PROCESS MINING AND PROCESS DISCOVERY SUI PROCESSI DI PROGRAMMAZIONE DI ROBOT LEGO

Progetto per il corso di Big Data Analytics e Machine Learning

Studenti

Michele Pasqualini (mat. 1101226) Denil Nicolosi (mat. 1100331)





Indice

- Introduzione
- Preprocessing
- Process discovery
- Risultati dell'analisi
- Conclusioni





In questo progetto ci si è posti l'obiettivo di analizzare la metodologia di programmazione di robot di marca LEGO da degli studenti di diverse scuole e di diverso ordine e grado.

I ragazzi dovevano programmare il robot in modo da svolgere due esercizi:

- Percorrere la distanza di esattamente un metro dal punto iniziale
- Raggiungere una determinata distanza da un obiettivo sfruttando un sensore di distanza ad ultrasuoni



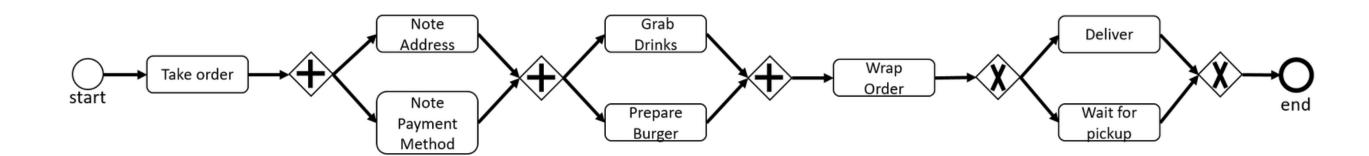
Obiettivo dell'analisi



L'obiettivo dell'analisi è quello di intercettare dei pattern significativi nelle esecuzioni dei diversi gruppi, quindi confrontare i comportamenti caratteristici di chi raggiunge l'obiettivo con successo e chi no.

Per fare questo, vengono utilizzate tecniche di process discovery a partire dal log delle operazioni di programmazione eseguite.

Si vuole quindi ottenere dei modelli di processo in grado di descrivere le attività di programmazione eseguite.



Preprocessing



Dataset

Log contenente lo stato di programmazione del robot per ogni tentativo

Confronto

confronta ogni singola esecuzione con la successiva, separate dal comando "STOP PROGRAM"

Esecuzioni uguali

non è stata eseguita nessuna variazione sulla programmazione del robot, ma solo un nuovo tentativo

Esecuzioni differenti

Il gruppo ha modificato la programmazione del robot

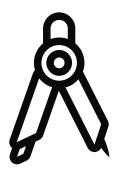
Identificazione cambiamenti

blocco aggiunto o rimosso, aumento o dimuzione dei parametri

Preprocessing

Operazioni principali

- Rimozione di tutti i cicli e parametro condition
- Aggiunta attività fittizie
- Rimozione dei blocchi duplicati di Ultrasonic Sensor Compare
- Calcolo di un punteggio per trovare esecuzioni con istruzioni simili
- Calcolo delle differenze tra i parametri
- Aggiunta di un timestamp
- Divisione dell'output





Sono state utilizzate due metodologie differenti per ottenere i modelli:

Utilizzando la libreria python **PM4PY** e l'algoritmo **Inductive Miner**



Utilizzando il software **Disco** e
 l'algoritmo **Fuzzy Miner**



6

PM4PY

Dataset

lettura del file csv e formattazione del dataframe secondo lo standard di pm4py

Event Log

creazione del file xes di log

Inductive miner

realizzazione di 9 modelli di processo diversi con incrementi di 0.1 sul parametro "NOISE THRESHOLD"

Metriche

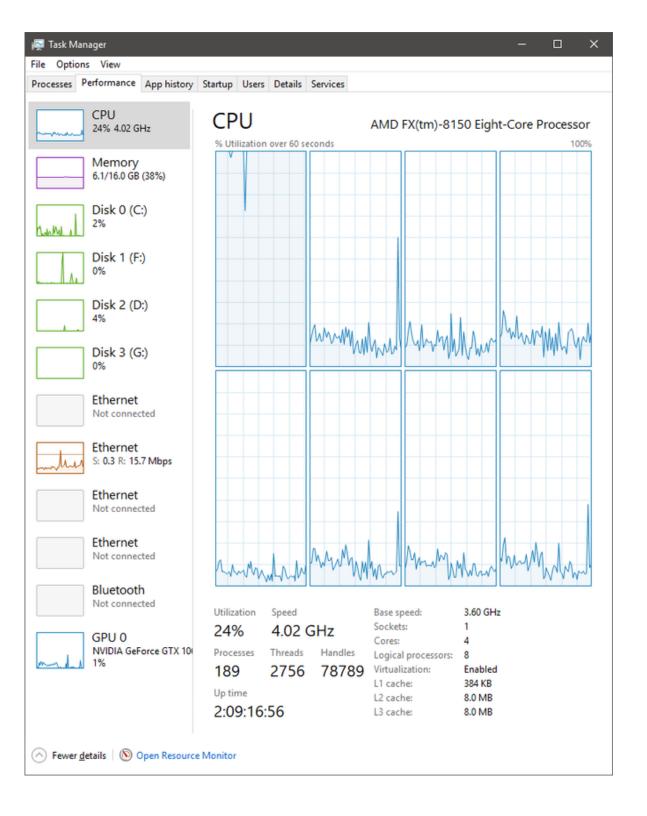
calcolo delle metriche di fitness, precision, generalization, simplicity e sound

60

PM4PY

Metriche

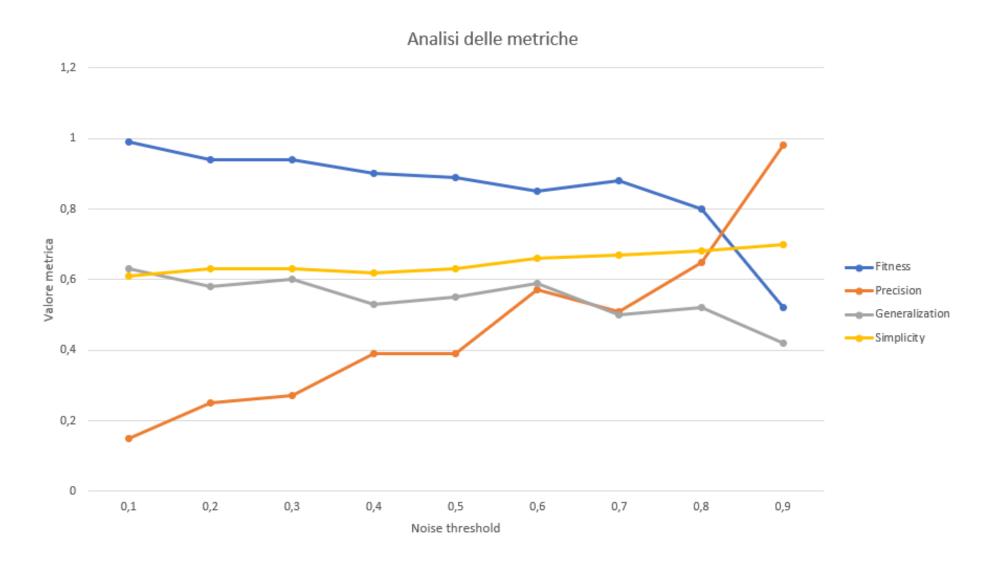
Considerato il calcolo della fitness molto impegnativo, per migliorare le performance è stato implementato il multithreading che ci ha permesso di elaborare le metriche in tempi molto più rapidi, ottenendo una velocità di esecuzione quasi 5 volte superiore rispetto all'esecuzione single-thread





