

SPERIMENTAZIONE

NICOLA SANITATE

REDISCOVERING WORKFLOW MODELS FROM EVENT-BASED DATA

PROGETTO DI INTELLIGENZA ARTIFICIALE

A.A. 2010 - 2011

UNIVERSITÀ DEGLI STUDI DI BARI

SPERIMENTAZIONE

INDICE

| | |
|------------------------------------|-----------|
| OBIETTIVO..... | 3 |
| INTRODUZIONE..... | 4 |
| GENERATORE DI SEQUENZE..... | 5 |
| COMPARATORE DI MODELLI..... | 10 |
| ESPERIMENTO 1..... | 16 |
| ESPERIMENTO 2..... | 19 |
| ESPERIMENTO 3..... | 22 |
| ESPERIMENTO 4..... | 25 |
| ESPERIMENTO 5..... | 28 |
| ESPERIMENTO 6..... | 30 |
| CONCLUSIONI..... | 32 |

SPERIMENTAZIONE

OBIETTIVO

Si vuole verificare che la tecnica presentata nell'articolo Rediscovering Workflow Models from Event-Based Data produca, tramite l'analisi dei log di workflow estrapolati da sistemi informativi transazionali, dei modelli di workflow che rispettino i requisiti di completezza e correttezza nei confronti del modello di workflow reale da cui derivano i log analizzati. Inoltre si vuole testare la robustezza della tecnica rispetto a diversi livelli di rumore all'interno dei log di workflow.

INTRODUZIONE

Per questa sperimentazione si sceglie di seguire l'iter sperimentale adottato dagli autori dell'articolo. Questo consiste nel testare il sistema su sei diversi modelli di workflow comprendenti dai 13 ai 16 task. Ogni modello di workflow presenterà sia processi concorrenti che cicli. Da ogni modello verranno generati 3 log di workflow da 1000 sequenze ciascuno: uno non presenterà rumore, uno presenterà il 5% di rumore e l'ultimo presenterà il 10% di rumore. La generazione delle sequenze sarà casuale, ma dovrà rispettare il modello di base.

Per generare rumore nei log si sono ideate quattro operazioni che simulano il rumore che è possibile riscontrare nei log ottenibili dai sistemi informativi transazionali:

1. cancellazione della testa di una sequenza di eventi;
2. cancellazione della coda di una sequenza di eventi;
3. cancellazione del corpo di una sequenza di eventi;
4. scambio di due eventi casuali in una sequenza di eventi.

Il numero di eventi da cancellare varia da un minimo di 1 ad un massimo di $\frac{1}{3}$ della lunghezza della sequenza.

Al fine di verificare la completezza e la correttezza dei modelli rigenerati è necessario formalizzare i concetti di completezza e correttezza. Dato che, come affermato nel documento di analisi del progetto, la reale complessità nella ricostruzione del modello di workflow a partire dalle sequenze di workflow sta nell'individuazione delle tipologie di connessione, si effettuerà il confronto tra il modello di partenza ed il modello ottenuto sulla base degli split e delle join correttamente individuati.

Per completezza si intende che tutti gli split ed i join presenti nel modello di workflow iniziale siano coperti da degli split e dei join del modello di workflow ottenuto dal sistema.

Per correttezza si intende che tutti gli split ed i join presenti nel modello di workflow ottenuto dal sistema siano coperti da degli split e dei join del modello di workflow iniziale.

Infine, per copertura si intende che tutte le transizioni partecipanti ad uno split (join) in un modello partecipino ad uno split (join) nel modello da confrontare mantenendo gli stessi ruoli, ossia di transizioni di partenza o transizioni di destinazione. Ciò si rende necessario in quanto, dai log di workflow, è possibile mantenere traccia solo del nome delle transizioni e non del nome dei posti.

Nel resto del documento verranno presentati dei metodi automatici di generazione delle sequenze e comparazione dei modelli, seguite dagli esperimenti e dalle conclusioni.

SPERIMENTAZIONE

GENERATORE DI SEQUENZE

È stato realizzato un algoritmo per generare delle sequenze casuali che siano coerenti con un modello di workflow e che stampi tali sequenze su un file di output dopo aver applicato uno dei quattro tipi di rumore illustrati precedentemente ad un numero di sequenze che varia a seconda della percentuale di rumore desiderata.

Al fine di facilitare la comparazione, per la rappresentazione del modello di workflow si è scelto di attenersi alla rappresentazione adottata per il file di output del sistema realizzato, ossia due insiemi di fatti raffiguranti gli archi in entrata ed un uscita:

in(PostoPrecedente, Transizione), out(Transizione, PostoSuccessivo).

Ai fini della sperimentazione il modello di workflow deve essere intercambiabile, quindi si sceglie di salvare il modello in un file “wf-net-in” esterno e consultarlo prima della generazione delle sequenze.

Per la rappresentazione interna delle sequenze le opzioni individuate sono 2:

- sequenza([A,B,...]).
- [[A₁,B₁,...],[A₂,B₂,...],...].

La scelta è da effettuare tra un insieme di fatti raffiguranti le singole sequenze, e una liste che raggruppa tutte le sequenze create. La scelta è ricaduta sul secondo formalismo in quanto, ai fini della stampa, le sequenze verranno visitate in modo sequenziale.

Infine, per la rappresentazione delle sequenze su file di output “workflow.log” si è scelto ancora di attenersi alla rappresentazione adottata per il file di input del sistema realizzato per gli stessi motivi:

- coerenza con l'articolo
- necessità di scegliere un formalismo che sia facilmente ricavabile dai log dei sistemi informativi transazionali.

Di seguito verrà illustrata la soluzione trovata per il generatore. Questa sarà divisa in due parti: la prima prevederà la generazione delle sequenze, mentre la seconda provvederà all'applicazione del giusto grado di rumore e alla stampa su file.

Si noti che la frequenza con cui applicare il rumore alle sequenze si può ricavare come:

$$\begin{aligned} \#Righe &= 1000 \\ \#RigheConRumore &= \frac{\#Righe \cdot \%Rumore}{100} \rightarrow FrequenzaRumore = \frac{100}{\%Rumore} \\ FrequenzaRumore &= \frac{\#Righe}{\#RigheConRumore} \end{aligned}$$

SPERIMENTAZIONE

Di seguito si riporta la soluzione trovata per la generazione delle sequenze.

Selezione di un elemento casuale in una lista:

- ottiene la lunghezza L della lista da cui prelevare l'elemento, ottiene un numero casuale N compreso tra $[0, L)$, seleziona e restituisce l'elemento situato al N -esimo posto nella lista.

Differenza tra liste:

- termina non ci sono più elementi nella lista sottraendo, e in tal caso restituisce la lista minuendo come lista differenza;
- per gli altri casi aggiorna la lista minuendo rimuovendo un istanza dell'elemento di testa della lista sottraendo e richiama la differenza tra liste tra il resto della lista sottraendo e la lista minuendo aggiornata (il fallimento della nuova differenza è indice del fatto che la lista sottraendo non è sottoinsieme della lista minuendo, dunque deve fallire anche la differenza corrente in quanto l'aggiornamento della lista minuendo non fornirebbe altri valori validi); infine restituisce la lista differenza ottenuta.

Generazione di una sequenza:

- termina se il posto di partenza corrisponde al posto di arrivo, e in tal caso restituisce lista di transizioni attraversate vuota.
- raggruppa tutte le transizioni raggiungibili da un posto di partenza, da questa lista seleziona una transizione casuale che si desidera attraversare, in seguito raggruppa tutti i posti che precedono la transizione estratta, tali posti vengono sottratti dalla lista dei token disponibili (questa operazione serve a valutare se far scattare una transizione a seconda dei posti che la precedono, che sono prerequisiti indispensabili affinché lo scatto possa avvenire); se la differenza non fallisce allora attraversa la transizione raggruppando i posti di destinazione della transizione e aggiungendoli ai token rimanenti dalla differenza effettuata precedentemente; infine seleziona un posto casuale dalla lista dei token e richiama la generazione di una sequenza che abbia come posto di partenza il posto appena estratto; alla sequenza ottenuta aggiungerà in testa la transizione attraversata;
- se la differenza fallisce allora seleziona un posto casuale dalla lista dei token e richiama la generazione di una sequenza che abbia come posto di partenza il posto appena estratto; la sequenza ottenuta verrà restituita come sequenza corrente;

SPERIMENTAZIONE

Generazione delle sequenze:

- termina se il numero di sequenze da generare è 0, e in tal caso restituisci la lista delle sequenze vuota;
- per gli altri casi decrementa di 1 il numero di sequenze da generare, chiama la generazione di una sequenza dal posto “sb” al posto “se” con una lista di token popolata dal solo “sb” e richiama la generazione delle sequenze con il numero delle sequenze da generare decrementato; alle sequenze ottenute aggiungerà in testa la sequenza generata.

Di seguito si riporta la soluzione trovata per l'applicazione del rumore alle sequenze ed alla loro successiva stampa.

Divisione di una lista:

- termina se la posizione dell'elemento di divisione è 0, e in tal caso restituisce una lista degli elementi precedenti vuota, l'elemento di testa della lista da dividere come elemento medio, e gli elementi rimanenti della lista da dividere come lista degli elementi successivi;
- per gli altri casi decrementa la posizione dell'elemento di divisione di 1 e richiama la divisione della lista sulla lista da dividere escludendo la testa e sulla nuova posizione dell'elemento di divisione; prima di restituire la lista degli elementi precedenti ottenuta vi aggiungerà l'elemento di testa della lista da dividere, mentre l'elemento medio e la lista degli elementi successivi ottenuti vengono restituiti senza modifiche.

Rimozione della testa di una sequenza:

- termina se il numero di elementi da eliminare è uguale a 0, in tal caso restituisce la sequenza di partenza come sequenza con rumore;
- per gli altri casi decrementa il numero di elementi da eliminare di 1, e richiama la rimozione della testa di una lista sulla sequenza di partenza escludendo la testa; la sequenza con rumore ottenuta viene restituita senza modifiche.

