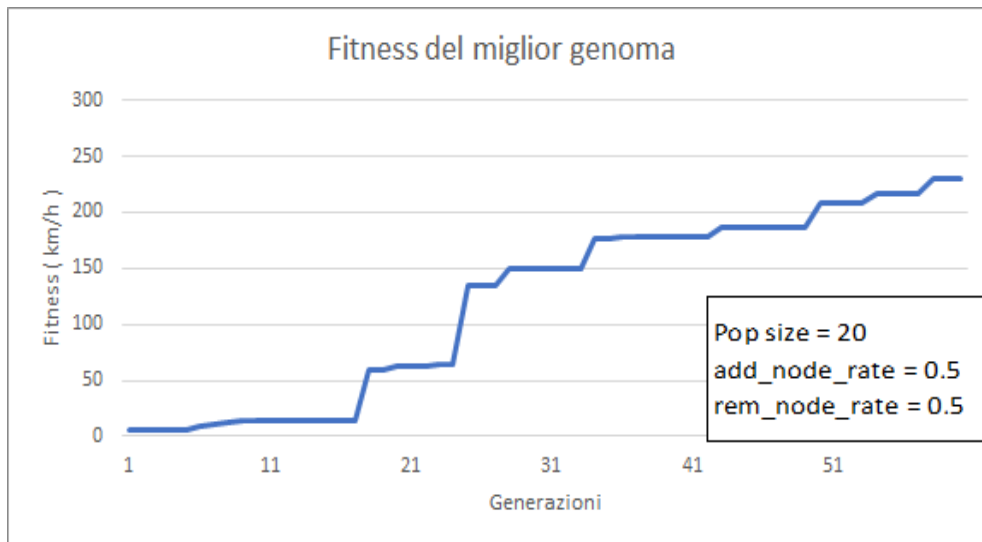


Figura 5.1: Andamento della fitness del miglior genoma nelle prime 60 generazioni dell'addestramento illustrato nel paragrafo 4.2.

Si consideri ora la variazione della population size a 50. Come si è visto nel terzo capitolo, una variazione di questo iperparametro non ha portato cambiamenti significativi nelle performance ottenute, ma l'andamento era comunque crescente al crescere del valore considerato e, dunque, ci si aspetta un miglioramento anche in questa nuova sperimentazione.

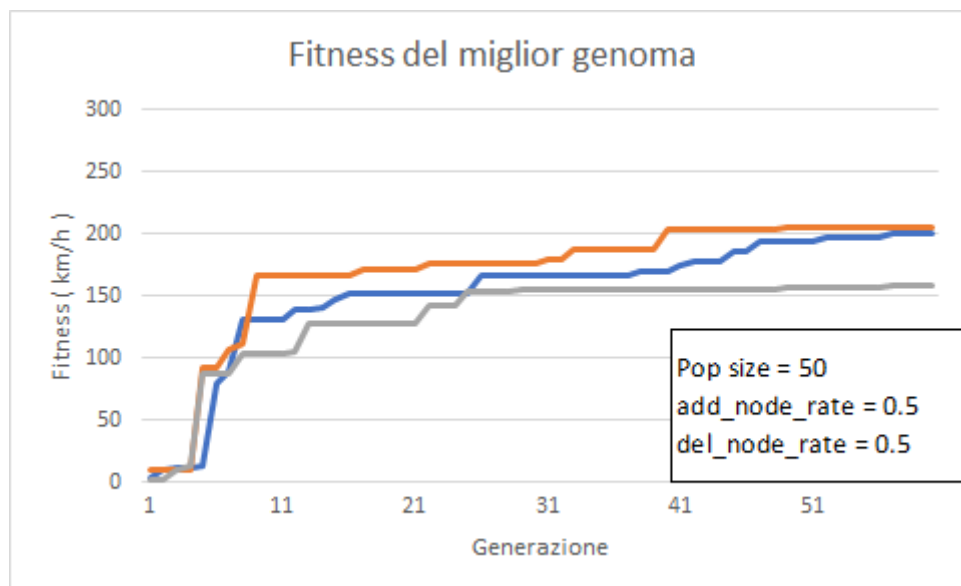


Figura 5.2: Andamento della fitness del miglior genoma dei tre addestramenti effettuati nelle prime 60 generazioni dell'addestramento con configurazione dei parametri mostrata in figura.