PM4PY

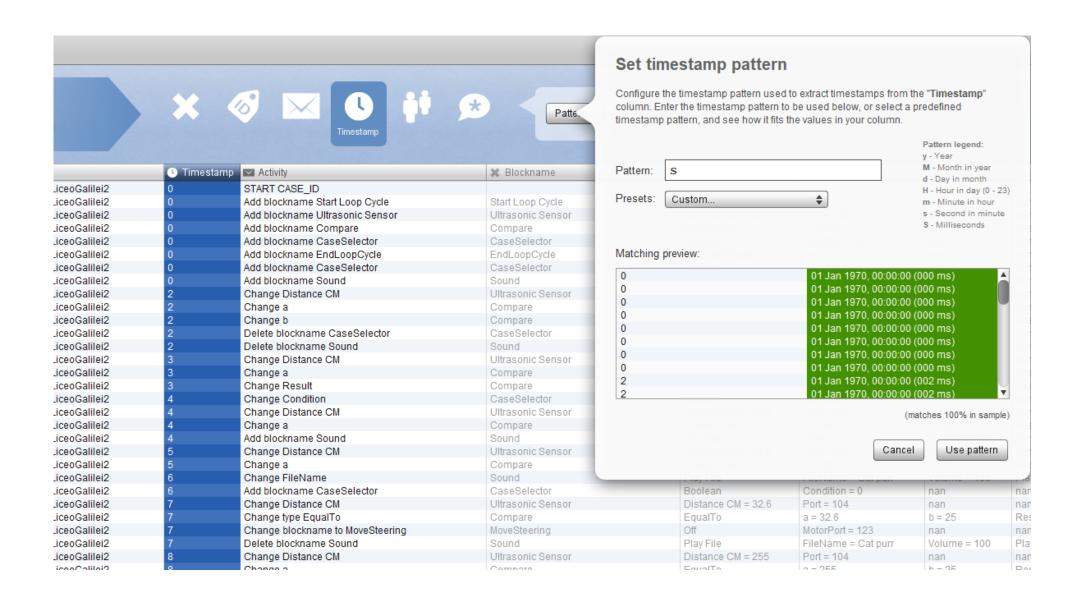
Noise	Fitness	Precision	Generalization	Simplicity	Sound
0,1	0,99	0,18	0,61	0,6	True
0,2	0,93	0,43	0,59	0,61	True
0,3	0,93	0,43	0,6	0,62	True
0,4	0,88	0,63	0,53	0,63	True
0,5	0,87	0,64	0,54	0,63	True
0,6	0,72	0,51	0,58	0,66	True
0,7	0,66	0,51	0,55	0,67	True
0,8	0,63	0,51	0,57	0,67	True
0,9	0,67	0,53	0,6	0,66	True





Disco

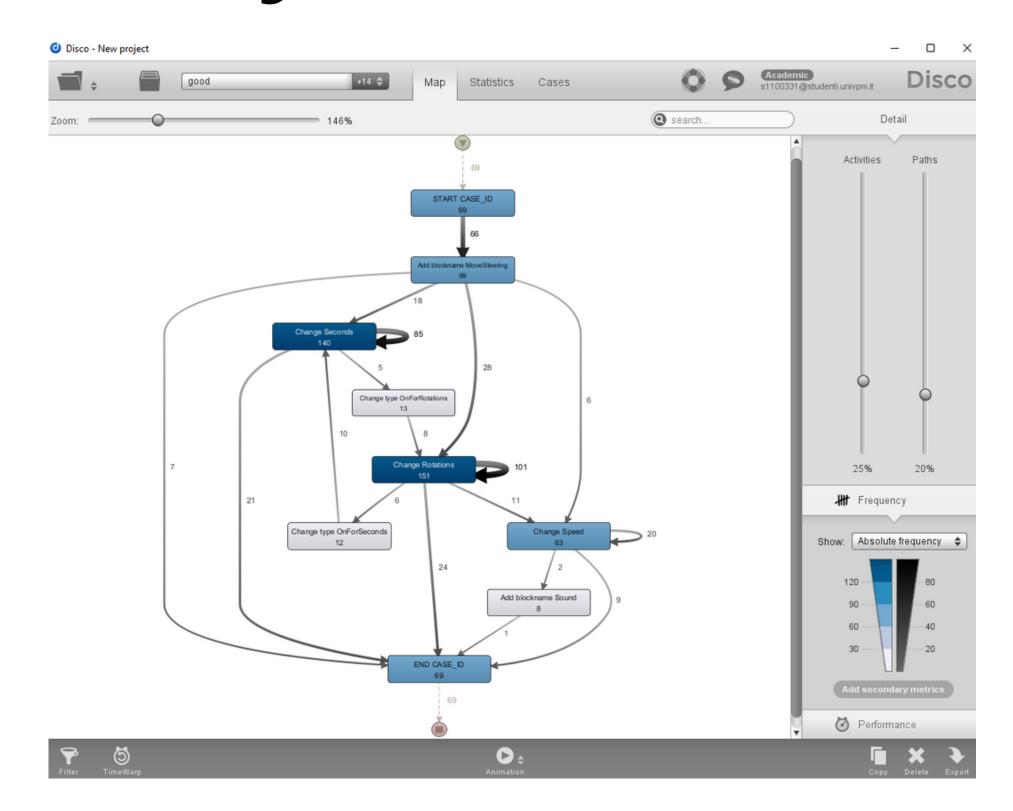
- Importazione dell'event log
- Impostazione del pattern del timestamp





Disco

Regolazione dei parametri Activities e Paths





PM4PY - Esercizio A - Good

Sono stati ottenuti 9 modelli diversi per ogni soglia di noise threshold.

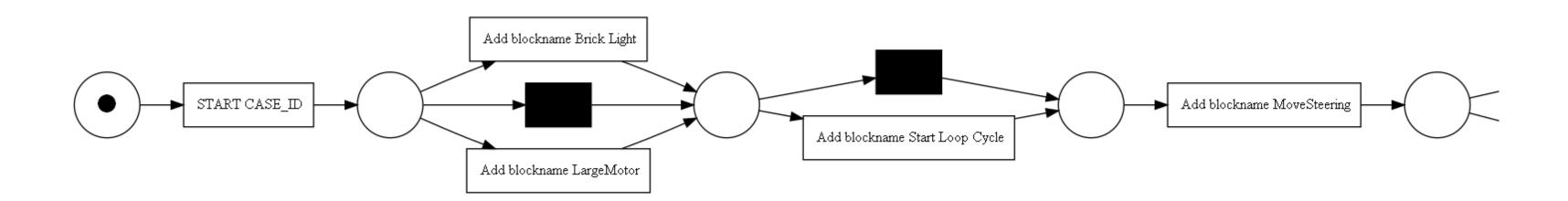
E' stato scelto il modello con noise a 0,7 perché permette di avere il miglior compromesso con tutte le metriche, privilegiando la fitness.