Rimozione della coda di una sequenza:

- termina se il numero di elementi da eliminare è uguale a 0, in tal caso restituisce la sequenza di partenza come sequenza con rumore;
- per gli altri casi calcola la posizione dell'ultimo elemento della lista e lo elimina, decrementa il numero di elementi da eliminare di 1 e richiama la rimozione della coda di una lista sulla lista rimanente; la sequenza con rumore ottenuta viene restituita senza modifiche.

SPERIMENTAZIONE

Rimozione del corpo di una sequenza:

- ottiene la posizione dell'elemento medio della sequenza di partenza, divide la sequenza di partenza usando come elemento di divisione l'elemento nella posizione ottenuta prima, dalla lista degli elementi successivi ottenuta dalla divisione elimina la testa per un numero di elementi pari alla metà del numero di elementi da eliminare, dalla lista degli elementi precedenti ottenuta dalla divisione elimina la coda per un numero di elementi pari alla metà del numero di elementi da eliminare, e infine restituisce la concatenazione delle liste ottenute dagli ultimi due passi.

Scambio di due elementi di una sequenza:

- ottiene la lunghezza L della sequenza, ottiene un numero casuale N compreso tra $[1, L)$, ottiene un altro numero casuale M compreso tra $[0, N)$, divide la sequenza di partenza usando come elemento di divisione l'elemento in posizione N , divide la lista degli elementi precedenti ad N usando come elemento di divisione l'elemento in posizione M , infine concatena la lista degli elementi precedenti ad M , l'elemento N , la lista degli elementi successivi ad M , l'elemento M e gli elementi successivi all'elemento N .

Applicazione di una tipologia di rumore:

- scegli un numero casuale compreso tra $[0,4)$ e richiama l'applicazione di una tipologia di rumore sul numero estratto; la sequenza con rumore ottenuta verrà restituita senza modifiche;
- se il numero estratto è 0 allora ottiene la lunghezza della sequenza, ottiene un numero casuale compreso tra 0 e un terzo della lunghezza della sequenza e restituisce la sequenza con errore rimuovendo dalla testa della sequenza di partenza un numero di elementi pari al numero estratto;
- se il numero estratto è 1 allora ottiene la lunghezza della sequenza, ottiene un numero casuale compreso tra 0 e un terzo della lunghezza della sequenza e restituisce la sequenza con errore rimuovendo dalla coda della sequenza di partenza un numero di elementi pari al numero estratto;
- se il numero estratto è 2 allora ottiene la lunghezza della sequenza, ottiene un numero casuale compreso tra 0 e un terzo della lunghezza della sequenza e restituisce la sequenza con errore rimuovendo dal corpo della sequenza di partenza un numero di elementi pari al numero estratto;
- se il numero estratto è 3 allora restituisce la sequenza con errore scambiando due elementi a caso nella sequenza di partenza;

SPERIMENTAZIONE

Stampa di una sequenza:

- termina se non ci sono più transizioni nella sequenza;
- per gli altri casi stampa una virgola e la transizione di testa della sequenza sullo stream di output e richiama la stampa della sequenza sulle transizioni rimanenti.

Stampa delle sequenze:

- termina se non ci sono più sequenze nella lista delle sequenze;
- se la frequenza del rumore corrente è uguale ad 1 allora applica del rumore alla sequenza di testa della lista delle sequenze, stampa la transizione di testa della sequenza con rumore ottenuta su stream di output, stampa il resto della sequenza e richiama la stampa delle sequenze sulle sequenze rimanenti e su una frequenza del rumore corrente ripristinato alla frequenza iniziale;
- per gli altri casi stampa la transizione di testa della sequenza di testa della lista delle sequenze su stream di output, stampa il resto della sequenza, decrementa la frequenza del rumore corrente di 1 e richiama la stampa delle sequenze sulle sequenze rimanenti e sulla frequenza corrente decrementata;

L'accorpamento delle due parti è il seguente.

Start:

- consulta il file di input “wf-net-in”, genera 1000 sequenze seguendo il modello appena importato,
 - se la percentuale di rumore è 0 allora stampa le sequenze create applicando un rumore con frequenza pari a 0;
 - se la percentuale di rumore non è 0 allora calcola la frequenza con cui applicare il rumore alle sequenze dividendo 100 per la percentuale di rumore, e stampa le sequenze create applicando un rumore con la frequenza calcolata;infine cancella tutto ciò che è stato asserito durante l'esecuzione per permettere ulteriori esecuzioni successive.

Rispetto all'implementazione del sistema, per questa soluzione è stata utilizzata in aggiunta la libreria standard “random”, che fornisce degli strumenti per ottenere dei valori numerici casuali.

COMPARATORE DI MODELLI

È stato realizzato un algoritmo per confrontare automaticamente il modello di workflow ottenuto dal sistema (salvato nel file “wf-net-out”) con il modello di workflow di originale (salvato nel file “wf-net-in”) allo scopo di valutarne la completezza e la correttezza.

Per la rappresentazione dei due modelli di workflow la scelta è ricaduta obbligatoriamente sugli insiemi di fatti raffiguranti gli archi in entrata ed un uscita, ossia sulla rappresentazione adottata per il file di output del sistema realizzato nonché formalismo adottato per il modello iniziale su cui si basa il generatore di sequenze:

in(PostoPrecedente, Transizione), out(Transizione, PostoSuccessivo).

Come spiegato nell'introduzione, i concetti di completezza e correttezza si basano sul concetto di copertura tra split e join dei due modelli. Di conseguenza si rende indispensabile scegliere una rappresentazione adeguata di entrambe le tipologie di questi due oggetti. Le opzioni individuate sono 2:

1. `or_split(A,[[B1,B2],[B3,...],...]), and_split(A,[[B1,B2],[B3,...],...]),`
`or_join(A,[[B1,B2],[B3,...],...]), and_join(A,[[B1,B2],[B3,...],...]).`
2. `[[A1,[[B1,B2],[B3,...],...]],[A2,[[B1,B2],[B3,...],...],...]] e [[A1,[[B1,B2],[B3,...],...]],[A2,[[B1,B2],[B3,...],...],...]] e`
`[[A1,[[B1,B2],[B3,...],...]],[A2,[[B1,B2],[B3,...],...],...]] e [[A1,[[B1,B2],[B3,...],...]],[A2,[[B1,B2],[B3,...],...],...]].`

La scelta è da effettuare tra 4 insiemi di fatti raffiguranti OR-split, AND-split, OR-join e AND-join rispetto ad una singola transizione, e 4 liste, ognuna contenente tutti gli split o join di una data tipologia. La scelta è ricaduta sul secondo formalismo in quanto, trattandosi di un confronto, da un lato assicura che non ci sia confusione tra oggetti del primo modello e oggetti del secondo modello, e dall'altro lato risulta più adeguato per un accesso sequenziale, necessario visto ogni oggetto dovrà essere analizzato.

Di seguito si riporta la soluzione trovata per la comparazione dei due modelli.

Prelevamento delle transizioni di un modello:

- termina se non ci sono più archi nell'insieme degli archi di tipo in, e in tal caso restituisce l'insieme delle transizioni vuoto;
- per gli altri casi richiama il prelevamento delle transizioni di un modello sull'insieme degli archi di tipo in escludendo la testa; alle transizioni ottenute verrà aggiunta in testa la transizione dell'arco di testa dell'insieme degli archi di tipo in.

SPERIMENTAZIONE

Prelevamento delle transizioni di destinazione di un AND-split:

- termina se non ci sono più posti nell'insieme dei posti coinvolti nell'AND-split, e in tal caso restituisce la lista delle transizioni di destinazione vuota;
- per gli altri casi ottiene la lista delle transizioni di destinazione degli archi in uscita dal posto di testa dell'insieme dei posti coinvolti nell'AND-split, richiama il prelevamento delle transizioni di destinazione di un AND-split sui posti coinvolti rimanenti e restituisce la concatenazione delle transizioni di destinazione ottenute dai due passi precedenti.

Ritrovamento di tutti gli AND-split:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, e in tal caso restituisce la lista degli AND-split vuota;
- per gli altri casi ottiene la lista dei posti di destinazione degli archi in uscita dalla transizione di testa della lista delle transizioni distinte, se questa lista conta più di un posto allora si prelevano le transizioni di destinazione per ogni posto ottenuto in modo da ottenere l'AND-split corrente, e infine richiama il ritrovamento di tutti gli AND-split sulle transizioni distinte rimanenti; agli AND-split ottenuti verrà aggiunto in testa l'AND-split corrente;
- se la lista dei posti di destinazione non conta più di un posto allora richiama il ritrovamento di tutti gli AND-split sulle transizioni distinte escludendo la testa; gli AND-split ottenuti verranno restituiti senza modifiche.

Ritrovamento degli OR-split a partire da alcuni posti di partenza:

- termina se non ci sono più posti nella lista dei posti di partenza, e in tal caso restituisci la lista degli OR-split vuota;
- per gli altri casi ottiene la lista delle transizioni di destinazione degli archi in uscita dal posto di testa della lista dei posti di partenza, se questa lista conta più di una transizione allora queste transizioni partecipano all'OR-split corrente, e infine richiama il ritrovamento degli OR-split a partire dai posti di partenza rimanenti; agli OR-split ottenuti verrà aggiunto in testa l'OR-split corrente;
- se la lista delle transizioni di destinazione non conta più di un posto allora richiama il ritrovamento degli OR-split a partire dai posti di partenza escludendo la testa; gli OR-split ottenuti verranno restituiti senza modifiche.

SPERIMENTAZIONE

Ritrovamento di tutti gli OR-split:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, e in tal caso restituisce la lista degli OR-split vuota;
- per gli altri casi ottiene la lista dei posti di destinazione degli archi in uscita dalla transizione di testa della lista delle transizioni distinte, ottiene una lista di OR-split a partire dalla lista di posti ottenuta precedentemente, se questa lista conta almeno un posto allora richiama il ritrovamento di tutti gli OR-split sulle transizioni distinte rimanenti; la lista degli OR-split restituita sarà una concatenazione degli OR-split ritrovati negli ultimi due passi.
- se la lista degli OR-split non conta almeno un posto allora richiama il ritrovamento di tutti gli OR-split sulle transizioni distinte escludendo la testa; gli OR-split ottenuti verranno restituiti senza modifiche.