Come si nota dal grafico, si ha una crescita della fitness del miglior genoma molto più rapida nelle fasi iniziali dell'addestramento, invece nella seconda parte si nota nuovamente un andamento piatto. Allo stesso modo, i grafici della media e della deviazione standard della fitness all'interno della popolazione non vengono riportati,

in quanto mantengono il solito andamento, con una crescita quasi lineare all'inizio e una stabilizzazione a circa metà addestramento.

Proseguendo con le sperimentazioni, essendo il task molto complesso, si è provata una configurazione del tasso di aggiunta e rimozione nodo che andasse a aumentare la complessità della rete, in particolare si è scelto 0.1 per il tasso di rimozione al posto di 0.5, mentre il tasso di aggiunta è stato lasciato invariato a 0.5.

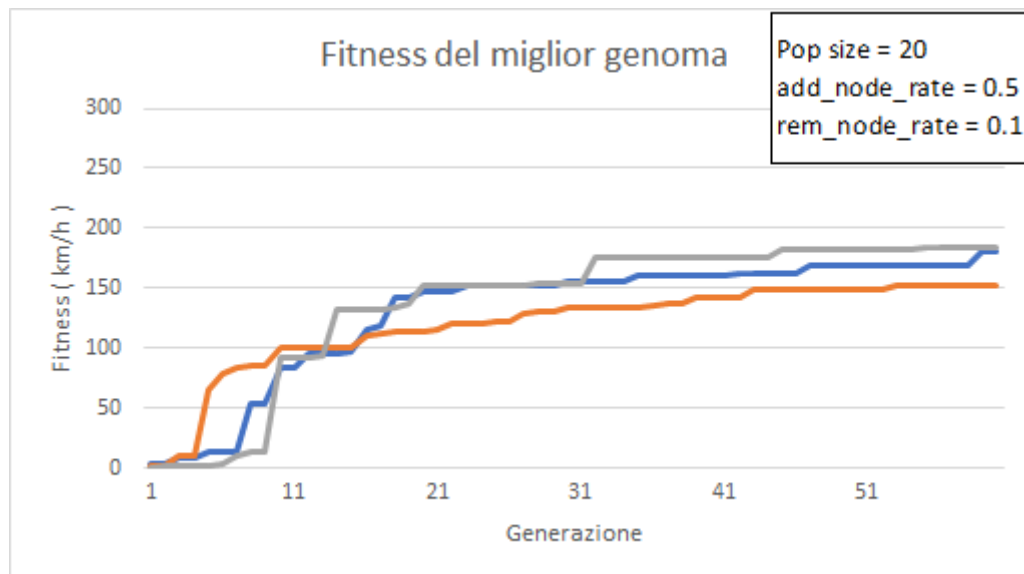


Figura 5.3: Andamento della fitness del miglior genoma dei tre addestramenti effettuati nelle prime 60 generazioni dell'addestramento con configurazione dei parametri mostrata in figura.

Dalla figura si nota subito un peggioramento delle prestazioni rispetto al caso base in cui i due tassi erano uguali. L'idea era però quella di ottenere una rete complessa e, infatti, il miglior genoma dell'ultima generazione della sperimentazione rappresentata dalla linea blu nel grafico, consiste in una rete con 12 neuroni aggiunti mostrata di seguito in figura 5.4.