Noise	Fitness	Precision	Generaliz	Simplicity	Sound
0,1	0,97	0,45	0,58	0,63	True
0,2	0,95	0,41	0,56	0,62	True
0,3	0,92	0,46	0,59	0,64	True
0,4	0,91	0,61	0,51	0,65	True
0,5	0,91	0,62	0,52	0,65	True
0,6	0,9	0,64	0,58	0,68	True
0,7	0,9	0,64	0,58	0,68	True
0,8	0,79	0,92	0,53	0,69	True
0,9	0,75	0,95	0,49	0,7	True



PM4PY - Esercizio A - Good

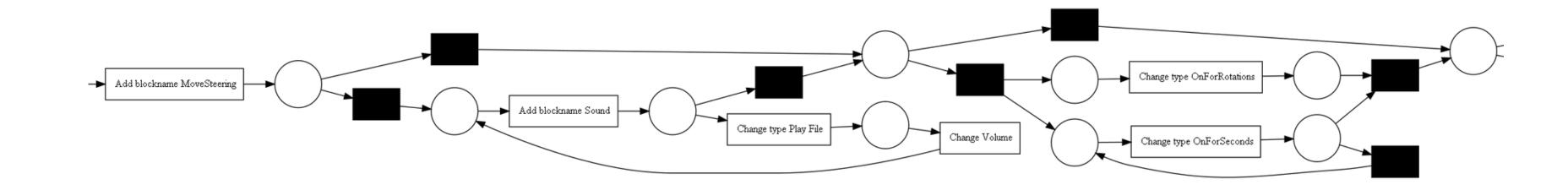
Non essendoci transizioni fittizie in corrispondenza di "Add Blockname MoveSteering", tutti i gruppi hanno sicuramente eseguito tale azione.





PM4PY - Esercizio A - Good

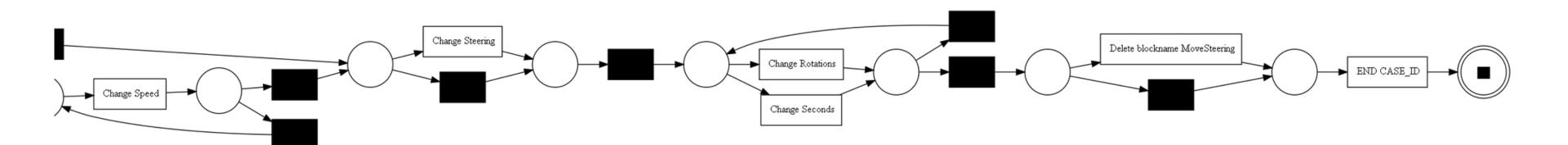
Successivamente abbiamo due diramazioni, in un path è possibile eseguire un cambiamento, il "Change Type OnForRotations" oppure il "Change Type OnForSeconds", entrambi riferiti al blocco "MoveSteering".





PM4PY - Esercizio A - Good

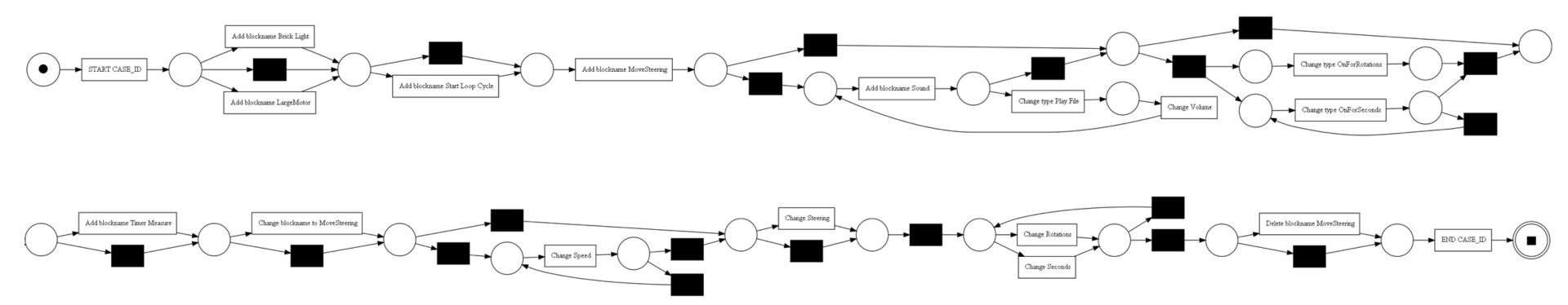
E' possibile effettuare un ciclo su "Change Speed", per poi andare ad eseguire nuovamente un ciclo su "Change Rotations" e "Change Seconds".





PM4PY - Esercizio A - Good

Si può concludere quindi che vi sono delle azioni mandatorie che hanno dovuto eseguire tutti in qualche modo. Queste azioni sono: "Add blockname MoveSteering" insieme all'esecuzione di una o entrambe le attività di "Change Rotations" e "Change Seconds".





PM4PY - Esercizio A - Wrong

Sono stati ottenuti 9 modelli diversi per ogni soglia di noise threshold.

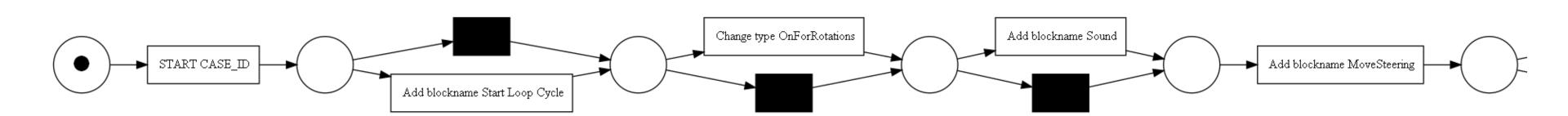
E' stato scelto il modello con noise a 0,8 perché abbiamo accettato il compromesso di perdere 0.05 sulla fitness, ma guadagnare un buona quota, quasi del 50% in più, sulla precision.

Noise	Fitness	Precision	Generaliz	Simplicity	Sound
0,1	1	0,41	0,61	0,67	True
0,2	1	0,41	0,61	0,67	True
0,3	0,96	0,47	0,59	0,66	True
0,4	0,96	0,47	0,59	0,66	True
0,5	0,91	0,55	0,58	0,67	True
0,6	0,91	0,55	0,58	0,67	True
0,7	0,86	0,8	0,54	0,68	True
0,8	0,86	0,8	0,54	0,68	True
0,9	0,75	0,49	0,47	0,72	True



PM4PY - Esercizio A - Wrong

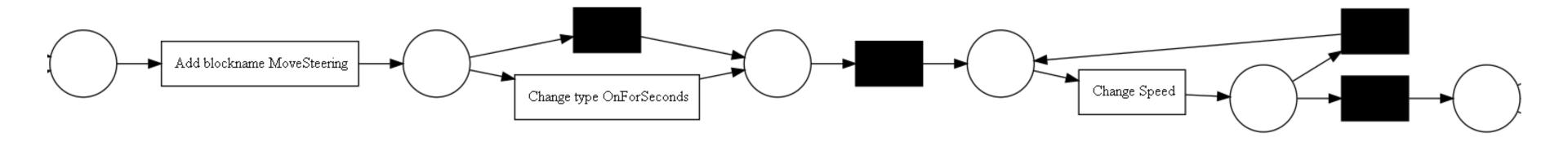
Non essendoci transizioni fittizie in corrispondenza di "Add Blockname MoveSteering", tutti i gruppi hanno sicuramente eseguito tale azione.