Prelevamento delle transizioni di partenza di un AND-join:

- termina se non ci sono più posti nell'insieme dei posti coinvolti nell'AND-join, e in tal caso restituisce la lista delle transizioni di partenza vuota;
- per gli altri casi ottiene la lista delle transizioni di partenza degli archi in entrata al posto di testa dell'insieme dei posti coinvolti nell'AND-join, richiama il prelevamento delle transizioni di partenza di un AND-join sui posti coinvolti rimanenti e restituisce la concatenazione delle transizioni di destinazione ottenute dai due passi precedenti.

Ritrovamento di tutti gli AND-join:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, e in tal caso restituisce la lista degli AND-join vuota;
- per gli altri casi ottiene la lista dei posti di partenza degli archi in entrata alla transizione di testa della lista delle transizioni distinte, se questa lista conta più di un posto allora si prelevano le transizioni di partenza per ogni posto ottenuto in modo da ottenere l'AND-join corrente, e infine richiama il ritrovamento di tutti gli AND-join sulle transizioni distinte rimanenti; agli AND-join ottenuti verrà aggiunto in testa l'AND-join corrente;
- se la lista dei posti di partenza non conta più di un posto allora richiama il ritrovamento di tutti gli AND-join sulle transizioni distinte escludendo la testa; gli AND-join ottenuti verranno restituiti senza modifiche.

SPERIMENTAZIONE

Ritrovamento degli OR-join a partire da alcuni posti di destinazione:

- termina se non ci sono più posti nella lista dei posti di destinazione, e in tal caso restituisci la lista degli OR-join vuota;
- per gli altri casi ottiene la lista delle transizioni di partenza degli archi in entrata al posto di testa della lista dei posti di destinazione, se questa lista conta più di una transizione allora queste transizioni partecipano all'OR-join corrente, e infine richiama il ritrovamento degli OR-join a partire dai posti di destinazione rimanenti; gli OR-join ottenuti verrà aggiunto in testa l'OR-join corrente;
- se la lista delle transizioni di partenza non conta più di un posto allora richiama il ritrovamento degli OR-join a partire dai posti di destinazione escludendo la testa; gli OR-join ottenuti verranno restituiti senza modifiche.

Ritrovamento di tutti gli OR-join:

- termina se non ci sono più transizioni nella lista delle transizioni distinte, e in tal caso restituisce la lista degli OR-join vuota;
- per gli altri casi ottiene la lista dei posti di partenza degli archi in entrata alla transizione di testa della lista delle transizioni distinte, ottiene una lista di OR-join a partire dalla lista di posti ottenuta precedentemente, se questa lista conta almeno un posto allora richiama il ritrovamento di tutti gli OR-join sulle transizioni distinte rimanenti; la lista degli OR-join restituita sarà una concatenazione degli OR-join ritrovati negli ultimi due passi.
- se la lista degli OR-join non conta almeno un posto allora richiama il ritrovamento di tutti gli OR-join sulle transizioni distinte escludendo la testa; gli OR-join ottenuti verranno restituiti senza modifiche.

Controllo di uno split (join) tra gli split (join) di paragone:

- termina se non ci sono più split (join) nella lista degli split (join) di paragone, e in tal caso restituisce lo split (join) corrente come split (join) non coperto;
- termina se, confrontando la lista ordinata di transizioni dello split (join) in esame e la lista ordinata di transizioni dello split (join) di testa della lista degli split (join) di paragone, queste risultino identiche (se il confronto va a buon fine non deve essere possibile tornare su questa decisione in quanto gli ordinamento non fornirebbe altri valori validi); in tal caso restituisce la lista vuota come split (join) non coperto;
- se il confronto non va a buon fine si richiama il controllo di uno split (join) tra gli split (join) di paragone escludendo la testa; lo split (join) non coperto verrà restituito senza modifiche.

SPERIMENTAZIONE

Comparazione tra una lista di split (join) ed un'altra di paragone:

- termina se non ci sono più split (join) nella lista di split (join) in esame, e in tal caso restituisce la lista di split (join) non coperti vuota;
- per gli altri casi effettua il controllo dello split (join) di testa della lista di split (join) in esame tra gli split (join) di paragone, se lo split (join) non coperto ottenuto non è una lista vuota allora sarà lo split (join) non coperto corrente; infine richiama la comparazione tra una lista di split (join) ed un'altra di paragone sugli split (join) in esame rimanenti; agli split (join) non coperti ottenuti aggiungerà lo split (join) non coperto corrente in testa;
- se lo split (join) non coperto ottenuto è una lista vuota allora richiama la comparazione tra una lista di split (join) ed un'altra di paragone sugli split (join) in esame escludendo la testa; gli split (join) non coperti ottenuti verranno restituiti senza alcuna modifica.

Stampa di una lista di transizioni:

- termina se non ci sono più transizioni nella lista di transizioni;
- per gli altri casi stampa la transizione di testa della lista di transizioni seguita da uno spazio e richiama la stampa di una lista di transizioni sulle transizioni rimanenti.

Stampa della lista di AND-split non coperti:

- termina se non ci sono più AND-split nella lista degli AND-split non coperti;
- per gli altri casi stampa la scritta "Divisione AND", il nome della transizione di partenza, una freccia, la lista delle transizioni di destinazione e la scritta "non coperta"; infine richiama la stampa della lista di AND-split non coperti sugli AND-split rimanenti.

Stampa della lista di OR-split non coperti:

- termina se non ci sono più OR-split nella lista degli OR-split non coperti;
- per gli altri casi stampa la scritta "Divisione OR", il nome della transizione di partenza, una freccia, la lista delle transizioni di destinazione e la scritta "non coperta"; infine richiama la stampa della lista di OR-split non coperti sugli OR-split rimanenti.

SPERIMENTAZIONE

Stampa della lista di AND-join non coperti:

- termina se non ci sono più AND-join nella lista degli AND-join non coperti;
- per gli altri casi stampa la scritta “Unione AND”, la lista delle transizioni di partenza, una freccia, il nome della transizione di destinazione e la scritta “non coperta”; infine richiama la stampa della lista di AND-join non coperti sugli AND-join rimanenti.

Stampa della lista di OR-join non coperti:

- termina se non ci sono più OR-join nella lista degli OR-join non coperti;
- per gli altri casi stampa la scritta “Unione OR”, la lista delle transizioni di partenza, una freccia, il nome della transizione di destinazione e la scritta “non coperta”; infine richiama la stampa della lista di OR-join non coperti sugli OR-join rimanenti.

Start:

- esegue questi passi prima per il file “wf-net-in” e dopo per il file “wf-net-out” :
 - consulta il file,
 - ottiene una lista di tutti gli archi di tipo in,
 - preleva le transizione e rimuove i duplicati per ottenere le transizioni distinte,
 - ottiene una lista per ogni tipologia di split e join,
 - somma le lunghezze delle liste per ottenere il numero totale di split e join,
 - cancella tutto ciò che è stato asserito durante la consultazione del file,
- per valutare la completezza:
 - per ogni tipologia di split e join compara la lista ottenuta dal primo file usando la lista del secondo file come lista di paragone, ottenendo gli split e i join del modello originale non coperti dal modello rigenerato,
 - stampa gli split e i join non coperti,
 - calcola e stampa la percentuale di split e join coperti rispetto al numero totale
- per valutare la completezza:
 - per ogni tipologia di split e join compara la lista ottenuta dal secondo file usando la lista del primo file come lista di paragone, ottenendo gli split e i join del modello rigenerato non coperti dal modello originale,
 - stampa gli split e i join non coperti,
 - calcola e stampa la percentuale di split e join coperti rispetto al numero totale

SPERIMENTAZIONE

ESPERIMENTO 1

Il primo esperimento è stato effettuato sul seguente modello di workflow.

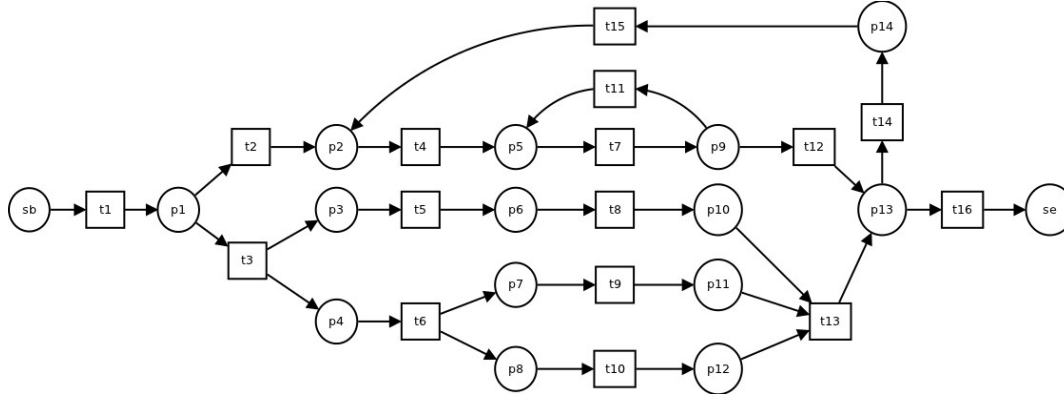


Immagine 1: Modello di workflow per l'esperimento 1

Questo modello presenta le seguenti caratteristiche:

- 2 AND-split:
 - da t3 alla coppia {t5,t6};
 - da t6 alla coppia {t9,t10};
- 4 OR-split:
 - da t1 alla coppia {t2,t3};
 - da t7 alla coppia {t11,t12};
 - da t12 alla coppia {t14,t16};
 - da t13 alla coppia {t14,t16};
- 1 AND-join:
 - dalla tripla {t8,t9,t10} a t13;
- 4 OR-join:
 - dalla coppia {t2,t15} a t4;
 - dalla coppia {t4,t11} a t7;
 - dalla coppia {t12,t13} a t14;
 - dalla coppia {t12,t13} a t16;
- 1 grande ciclo da p13 a p2 passando per t14, p14 e t15;
- 1 piccolo ciclo da p9 a p5 passando per t11.

