



**XXXII. Országos Tudományos
Diákköri Konferencia
2015**



**Szekvencia kereső algoritmus fejlesztése
folyamatbiztonság növelése céljából**

Készítette: Károly Béla Richárd

Témavezető: dr. Abonyi János

Mérnöki Kar, Folyamatmérnöki Intézeti Tanszék

Tartalom

1.	Bevezetés	2
2.	Több időbeli mintákat feltáró algoritmus	5
2.1.	Több időbeli minta definíciója	5
2.2.	Több időbeli minták tulajdonságai	11
2.3.	A feltáró algoritmus	16
3.	Az algoritmus implementálása MATLAB környezetben	21
3.1.	Változtatások az algoritmusban	21
3.2.	Az eseménynapló felépítése	23
4.	Az eszköz alkalmazása	24
4.1.	Validáció	24
	A kimeneti fájl felépítése	25
4.2.	Vinil-acetát előállítási folyamat alarm menedzsment rendszerének fejlesztése több időbeli minták segítségével	26
4.3.	Az eredmények kiértékelése	31
4.4.	További lehetőség: tanító minták feltárása	31
5.	Összefoglalás	32
6.	Irodalomjegyzék	33
7.	Köszönetnyilvánítás	34

1. Bevezetés

A komplex termelő rendszerek hatékony és biztonságos üzemeltetését a modern folyamatirányító rendszereken túl a jól képzett operátók (a technológia működtetéséért felelős szakemberek) biztosítják.

A folyamatirányító rendszer az operátorok munkáját riasztásokkal (alarmokkal) és figyelmeztetésekkel (alertek, warningok) támogatja [1]. A figyelmeztetések olyan jelzések, melyek nem igényelnek azonnali beavatkozást, a riasztások elhárításához azonban szükség van az operátor beavatkozására. A jelzések segítségével az üzemeltetők képesek beavatkozni a folyamat működésébe, hogy visszatérítsék a folyamatot a normál működésébe, vagy megelőzzék, hogy a folyamat abnormális vagy veszélyes állapotba kerüljön. Alarmjelzéseket alkalmazva képesek vagyunk összetett technológiák biztonságos üzemeltetésére, éppen ezért rendkívül elterjedt a használatuk. Nagyon fontos, hogy a biztonsági riasztások akkor igazán hasznosak, ha átfogó képet adnak a folyamat működéséről. Egy jól definiált alarmnak fel kell hívnia az operátor figyelmét a nem kívánt eseményekre, informálnia kell annak természetéről, valamint útmutatóul kell szolgálnia a megfelelő közbelépési művelethez.

A digitális folyamatirányító rendszerek megváltoztatták a riasztások szerepét. Évekkel ezelőtt egy jelzés beiktatásának költsége meglehetősen magas volt. A jelenlegi automatizálási rendszerek gyakorlatilag eltörölték a jelzések költségeit, és ezzel az ösztönző erőt, hogy korlátozzuk a számukat. Ennek köszönhetően, többnyire a kelleténél jóval több riasztás és figyelmeztetés terheli az operátorokat, megnehezítve munkájuk végzését. Az alarm menedzsment rendszerek azonban nem csak folyamatirányítási alkalmazások esetén hatékonyak, informatikai alkalmazásuk is elterjedt [2]. Telekommunikációs hálózatok jelzéseinek összefüggéseiről Saeed Salah és kutatótársai adnak átfogó képet [3]. Kimutatták, hogy a jelzési rendszerek elégtelen működése jelentős mértékben járul hozzá az ipari balesetek, katasztrófák kialakulásához [4]. A riasztások felhalmozódása, az úgynevezett alarm flood, az egyik leggyakrabban előforduló hiba az alarm menedzsment rendszerekkel kapcsolatban.

A legfontosabb lépés a riasztások és figyelmeztetések racionalizálása. Ebben a folyamatban optimalizáljuk a riasztások adatbázisát. Ezt többnyire az alarmok számának csökkentésével, prioritásuknak meghatározásával, a paraméterek validálásával tudjuk elérni. Fontos fejlesztési szempont továbbá, hogy az operátorokhoz jutó jelzéseket értelmezhetővé és informatívvá kell tenni. Fontos, hogy a riasztásokat úgy definiáljuk, hogy pontosan a normális és abnormális működés határán legyenek, így elkerülve a felesleges riasztásokat, valamint azt, hogy a jelzés

beérkezésével ne legyen idő a megfelelő fellépésre. Jelenleg nincs olyan, az alarmok mély összefüggéseit feltáró eszköz, mely a folyamatirányítási rendszerek fejlesztőinek és azok üzemeltetőinek segítségére lehet abban, hogy a meglévő jelzési rendszereiket optimalizálhassák, azok számát racionalizálhassák. Hasonló gondolatmenetet valósítanak meg Sandeep R. Kondaveeti és munkatársai, cikkükben egy alarm menedzsment rendszer racionalizálását oldják meg oly módon, hogy az eseménysorok hasonlósága alapján grafikus megjelenítést adnak [5]. Ez a megközelítés azonban az összetett szekvenciák összefüggéseit nem képes megállapítani.

Célunk egy olyan eszköz kialakítása, mely az üzemek eseménynaplóit elemezve képes támogatni ezt a folyamatot. Megközelítésmódunk alapja az operátori jelzések alapján definiált események naplófájljait adatbázisba szervezzük annak érdekében, hogy az események időbeni sorrendjét kezdő- illetve végpontjainak segítségével meghatározzuk a gyakori eseménysorokat, feltárva azok tipikus bekövetkezési sorrendjét, meghatározva ez alapján a felesleges jelzéseket. A feltárt minták megbízhatóságának kiértékelése szintén támogatja azt a jelzések számának csökkentését, a megbízható alarm menedzsment rendszer kialakíthatását. Tongyan Li és Xingming Li felismerték, hogy a gyakori elemhalmaz kereséssel támogatni tudják az alarm menedzsment rendszereket [6]. A gyakori elemekből kialakított szekvenciák felállítása több információt szolgáltat azáltal, hogy az elemek közötti relációkat is definiálja.

Annak érdekében, hogy a kialakított eszköz minden felállított kritériumnak megfeleljen, olyan algoritmus fejlesztésére van szükség, mely képes több változó együttes kezelésére, fel tudja tárni az események között előforduló időbeli relációkat, és ezeket segítségével képes több-időbeli (úgynevezett multi-temporal) minták kialakítására és ezeknek ellenőrzésére. A fejlesztés során Xiaoxiao Kong, Qiang Wei és Guoqing Chen által definiált több-időbeli minták feltárására alkalmas algoritmust implementáltuk és fejlesztettük tovább MATLAB környezetben, valamint létrehoztunk a szükséges programokat, melyek megvalósítják az eseménynaplók definiált bemeneti formára alakítását [7]. A felhasznált algoritmus a vizsgált rendszer állapotaiból állít fel szekvenciákat, melyeket az eseménynaplóban előforduló események idéznek elő. Megvalósításunkban eltértünk a szerzők definícióitól: más jelöltállítási feltételeket és struktúrát alkalmazunk, valamint a hatékonyság érdekében nem tároljuk el az egyes állapotokat támogató eseményszekvenciákat. A megfelelő állapotszekvenciák létrehozásához elegendő az egyes eseményszekvenciák effektív időtartamát és az egyes események legutolsó előfordulását eltárolni, így gyorsítva a futási időt és csökkentve az algoritmus memóriaigényét.

Az eredmények validálása és statisztikai kiértékelhetősége érdekében elsőként egy, a műszaki gyakorlatban széles körben alkalmazott sztochasztikus folyamatmodellre vonatkozó alarmjelzéseket vizsgáltunk. A megközelítésmód alapja azonos a statisztikai folyamatirányításban (SPC) alkalmazott jelzésekkel (Western Electric) [8]. A szekvenciák előrejelző képességét a megbízhatóságukkal jellemeztük. Kimutattuk, hogy mely jelzések igazán informatívak, és mekkora az esélye annak, hogy egy adott jelzést követően a folyamat nem kívánt állapotba jut.

A dolgozat második fejezete az alkalmazott algoritmust ismerteti. Ezt követően a harmadik fejezet tárgyalja az algoritmus továbbfejlesztése érdekében tett módosításokat. Definiáljuk, hogy, a létrehozott eszköz alkalmazása milyen bemeneti adatstruktúrát igényel, és bemutatjuk a fejlesztés eredményeként létrehozott MATLAB multi-temporal toolboxot, melynek kézikönyve a mellékletben érhető el. A negyedik fejezetben a kialakított eszköz validációjára kerül sor, majd bemutatjuk működését egy valós műszaki rendszer modelljén alapuló szimulátor segítségével.

2. Több időbeli mintákat feltáró algoritmus

Kutatásunk alapgondolata, hogy a valós műszaki rendszerek különböző állapotait több változó együttese határozza meg. Ezen tény figyelembevételével esett választásunk a több időbeli mintákat feltárni képes algoritmus célirányos alkalmazására, mely segítségével képesek vagyunk több változó között fellépő összefüggéseket megismerni.

