

Documentazione dell'analisi effettuata sui due algoritmi di scheduling

1. Introduzione:

Con il presente documento si vuole spiegare la natura degli esperimenti effettuati per verificare i due algoritmi di scheduling. Si vuole inoltre andare a spiegare lo schema del sistema di analisi e quali lavori si potrebbero fare, per ampliare l'analisi dello scheduler.

2. Analizzatore:

Il sistema di analisi che si trova sulla sottocartella "Analyzer" è un insieme di file Python ognuno con dei compiti specifici, in particolare:

- Il file "measuererAndAnalyzer.py" è un file Python che ha il compito di collegarsi al sistema ed effettuare chiamate HTTP sull'indirizzo "http://localhost:3000/api/schedule/generation" in modo tale da creare N schedulazioni ognuna da 1 mese. In seguito, esegue query direttamente al database per prendere le metriche e tirar fuori le statistiche desiderate; quindi, prelevate le metriche, le aggrega e le memorizza nella sottocartella Statistiche tramite uno schema di file json. Questo procedimento viene applicato prima per il nuovo scheduler e poi per il vecchio scheduler. Tra l'invocazione di uno scheduler e l'altro, vi è una fase di reset del sistema fatto tramite chiamata http per ogni schedula messa precedentemente, in modo da avere il sistema il più fair possibile.
- Il file "plotter.py" ha il compito di leggere le statistiche scritte nei file json e, tramite la libreria matplotlib, crea dei grafici ad hoc. Ovviamente questo file ha una dipendenza con "measuererAndAnalyzer" in quanto deve essere eseguito dopo di lui per produrre dei risultati.
- Il file "HypothesisTest.py" si occupa invece di prelevare dai file json le metriche ottenute e studiare il comportamento della variabile aleatoria in modo da avere un confronto matematico-statistico sui comportamenti dei due algoritmi di schedulazione. Il confronto e lo studio delle variabili aleatore vogliono essere catturati mediante test di ipotesi. Per ora vengono effettuati i seguenti test:
 - T-test
 - Test di Kolmogorov-Smirnov
 - Test di Shapiro-will
 - Test di ChiSquare

3. Esperimenti:

In questa sezione, si vogliono elencare tutte le variabili che dovrebbero far parte del sistema per ottenere un confronto più veritiero possibile.

Variabili in gioco:

La tabella seguente rappresenta le variabili del sistema che sono state fissate all'inizio dei vari esperimenti.

Nome variabile	Significato	Valore Assegnato
N	Numero totale di medici all'interno dell'ospedale	42
Nm	Numero di medici di tipo "Strutturato" all'interno dell'ospedale	14
Ns	Numero di medici di tipo "Specializzando senior" all'interno dell'ospedale	14
Nj	Numero di medici di tipo "Specializzando junior" all'interno dell'ospedale	14
N62	Numero di medici di tipo over 62	0
Ninc	Numero di medici incinta	0
Nmal	Numero di medici in malattia	0
Nmat	Numero di medici in maternità	0
Nfer	Numero di medici in ferie	0
F	Numero di festività durante l'anno	14
R	Numero di Servizi (Reparti)	1
M	Numero di Mansioni (Task)	4
Mr	Numero di Mansioni per reparto	4
Tn	Numero di Turni notturni per reparto	1
TI	Numero di Turni lunghi per reparto	1
Tm	Numero di Turni mattinieri per reparto	1
Tp	Numero di Turni pomeridiani per reparto	1
Td	Numero di Turni domenicali	3
Njmt	Numero di medici di tipo "Specializzando junior" per mansione per turno	1
Nsmt	Numero di medici di tipo "Specializzando senior" per mansione per turno	1
Nmmt	Numero di medici di tipo "Strutturato" per mansione per turno	1
DimS	Dimensione della schedulazione (in mesi)	1
NumS	Numero di schedulazioni	48

4. Risultati Ottenuti:

Per il confronto dei due algoritmi si è seguita questa idea:

"l'algoritmo migliore è quello più fair tra i medici"

Ovvero, abbiamo deciso di confrontare quale fosse l'andamento della priorità di ogni medico per i due scheduler. Ovviamente i due algoritmi hanno definizioni differenti di cosa sia una priorità di un medico. Infatti, il vecchio scheduler considerava la priorità come gli Uffa point accumulati: più se ne aveva, meno il medico ha priorità all'interno dell'assegnazione dei turni. Invece, il nuovo algoritmo utilizza tre code di priorità per i vari tipi di scocciatura e la mia posizione nella coda è il livello di priorità dell'assegnazione del turno (in particolare più il livello è basso, più la mia priorità di assegnazione sarà alta in merito a quella coda).

Il grafico in figura 1 ha l'obiettivo di rappresentare per ogni schedulazione la distanza dalla media di ogni medico rispetto al suo livello di priorità. Tutto attraverso l'idea sarebbe che più questo andamento tende a essere basso (quindi a 0) più l'algoritmo sarà fair nel rispetto di ogni medico.