SPERIMENTAZIONE

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 11 su 11 | 11 su 11 | 100% | 100% |
| 5 | 9 su 11 | 9 su 9 | 82% | 100% |
| 10 | 9 su 11 | 9 su 11 | 82% | 82% |

Tabella 1: Risultati dell'esperimento 1

Il modello ottenuto dalle sequenze senza rumore corrisponde in pieno al modello di partenza. Tutte le connessioni sono state individuate dalla euristica 1, tranne la connessione da t7 a t11 e quella da t11 a t7, individuate invece dalla euristica 3: ciò era prevedibile dato che queste due transizioni sono impegnate nel piccolo ciclo sopra descritto e l'euristica 3 è stata ideata proprio per identificare queste situazioni.

Il modello ottenuto dalle sequenze con il 5% di rumore corrisponde solo in parte al modello di partenza. Di seguito si mostra il risultato del processo di Process Mining.

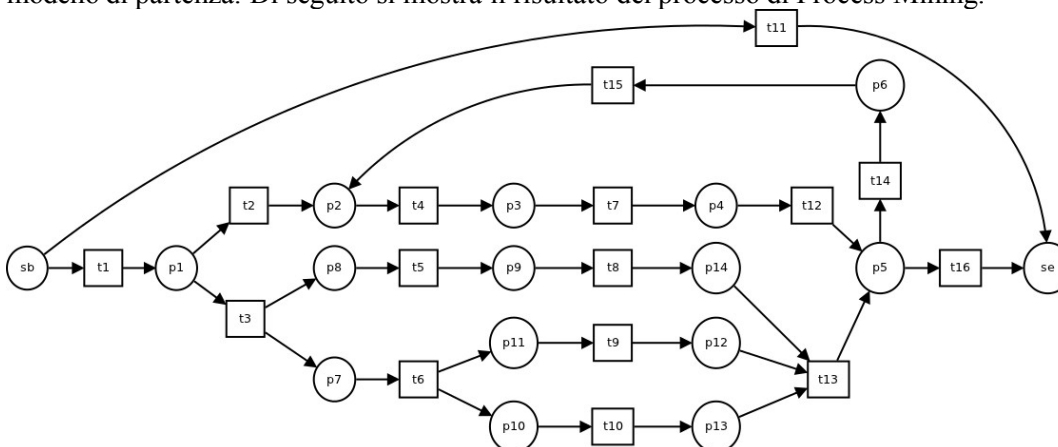


Immagine 2: Modello di workflow ottenuto dalle sequenze con il 5% di rumore

Come si può notare c'è una mancanza riguardante il piccolo ciclo in cui dovrebbero essere impegnate le transizioni t7 e t11. Questa volta infatti l'euristica 3 non è riuscita ad individuare una connessione tra queste due transizioni, ed il sistema, non riuscendo ad individuare altre connessioni entranti o uscenti da t11, ha fornito erroneamente l'unica soluzione possibile, ossia che t11 fosse connesso sia al posto di ingresso che a quello di uscita. Queste connessioni errate però non sono da attribuire alla tecnica ma alla sua implementazione, dunque non verranno considerate come errori di correttezza. Al contrario, la mancata individuazione del piccolo ciclo è un errore dovuto alla euristica 3, quindi viene considerato come errore di completezza.

SPERIMENTAZIONE

L'ultimo modello è quello ottenuto dalle sequenze con il 10% di rumore. Questo presenta diversi errori rispetto al modello di partenza.

Di seguito si mostra il risultato del processo di Process Mining.

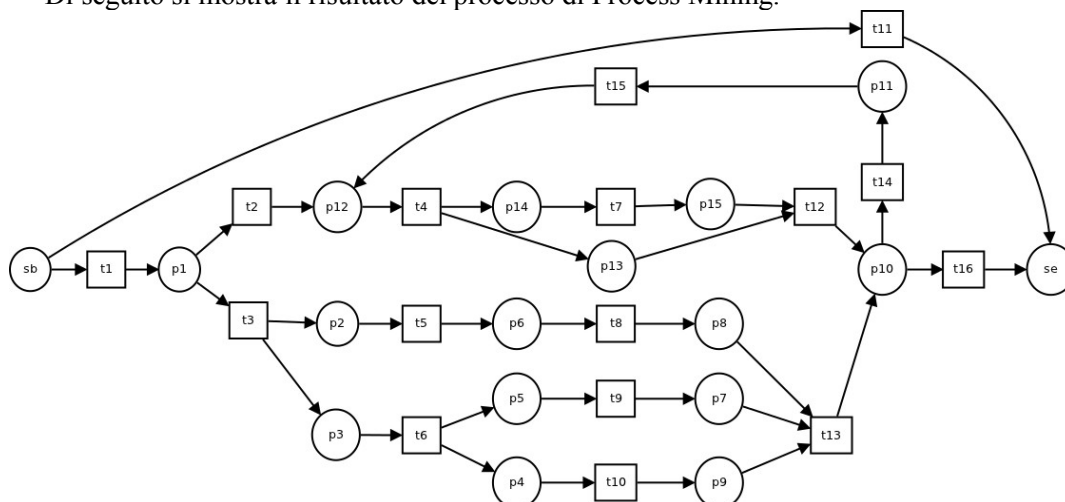


Immagine 3: Modello di workflow ottenuto dalle sequenze con il 10% di rumore

Nella parte inferiore del modello così come per la parte riguardante il grande ciclo non ci sono né connessioni mancanti né connessioni scorrette. Ancora una volta la parte interessata dagli errori è quella inerente al piccolo ciclo. Le considerazioni fatte per la parte precedente dell'esperimento valgono anche per questa parte, ragione per cui si spiegano i due punti in meno riguardanti la completezza. Questa volta inoltre, il modello ottenuto presenta 2 errori di correttezza sempre nella zona del piccolo ciclo. È stata individuata erroneamente una connessione diretta dalla transizione t4 alla transizione t12 dovuta all'euristica 1.

L'ultimo problema descritto è da imputare senza dubbio al rumore infatti, se si richiama il modello di Process Mining fissando il fattore di rumore al 10% sulle stesse sequenze appena analizzate, il risultato è esattamente lo stesso di quello ottenuto analizzando le sequenze con il 5% di errore: migliora la correttezza ma non la completezza.

Se da un lato per questo esperimento si sono ottenuti ottimi risultati per quanto riguarda le sequenze con assenza di rumore, dall'altro lato l'introduzione di rumore ha mandato in crisi la tecnica.

SPERIMENTAZIONE

ESPERIMENTO 2

Il secondo esperimento è stato effettuato sul seguente modello di workflow.

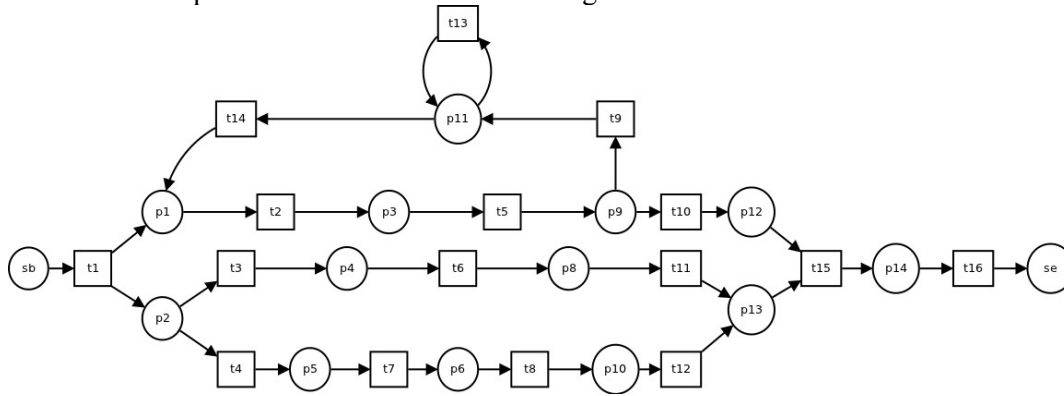


Immagine 4: Modello di workflow per l'esperimento 2

Questo modello presenta le seguenti caratteristiche:

- 1 AND-split:
 - da t1 alla tripla {t2,t3,t4};
- 4 OR-split:
 - da t1 alla coppia {t3,t4};
 - da t5 alla coppia {t9,t10};
 - da t9 alla coppia {t13,t14};
 - da t13 alla coppia {t13,t14};
- 1 AND-join:
 - dalla tripla {t10,t11,t12} a t15;
- 4 OR-join:
 - dalla coppia {t1,t14} a t2;
 - dalla coppia {t11,t12} a t15;
 - dalla coppia {t19,t13} a t13;
 - dalla coppia {t19,t13} a t14;
- 1 grande ciclo da p9 a p1 passando per t19, p11, t13 e t14;
- 1 ricorsione su t13 passando da p11.

SPERIMENTAZIONE

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 6 su 10 | 6 su 14 | 60% | 43% |
| 5 | 6 su 10 | 6 su 17 | 60% | 35% |
| 10 | 5 su 10 | 5 su 13 | 50% | 38% |

Tabella 2: Risultati dell'esperimento 2

Il modello ottenuto dalle sequenze senza rumore presenta già diverse imprecisioni. Di seguito si mostra il risultato del processo di Process Mining.

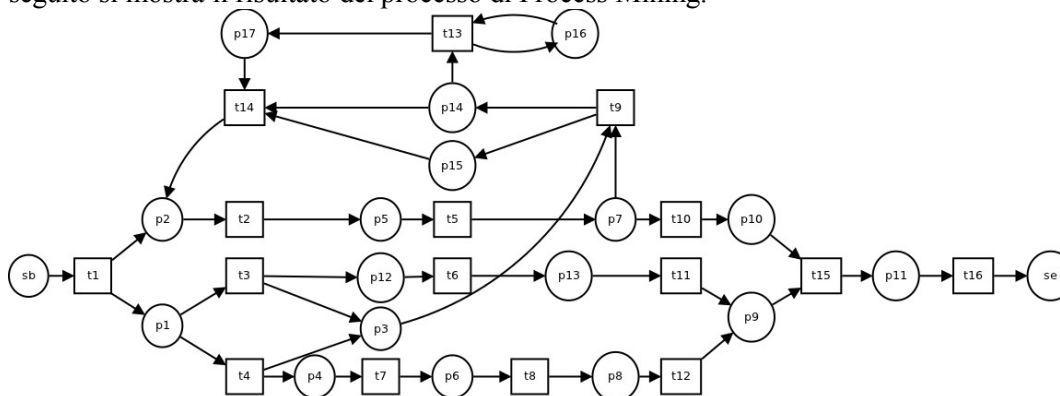


Immagine 5: Modello di workflow ottenuto dalle sequenze con lo 0% di rumore

In questo caso c'è una complicità di errori da parte sia delle euristiche per la induzione del D/F-graph che da parte dell'algoritmo per il riconoscimento della tipologia di split e join.