Ebben a fejezetben a [7] irodalomra támaszkodva definiáljuk a több időbeli mintákat, ismertetjük tulajdonságaikat, valamint következő fejezetben a fejlesztésre kerülő, a mintákat előállító feltáró algoritmust.

2.1. Több időbeli minta definíciója

Objektumok a való világban (pl. emberek, gépek, növények, stb.) leírhatóak a tulajdonságaik alapján. Egy tulajdonság értéke az objektum jellegzetes állapotát tükrözi. Más szavakkal, használhatunk $s = \{o, a, v\}$ rekordokat egy állapot ábrázolására, ahol o egy objektum, a egy tulajdonsága o -nak és v pedig az a tulajdonság értéke. Egy ilyen állapot lehet például: $\{A \text{ részvény, ár, növekedés}\}$.

Sok esetben egy állapot bizonyos időintervallumban helyezkedik el, melyet eseménynek nevezünk és e -vel jelölünk. Egy eseményt ugyancsak egy rekord segítségével írhatunk le: $e = \{s, st, et\}$, ahol s egy állapot, st az állapot kezdeti ideje és et az állapot végideje. Például az $s = \{A \text{ részvény, ár, növekedés}\}$, $e = \{s, 1, 3\}$ azt jelenti, hogy az A részvény ára az $[1, 3]$ időintervallum alatt *növekszik*.

Tegyük fel, hogy S n darab állapot sorozata. pl. $S = \{s_1, s_2, \dots, s_n\}$. Minden egyes állapot $s_i (s_i \in S, 1 \leq i \leq n)$ több e_{ij} eseménnyel lehet összhangban, ahol $j = 1, 2 \dots m_i$. Egy időbeli D_T adatbázis előállítható oly módon, hogy egy eseményhalmazt növekvő sorrendben helyezünk el azok kezdeti időpontjaik szerint. Az 1-es táblázat egy példa D_T időbeli adatbázisra.

1. Definíció

Az e_A eseményre és annak időintervallumára (st_A, et_A) , adott egy előre meghatározott tolerancia állandó, amit ablaknak (window) nevezünk. Azt mondjuk, hogy $C_A = [st_A, et_A + ablak]$ az e_A ablak kényszere. Adott egy másik esemény, e_B . Ha $st_B \in C_A$, akkor e_B kielégíti e_A ablak kényszerét.

Az ablak kényszer egy érdekességi mérceként használatos, hiszen ha két esemény távol áll egymástól, akkor ez a két esemény nincs jelentéstanilag összekapcsolva. Ez a kényszer sokféle szituációban hasznos.

2. Definíció

Adott két esemény, e_A és e_B melyet rendre (st_A, et_A) és (st_B, et_B) időpontok határoznak meg. Ez a két esemény időbeli esemény szekvenciákat határoz meg, ha a négy időbeli reláció valamelyikével kapcsolódnak (*egyenlő, előtt, közben, átfedés*) a következők szerint:

1. Ha $st_A = st_B$ és $et_A = et_B$, akkor e_A egyenlő e_B -vel
2. Ha $0 \leq st_B - et_A \leq ablak$, akkor e_A megelőzi e_B -t
3. Ha $st_B < st_A < et_A \leq et_B$ vagy $st_B \leq st_A < et_A < et_B$, akkor e_A közben e_B
4. Ha $st_A \leq st_B < et_A < et_B$ vagy $st_A < st_B < et_A \leq et_B$, akkor e_A átfedésben van e_B -vel

Elsősorban ezt a négy relációt tekintjük a legnépszerűbbnek. Az egyszerűség kedvéért rendre E, B, D, O rövidítéseket fogjuk használni. Így időbeli esemény szekvenciákat határozhatunk meg e_A és E_B események között a következőképpen: $\Phi := e_A \rightarrow R \rightarrow e_B$, ahol R egy időbeli reláció, pl., $R \in \{D, B, O, E\}$. Például az következő táblázatban $e_{31} \rightarrow D \rightarrow e_{11}, e_{61} \rightarrow B \rightarrow e_{23}$ és $e_{62} \rightarrow O \rightarrow e_{42}$.

1. Táblázat: Egy példa D_T időbeli adatbázisra

Esemény	Állapot	st	et	Esemény	Állapot	st	et
e_{11}	s_1	1	5	e_{32}	s_3	33	38
e_{31}	s_3	1	4	e_{62}	s_6	34	39
e_{51}	s_5	2	10	e_{42}	s_4	25	37
e_{21}	s_2	2	8	e_{52}	s_5	27	42
e_{61}	s_6	2	5	e_{23}	s_2	28	32
e_{22}	s_2	4	6	e_{12}	s_1	30	40
e_{41}	s_4	3	7	e_{33}	s_3	30	38
...	...	...	...	...	...	...	...

Továbbá, azon tény következtében, hogy egy időbeli esemény szekvencia egyfajta időbeli kapcsolatot tükröz két állapot között (pl. S_i és S_j), fontos megtalálnunk, hogy egy ilyen kapcsolat magában foglal-e további esemény szekvenciákat, hiszen ez a kapcsolat egy időbeli mintát képvisel $S_i \rightarrow R \rightarrow S_j$. Konkrétan, ha egy $S_i \rightarrow R \rightarrow S_j$ időbeli mintához létezik egy időbeli esemény szekvencia, pl. $e_{ip} \rightarrow R \rightarrow e_{jq}$, ahol e_{ip} és e_{jq} azon események, melyek rendre S_i -hez és S_j -hez kapcsolódnak, akkor azt mondjuk, hogy az időbeli mintát $S_i \rightarrow R \rightarrow S_j$ támogatja az $e_{ip} \rightarrow R \rightarrow e_{jq}$ időbeli esemény szekvencia. Például, az 1-es táblázatban az $S_1 \rightarrow O \rightarrow S_2$ időbeli mintát támogatja $e_{11} \rightarrow O \rightarrow e_{21}$, $e_{11} \rightarrow O \rightarrow e_{22}$ és $e_{12} \rightarrow O \rightarrow e_{23}$. Így az adatbányászat és a tudásfelfedezés szempontjából csak az időbeli esemény szekvenciák által gyakran támogatott időbeli mintákat minősítjük döntéshozatal számára hasznosnak.

3. Definíció

Adott egy n elemű állapot sorozat S , pl. $S = \{s_1, s_2, \dots, s_n\}$, és időbeli relációk sorozata $\{D, B, O, E\}$. Azt mondjuk, hogy egy minta amelyben $k + 1$ állapotot k időbeli reláció kapcsol össze azt k hosszú időbeli mintának nevezzük (a relációk száma k), vagy $(k + 1)$ -állapot időbeli mintának (az állapotok száma $k + 1$) és Φ_k -val jelöljük. Ebben az esetben megkülönböztetjük:

1. Amikor $k = 0$, $\Phi_0 := S_0, S_0 \in S$ (S_0 -t degenerált időbeli mintának nevezzük);
2. Amikor $k = 1$, $\Phi_1 := (\Phi_0 \rightarrow R_1 \rightarrow S_1) := (S_0 \rightarrow R_1 \rightarrow S_1), R_1 \in \{D, B, O, E\}$;
3. Amikor $k = 2$, $\Phi_2 := (\Phi_1 \rightarrow R_2 \rightarrow S_2) := (S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2), R_1, R_2 \in \{D, B, O, E\}$;
4. Amikor $k > 2$, $\Phi_k := (\Phi_{k-1} \rightarrow R_k \rightarrow S_k) := (S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, \text{ ahol } j = 1, 2, \dots, k$.

Tisztán látható, hogy $k = 0$ esetén a minta csak egy állapotot tartalmaz, pl. „A részvény árnövekedése”. $k = 1$ esetén két állapot között van egy időbeli reláció, pl. „A részvény árnövekedése” $\rightarrow B \rightarrow$ „Shanghai tőzsdeindex növekedés”. Ez azt jelenti, hogy az A jelű részvény ára azelőtt növekszik, hogy a tőzsdeindex növekedne. Továbbá, $k \geq 2$ esetén több állapot között áll fenn szekvenciális időbeli reláció, pl. „A részvény árnövekedése” $\rightarrow B \rightarrow$ „Shanghai tőzsdeindex növekedés” $\rightarrow B \rightarrow$ „Shenzhen tőzsdeindex növekedés”. Láthatóan ez az időbeli minta olyan tudást ad, mely hasznos a döntéshozók számára a kínai tőzsdepiac dinamikájának megértésében. Megjegyezzünk, hogy amennyiben egy minta csakis azonos relációkat tartalmaz, akkor a mintát egy-időbeli mintának nevezzük. Ha különböző relációkat

integrálunk egy időbeli mintába, akkor a mintát több időbeli mintának nevezzük. Példaképpen: „B részvény árcsökkenése” $\rightarrow D \rightarrow$ „A részvény árnövekedése” $\rightarrow B \rightarrow$ „Shanghai tőzsdeindex növekedés” egy több időbeli mintát reprezentál két relációval: D, B .