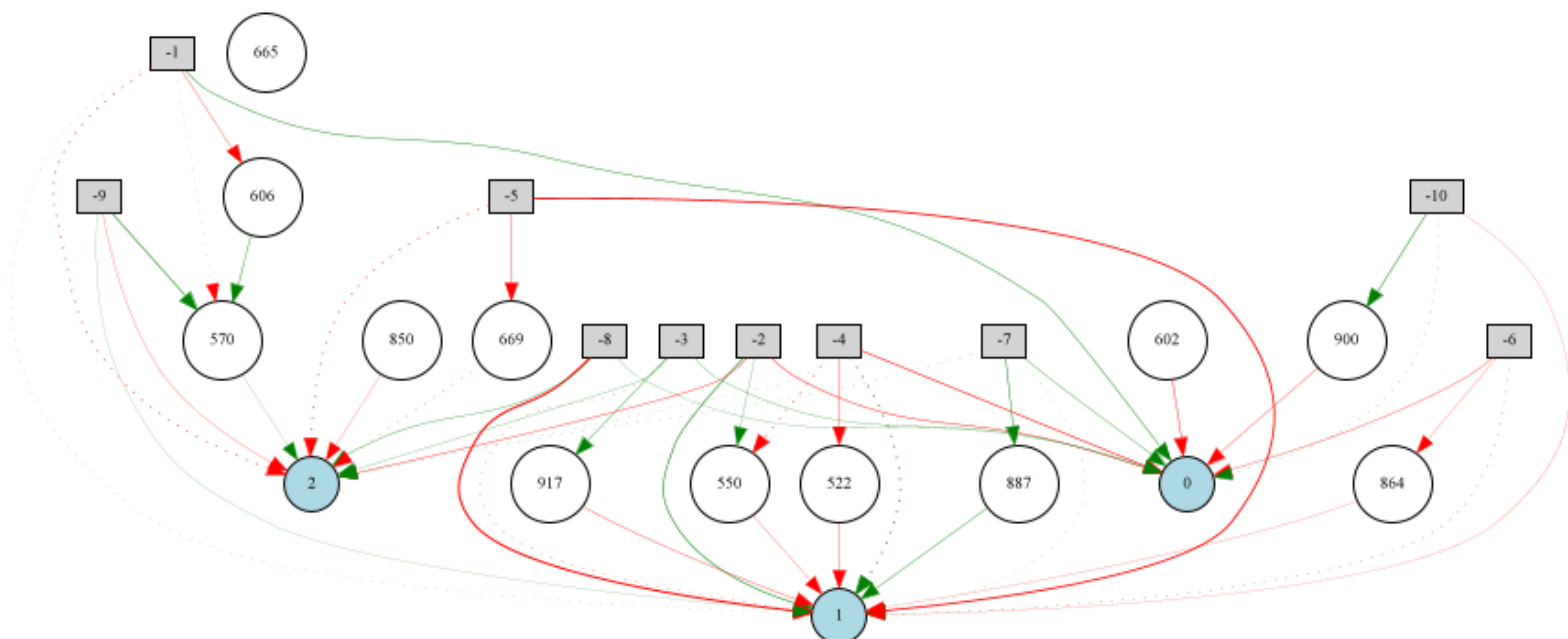


Figura 5.4: Immagine della rete ottenuta. In rosso i collegamenti con peso negativo. I collegamenti tratteggiati sono geni disabilitati nell'individuo.

Da questa sperimentazione si nota, quindi, come una rete decisamente più grande di quella ottenuta alla fine del capitolo 4 abbia prestazioni peggiori perché non risulta correttamente ottimizzata.

Inoltre, non c'è stata la formazione di nuove specie a causa probabilmente di una soglia di compatibilità troppo alta. Dunque, si è deciso di ritentare la stessa configurazione con una Compatibility threshold pari a 2.