In particolare, l'OR-join dalla coppia $\{t3, t4\}$ a $t9$ è dovuta all'euristica 1, che non è riuscita a riconoscere la casualità dei valori di sequenza dirette tra le transizioni $[t3, t9]$ e $[t4, t9]$, interpretandole come un indice di causalità. Questa connessione rende il modello non valido in quanto si possono creare condizioni di non terminazione, e questo va contro il principio di soundness.

Tutti gli errori della parte superiore invece sono dovuti all'algoritmo di riconoscimento. Anche in questo caso i valori della D/F-table risultano ingannevoli e l'algoritmo si rivela essere poco efficace.

C'è da segnalare però che l'euristica 2 è riuscita correttamente a individuare la ricorsione; purtroppo anche questa è stata male interpretata dall'algoritmo di riconoscimento, contribuendo all'invalidazione del modello (l'attivazione di $t13$ richiede un token in $p16$, il quale può riceverlo solo da $t13$; questo dunque diventa un punto morto).

SPERIMENTAZIONE

I modelli ottenuti dalle sequenze con il 5% e con il 10% di rumore presentano molti errori. Di seguito si mostra il risultato di entrambi i processi di Process Mining.

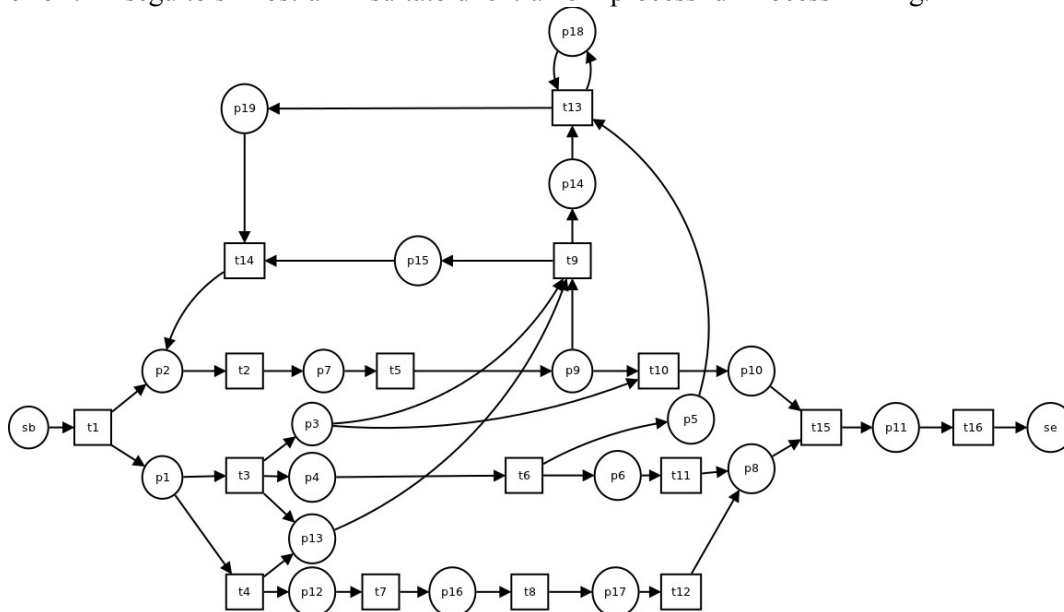


Immagine 6: Modello di workflow ottenuto dalle sequenze con il 5% di rumore

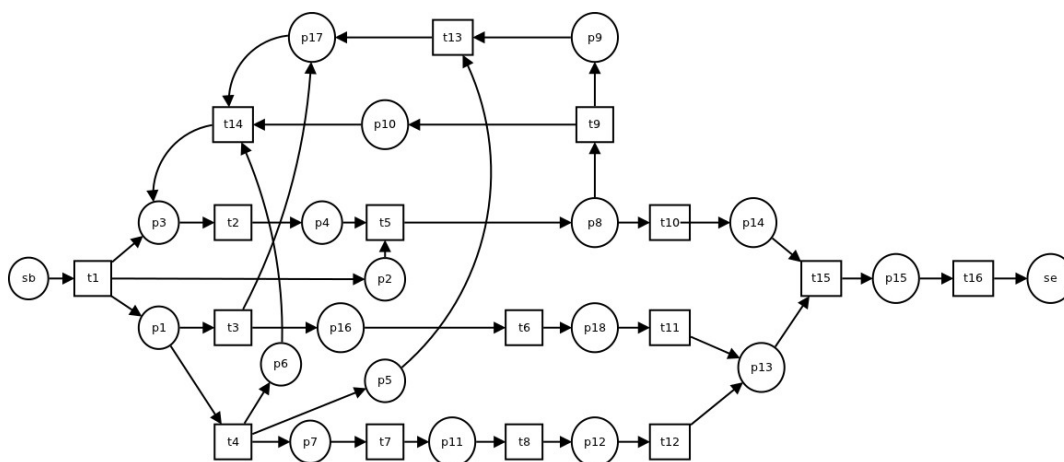


Immagine 7: Modello di workflow ottenuto dalle sequenze con il 10% di rumore

I problemi riscontrati nella parte precedente dell'esperimento vengono ulteriormente amplificati con l'aggiunta di rumore nelle sequenze. Rispetto ai risultati precedenti, questi mostrano anche l'inadeguatezza, in caso di rumore nei dati di input, della euristica 2 dedicata al riconoscimento delle ricorsioni,.

I risultati di tutto l'esperimento risultano quindi totalmente inaccettabili.

ESPERIMENTO 3

Il terzo esperimento è stato effettuato sul seguente modello di workflow.

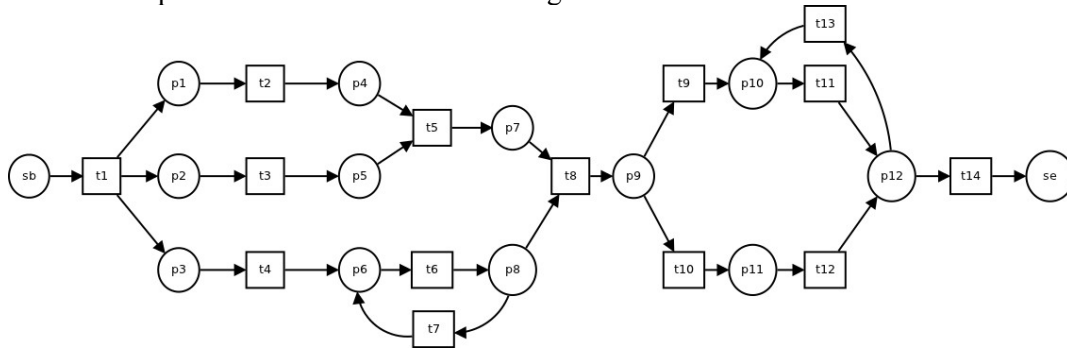


Immagine 8: Modello di workflow per l'esperimento 3

Questo modello presenta le seguenti caratteristiche:

- 1 AND-split:
 - da t1 alla tripla {t2,t3,t4};
- 4 OR-split:
 - da t6 alla coppia {t7,t8};
 - da t8 alla coppia {t9,t10};
 - da t10 alla coppia {t13,t14};
 - da t11 alla coppia {t13,t14};
- 2 AND-join:
 - dalla coppia {t2,t3} a t5;
 - dalla coppia {t5,t6} a t7;
- 4 OR-join:
 - dalla coppia {t4,t7} a t6;
 - dalla coppia {t9,t13} a t11;
 - dalla coppia {t11,t12} a t13;
 - dalla coppia {t11,t12} a t14;
- 2 piccoli cicli:
 - da p8 a p6 passando per t7;
 - da p12 a p10 passando per t13.

SPERIMENTAZIONE

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 6 su 11 | 6 su 8 | 55% | 75% |
| 5 | 6 su 11 | 6 su 8 | 55% | 75% |
| 10 | 8 su 11 | 8 su 8 | 73% | 100% |

Tabella 3: Risultati dell'esperimento 3

I modelli ottenuti dalle sequenze con 0% e 5% di rumore presentano entrambe le stesse imprecisioni. Di seguito si mostra il risultato dei processi di Process Mining.

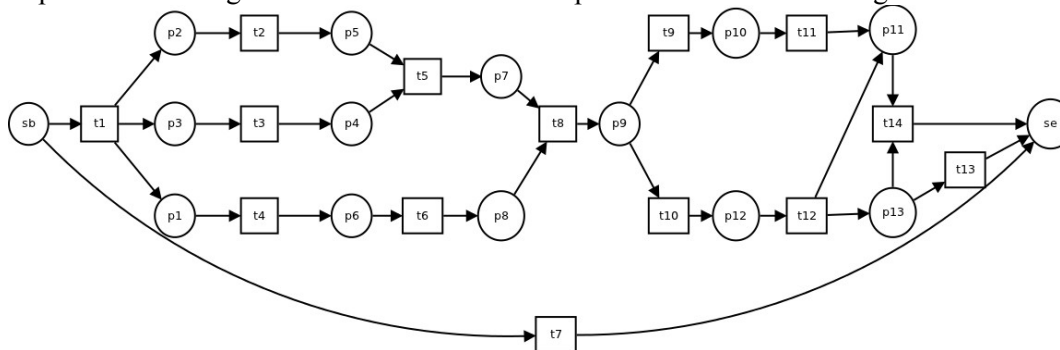


Immagine 9: Modello di workflow ottenuto dalle sequenze con lo 0% e il 5% di rumore

Rispetto al primo esperimento, questa volta la tecnica non è precisa per quanto riguarda i piccoli cicli. L'euristica 3 infatti non è stata attivata né sul ciclo da p8 a p6, né sul ciclo da p12 a p10.