PM4PY - Esercizio A - Wrong

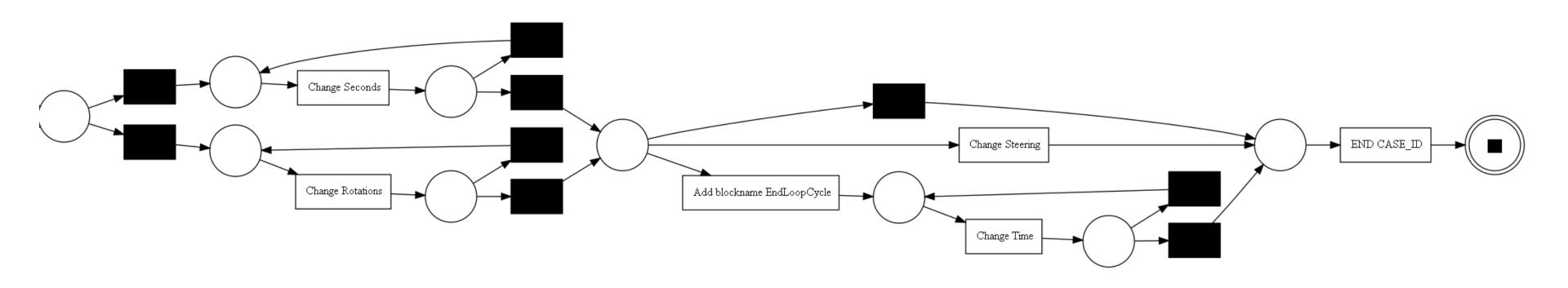
Possibile cambiamento "Change Type OnForSeconds" sulla modalità di movimento del blocco "MoveSteering". In seguito, tutti i gruppi hanno eseguito almeno una volta "Change Speed", avendo anche la possibilità di eseguire l'operazione più volte.





PM4PY - Esercizio A - Wrong

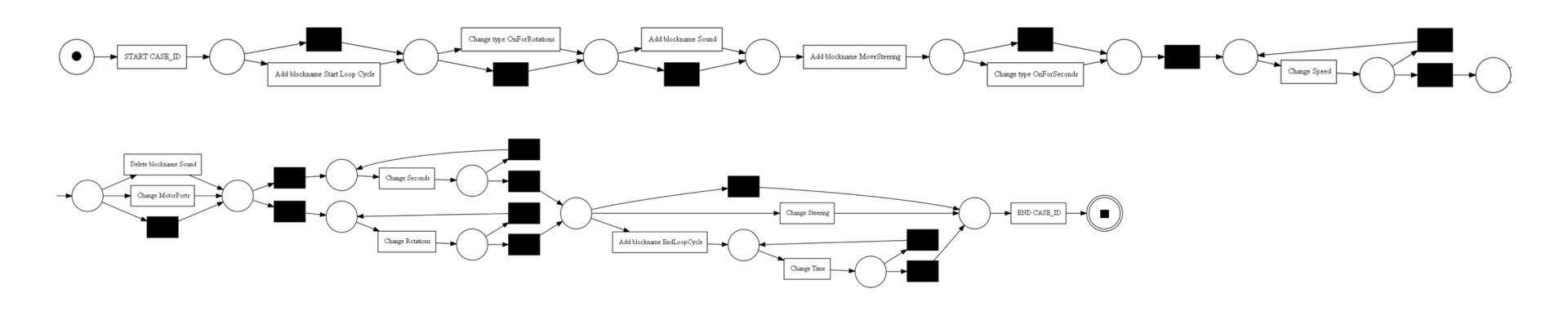
Devono essere eseguite almeno una delle due attività fra "Change Rotations" e "Change Seconds", in cui per ognuna vi è la possibilità di rieseguire l'operazione più volte.





PM4PY - Esercizio A - Wrong

Si può concludere quindi che vi sono delle azioni mandatorie che hanno dovuto eseguire tutti in qualche modo. Queste azioni sono: "Add blockname MoveSteering", "Change Speed", insieme all'esecuzione di una tra le attività di "Change Rotations" e "Change Seconds".

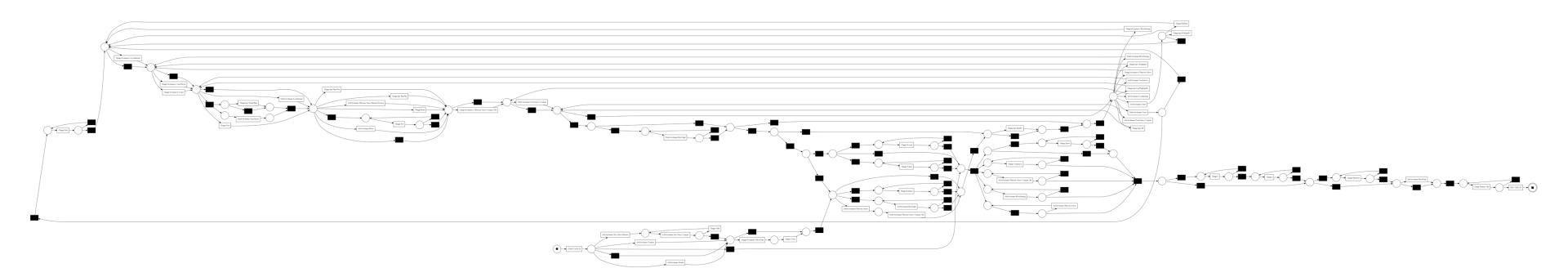




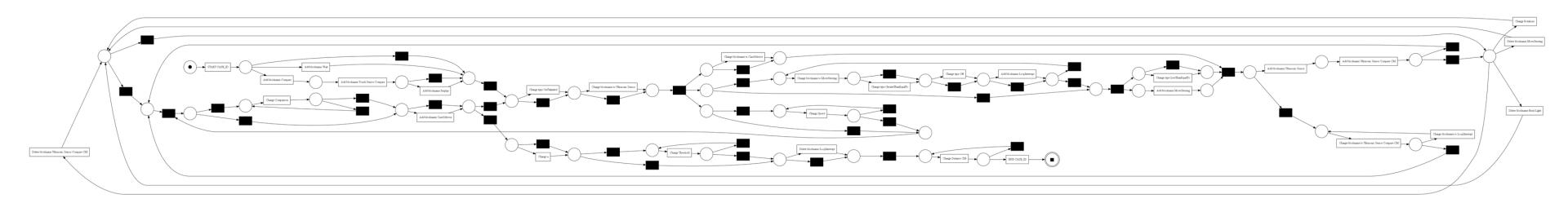
PM4PY - Esercizio A - Confronto

- "Wrong": dopo "MoveSteering", il modello presenta solamente "Change Type OnForSeconds".
 Questa modalità è un possibile motivo per cui i gruppi sbagliano l'esercizio.
 "Good": dopo "MoveSteering", il modello presenta sia "Change Type OnForSeconds" che "Change Type OnForRotations".
- "Wrong": l'attività "Change Speed" deve essere eseguita obbligatoriamente almeno una volta.
 "Good": l'attività "Change Speed" non è obbligatoria e può essere non eseguita.