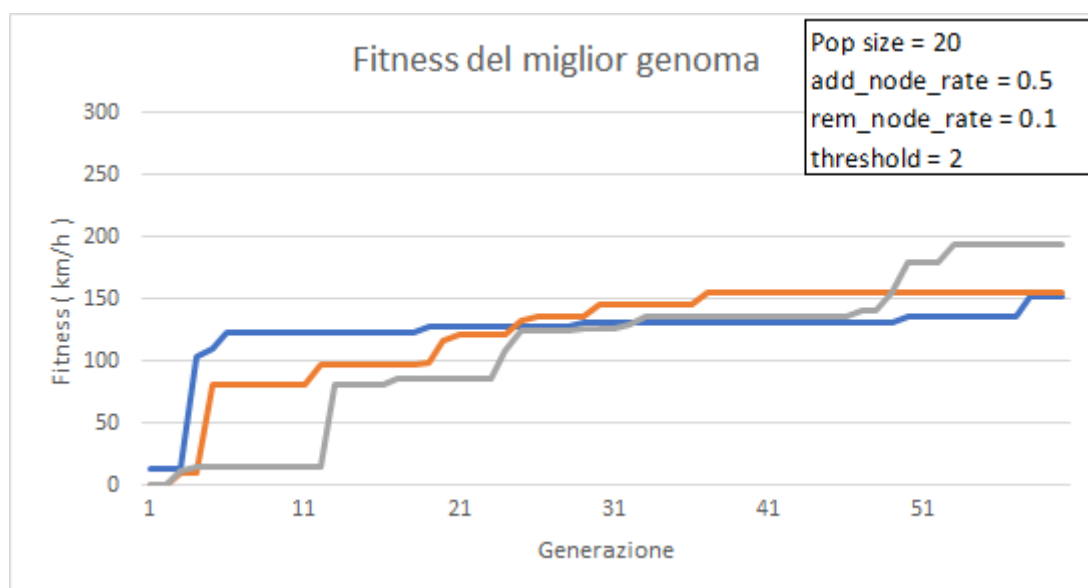
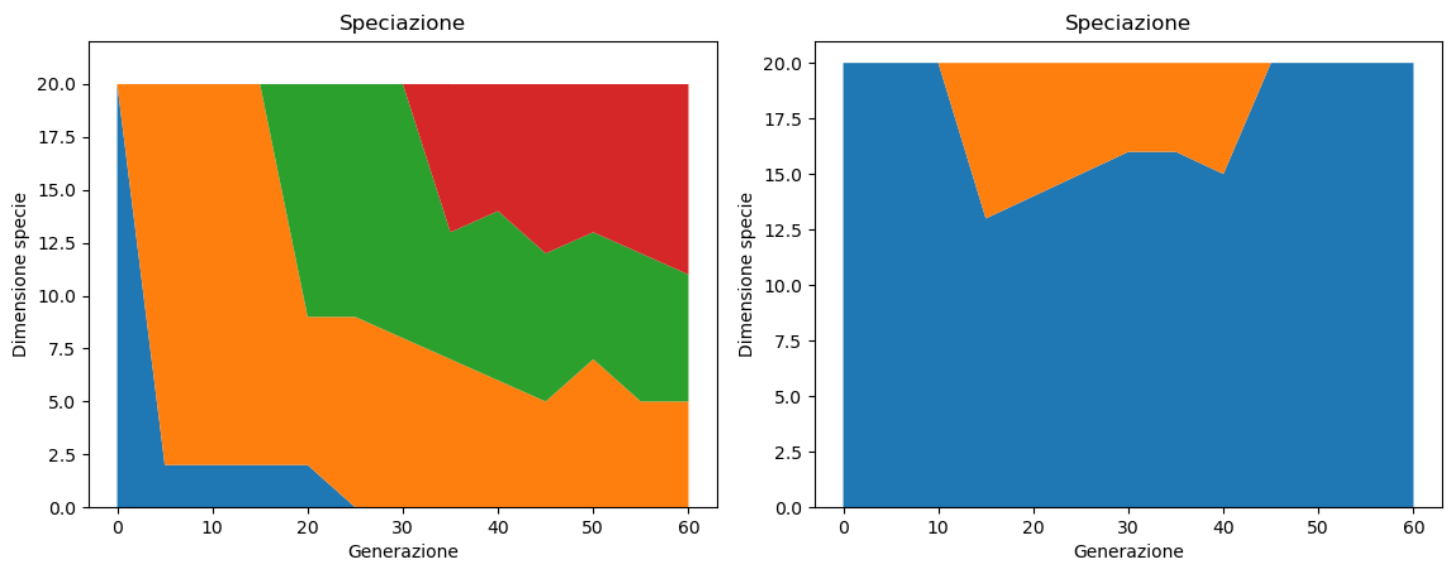


Figura 5.5: Andamento della fitness del miglior genoma dei tre addestramenti effettuati nelle prime 60 generazioni dell'addestramento con configurazione dei parametri mostrata in figura.