Nel primo caso, come successo nel primo esperimento in caso di rumore, la transizione coinvolta nel ciclo non è stata associata ad alcuno split o join. In questo modo il sistema vi ha connesso i posti iniziali e finali. Come detto precedentemente questo non può essere considerato un errore della tecnica, ma dell'implementazione. Il mancato riconoscimento del ciclo invece viene considerato come errore sulla completezza.

Nel secondo caso invece, il ciclo si trova in una posizione più critica, in quanto parte da un posto di snodo tra diversi OR-split e OR-join. Il pessimo risultato raggiunto dimostra che la tecnica ha seri problemi nell'identificare cicli, in particolar modo in situazioni complesse. Per questo motivo quindi, oltre all'indice di completezza viene decrementato anche l'indice di correttezza.

SPERIMENTAZIONE

Il modello ottenuto dalle sequenze con il 10% di rumore presenta paradossalmente meno errori dei precedenti. Di seguito si mostra il risultato del processo di Process Mining.

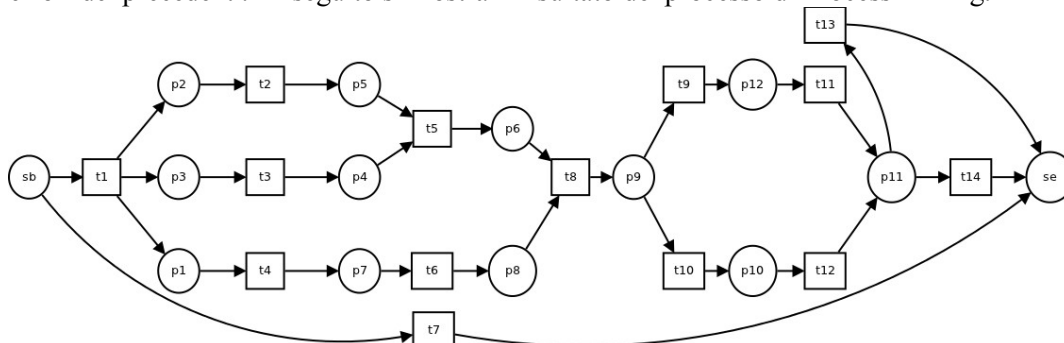


Immagine 10: Modello di workflow ottenuto dalle sequenze con il 10% di rumore

Come si può notare, ancora una volta i problemi di correttezza derivano dalla mancata attivazione della euristica 3: entrambi i piccoli cicli non vengono riconosciuti, o vengono riconosciuti solo in parte come per il caso di t13.

Il miglioramento dei risultati rispetto alle prove precedenti è facilmente spiegabile: la maggior parte degli errori precedenti ricadevano sulla interpretazione forzata del piccolo cicli comprendente t13. In questo caso, l'incremento del rumore, in particolare delle sequenze con rumore di tipo 2 (rimozione della coda di una sequenza), ha portato la D/F-table ad avere valori meno alti in quella zona del modello. Di conseguenza, invece di forzare una interpretazione, la tecnica collega t13 all'unico posto sicuro (il posto di ingresso), evitando di collegare t13 ad altri posti in uscita. Il collegamento al posto finale, come detto in precedenza, è da imputarsi all'implementazione.

Sebbene quest'ultimo risultato sia relativamente positivo, i modelli ottenuti sulle sequenze con lo 0% ed il 5% portano a concludere che anche i risultati di questo esperimento sono da ritenersi inaccettabili.

SPERIMENTAZIONE

ESPERIMENTO 4

Il quarto esperimento è stato effettuato sul seguente modello di workflow.

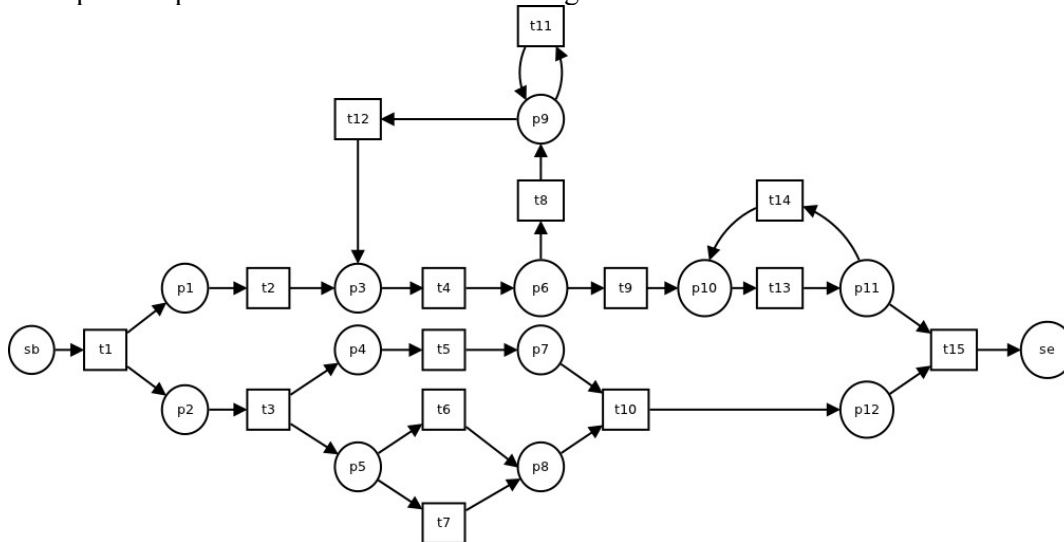


Immagine 11: Modello di workflow per l'esperimento 4

Questo modello presenta le seguenti caratteristiche:

- 2 AND-split:
 - da t1 alla coppia {t2,t3};
 - da t3 alla tripla {t5,t6,t7};
- 5 OR-split:
 - da t3 alla coppia {t6,t7};
 - da t4 alla coppia {t8,t9};
 - da t8 alla coppia {t11,t12};
 - da t11 alla coppia {t11,t12};
 - da t13 alla coppia {t14,t15};
- 2 AND-join:
 - dalla tripla {t5,t6,t7} a t10;
 - dalla coppia {t10,t13} a t15;

SPERIMENTAZIONE

- 5 OR-join:
 - dalla coppia {t2,t12} a t14;
 - dalla coppia {t6,t7} a t10;
 - dalla coppia {t8,t11} a t11;
 - dalla coppia {t8,t11} a t12;
 - dalla coppia {t9,t14} a t13;
- 1 grande ciclo da p6 a p3 passando per t8, p9, t11 e t12;
- 1 piccolo ciclo da p11 a p10 passando per t14;
- 1 ricorsione su t11 passando per p9.

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 7 su 14 | 7 su 16 | 50% | 44% |
| 5 | 7 su 14 | 7 su 19 | 50% | 37% |
| 10 | 7 su 14 | 7 su 19 | 50% | 37% |

Tabella 4: Risultati dell'esperimento 4

Il modello ottenuto dalle sequenze senza rumore presenta già diverse imprecisioni. Di seguito si mostra il risultato del processo di Process Mining.

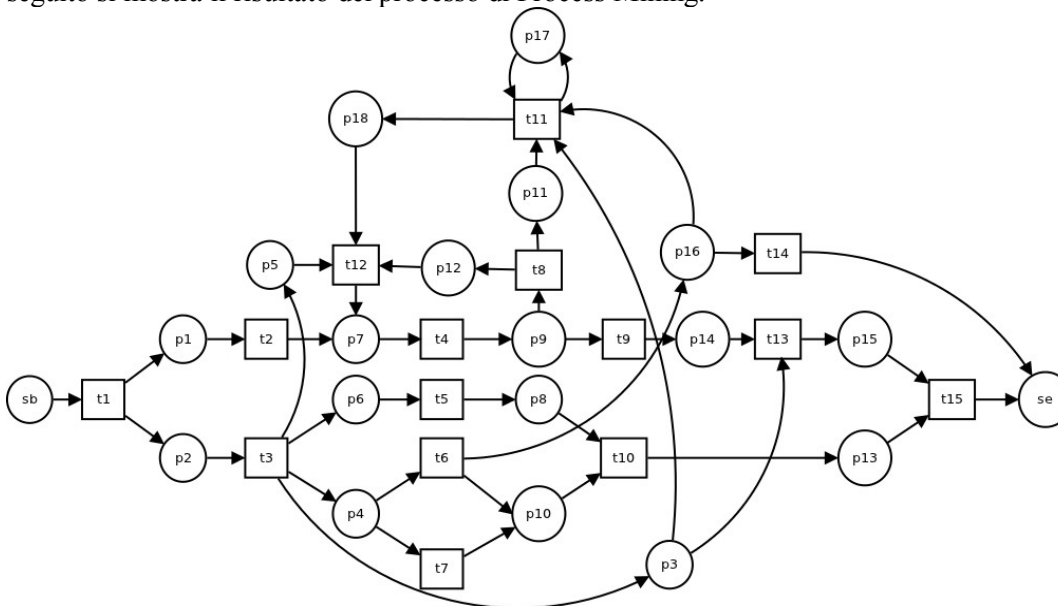


Immagine 12: Modello di workflow ottenuto dalle sequenze con lo 0% di rumore

SPERIMENTAZIONE

In questo caso c'è un fallimento completo sull'efficacia delle tre euristiche. Non sono, come nei casi precedenti, le euristiche 2 e 3 non sono riuscite a individuare rispettivamente la ricorsione e il piccolo ciclo, ma persino l'euristica 1 si è rivelata inefficace individuando archi inesistenti. Particolarmente evidenti sono gli archi uscenti da t3 e t6 verso la parte alta del modello.

Questi risultati dimostrano che in presenza di un alto tasso di concorrenza nel modello iniziale, la tecnica va in crisi anche sulla euristica che, fino ad ora, ha dimostrato di essere la più robusta.