PM4PY - Esercizio B - Good



PM4PY - Esercizio B - Wrong





PM4PY - Esercizio B

Non è stato possibile calcolare le metriche perché richiedevano tempi di elaborazione elevati. Possiamo supporre che questa problematica possa essere dovuta a:

Cicli

Presenza di molti cicli (circa 40) il quale molti di essi contengono ulteriori sotto cicli al loro interno.

Hidden transition

Rispetto al numero di attività, vi è un gran numero di hidden transition nel modello.

Complessità del modello

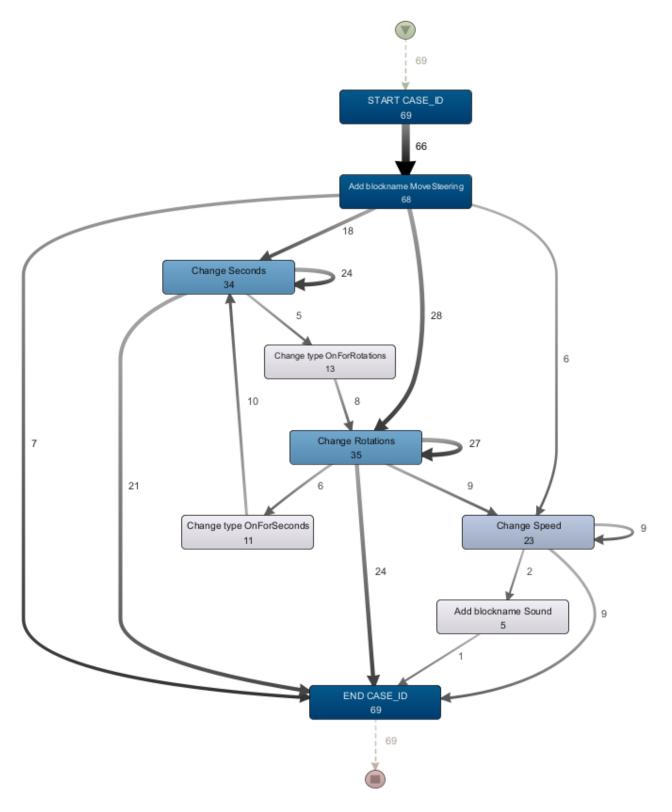
La presenza combinata di un elevata quantità di cicli ricorsivi e hidden transition aumenta notevolmente la complessità generale del modello. Inoltre ci sono comunque tantissime attività e path.

Nell'eseguire l'alignment based replay, devono essere testati tutti i possibili path. Quindi, in presenza di cicli ricorsivi, i path possibili aumentano a dismisura, rendendo l'operazione ad elevata complessità computazionale.

Disco - Esercizio A - Good

Per ottenere il modello viene impostato un valore di activities al 25% e path al 20%.

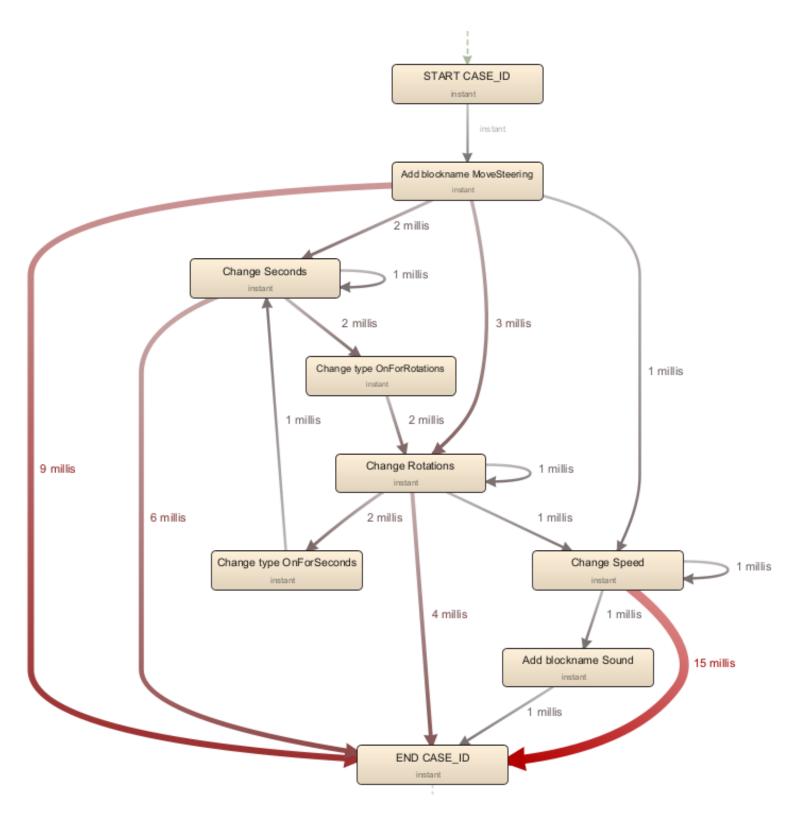
- Sette gruppi raggiungono immediatamente l'obiettivo impostando subito correttamente i parametri.
- 66 gruppi su 69 aggiungono come primo blocco il blocco MoveStering. Dopo, il 26% cambiano i secondi, i 40% cambiano le rotazioni e solo l'8% la velocità.
- Coloro che agiscono sui secondi e cambiano la velocità sono meno rispetto a quelli che agiscono sempre sul parametro Speed passando per il cambiamento delle rotazioni.



Disco - Esercizio A - Good

 A seguito dell'attività di cambiamento della velocità vengono effettuati in media 15 tentativi di esecuzione prima della conclusione dell'esercizio.

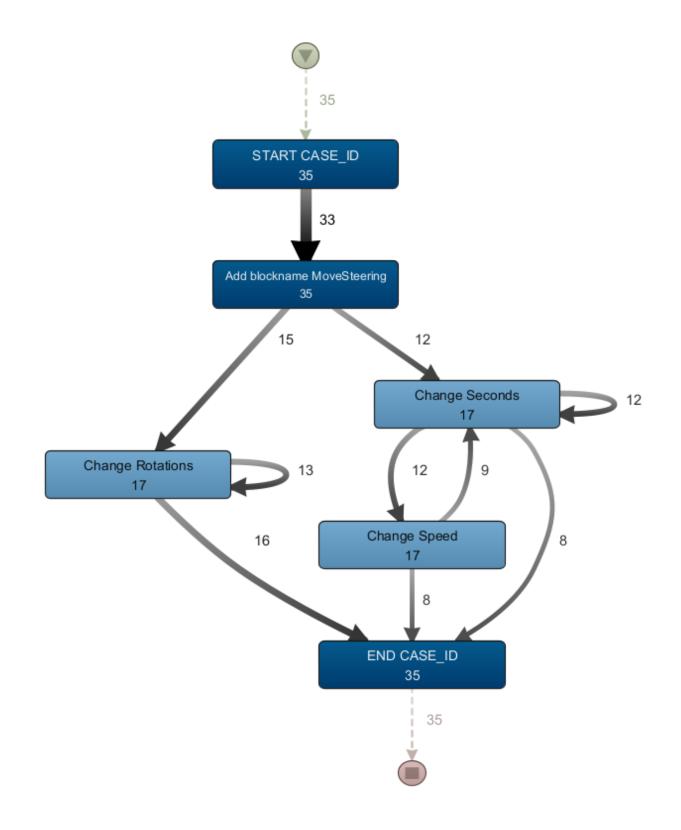
 A seguito dell'aggiunta del blocco MoveSteering vengano effettuati subito diversi tentativi prima della conclusione.