Il risultato in termini di fitness del miglior genoma è molto simile, nonostante si sia verificata la formazione di nuove specie. Si deduce, quindi, che l'effetto della

speciazione in NEAT è apprezzabile solo con popolazioni di individui molto ampie e con un numero di generazioni a disposizione maggiore, in modo che ogni specie sia formata da un numero adeguato di individui e possa evolvere al meglio.



*Figura 5.6: Andamento della dimensione di ogni specie identificate da colori differenti all'avanzare delle generazioni.*

Dei tre addestramenti effettuati, vengono mostrati solo i due grafici di speciazione più rappresentativi (figura 5.6). Si nota, infatti, che nella prima figura si arriva ad una formazione di 4 specie diverse, in particolare la specie iniziale tende ad estinguersi dopo poche generazioni mentre le altre tre perdurano fino alla fine dell'addestramento. Probabilmente, essendo la soglia di compatibilità molto bassa, ci sono state subito delle mutazioni strutturali che hanno portato alla creazione della seconda specie. Dal momento che il rapporto tra il tasso di aggiunta nodo e il tasso di rimozione risulta essere sbilanciato, il 90% della popolazione è risultato appartenere alla nuova specie creata nell'arco di 5 generazioni. Inoltre, si noti come la comparsa della terza specie ha causato l'estinzione della prima ed, infine, si noti come, data la comparsa di una nuova specie, la crescita della dimensione della stessa è molto rapida nelle prime generazioni successive alla sua formazione.

Dal secondo grafico, invece, si nota come nel secondo addestramento non ci siano state mutazioni di forte impatto, difatti è presente la comparsa di una seconda specie, ma questa si estingue nell'arco di 30-40 generazioni non andando mai a rappresentare la maggior parte della popolazione. L'avvenuta estinzione suggerisce che la seconda specie non ha portato benefici in termini di fitness. In particolare, essendo l'elitismo pari a 2, i due migliori genomi della popolazione sono appartenuti entrambi alla prima specie.

Una volta appurato che lo sbilanciamento del tasso di aggiunta o rimozione nodo non comportasse miglioramenti in termini di fitness, si è deciso di mantenere lo stesso

valore per i due tassi ma al posto di utilizzare 0.5 si è scelto un valore più basso, pari a 0.2. Di seguito si riportano i risultati ottenuti.