Si evidenzia anche per questo esperimento che il modello risultante non è un modello valido, sia perché la ricorsione è vittima delle considerazioni fatte per gli altri esperimenti, ma in più, in questo caso, si notano molte connessioni tra due parti divise da un AND-split, le quali porterebbero in molti casi a condizioni di non terminazione e quindi ad un mancato rispetto del requisito di soundness.

I modelli ottenuti dalle sequenze con il 5% e con il 10% risultano essere molto simili e con qualche errore in più rispetto al modello precedente. Di seguito si mostra il risultato di entrambi i processi di Process Mining.

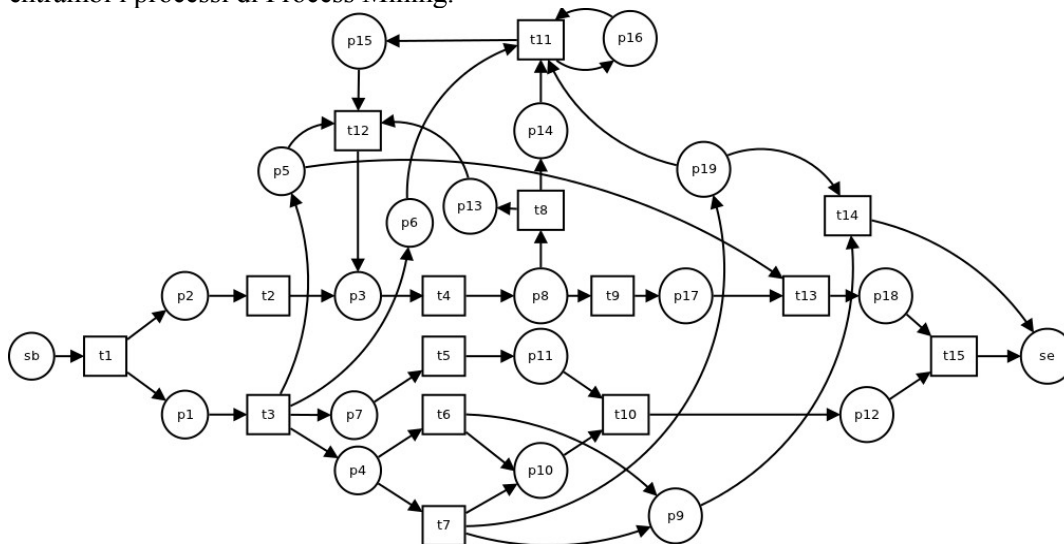


Immagine 13: Modello di workflow ottenuto dalle sequenze con il 5% di rumore

Le connessioni tra le due parti dell'AND-split continuano ad aumentare rendendo il modello risultante ancora meno corretto di quello precedente. Le considerazioni fatte in precedenza valgono anche qui, ma in questo caso il rumore incrementa l'inefficacia delle euristiche.

In conclusione anche i risultati di questo esperimento sono da ritenersi inaccettabili.

SPERIMENTAZIONE

ESPERIMENTO 5

Il quinto esperimento è stato effettuato sul seguente modello di workflow.

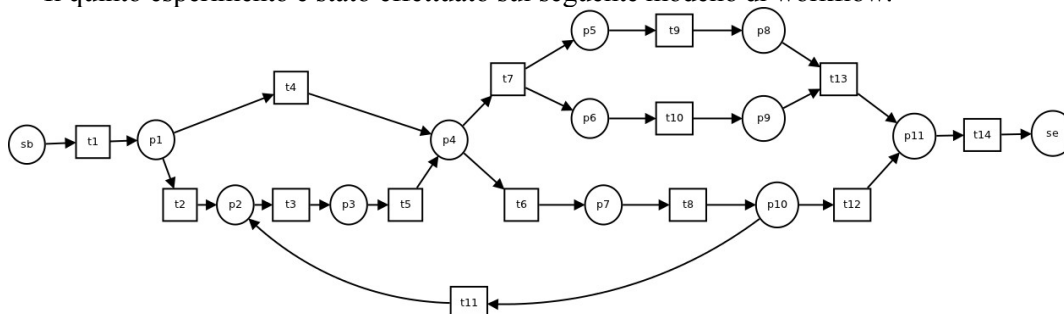


Immagine 14: Modello di workflow per l'esperimento 5

Questo modello presenta le seguenti caratteristiche:

- 1 AND-split:
 - da t7 alla coppia {t9,t10};
- 4 OR-split:
 - da t1 alla coppia {t2,t4};
 - da t4 alla coppia {t6,t7};
 - da t5 alla coppia {t6,t7};
 - da t8 alla coppia {t11,t12};
- 1 AND-join:
 - dalla coppia {t9,t10} a t13;
- 4 OR-join:
 - dalla coppia {t2,t11} a t3;
 - dalla coppia {t4,t5} a t6;
 - dalla coppia {t4,t5} a t7;
 - dalla coppia {t12,t13} a t14;
- 1 grande ciclo da p10 a p2 passando per t11;

SPERIMENTAZIONE

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 10 su 10 | 10 su 10 | 100% | 100% |
| 5 | 10 su 10 | 10 su 10 | 100% | 100% |
| 10 | 9 su 10 | 9 su 13 | 90% | 69% |

Tabella 5: Risultati dell'esperimento 5

I modelli ottenuti dalle sequenze con lo 0% ed il 5% di rumore corrispondono in pieno al modello di partenza. Questo modello di partenza è molto simile al modello di esempio dell'articolo in cui viene presentata la tecnica: non presenta né piccoli cicli né ricorsioni. Questo indica che, nel riconoscere il modello, la tecnica non ha dovuto utilizzare le euristiche 2 e 3.

Il modello ottenuto dalle sequenze con il 10% di rumore corrisponde solo in parte al modello di partenza. Di seguito si mostra il risultato del processo di Process Mining.

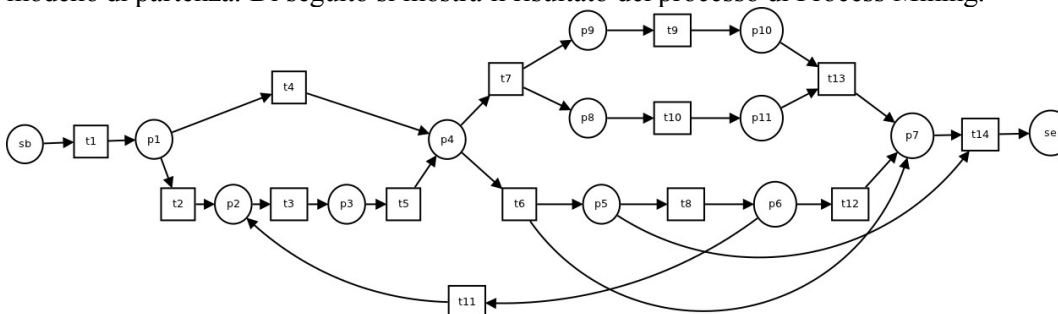


Immagine 15: Modello di workflow ottenuto dalle sequenze con il 10% di rumore

Come si può notare ci sono dei problemi nella parte inferiore destra relativi a 2 connessioni aggiunte erroneamente da t6 verso t14. Dalla D/F-table si nota come tra t6 e t14 compaiono erroneamente diverse connessioni dirette: queste vengono interpretate dall'euristica 1 come indice di causalità e quindi di una nuova connessione. Questo è senza dubbio riconducibile all'aumento del rumore nelle sequenze, in particolare alle sequenze con rumore di tipo 3 (rimozione del corpo di una sequenza).

Se si aumento il fattore di errore dal 5% al 10% nel processo di Process Mining, il modello risultante dalle sequenze con il 10% di rumore torna ad essere pienamente corrispondente al modello di partenza.

Questo esperimento dimostra che, nei casi in cui il riconoscimento richiede l'utilizzo della sola euristica 1, la tecnica si rivela essere efficace e robusta anche per alte soglie di rumore nei dati di input. Pertanto, i risultati di questo esperimento, sono da ritenersi pienamente accettabili.

SPERIMENTAZIONE

ESPERIMENTO 6

Il quinto esperimento è stato effettuato sul seguente modello di workflow.

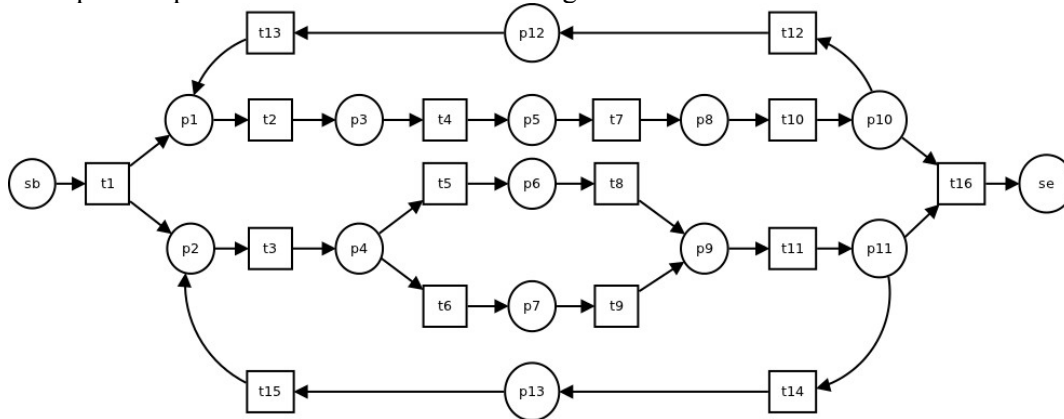


Immagine 16: Modello di workflow per l'esperimento 6

Questo modello presenta le seguenti caratteristiche:

- 1 AND-split:
 - da t1 alla coppia {t2,t3};
- 3 OR-split:
 - da t3 alla coppia {t5,t6};
 - da t10 alla coppia {t12,t16};
 - da t11 alla coppia {t14,t16};
- 1 AND-join:
 - dalla coppia {t10,t11} a t16;
- 3 OR-join:
 - dalla coppia {t1,t13} a t2;
 - dalla coppia {t1,t15} a t3;
 - dalla coppia {t8,t9} a t11;
- 2 grandi cicli:
 - da p10 a p1 passando per t12, p12 e t13;
 - da p11 a p2 passando per t14, p13 e t15.