Disco - Esercizio A - Wrong

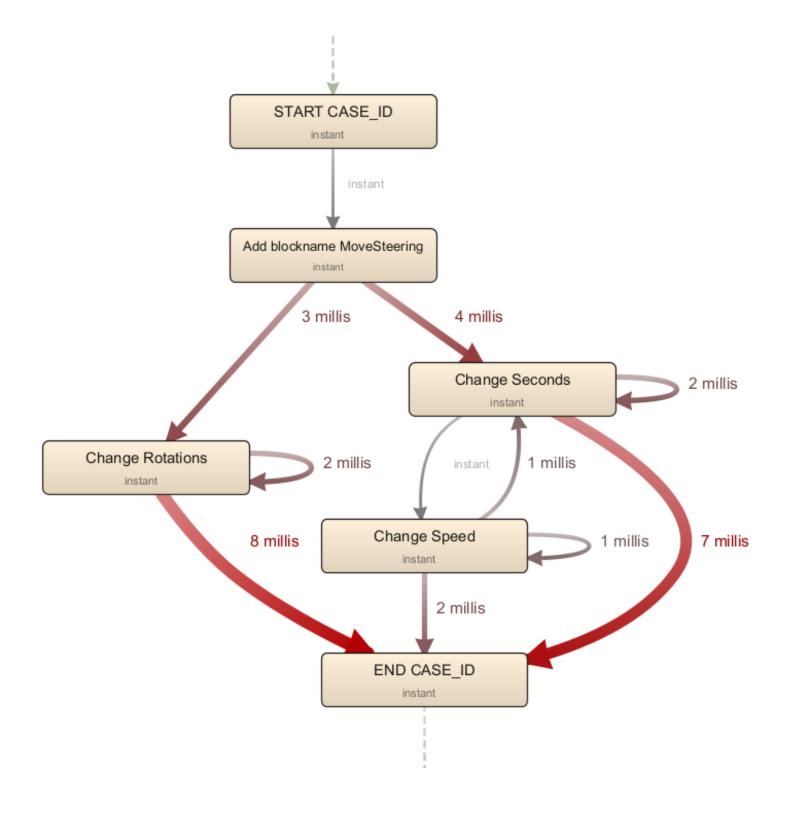
Per ottenere il modello viene impostato un valore di activities al 25% e path al 40%.

- I gruppi si dividono in chi cambia le rotazioni e chi cambia i secondi. Una buona parte di coloro che agiscono sui secondi decidono di cambiare anche il parametro Speed. La regolazione di due parametri contemporaneamente porta a sbagliare l'esercizio.
- Nella visualizzazione con la frequenza assoluta, il numero delle volte in cui vengono modificati i secondi è maggiore, di circa un terzo, delle volte in cui vengono modificate le rotazioni.



Disco - Esercizio A - Wrong

 In media, vengono eseguiti quasi lo stesso numero di tentativi sia dopo il cambio delle rotazioni che il cambio dei secondi.





Disco - Esercizio A - Confronto

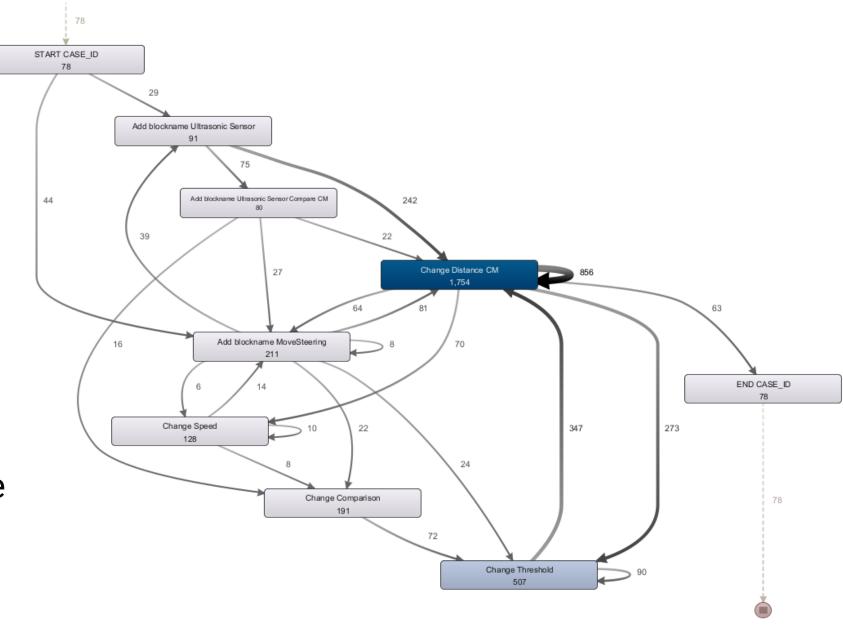
- "Wrong": cambiamento della velocità insieme alla regolazione dei secondi (variabili dipendenti).
 "Good": cambiamento della velocità insieme alla regolazione delle rotazioni (variabili indipendenti).
- "Wrong": divisione equa tra chi agisce sulle rotazioni e sui secondi.
 "Good": divisione netta tra chi agisce sulle rotazioni e sui secondi.
 Ciò potrebbe significare che è più semplice raggiungere l'obiettivo regolando le rotazioni del robot piuttosto che i secondi.
- "Wrong": numero dei tentavi prima di giungere alla conclusione omogeneo sulla regolazione dei secondi e rotazioni

"Good": numero dei tentativi eseguiti dopo il cambo della velocità è più alto rispetto agli altri parametri.

Disco - Esercizio B - Good

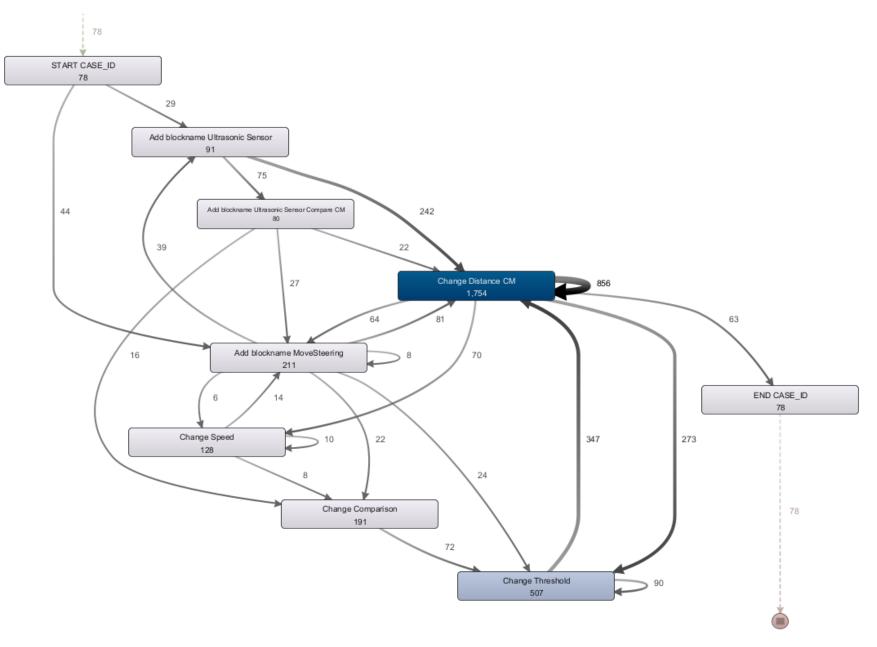
Per ottenere il modello viene impostato un valore di activities al 14% e path al 8%.