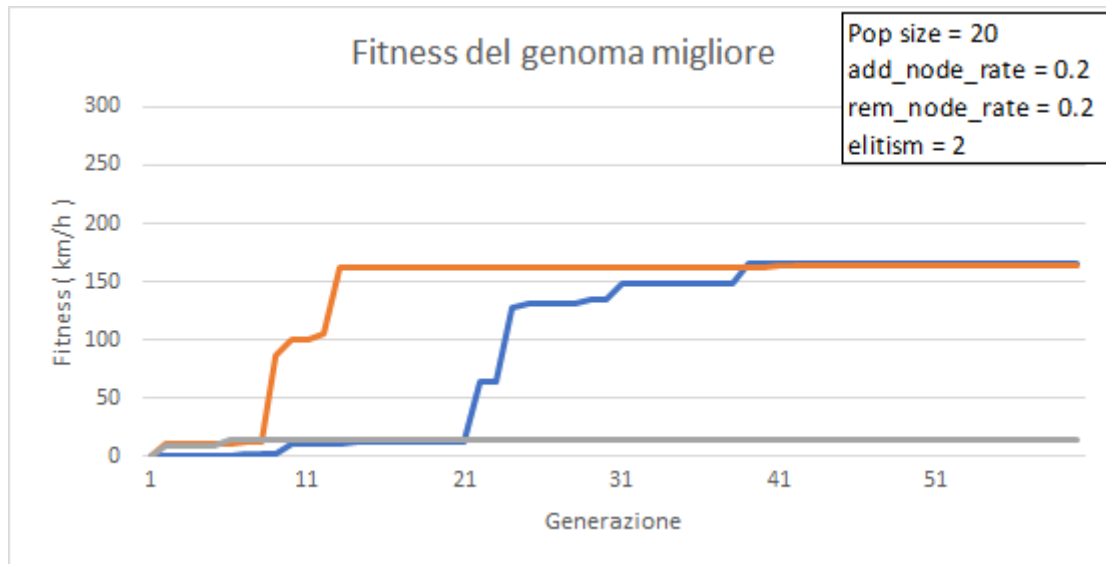


Figura 5.7: Andamento della fitness del miglior genoma dei tre addestramenti effettuati nelle prime 60 generazioni dell'addestramento con configurazione dei parametri mostrata in figura.

Dal grafico si evince che nessuno dei tre addestramenti fatti ha avuto un andamento migliore rispetto alla configurazione con tasso di aggiunta e rimozione nodo pari a 0.5, il quale aveva un valore di fitness superiore a 200 nelle 60 generazioni. Inoltre, uno dei tre addestramenti, riportato in grigio nel grafico, non è riuscito a soddisfare i requisiti minimi imposti dal calcolo della fitness per evitare che questa venga scalata di un fattore pari a 10, come illustrato nel [paragrafo 4.2](#).

Un' ultima analisi è stata invece condotta sul valore dell'elitismo. Questo è stato mantenuto pari a 2 in ottica di preservare, tra una generazione e l'altra, solo i due migliori individui che poi attraverso la riproduzione potessero migliorare la fitness della popolazione. Si è dunque provato ad aumentare questo valore a 4, ma il risultato, mostrato nella pagina successiva, non ha portato a cambiamenti netti rispetto ai precedenti.



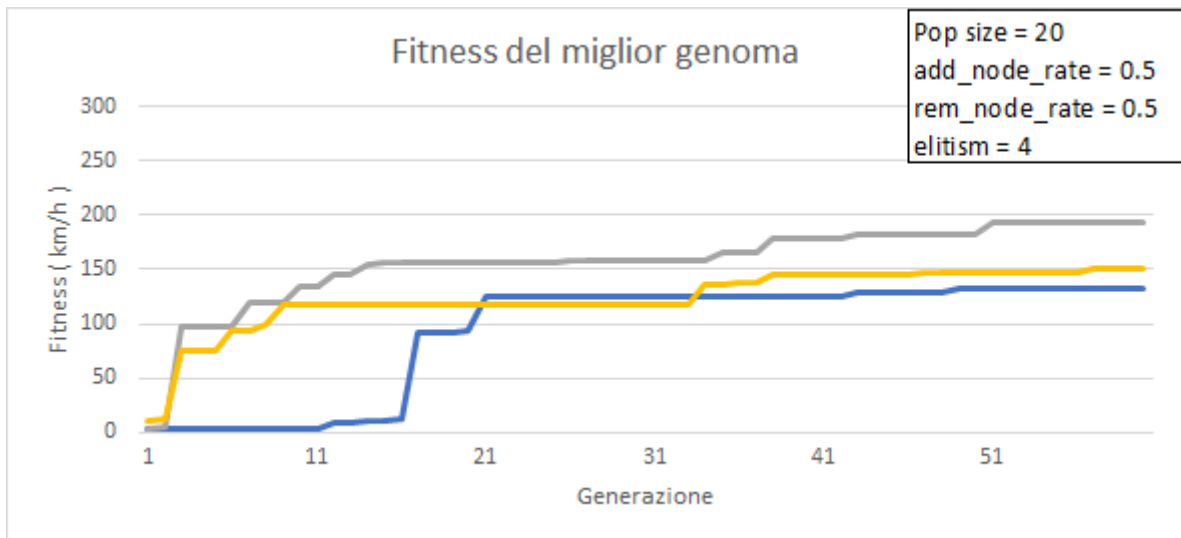


Figura 5.8: Andamento della fitness del miglior genoma dei tre addestramenti effettuati nelle prime 60 generazioni dell'addestramento con configurazione dei parametri mostrata in figura.