Hasonlóan a két állapotú mintákhoz és esemény szekvenciákhoz, egy több időbeli mintát $S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$ több időbeli esemény szekvencia is támogatathat, akár csak $e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kp k}$ esetén. Például az első táblázatban az időbeli $\Phi := S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ mintát a következő időbeli eseményszekvenciák támogatják: $e_{31} \rightarrow D \rightarrow e_{11} \rightarrow O \rightarrow e_{21}$ és $e_{33} \rightarrow D \rightarrow e_{12} \rightarrow O \rightarrow e_{23}$.

Megérzés: Ha egy időbeli mintát gyakran támogatnak időbeli esemény szekvenciák - egy bizonyos fokig -, akkor a minta alkalmasnak tekinthető. Az alkalmas minták felfedezése érdekében az effektív időintervallum elképzelését a következőekben tárgyaljuk.

4. Definíció

Adott egy időbeli minta $\Phi, \Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$, melyet támogat egy időbeli esemény szekvencia $\phi, \phi := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kp k}$. Az effektív időintervallumot ahol ϕ támogatja Φ -t, $t_{\Phi\phi}$ -vel jelöljük, és a következőképpen definiáljuk: $t_{\Phi\phi} = t_{kp k} \cup t_{(k-1)p(k-1)} \cup \dots \cup t_{1p1} \cup t_{0p0}$, ahol $t_{ipj} = (st_{ipj}, et_{ipj})$ az $e_{jpj}, j = 0, 2, \dots, k$. esemény időintervalluma. Ekkor $t_{\Phi\phi} = (st_{\Phi\phi}, et_{\Phi\phi})$, ahol $st_{\Phi\phi} = \min\{st_{jpj} | j = 0, 2, \dots, k\}$, $et_{\Phi\phi} = \max\{et_{jpj} | j = 0, 2, \dots, k\}$. Továbbá, $T(\Phi) = \{t_{\Phi\phi} | t_{\Phi\phi} \text{ az effektív időintervallum ahol } \phi \text{ időbeli esemény szekvencia támogatja } \Phi\text{-t}\}$ -t az összes időbeli esemény szekvencia effektív időintervallumának sorozatának nevezzük, melyek Φ időbeli mintát támogatják.

Például, az 1-es táblázatban, $\Phi := S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ időbeli mintát támogatja $\phi := e_{31} \rightarrow D \rightarrow e_{11} \rightarrow O \rightarrow e_{21}$ időbeli esemény szekvencia. Az effektív időintervallum, ahol ϕ támogatja Φ -t a következő: $t_{\Phi\phi} = (st_{\Phi\phi}, et_{\Phi\phi}) = (\min\{1, 1, 2\}, \max\{4, 5, 8\}) = (1, 8)$.

Az effektív időintervallum elképzelése alapján a következő tulajdonságokhoz juthatunk:

1. Tétel

Adott egy időbeli minta $\Phi, \Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$, melyet egy időbeli esemény szekvencia támogat: $\phi, \phi := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kp k}$, az effektív időintervallum, ahol ϕ támogatja Φ -t pl. $t_{\Phi\phi}$ a következő tulajdonságokkal rendelkezik:

1. $st_{jpj} \geq st_{\Phi\phi}, et_{jpj} \leq et_{\Phi\phi}$, és $et_{kpk} = et_{\Phi\phi}$, ahol $j = 0, 2, \dots, k$;
2. Ha $R_1 = R_2 = \dots = R_k = E$, akkor $st_{\Phi\phi} = st_{1p1} = st_{1p2} = \dots = st_{kpk}$, és $et_{\Phi\phi} = et_{1p1} = et_{1p2} = \dots = et_{kpk}$;
3. Ha $R_1 = R_2 = \dots = R_k = D$, akkor $st_{\Phi\phi} = st_{kpk}$, és et_{kpk} ;
4. Ha $R_1 = R_2 = \dots = R_k = B$ vagy D , akkor $st_{\Phi\phi} = st_{1p1}$, és $et_{\Phi\phi} = et_{kpk}$;

ahol az s_j állapotnak megfelelő e_{jpj} esemény időintervalluma $t_{jpj} = (st_{jpj}, et_{jpj})$.

Bizonyítás

1. A 4-es definíció alapján $st_{\Phi\phi} = \min\{st_{0p0}, st_{1p1}, \dots, st_{kpk}\}$, $et_{\Phi\phi} = \max\{et_{0p0}, et_{1p1}, \dots, et_{kpk}\}$.
Tehát $st_{jpj} \geq st_{\Phi\phi}, et_{jpj} \leq et_{\Phi\phi}, j = 0, 1, 2, \dots, k$.
A $\phi_1 := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1}$ által támogatott $\Phi_1 := s_0 \rightarrow R_1 \rightarrow s_1$ időbeli mintára a 2-es definíció szerint $et_{0p0} \leq et_{1p1}$, ekkor a 4-es definíció szerint $et_{\Phi_1\phi_1} = et_{1p1}$.
Hasonlóképpen a $\phi_1 := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2}$ által támogatott $\Phi_2 := \Phi_1 \rightarrow R_2 \rightarrow s_2$ időbeli mintára $et_{0p0} \leq et_{1p1} \leq et_{2p2}$, ekkor $et_{\Phi_2\phi_2} = et_{2p2}$. És így tovább a $\Phi := s_0 \rightarrow R_1 \rightarrow s_1 \rightarrow R_2 \rightarrow s_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow s_k = \Phi_{k-1} \rightarrow R_k \rightarrow s_k$ -ra $et_{0p0} \leq et_{1p1} \leq \dots \leq et_{kpk}$, ekkor $et_{\Phi\phi} = et_{kpk}$.
2. Közvetlenül a 2-es definícióból bizonyítható (1).
3. A 2-es definíció (3) alapján következtethetjük ki, hogy $et_{0p0} \leq et_{1p1} \leq \dots \leq et_{kpk}$, és $st_{kpk} \leq st_{(k-1)p(k-1)} \leq \dots \leq st_{0p0}$. Mivel $st_{\Phi\phi} = \min\{st_{0p0}, st_{1p1}, \dots, st_{kpk}\}$, $et_{\Phi\phi} = \max\{et_{0p0}, et_{1p1}, \dots, et_{kpk}\}$, ezért $st_{\Phi\phi} = st_{kpk}$, és $et_{\Phi\phi} = et_{kpk}$.
4. A 2-es definíció alapján ((2) és (4)), kikövetkeztethetjük, hogy $st_{0p0} < st_{1p1} < \dots < st_{kpk}$. Mivel $st_{\Phi\phi} = \min\{st_{0p0}, st_{1p1}, \dots, st_{kpk}\}$, ezért $st_{\Phi\phi} = st_{1p1}$, és $et_{\Phi\phi} = et_{kpk}$ (a bizonyítás (1) alapján).

5. Definíció

Egy időbeli adatbázisban egy adott Φ időbeli mintát támogató időbeli esemény szekvenciák számát Φ támogatottságának nevezzük, és $supp(\Phi)$ -vel jelöljük. Ekkor egy Φ időbeli minta támogatottságának mértékét a következőképpen definiáljuk:

$$support(\Phi) = supp(\Phi)/|E|,$$

ahol $|E| = \max(|E_j|), N$ ahol $j = 1, 2, \dots, N$ az adott D_T adatbázisban az állapotok száma, E_j az események száma, melyek az S_j állapotot támogatják, $|E_j|$ az E_j számossága, és e_{jp} egy esemény, mely az S_j állapotot támogatja, pl. $e_{jp} \in E_j$.

Például az 1-es táblázatban $\Phi := S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ több időbeli mintát az $e_{31} \rightarrow D \rightarrow e_{11} \rightarrow O \rightarrow e_{21}$ és $e_{33} \rightarrow D \rightarrow e_{12} \rightarrow O \rightarrow e_{23}$ időbeli esemény szekvenciák támogatják, így $supp(\Phi) = 2$. Szintén az 1-es táblázatban látható, hogy $E_1 = \{e_{11}, e_{12}\}, E_2 = \{e_{21}, e_{22}, e_{23}\}, E_3 = \{e_{31}, e_{32}, e_{33}\}, E_4 = \{e_{41}, e_{42}\}, E_5 = \{e_{51}, e_{52}\}$ és $E_6 = \{e_{61}, e_{62}\}$. Tehát $|E| = \max(2, 3, 3, 2, 2, 2) = 3$. Ekkor $support(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2) = 2/3$. A támogatottság mértékét arra használhatjuk, hogy megmutassuk egy több időbeli minta erősségét egy adott időbeli adatbázisban. Következésképpen a következőket kaphatjuk:

1. Tulajdonság

Bármely adott időbeli minta esetén $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$ ahol E_j támogatja S_j -t, $supp(\Phi) \leq |E_j| \leq |E|$.

Bizonyítás.

A 1. és 3. definíció alapján, ha az e esemény támogatja S_j -t, akkor lehetséges, hogy egy esemény szekvenciában helyezkedik el, hogy támogassa Φ -t. Azonban, ha e esemény nem támogatja S_j -t, akkor semmiképpen nem helyezkedhet el olyan esemény szekvenciában, amely támogatja Φ -t. Tehát $supp(\Phi) \leq |E_j|$.

Továbbá az 5. definíciónak megfelelően $|E_j| \leq |E|$.

Ebből kifolyólag $supp(\Phi) \leq |E_j| \leq |E|$.