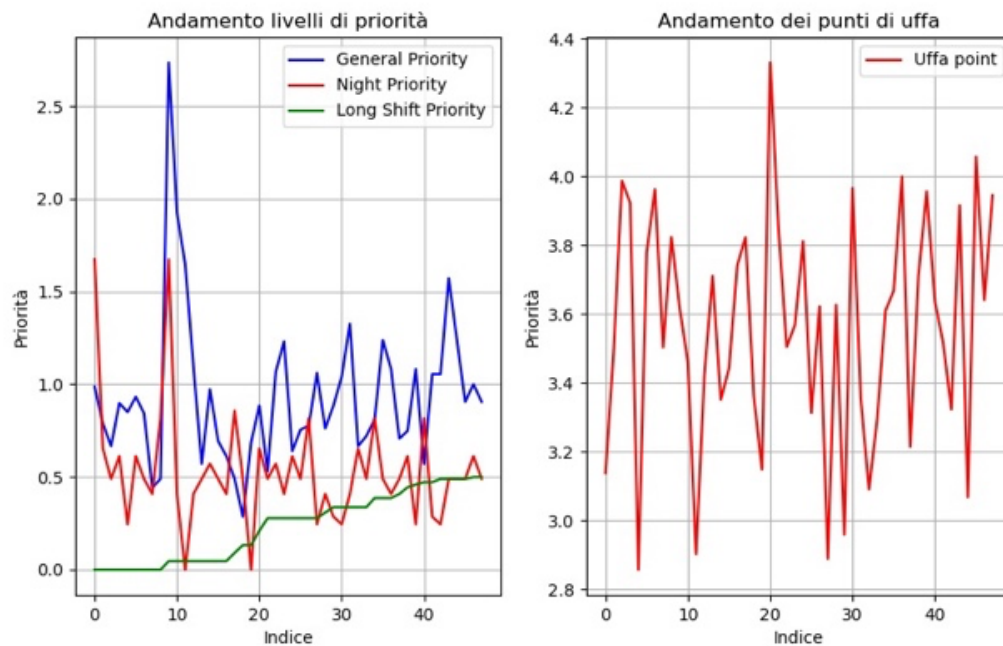


Figura 1

Nel grafico in figura 2 invece si mostra il comportamento dei due scheduler nell'assegnare ai medici in merito ai tipi di turni assegnati (e in base alla media dei turni assegnati per ogni tipo di turno). Poiché le schedulazioni sono tutte molto simili tra loro, variano solo per la durata: Più il grafico è ordinato, più è segno di una buona schedulazione; Questo perché ogni schedulazione ha quasi la stessa quantità di turni (variano di pochi giorni poiché i mesi hanno lunghezze differenti).

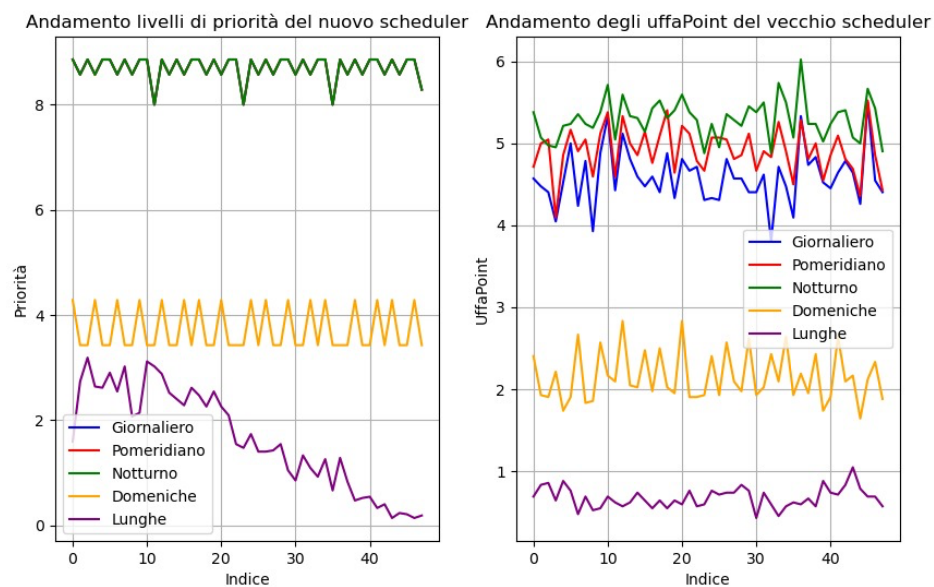


Figura 2

5. Sviluppi Futuri:

Per ampliare l'analisi degli scheduler c'è ancora molto lavoro da portare avanti, essendovi tante variabili in gioco, si potrebbero estendere gli esperimenti per aumentare la diversità del tipo di schedulazione che l'algoritmo è sottoposto a fare.

Per far questo si potrebbe pensare di creare un ulteriore componente che si occupa di sistemare l'architettura inizializzandola a configurazioni diverse rispetto a quella di default attuale (che al momento è statica rispetto allo start up del sistema). Quindi servirebbe un componente che, in base a dei file di configurazione inserisce i turni, i medici, gli stati dei medici, i reparti, le mansioni e così via, in modo da far variare gli esperimenti il più possibile per verificare in maniera più esaustiva le prestazioni dell'algoritmo.