SPERIMENTAZIONE

Si presenta la tabella dei risultati e in seguito i commenti a riguardo.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 8 su 8 | 8 su 8 | 100% | 100% |
| 5 | 8 su 8 | 8 su 8 | 100% | 100% |
| 10 | 8 su 8 | 8 su 8 | 100% | 100% |

Tabella 6: Risultati dell'esperimento 6

In questo caso, tutti i 3 modelli, ottenuti rispettivamente dalle sequenze con con lo 0%, il 5% ed il 10% di rumore, corrispondono in pieno al modello di partenza. Ancora una volta, tutte le connessioni sono state individuate tramite l'euristica 1, quindi, i risultati raggiunti rispecchiano le considerazioni fatte per l'esperimento 5.

A titolo puramente dimostrativo, di seguito si riporta il modello risultante dalle sequenze con il 10% di errore.

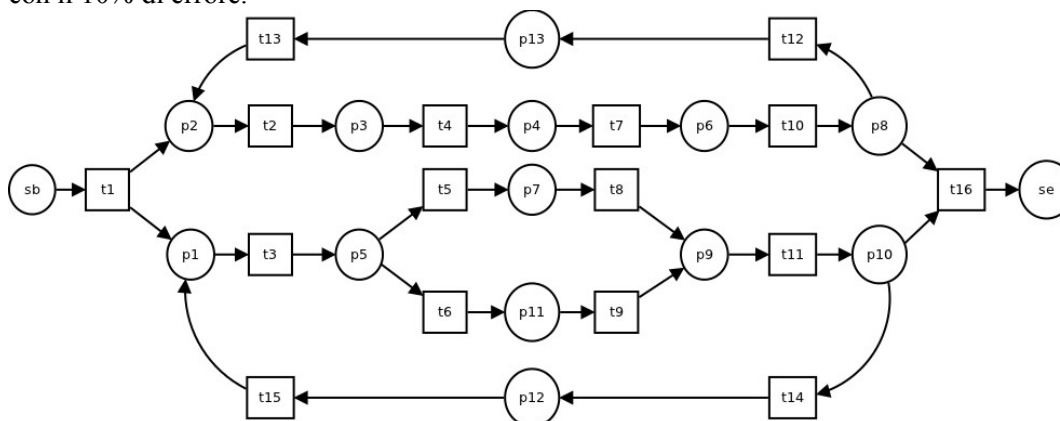


Immagine 17: Modello di workflow ottenuto dalle sequenze con il 10% di rumore

Il risultato migliore sulle sequenze con il 10% di errore rispetto all'esperimento 5 lo si deve ad una maggior lunghezza nelle sequenze, e quindi una maggior quantità di dati per la popolazione della D/F-table. La lunghezza delle sequenze non è casuale, ma la si deve alla struttura di questo modello: se da un lato il modello dell'esperimento 5 presentava due grandi OR-split che permettono di scegliere uno dei due percorsi puntati dalla divisione, dall'altro lato questo modello presenta un grande AND-split, che obbliga a percorrere ambedue i percorsi puntanti.

Come affermato per l'esperimento 5, anche questo esperimento dimostra che in caso di utilizzo della sola euristica 1, la tecnica si rivela essere efficace e robusta anche per alte soglie di rumore nei dati di input. Pertanto, i risultati di questo esperimento, sono da ritenersi pienamente accettabili.

SPERIMENTAZIONE

CONCLUSIONI

La tecnica testata si è rivelata molto efficace per quei modelli non troppo complicati. In particolare riesce a riconoscere il modello di partenza, anche su sequenze con un fattore di errore molto alto, qualora questo non presenti ricorsioni e piccoli cicli.

Dall'esperimento 2 si nota che anche un forte grado di concorrenza nei modelli può provocare ulteriori problemi alle euristiche, mentre tutti gli esperimenti ad esclusione dell'esperimento 3 presentano, com'era prevedibile, un decremento dell'efficacia della tecnica all'aumentare del grado di rumore all'interno dei dati di input.

Il problema è da imputarsi principalmente alla scarsa efficacia delle euristiche 2 e 3, le quali sono responsabili rispettivamente dell'individuazione di ricorsioni e piccoli cicli.

Si riportano di seguito le euristiche evidenziando i punti critici che non ne hanno quasi mai permesso l'attivazione durante gli esperimenti eseguiti:

IF(($A \rightarrow A \approx 0$) AND ($A < A + A > A > 0,5 * \#A$) AND ($A < A - A > A \approx 0$)) THEN $\langle A, A \rangle \in G$

IF(($A \rightarrow B \approx 0$) AND ($A > B \geq \sigma$) AND ($B < A \approx A > B$) AND ($A > B \geq 0,4 * \#A$) AND ($B < A \approx A > B$))) THEN $\langle A, B \rangle \in G$

In particolare, si critica la scelta di applicare il concetto di differenza relativa per quantificare il concetto di similitudine in queste tre componenti critiche.

Nel primo e nel terzo caso non ha senso rendere relativa la differenza tra il fattore di causalità e 0 in quanto, tale fattore, è già un rapporto dei fattori di incremento e decremento sulla la frequenza assoluta della prima transizione. In questi due casi sarebbe opportuno sostituire il concetto di differenza relativa con il concetto di differenza assoluta.

Nel secondo caso invece risulta praticamente impossibile che la differenza relativa tra i due membri della uguaglianza sia inferiore al fattore di rumore. L'idea di verificare che la differenza tra precedenza diretta e successione diretta sia prossima allo 0 non è del tutto sbagliata, ma dovrebbe essere sostituita dalla loro differenza relativa. In questo caso sarebbe opportuno sostituire la similitudine tra la differenza e 0, con una similitudine tra minuendo e sottraendo della differenza, applicando in seguito il concetto di differenza relativa.

Di seguito si riportando le tre reinterpretazioni delle componenti appena descritte.

$$(A \rightarrow A \approx 0) \equiv (|A \rightarrow A| \leq N) \quad .$$

$$(A < A \approx A > A) \equiv \left(|A < A - A > A| \leq N \cdot \frac{A < A + A > A}{2} \right) \quad .$$

$$(A \rightarrow B \approx 0) \equiv (|A \rightarrow B| \leq N) \quad .$$

SPERIMENTAZIONE

Seguendo le considerazioni precedenti è stata prodotta una implementazione alternativa dell'algoritmo. Di seguito si presentano i dati dell'applicazione di tale algoritmo sui modelli degli esperimenti visti in precedenza.

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 11 su 11 | 11 su 17 | 100% | 65% |
| 5 | 11 su 11 | 11 su 15 | 100% | 73% |
| 10 | 11 su 11 | 11 su 17 | 100% | 65% |

Tabella 7: Risultati dell'esperimento 1 con l'algoritmo alternativo

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 6 su 10 | 6 su 14 | 60% | 43% |
| 5 | 6 su 10 | 6 su 17 | 60% | 35% |
| 10 | 5 su 10 | 5 su 14 | 50% | 36% |

Tabella 8: Risultati dell'esperimento 2 con l'algoritmo alternativo

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 6 su 11 | 6 su 8 | 55% | 75% |
| 5 | 6 su 11 | 6 su 8 | 55% | 75% |
| 10 | 8 su 11 | 8 su 14 | 73% | 57% |

Tabella 9: Risultati dell'esperimento 3 con l'algoritmo alternativo

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 7 su 14 | 7 su 16 | 50% | 44% |
| 5 | 7 su 14 | 7 su 19 | 50% | 37% |
| 10 | 7 su 14 | 7 su 20 | 50% | 35% |

Tabella 10: Risultati dell'esperimento 4 con l'algoritmo alternativo

SPERIMENTAZIONE

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 10 su 10 | 10 su 14 | 100% | 71% |
| 5 | 10 su 10 | 10 su 14 | 100% | 71% |
| 10 | 9 su 10 | 9 su 17 | 90% | 53% |

Tabella 11: Risultati dell'esperimento 5 con l'algoritmo alternativo

| Rumore % | Completezza | Correttezza | Completezza % | Correttezza % |
|----------|-------------|-------------|---------------|---------------|
| 0 | 8 su 8 | 8 su 32 | 100% | 25% |
| 5 | 8 su 8 | 8 su 30 | 100% | 27% |
| 10 | 8 su 8 | 8 su 28 | 100% | 29% |

Tabella 12: Risultati dell'esperimento 6 con l'algoritmo alternativo

Con la nuova interpretazione delle euristiche si ha un lieve miglioramento solo per quanto riguarda i risultati inerenti alla completezza del primo esperimento, in particolare quelli riguardanti le sequenze con il 10% di rumore. Di contro però peggiorano di molto i valori di correttezza dei modelli trovati per tutti gli esperimenti, anche quelli che in precedenza avevano fornito valori accettabili.

L'unico miglioramento alla correttezza si ottiene sulle sequenze senza errore eseguendo il sistema con un fattore di errore pari a 0. Questo fa da filtro alle connessioni, impedendo di accettare nel D/F-graph quelle incerte. Si deduce quindi che uno dei punti da migliorare è senza dubbio un più efficace utilizzo del fattore di rumore all'interno delle euristiche.

D'altro canto, come accennato durante gli esperimenti, anche l'algoritmo per il riconoscimento della tipologia di split e di join ha fallito diverse volte. Le poche volte che le euristiche 2 e 3 hanno individuato delle ricorsioni e dei piccoli cicli, l'algoritmo di riconoscimento di split e join li ha quasi sempre male interpretati, invalidando il modello rigenerato per il requisito di soundness. Dato che nell'articolo vengono presentate tre soluzioni equivalenti, si potrebbe considerare l'idea di valutare gli split e i join sulla base del risultato di tutte, oppure si potrebbero trattare gli archi del D/F-graph ottenuti mediante le euristiche 2 e 3 con un algoritmo ad hoc.

I pessimi risultati riscontrati durante questa sperimentazione indicano che il sistema andrebbe migliorato, in particolar modo nei punti sopra citati. Il tentativo di miglioramento adottato dall'implementazione alternativa non ha dato i risultati sperati, ma l'intuizione alla base dovrebbe essere presa in considerazione durante uno studio per il miglioramento complessivo del sistema.