6. Definíció

Egy időbeli adatbázisban egy adott időbeli minta $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$ esetén, legyen Φ_{k-1} a Φ $k - 1$ elemű részmintája, pl. $S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow$

$S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-1} \rightarrow S_{k-1}$, ekkor Φ -t jelölhetjük a következőképpen: $\Phi_{k-1} \rightarrow R_k \rightarrow S_k$.

Továbbá Φ bizalmi foka a következőképpen definiálható:

$$confidence(\Phi) = \begin{cases} \frac{supp(\Phi)}{supp(\Phi_{k-1})} \times confidence(\Phi_{k-1}) & R_k \neq D \\ \frac{supp(\Phi)}{supp(S_k)} \times confidence(S_k) = \frac{supp(\Phi)}{supp(S_k)} & R_k = D \end{cases}$$

1

Például az 1-es táblázatban bármely $\Phi := S_i$, $i = 1, 2, \dots, N$, esetén $confidence(\Phi) = 1$, hiszen Φ egy degenerált időbeli minta előzmények nélkül. $\Phi := S_3 \rightarrow S_1$ ($R_k = D, k = 1$) esetén $supp(S_3 \rightarrow D \rightarrow S_1) = 2, supp(S_1) = 2$, és $confidence(S_1) = 1$. Ekkor

$confidence(S_3 \rightarrow D \rightarrow S_1) = \frac{supp(S_3 \rightarrow D \rightarrow S_1)}{supp(S_1)} = \frac{2}{2} = 100\%$, ami azt jelenti, hogy S_3 100% eséllyel fog bekövetkezni S_1 bekövetkezése esetén.

Továbbá, $\Phi := S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ több időbeli minta esetén ($R_k \neq D, k = 2$), $supp(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2) = 2, supp(S_3 \rightarrow D \rightarrow S_1) = 2$ és $confidence(S_3 \rightarrow D \rightarrow S_1) = 1$. Ekkor $confidence(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2) = \frac{supp(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2)}{supp(S_3 \rightarrow D \rightarrow S_1)} \times$

$confidence(S_3 \rightarrow D \rightarrow S_1) = \frac{2}{2} \times 100\% = 100\%$. A bizalmi fok alapvetően arra használható, hogy megbecsüljük egy adott időbeli adatbázison belül egy több időbeli minta fontosságát.

2.2. Több időbeli minták tulajdonságai

Mielőtt felállítanánk a feltáró algoritmust alkalmas több időbeli minták feltárásának céljából, több hasznos tulajdonságot is származtathatunk és használhatunk a további algoritmus tervezésben a hatékonyság fejlesztésének érdekében.

7. Definíció

Adott egy Φ több időbeli minta, $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$, ekkor Φ' mintát $\Phi' := S_{j_1} \rightarrow R_{j_2} \rightarrow S_{j_2} \rightarrow R_{j_3} \rightarrow S_{j_3} \rightarrow \dots \rightarrow R_{j_m} \rightarrow S_{j_m}$, ahol $0 \leq j_1 \leq j_2 \leq \dots \leq j_m \leq k$ és $0 \leq m \leq k$ Φ m -elemű részmintájának nevezzük. Ha $j_1, j_2, \dots, j_m$ egymást követő természetes számok sorozata (pl. $j_p = j_{p-1} + 1, p = 1, 2, \dots, m$ és $j_1 = 0$ akkor Φ' -t Φ szekvenciális részmintájának nevezzük.

8. Definíció

Adott egy Φ időbeli minta és egy Φ' részminta. Φ bármely ϕ esemény szekvenciájára és effektív időintervallumára $t_{\Phi\phi} \in T(\Phi)$ létezik Φ' -nek egy ϕ' és annak effektív időintervalluma $t_{\Phi'\phi'} \in T(\Phi')$ oly módon, hogy $t_{\Phi'\phi'} \subseteq t_{\Phi\phi}$, ekkor Φ' -t Φ támogató részmintájának nevezzük.

Például, az 1-es táblázatban $S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$, mintának $S_3 \rightarrow D \rightarrow S_1, S_1 \rightarrow O \rightarrow S_2$ és $S_3 \rightarrow O \rightarrow S_2$ részmintái, miközben $S_3 \rightarrow D \rightarrow S_1$ és $S_1 \rightarrow O \rightarrow S_2$ szekvenciális részmintái. Továbbá az 1-es táblázatban $\Phi := S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ minta két esemény szekvenciával rendelkezik: $\phi_1 := e_{31} \rightarrow D \rightarrow e_{11} \rightarrow O \rightarrow e_{21}$ és $\phi_2 := e_{33} \rightarrow D \rightarrow e_{12} \rightarrow O \rightarrow e_{23}$ melyek effektív időintervallumai rendre $t_{\Phi\phi_1} = (1, 8)$ és $t_{\Phi\phi_2} = (28, 40)$. Tekintsük $\Phi' := S_3 \rightarrow O \rightarrow S_2$ részmintát. Ennek esemény szekvenciái: $\phi'_1 := e_{31} \rightarrow O \rightarrow e_{21}$ és $\phi'_2 := e_{33} \rightarrow O \rightarrow e_{23}$, ezek effektív idő intervallumai rendre: $t_{\Phi'\phi'_1} = (1, 8)$ és $t_{\Phi'\phi'_2} = (28, 38)$. Mivel $(1, 8) = t_{\Phi'\phi'_1} \subseteq t_{\Phi\phi_1} = (1, 8)$ és $(28, 38) = t_{\Phi'\phi'_2} \subseteq t_{\Phi\phi_2} = (28, 40)$ ezért Φ' -t Φ támogató részmintájának nevezzük.

2. Tulajdonság

Ha Φ' támogató részmintája Φ -nek, akkor $\text{supp}(\Phi') \geq \text{supp}(\Phi)$.

Bizonyítás

Ha $\text{supp}(\Phi)$ az esemény szekvenciák száma, melyek támogatják Φ -t, akkor ez megegyezik az effektív időintervallumok számával. Mivel Φ' egy részmintája Φ -nek, ezért egy ϕ esemény szekvencia (mely támogatja Φ -t) effektív időintervalluma $t_{\Phi\phi}$ lefed egy $t_{\Phi'\phi'}$ időintervallumot, ahol ϕ' támogatja Φ' -t. Így $\text{supp}(\Phi') \geq \text{supp}(\Phi)$.

Például az 1-es táblázatban $\text{supp}(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2) = 2$. Ennek támogató részmintájára $(S_3 \rightarrow O \rightarrow S_2)$ $\text{supp}(S_3 \rightarrow O \rightarrow S_2) = 2 = \text{supp}(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2)$. Azt is észrevehetjük, hogy $S_1 \rightarrow O \rightarrow S_2$ a 8-as definíció alapján támogató részmintája az előbbinek. Továbbá $\text{supp}(S_1 \rightarrow O \rightarrow S_2) = 3 > 2 = \text{supp}(S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2)$. Általánosságban a 2-es tulajdonság azt jelenti, hogy egy időbeli minta supp értéke nem haladja meg támogató részmintájának supp értékét. Ebből következik, hogy ha egy minta gyakori jelölt, akkor ennek bármely részmintája is gyakori jelölt lesz, hiszen $\text{support}(\Phi') = \frac{\text{supp}(\Phi')}{|E|} \geq \frac{\text{supp}(\Phi)}{|E|} = \text{support}(\Phi) \geq \alpha$. Ez egy nagyon fontos tulajdonság, és egyfajta ritkító stratégiaként alkalmazható, az algoritmus hatékonyságának növelése érdekében.

3. Tulajdonság

Ha Φ' egy szekvenciális részmintája Φ -nek, akkor Φ' támogató részmintája Φ -nek.

Bizonyítás

Tekintsük Φ' -t Φ szekvenciális részmintájaként. Azt mondjuk, hogy létezik olyan $t_{\Phi'\phi'} \in T(\Phi')$, hogy $t_{\Phi'\phi'} \subseteq t_{\Phi\phi}, t_{\Phi\phi} \in T(\Phi)$, ahol ϕ és ϕ' két esemény szekvencia, melyek rendre Φ -t és Φ' -t támogatják, és $t_{\Phi\phi}, t_{\Phi'\phi'}$ pedig rendre a megfelelő effektív időintervallumai ϕ -nek és ϕ' -nek. A 8-as definícióból származtathatjuk, hogy Φ' támogató részmintája Φ -nek.

Példaképpen az 1-es táblázatból: $S_3 \rightarrow D \rightarrow S_1$ és $S_1 \rightarrow O \rightarrow S_2$ szekvenciális részmintái $S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ -nek, ami azt is jelenti, hogy mindkettő támogató részmintája $S_3 \rightarrow D \rightarrow S_1 \rightarrow O \rightarrow S_2$ -nek.