- Il 50% rispetto al totale dei casi, aggiunge il blocco "MoveSteering". Di questi, la maggior parte aggiunge il blocco "Ultrasonic Sensor" e poi il blocco "Ultrasonic Sensor compare CM.
- Divisione quasi equa tra modifica il parametro "Threshold" e chi il parametro "Comparison".
- Presenza di un ciclo tra "Add BlockName MoveSteering" e "Change Speed". Anomalia in cui 13 gruppi aggiungono MoveSteering con type = Off, poichè in precedenza aveno impostato type = OnUlimited.



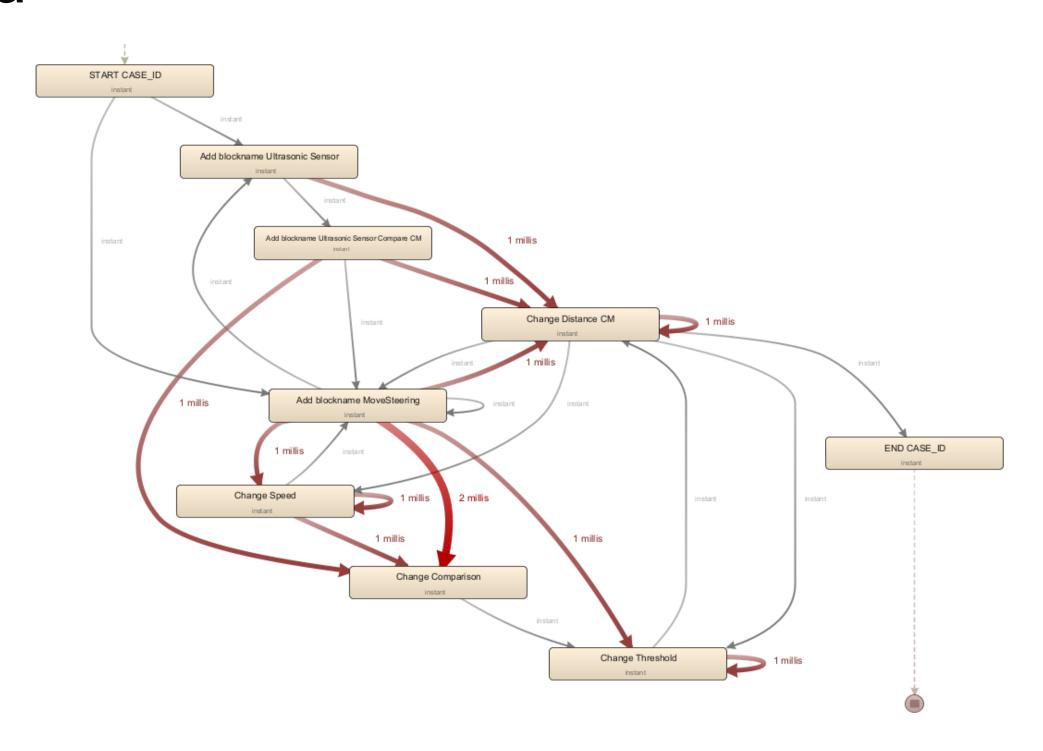
Disco - Esercizio B - Good

- Presenza di cicli su "Change threshold" e "Change distance CM". Coloro che rieseguono queste operazioni sono rispettivamente il 35% e il 91% rispetto al totale dei gruppi.
- Dalla vista "Case Coverage", si nota che "Move-Steering" e "Change Distance CM" vengono utilizzati dal 100% dei gruppi. Altre attività come "Add Blockname Ultrasonic Sensor" e "Add Blockname Ultrasonic Sensor Compare CM" vengono utilizzati molto spesso, rispettivamente dal 96% e 91% dei gruppi.
- Dalla vista "Absolute Frequency", si nota che esiste un ciclo sull'attività "Change Distance CM" effettuato 856 volte. La sola attività viene eseguita ben 1754 volte.



Disco - Esercizio B - Good

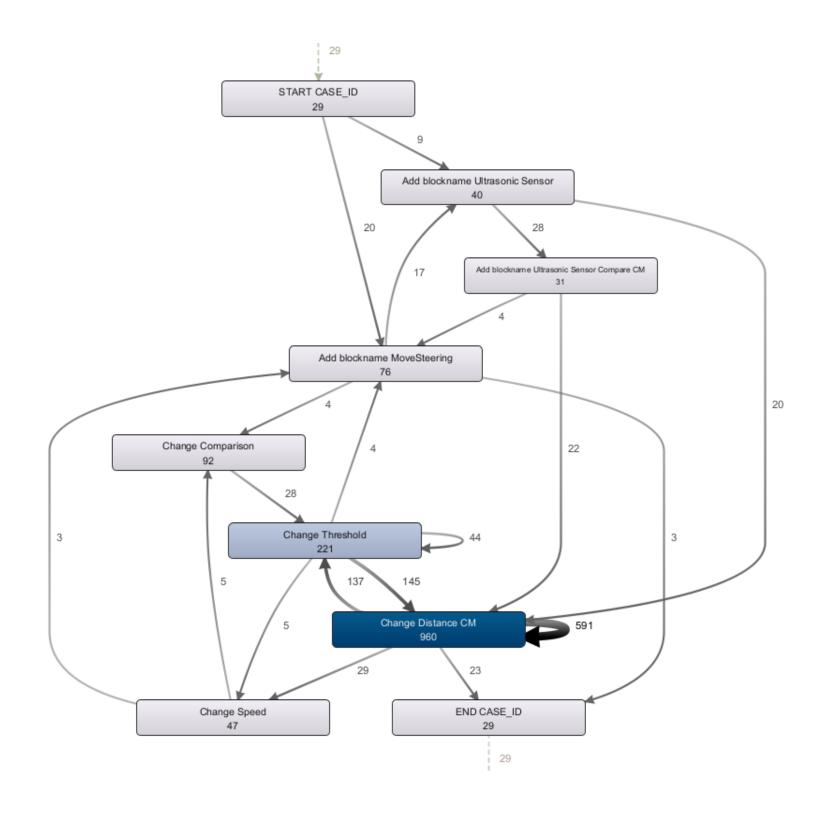
 Dalla vista performance non emergono particolari considerazioni sulla media dei tentativi effettuati.