In conclusione, in questo capitolo, si è riusciti ad apprezzare l'impatto delle variazioni dei parametri di NEAT sull'addestramento e sulle prestazioni finali. In particolare, si è visto come una popolazione di dimensioni maggiori abbia portato ad un addestramento che raggiunge buoni valori di fitness in meno generazioni, mentre per apprezzare miglioramenti sul valore finale sarebbero necessari addestramenti ben più lunghi. Per quanto riguarda invece i valori dei tassi di rimozione e aggiunta nodo, si è visto come provare ad ottenere reti maggiormente complesse non ha portato benefici nel numero di generazioni disponibili essendo incapace di ottimizzare i pesi dei collegamenti tra i neuroni, e quindi, non è stato necessario eseguire sperimentazioni aggiuntive per ottenere un controllore con prestazioni ancora migliori.

# Conclusioni

Traendo le conclusioni, il progetto descritto in questo elaborato ha permesso di apprezzare al meglio le tecniche di computazione naturale utilizzate. Attraverso NEAT è stata fatta evolvere una popolazione di reti neurali artificiali al fine di risolvere il task di “reinforcement learning”. In particolare, è stato possibile apprezzare come un task così complesso come il training contemporaneo su circuiti diversi è stato risolto con una rete molto semplice, che non è stata predeterminata e che è semplicemente risultata la migliore in seguito all'evoluzione.

Risulta, inoltre, opportuno sottolineare la necessità di una accurata interpretazione ed astrazione del problema. Questa è, infatti, risultata una fase importantissima in cui sono stati scelti i parametri di input e output della rete, ovvero la sua struttura base, e in che modo integrarla con il controllore base considerato. Un approccio diverso basato sulla semplice applicazione di tecniche standard al problema, senza uno studio preliminare dello stesso, sarebbe stato senza dubbio fallace, determinando scarse performance finali e/o tempi di addestramento insostenibili.

Il controllore descritto in questo documento raggiunge gli obiettivi proposti. Esso risulta guadagnare secondi sul tracciato Ruudskogen rispetto al controllore base e risulta mantenere sempre la pista a prescindere dal circuito testato, evitando quindi di provocare elevati danni alla vettura sui diversi tracciati. Ovviamente non è stata trovata una soluzione generica per ogni circuito, le prestazioni dipendono fortemente dal circuito per il quale il controllore è stato addestrato.

Un miglioramento possibile consiste proprio nell'addestrare il controllore su più circuiti con caratteristiche diverse. In questo modo, sarebbe possibile ottenere una rete neurale artificiale abbastanza complessa da garantire performance migliori per ogni tracciato. Purtroppo, però, un approccio di questo tipo comporterebbe tempi di addestramento insostenibili per quelli che sono gli obiettivi di questo progetto.

Un ulteriore miglioramento consisterebbe nell'effettuare un tuning degli iperparametri più approfondito. Per motivi legati strettamente alle risorse limitate a disposizione, infatti, in questo documento non è stato possibile considerare un numero adeguato di inizializzazioni casuali diverse a seconda delle varie configurazioni degli iperparametri, ma nemmeno far variare in maniera più raffinata e precisa le configurazioni stesse al fine di ottenere un panorama completo degli effetti provocati dalle variazioni degli iperparametri.

Concludendo si ritengono i risultati ottenuti soddisfacenti rispetto agli obiettivi prefissati, seppur migliorabili nei modi descritti pocanzi.