4. Tulajdonság

Ha egy ϕ esemény szekvencia $\phi := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kpk}$ létezik, $R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$, akkor;

1. Ha $R_j = D, j = 1, 2, \dots, k$, akkor $e_{ipi} \rightarrow R_k \rightarrow e_{kpk}$ létezik, $i = 0, 1, 2, \dots, k - 1$.
2. Ha $R_j = B, j = 1, 2, \dots, k$, akkor $e_{(k-1)p(k-1)} \rightarrow R_k \rightarrow e_{kpk}$ létezik. Továbbá ha e_{kpk} megtörténik $e_{ipi}, i = 0, 1, 2, \dots, k - 1$ ablak kényszerében, akkor $e_{ipi} \rightarrow R_k \rightarrow e_{kpk}$ létezik, $i = 0, 1, 2, \dots, k - 1$.

Bizonyítás

1. Ha $R_j = D, j = 1, 2, \dots, k$, akkor $\phi := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kpk}$ esemény szekvencia időintervallumát (st_ϕ, et_ϕ) -nek nevezzük. A 1-es tétel ((1) és (3)) alapján $st_{ipi} \leq st_\phi = st_{kpk} < et_{ipi} \leq et_\phi = et_{kpk}$. Így az esemény szekvencia $e_{ipi} \rightarrow R_k \rightarrow e_{kpk}$ létezik $i = 1, 2, \dots, k - 1$.
2. Ha $R_j = B, j = 1, 2, \dots, k$ akkor $\phi' := e_{0p0} \rightarrow R_1 \rightarrow e_{1p1} \rightarrow R_2 \rightarrow e_{2p2} \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow e_{kpk}$ időintervallumát (st'_ϕ, et'_ϕ) -nek nevezzük. A 1-es tétel (4) alapján $et_{(k-1)p(k-1)} = et'_\phi$ és $et_{ipi} \leq et'_\phi, i = 0, 2, \dots, k - 1$. 2-es definícióban a *közben* elképzelés alapján $0 \leq st_{kpk} - et'_\phi \leq \text{ablak}$. Ezek tudatában $e_{(k-1)p(k-1)} \rightarrow R_k \rightarrow e_{kpk}$ létezik.

Továbbá kikövetkeztethető, hogy $st_{kpk} \geq et_{(k-1)p(k-1)} = et'_\phi \geq et_{ipi}$, ha $st_{kpk} - et_{ipi} \leq ablak$, ami azt jelenti, hogy e_{kpk} és e_{ipi} események ablak kényszere teljesül, ekkor $e_{ipi} \rightarrow R_k \rightarrow e_{kpk}$ létezik, $i = 0, 1, \dots, k-1$, különben nem.

Megemlítendő, hogy ha $R_k = O$ vagy E , akkor $e_{ipi} \rightarrow R_k \rightarrow e_{kpk}$ esemény szekvencia nem biztosan következik be. Ezután a következőkre következtethetünk:

5. Tulajdonság

Adott egy Φ több időbeli minta: $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$. Ekkor

1. Ha $R_k = D$, az 1 elemű részminta $S_i \rightarrow R_k \rightarrow S_k, i = 0, 1, 2, \dots, k-1$, Φ támogató részmintája.
2. Ha $R_k = B$, az 1 elemű részminta $S_{k-1} \rightarrow R_k \rightarrow S_k$ Φ támogató részmintája.
Továbbá ha S_i és S_k teljesíti az ablak kényszert, akkor az $S_i \rightarrow R_k \rightarrow S_k$ 1 elemű részminta támogató részmintája Φ -nek, $i = 0, 2, \dots, k-1$.

Bizonyítás

A tulajdonság közvetlenül bizonyítható a 4-es tulajdonság alapján.

6. Tulajdonság

Adott Φ több időbeli minta $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$, ha $R_k = D$, akkor az részminta $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-2} \rightarrow S_{k-2} \rightarrow R_k \rightarrow S_k$ Φ támogató részmintája.

Bizonyítás

Az általánosság elvesztése nélkül $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-2} \rightarrow S_{k-2} \rightarrow R_k \rightarrow S_k$ -t elnevezzük Φ' -nek. Ekkor Φ -t értelmezhetjük a következőképpen: $\Phi' \rightarrow R_{k-1} \rightarrow S_{k-1} \rightarrow R_k \rightarrow S_k$. A 5-ös tulajdonság alapján kikövetkeztethetjük, hogy $\Phi' \rightarrow R_k \rightarrow S_k$ támogató részmintája Φ -nek. Így $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-2} \rightarrow S_{k-2} \rightarrow R_k \rightarrow S_k$ támogató részmintája Φ -nek.

7. Tulajdonság

Adott $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k$. Ha létezik bármely $R_j = D$ vagy $B, j = 1, 2, \dots, k-1$, akkor $S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow$

$\dots \rightarrow R_k \rightarrow S_k$ támogató részmintája Φ -nek; máskülönben, csak akkor támogató részmintája $S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$ Φ -nek, ha $R_k = D$ vagy B .

Bizonyítás

A tulajdonság közvetlenül bizonyítható Tulajdonság 6. alapján.

2.3. A feltáró algoritmus

Ebben a fejezetben bemutatásra kerülő program kiválóan alkalmazható több időbeli minták feltárására, mely műveletre olyan erőt próbáló esetekben van szükség, mint a pénzügyi előrejelzések, vagy az egészségügyi kórtörténetek feltárása. Az időbeli minták feltárása effektív és informatív egyaránt, hiszen nem csak az *egyenlő* és *előtte* relációk feltárására alkalmas, valamint kezeli az események távolságának toleranciáját, tehát az egymástól túlzottan távoli eseményeket nem tartja összefüggőnek.

2 elemű időbeli minták létrehozása ($k = 1$)

A feltáró folyamat és a megfelelő algoritmus megkönnyítésének érdekében az időbeli adatbázist újra kell szervezni. Először egy új adattáblát készítünk, melynek mezői állapotokat reprezentálnak. Ezután minden időbeli eseményt a megfelelő oszlopba helyezünk. Például az 1. táblázatot átalakítjuk a 2. táblázattá. Csak azokat az eseményeket vesszük figyelembe, melyek elfogadható időn belül következnek be, és kielégítik az ablak kényszert. Adott egy D_T adatbázis melyben S állapotok helyezkednek el. Eleinte degenerált ($k = 0$) mintákat készítünk el, pl. $C_0 = \{S_j | S_j \in S\}$. Ezután megvizsgáljuk, hogy az egyes S_j állapotok gyakori előfordulásúak-e. Ennélfogva 1 elemű ($k = 0$) degenerált minták sorozatát kapjuk pl. $F_0 = \{S_j | \text{support}(S_j) \geq \alpha, S_j \in S\}$. Tegyük fel, hogy n esemény helyezkedik el az adatbázisban. A folyamat számítási komplexitása $O(n)$, vagyis ennyi adatbázis átvizsgálást igényel, hogy kiszámítsa az összes állapot *supp* értékét. Továbbá a következő tulajdonságra alapozva, 2 elemű ($k = 1$) gyakori minták hozhatók létre az 1 elemű minták alapján.

2. Táblázat: Átalakított időbeli adatbázis D'_T

$T(S_1)$	$T(S_2)$	$T(S_3)$	$T(S_4)$	$T(S_5)$	$T(S_6)$	...
(1, 5)	(2, 8)	(1, 4)	(3, 7)	(2, 10)	(2, 5)	...
(30, 40)	(4, 6)	(33, 38)	(25, 27)	(27, 42)	(34, 39)	...
...	(28, 32)	(30, 38)	...	...	...	...
...	...	...	...	...	...	...

8. Tulajdonság

Egy Φ 2 elemű időbeli mintára pl. $\Phi := S_i \rightarrow R \rightarrow S_j, R = \{D, B, O, E\}, \text{supp}(\Phi) \leq \text{supp}(S_j)$.

Bizonyítás

Mivel S_j egy degenerált minta, ezért szekvenciális részmintája Φ -nek. A 3-as tulajdonság szerint S_j támogató részmintája Φ -nek. Továbbá $\text{supp}(\Phi) \leq \text{supp}(S_j)$ (2. tulajdonság).

A 8-as tulajdonság szerint a 2 elemű ($k = 1$) jelölt mintákat például a következőképpen hozhatjuk létre: $C_1 = \{S_i \rightarrow R \rightarrow \frac{S_j}{S_i} \in S, S_j \in F_0\}$. Így növekszik a hatékonyság, hiszen nincs szükség a $C'_1 = \{S_j | S_i \in S, S_j \notin F_0\}$ komplementer generálására, vagy azzal való számolásra. Ezután megkeressük a 2 elemű gyakori mintákat az adatbázisban pl.

$$F_1 = \{S_1 \rightarrow S_j | S_i \in S, S_j \in F_0, \text{support}(S_i \rightarrow R \rightarrow S_j) \geq \alpha\}.$$

Továbbá a $(k + 1)$ elemű jelölt minták sorozata pl. C_k legenerálható a k elemű gyakori minták sorozata alapján.

$(k + 1)$ elemű időbeli minták létrehozása ($k \geq 1$)

9. Tulajdonság

Tegyük fel, hogy $R_j \in \{D, B, O, E\}, j = 1, 2, \dots, k, k \geq 2$. Adott egy $(k - 1)$ elemű Φ időbeli minta $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$, és Φ három részmintája, Φ_1, Φ_2, Φ_3 , $\Phi_1 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-1} \rightarrow S_{k-1}$, $\Phi_2 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$, és $\Phi_3 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-2} \rightarrow S_{k-2}$.