Disco - Esercizio B - Wrong

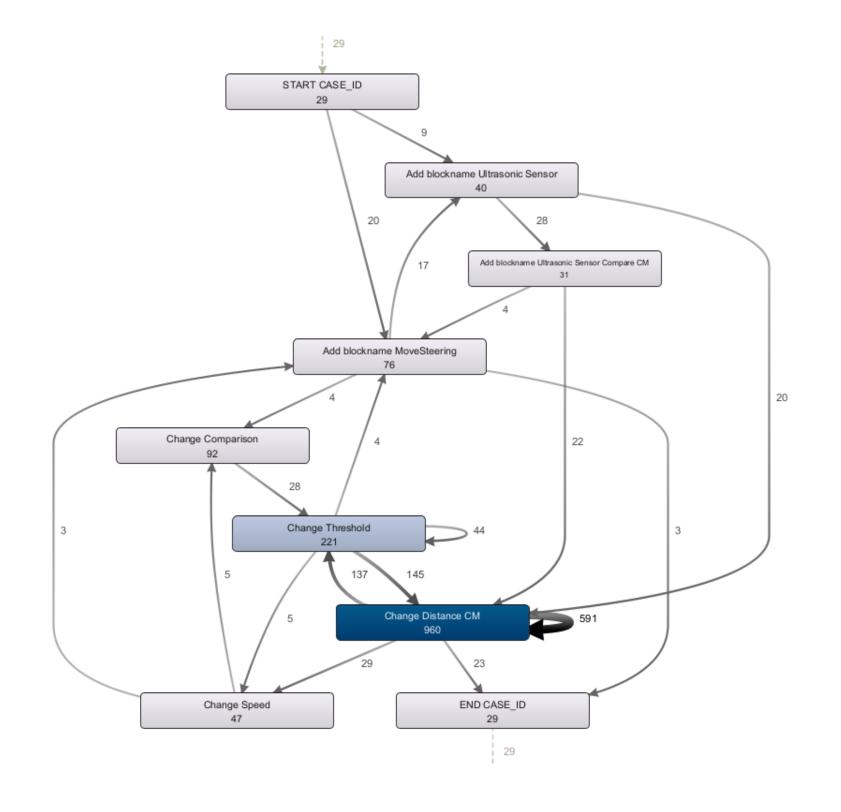
Per ottenere il modello viene impostato un valore di activities al 20% e path al 8%.

- La maggior parte inserisce il blocco "MoveSteering" e in seguito "Ultrasonic sensor" e "Ultrasonic sensor compare CM".
- Dopo l'aggiunta dei blocchi iniziali, nella maggior parte dei casi segue l'attività "Change Distance CM".
- Divisone quasi equa tra chi modifica il parametro "Speed" e chi modifica il parametro "Threshold".



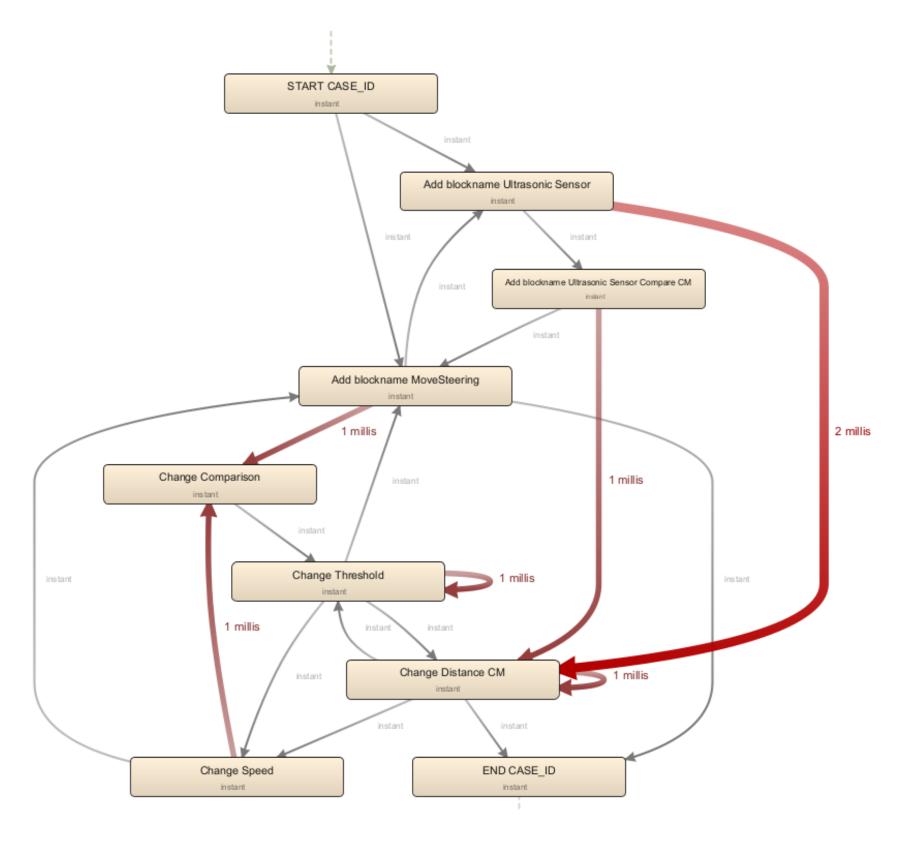
Disco - Esercizio B - Wrong

- Presenza di cicli su "Change threshold" e "Change distance CM". Coloro che rieseguono queste operazioni sono rispettivamente il 24% e l'86% rispetto al totale dei gruppi.
- Dalla vista "Case Coverage" si nota che "Add Blockname MoveSteering", "Change Distance CM" e "Add Blockname Ultrasonic Sensor" vengono utilizzate dal 100% dei gruppi. Altre attività come "Add Blockname Ultrasonic Sensor Compare CM" e "Change comparison" vengono utilizzati dall'83% dei gruppi.
- Con la vista "Absolute Frequency", si nota un ciclo sull'attività "Change Distance CM" che viene effettuato 591 volte. La sola attività, invece, viene eseguita ben 960 volte.



Disco - Esercizio B - Wrong

 Dalla vista performance non emergono particolari considerazioni sulla media dei tentativi effettuati.





Disco - Esercizio B - Confronto

- In entrambi i casi prima inseriscono il blocco "MoveSteering", poi aggiungono "UltraSonic Sensor" e successivamente il blocco "Ultrasonic Sensor compare CM".
- In entrambi i casi l'azione che viene eseguita subito dopo più spesso è quella di "Change Distance CM".
- "Wrong": i ragazzi modificano i parametri "speed" e "threshold".
 "Good": i ragazzi modificano i parametri "threshold" e "comparison".
- "Good": in media, ogni gruppo esegue l'attività "Change distance" circa 22 volte e la ripete per 11 volte. "Wrong": in media, ogni gruppo esegue l'attività "Change distance" circa 33 volte e la ripete per 22 volte (il doppio).
- Infine, non emergono altri particolari differenze di comportamento tra chi raggiunge l'obiettivo e chi no, quindi si può dedurre che le differenze sostanziali risiedono nella regolazione dei parametri.

Conclusioni

Differenze tra PM4PY e Disco

PM4PY:

- + reti di petri dettagliate.
- filtro attraverso un solo parametro "noise threshold".
- leggibilità della rete.
- elevata complesstà del modello.

Disco:

- + filtro più stringente e separato per activities e paths
- + modello più leggibile con poche attività
- + interfaccia intuitiva anche per analizzare i vari cases
- non si ottengono metriche che descrivono il modello

Conclusioni

Risultati conclusivi

Ex A:

"OnForRotations" fa si che lo spostamento del robot dipende unicamente dalla variabile "Rotations", mentre con "OnForSeconds" vengono regolati sia il parametro "Speed" che "Seconds" aumentando cosi la difficoltà.

• Ex B:

Non sono state identificate differenze significative, anche a causa dei dati altamente variabili e l'assenza di pattern (l'esercizio poteva essere svolto in diversi modi). A fare quindi la differenza è per lo più l'impostazione dei parametri in modo corretto.

Grazie per l'attenzione