Ekkor:

Ha $R_k = D$, akkor, ha Φ_1, Φ_2, Φ_3 közül valamelyik nem gyakori pl. $\text{support}(X) < \alpha, X = \Phi_1, \Phi_2, \Phi_3$, akkor Φ nem gyakori, azaz $\text{support}(\Phi) < \alpha$.

Ha létezik bármely $R_j = D$ vagy $E, j = 1, 2, \dots, k - 1$, és Φ_1 vagy Φ_2 nem gyakori, akkor Φ nem gyakori, azaz $\text{support}(\Phi) < \alpha$.

Ha $R_j \neq D$ és $R_j \neq E$, akkor $\forall j = 1, 2, \dots, k - 1$, és $R_k = B$, és Φ_1 vagy Φ_2 nem gyakori, akkor Φ nem gyakori, azaz $\text{support}(\Phi) < \alpha$.

Bizonyítás

1. A 3-as tulajdonság alapján, Φ_1 támogató részmintája Φ -nek. Az 5-ös tulajdonság alapján Φ_2 támogató részmintája Φ -nek. A 6-os tulajdonság alapján Φ_3 támogató részmintája Φ -nek.

Tehát $\text{supp}(\Phi_1) \geq \text{supp}(\Phi)$, $\text{supp}(\Phi_2) \geq \text{supp}(\Phi)$, és $\text{supp}(\Phi_3) \geq \text{supp}(\Phi)$.

Ez azt jelenti, ha Φ_1 , Φ_2 , és Φ_3 közül valamelyik nem gyakori, akkor Φ nem gyakori.

2. Ez hasonlóképpen bizonyítható az 5-ös és 6-os tulajdonságok alapján.

3. Ez szintén bizonyítható az 5-ös és 6-os tulajdonságok alapján.

F_{k-1} alapján C_k létrehozásában felhasználhatjuk a 9-es tulajdonságot, melynek segítségével kiszűrhetjük a nem gyakori mintákat az adatbázis átvizsgálása nélkül. Az eljárás a következő:

1. F_{k-1} -ben minden egyes k elemű gyakori mintához pl. $\Phi_1 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-1} \rightarrow S_{k-1}$, címezzük meg, hogy a minta tartalmazza-e D -t vagy E -t.
2. F_{k-1} -ben keressük meg $\Phi_2 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$ -t, ahol R_k és S_k új időbeli relációk és állapotok melyek Φ_1 -hez tartoznak. Azt mondjuk, hogy Φ_2 egyenrangú Φ_1 -gyel. Ekkor egy új $(k + 1)$ elemű jelölt mintát hozhatunk létre, pl. $\Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k$. Így a $(k + 1)$ elemű jelölt minták sorozatát hozhatjuk létre, melyet C_k -val jelölünk, pl. $C_k = \{\Phi | \Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k, \text{ ahol } \Phi_1 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-1} \rightarrow S_{k-1}, \Phi := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_k \rightarrow S_k \text{ és } \Phi_1, \Phi_2 \in F_{k-1}\}$.
3. A létrehozott C_k -ra alapján, mielőtt átvizsgálnánk az adatbázist és kiszámítanánk a támogatottsági és bizalmi fokokat minden egyes Φ , $\Phi \in C_k$ -ra, hogy meghatározzuk, hogy mely Φ több időbeli minta alkalmas, alkalmazhatunk néhány szűrő stratégiát.
 - (a) Ha $R_k = D$, ez azt jelenti, hogy Φ_3 -nak, pl. $\Phi_3 := S_0 \rightarrow R_1 \rightarrow S_1 \rightarrow R_2 \rightarrow S_2 \rightarrow R_3 \rightarrow \dots \rightarrow R_{k-2} \rightarrow S_{k-2} \rightarrow R_k \rightarrow S_k$, hasonlóan Φ -hez, F_{k-1} -ben kell elhelyezkednie, különben Φ nem lesz alkalmas minta (9-es tulajdonság (1)). Ezután átvizsgáljuk F_{k-1} -et, hogy meghatározzuk, hogy $\Phi_3 \in F_{k-1}$ teljesül-e. Ha nem, akkor töröljük C_k -ből, pl. $C_k = C_k - \{\Phi_3\}$.
 - (b) Ha $R_k \neq D$ és létezik bármely $R_j = D$ vagy E , $j = 1, 2, \dots, k - 1$, akkor C_k maradjon változatlan (9-es tulajdonság (2)).
 - (c) Ha $R_j \neq D$ és $R_j \neq E$, akkor $\forall j = 1, 2, \dots, k - 1$, és $R_k = B$ esetén C_k maradjon változatlan (9-es tulajdonság (2)).

Ezután a módosított C_k alapján átvizsgáljuk az adatbázist, és kiszámoljuk a supp értéket minden $(k + 1)$ elemű Φ jelöltnek a C_k -ban. Így végül megkapjuk F_k -t. Továbbá F_k alapján legenerálhatjuk C_{k+1} -et, és hasonlóképpen szűrhetjük is. Amikor $C_k = 0$ megállunk, ami azt jelenti, hogy minden gyakori mintát megtaláltunk. A létrehozott gyakori időbeli mintákra és

supp értékekre alapozva minden egyes időbeli minta bizalmi fokát kiszámolhatjuk. Végül az alkalmas időbeli minták sorozatát is megkapjuk.

Az algoritmus pszeudo kódja:

1	$S = \{s s \text{ is a state in the } D_T\};$
2	$C_0 = \{s_i s_i \in S\}$
3	$F_0 = 0;$
4	$m = S ;$
5	FOR $s_j \in S$ DO
6	$F_0 = F_0 \cup \{s_j \text{support}(s_j) \geq \alpha, s_j \in S\}$
7	ENDFOR
8	$C_1 = \{s_i \rightarrow R \rightarrow s_j s_i \in S, s_j \in F_0\};$
9	$k = 1;$
10	WHILE $C_k \neq 0$ DO
11	$F_k = 0;$
12	FOR $\Phi_k \in C_k$ DO
13	Calculating $\text{support}(\Phi_k);$
14	IF $\text{support}(\Phi_k) \geq \alpha$ THEN
15	$F_k = F_k \cup \{\Phi_k\};$
16	END IF
17	END FOR
18	$C_k + 1 = 0;$
19	FOR $\Phi_1 \in F_k$ DO
20	IF Φ_1 contains D or E THEN
21	Flag = 1;
22	END IF
23	FOR $\Phi_2 \in F_k$ DO
24	IF Φ_2 matches Φ_1 AND $\Phi_3 \in F_k$ THEN
25	$C_{k+1} = C_{k+1} \cup \Phi;$
26	END IF
27	Filter $C_{k+1};$
28	END FOR
29	END FOR
30	END WHILE

A megbízhatóság számolás pszeudo kódja:

1	$Q = 0;$
2	$k = 0;$
3	<i>WHILE $F_k \neq 0$ DO</i>
4	<i>FOR $\Phi \in F_k$</i>
5	<i>Calculating confidence(Φ);</i>
6	<i>IF confidence(Φ) $\geq \beta$ THEN</i>
7	<i>$Q = Q \cup \{\Phi\};$</i>
8	<i>END IF</i>
9	<i>END FOR</i>
10	$k = k + 1;$
11	<i>END WHILE</i>

3. Az algoritmus implementálása MATLAB környezetben

Ez a fejezet az előzőekben ismertetett algoritmus megvalósítását mutatja be. Megfogalmazzuk, hogy az elérni kívánt cél érdekében mely pontokban térünk el a szerzők által definiált algoritmustól, valamint ismertetjük, hogy pontosan milyen követelményeknek kell megfelelnie a létrehozott program bemeneti fájljának.

3.1. Változtatások az algoritmusban

A program implementálásának kezdetén megvizsgáltuk, hogy leírásának melyek azon pontjai, melyeken érdemes változtatni az elérni kívánt cél érdekében. A definícióktól két alkalommal térünk el, melyek a jelöltállítási módszer, illetve az állapotszekvenciákat támogató eseményszekvenciák tárolásának módja.

A lehetséges gyakori szekvenciák előállításának stratégiája az algoritmus leírásában az „2.3. A feltáró algoritmus” olvasható. A szerzők a részletezett tulajdonságokban nem alkalmazzák az adatbányászat egyik legfontosabb alapelemét, az *a priori* elvet, mely kimondja, hogy egy adott szekvencia akkor, és csak akkor lehet gyakori, amennyiben minden részszekvenciája is gyakori. Ennek megfelelően a feltárást megelőzően töröljük a nem gyakori állapotokat, ily módon megelőzve a felesleges jelöltállítást.

Az alkalmazott lehetséges eseménynaplók tulajdonságát felhasználva, – mi szerint párhuzamosan ugyanazon állapot előfordulása nem lehetséges – a jelöltállítási műveletet lényegesen le tudtuk egyszerűsíteni, mely feltételezte az adataink rögzítési módszerének megváltoztatását. A feltárt állapotszekvenciákat támogató eseményszekvenciákban lévő események azonosítói közül csakis az egyes állapotokat előidéző események legmagasabb azonosítója informatív az effektív jelöltállítás szempontjából, hisz az algoritmus nem definiál olyan relációt, mely szerint a bővített szekvencia a jelöltállításban részvevő gyakori elem után következne be. E tulajdonságokat felhasználva a következő, rendkívül egyszerű lépésekre felbontható a jelöltállítás:

- vegyünk a gyakori $k - 1$ elemű, és az egy elemű állapotszekvenciák előfordulásainak teljes kombinációját,
- amennyiben i gyakori állapotszekvencia l , i -t támogató eseményszekvenciájára, valamint j gyakori állapotot támogató m eseményre igaz, hogy:

- amennyiben i degenerált szekvencia, akkor i állapotszekvencia és j állapot effektív időtartamai alapján meghatározott relációval k elemű jelölt alkalmas,
- amennyiben i nem degenerált minta, abban az esetben le kell ellenőriznünk, hogy az állapotot előidéző esemény azonosítója, mellyel bővíteni szeretnénk a szekvenciát nagyobb-e annál, mint a szekvenciában lévő, ugyanazon állapotot előidéző események legnagyobb azonosítója,
 - amennyiben előző feltétel teljesül, úgy ugyanazon jelöltállítási stratégiát alkalmazzuk, akár csak a degenerált szekvenciák esetén.

3.2. Az eseménynapló felépítése

Az algoritmus bemeneti fájlja, vagyis az eseménynapló egy Excel fájl (.xls kiterjesztésű). Az eseménynapló négy oszlopból áll: *event*, *state*, *st* és *et*.

<i>event</i>	<i>state</i>	<i>st</i>	<i>et</i>
...	...	...	...

- *event* oszlop tartalmazza az egyes állapotokat előidéző események azonosítóit
- *state* oszlopban határozzuk meg, hogy egy esemény mely állapotra vonatkozik (állapot azonosító)
- *st* az esemény kezdőpontját, míg *et* a végpontját tartalmazza

4. Az eszköz alkalmazása

Ebben a fejezetben tesztelés alá vetjük az általunk fejlesztett eszközt. Az algoritmus leírásában szereplő minta adatbázison hitelesítjük, majd egy valós műszaki rendszer szimulációja során generált eseménynaplón mutatjuk be működését.

4.1. Validáció

Az algoritmus leírásában is szereplő D_T minta adatbázist bemeneti fájlként megadva a programnak, annak lefutásával a következő eredményekre jutottunk:

Gyakoriság-küszöb: $\alpha = 0,66$, események távolságának toleranciája: $window = 3$.

3. Táblázat: A bemeneti Excel fájl:

<i>event</i>	<i>state</i>	<i>st</i>	<i>et</i>
1	1	1	5
1	3	1	4
1	5	2	10
1	2	2	8
1	6	2	5
2	2	4	6
1	4	3	7
2	3	33	38
2	6	34	39
2	4	25	37
2	5	27	42
3	2	28	32
2	1	30	40
3	3	30	38

A program által feltárt gyakori állapotszekvenciák a következők:

4. Táblázat: A kimeneti fájl

Sequence	Support	Conf.	Effective Time Spans			Sequence	Support	Conf.	Effective Time Spans	
1	0,67	1	[1 5]	[30 40]		[3 0 6 D 5]	0,67	1	[27 42]	[27 42]
2	1	1	[2 8]	[4 6]	[28 32]	[4 0 3 0 1]	0,67	1	[25 40]	[25 40]
3	1	1	[1 4]	[33 38]	[30 38]	[4 0 3 0 5]	0,67	1	[25 42]	[25 42]
4	0,67	1	[3 7]	[25 37]		[4 0 3 0 6]	0,67	1	[25 39]	[25 39]
5	0,67	1	[2 10]	[27 42]		[6 D 1 0 2]	0,67	1	[1 8]	[1 6]
6	0,67	1	[2 5]	[34 39]		[2 D 4 0 3 0 1]	0,67	1	[25 40]	[25 40]
[1 0 2]	0,67	1	[1 8]	[1 6]		[2 D 4 0 3 0 5]	0,67	1	[25 42]	[25 42]
[2 D 4]	0,67	1	[3 7]	[25 37]		[2 D 4 0 3 0 6]	0,67	1	[25 39]	[25 39]
[2 D 5]	1	2	[2 10]	[2 10]	[27 42]	[3 D 1 0 2 0 5]	0,67	1	[1 10]	[1 10]
[3 D 1]	1	2	[1 5]	[30 40]	[30 40]	[3 0 6 D 1 D 5]	0,67	1	[27 42]	[27 42]
[3 D 5]	0,67	1	[27 42]	[27 42]		[3 0 6 0 2 0 5]	0,67	0,67	[1 10]	[1 10]
[3 0 6]	1	1	[1 5]	[33 39]	[30 39]	[4 0 3 0 1 0 5]	0,67	1	[25 42]	[25 42]
[4 0 3]	0,67	1	[25 38]	[25 38]		[4 0 3 0 6 0 1]	0,67	1	[25 40]	[25 40]
[6 D 1]	0,67	1	[1 5]	[30 40]		[4 0 3 0 6 0 5]	0,67	1	[25 42]	[25 42]
[6 D 5]	0,67	1	[2 10]	[27 42]		[6 D 1 0 2 0 5]	0,67	1	[1 10]	[1 10]
[1 0 2 0 5]	0,67	1	[1 10]	[1 10]		[2 D 4 0 3 0 1 0 5]	0,67	1	[25 42]	[25 42]
[2 D 4 0 3]	0,67	1	[25 38]	[25 38]		[2 D 4 0 3 0 6 0 1]	0,67	1	[25 40]	[25 40]
[3 D 1 0 2]	0,67	1	[1 8]	[1 6]		[2 D 4 0 3 0 6 0 5]	0,67	1	[25 42]	[25 42]
[3 D 1 D 5]	0,67	1	[27 42]	[27 42]		[4 0 3 0 6 0 1 0 5]	0,67	1	[25 42]	[25 42]
[3 0 6 D 1]	0,67	1	[30 40]	[30 40]		[2 D 4 0 3 0 6 0 1 0 5]	0,67	1	[25 42]	[25 42]
[3 0 6 0 2]	0,67	0,67	[1 8]	[1 6]						

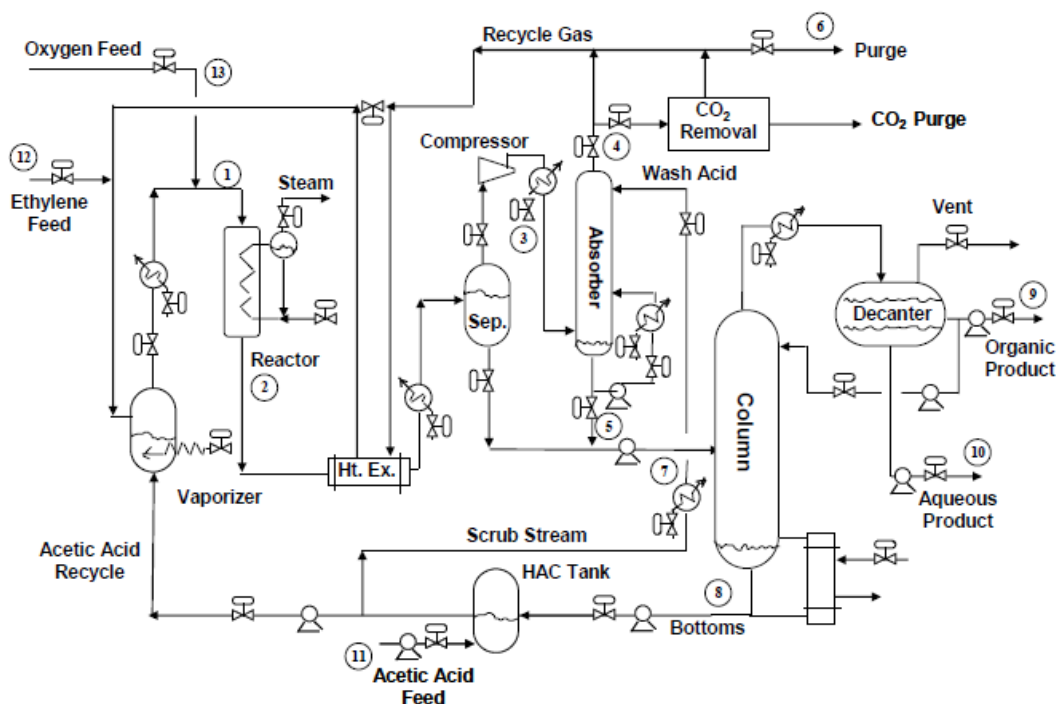
Ahogy az a 4. Táblázaton jól látható, a program rendkívül egyszerűen olvasható, és informatív kimenetet ad vissza a felhasználónak.

A kimeneti fájl felépítése

A táblázat első oszlopa megadja egy egyes gyakori állapotszekvenciákat, melyek a definíció szerint értelmezhetőek. Minden második elem a négy értelmezett reláció egyike, a jó olvashatóság érdekében visszaalakítva betűkké jelölésüket. A második oszlopban a szekvenciák gyakorisága található, mely definíció szerint az aktuális szekvencia támogatottságának, és a leggyakoribb degenerált szekvencia támogatottságának hányadosa. A kimenet harmadik oszlopa megadja az egyes feltárt gyakori szekvenciák megbízhatóságát, melyet a leírásnak megfelelően számítunk. A további oszlopokban az egyes gyakori állapotszekvenciákat támogató eseménysorok effektív időintervallumai olvashatóak le.

4.2. Vinil-acetát előállítási folyamat alarm menedzsment rendszerének fejlesztése több időbeli minták segítségével

A kidolgozott eszköz alkalmazását egy vinil-acetát (VAC) előállítására alkalmas technológia esetén mutatom be, mely folyamat elég komplex a sok folyamatváltozó együttes kezelésében rejlő lehetőségek illusztrálására. Az alkalmazott szimulátor alapját képző folyamatmodell 26 folyamatváltozót tartalmaz, melyek többsége között az összefüggések többnyire nemlineárisak. A szimulációs program – melyet Rong Chen és Kedar David készített 2002-ben –, nyílt forráskódú, így képesek voltunk a rendszerbe zavarásokat vinni, majd az eredményeket kiértékelni.



1. Ábra: A modellezett vinil-acetát előállítás folyamatábrája

A szimuláció során alkalmazott zavarások

A szimulátorhoz annak szerzői olyan MATLAB programot mellékeltek, melyben minden szükséges paramétert deklarálnak a szimulációhoz: a technológiai elemek mérete, a bemeneti áramok, a vezérlők paraméterei, valamint a jeladók késleltetése.

Az említett programot a szerzők úgy alakították ki, hogy meghívása során különböző zavarási azonosítókat adhatunk meg 1-től 8-ig. Ezek a zavarások a következők:

1. A reaktor kimeneti hőmérsékletének alapjele 8°C -al csökken (159-ről 151-re)
2. A reaktor kimeneti hőmérsékletének alapjele 6°C -al megnő (159-ről 165-re)
3. Az oszlop alján a H_2O tartalom alapjele 9%-al megnő (9%-ról 18%-ra)
4. A párologtató folyadék bemeneti tömegárama 0.44 kmol/perc -el megnő (2,2-ről 2,64-re)
5. A friss HAc betáplálás megszűnik
6. A friss O_2 betáplálás megszűnik
7. A C_2H_6 összetétel 0,001-ről 0,003-ra változik a friss C_2H_4 betáplálásban
8. Az oszlop táplálása megszűnik

Ezek a zavarások a szimuláció 10. percében következnek be, és alapértelmezetten a szimuláció végéig, valamint az 5-ös, 6-ös illetve a 8-as zavarás esetében 5 percig tartanak. Ezen a ponton módosítottuk a programot – hogy a hibák által előidézett szekvenciák jól definiálhatóak legyenek, illetve két hiba előfordulása között a folyamat újra stacioner állapotba jusson – az egyes zavarások hosszát paraméterben megadhatóvá tettem.

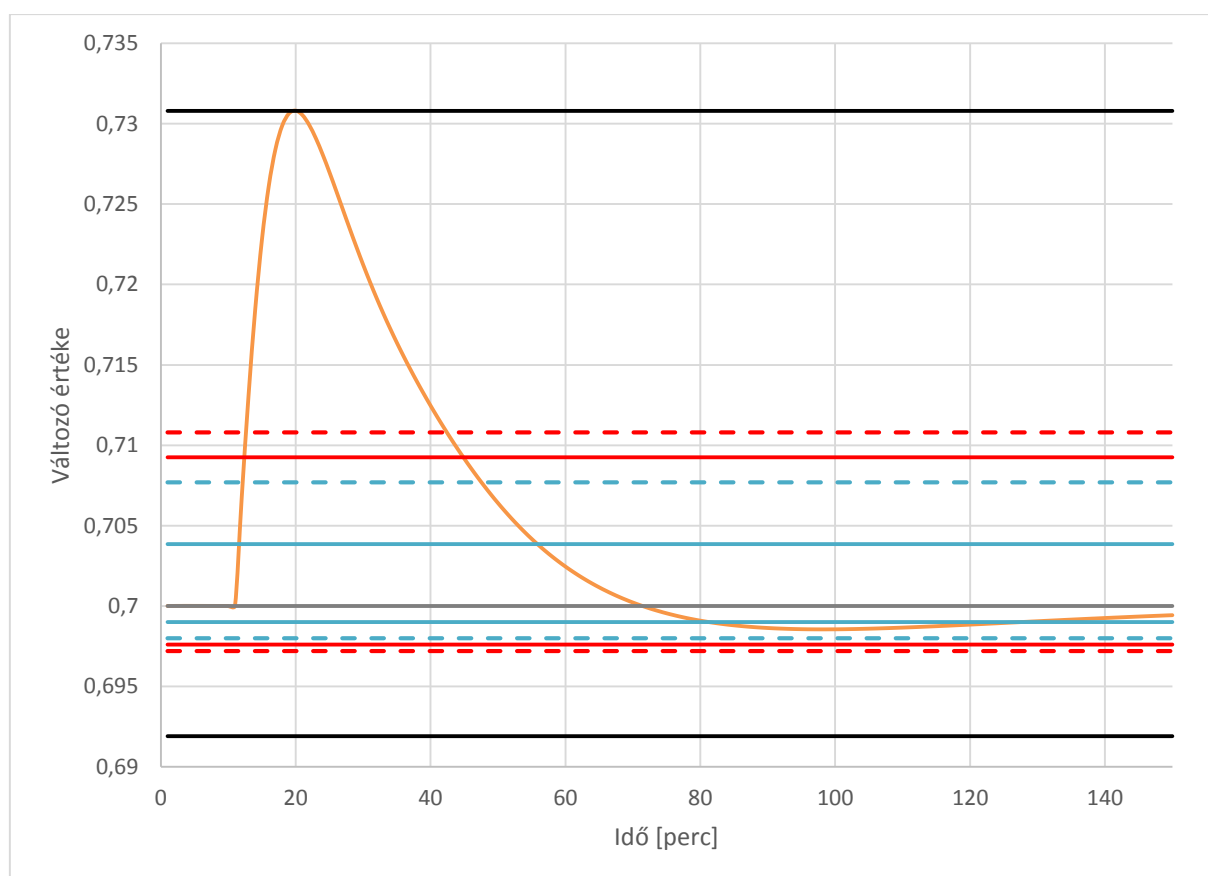
A rendszerváltozók vizsgálata

Ahhoz, hogy képesek legyünk alarmokat definiálni az egyes folyamatváltozókon, meg kellett vizsgálnunk azoknak viselkedését a különböző zavarások során, majd összevetni ezeket a stacioner értékekkel. Alarmok azokra a változókra definiálhatóak, melyek viselkedése olyan, hogy a zavarás megszűntével stacioner állapotba visszajutnak. A változók elemzéséhez zavarás nélküli esetben, valamint minden egyes zavarás esetén egy 1000 perces vizsgálatot végeztem, a kapott adatsorokat grafikonokon ábrázoltam, majd ezeket összevettem. A vizsgálat során megfigyelhető volt, hogy a 26 folyamatváltozó közül nem mindegyik viselkedése megfelelő ahhoz, hogy azokon alarmokat definiálni tudjunk. Számos változó a zavarások bekövetkezése után új stacioner értékre állt be. Ezeken a változókon minden zavarás esetében ezt a tendenciát tudtuk megfigyelni, így a vizsgált változók számát lecsökkentettük 12-re (melyek mindegyikére és minden zavarás esetén igaz, hogy visszaállnak a zavarás előtti állapotra).

Eseménynaplók készítése

A szimulátor által generált adatsorok alapvetően még nem alkalmasak a szekvencia elemzés elvégzésére, így további lépésként értelmezni kellett rajtuk a különböző állapotokat (alarmok, figyelmeztetések), majd a javasolt eszköz bemeneti formátumára kellett alakítani a kapott a log fájlokat. A feladat elvégzésére MATLAB programokat hoztunk létre.

A figyelmeztetések és alarmok definiálásához első lépésként egy struktúrába beolvastuk a szimulátor által generált adattáblákat (minden zavarásra egy szimuláció), melyek minden oszlopa egy folyamatváltozó alakulása az idő függvényében. A struktúrából minden egyes változóra meghatározható annak minimális és maximális értéke. Ezek meghatározása után az adatsorok elemeit interpoláció segítségével kódoltuk. Két-két interpolációs csomópontot határoztunk meg a stacioner illetve a minimum és a maximum értékek között. Az adatsor értékeit az interpolációs függvénnyel a legközelebbi ponthoz rendeltük, vagyis a határok a pontok közötti intervallumok felénél jöttek létre.



2. Ábra: Egy változó értékeinek alakulása az állapothatárokon

A 2. ábrán megfigyelhetők az egyik folyamatváltozóra meghatározott interpolációs pontok és határok. Szürkével a változó stacioner értéke, feketével a változó által felvett minimum és

maximum értékek. Piros, illetve kék vonallal láthatóak rendre az alarm, valamint a figyelmeztetési határok. Szaggatott vonallal a meghatározott interpolációs pontokat jelöltem.

A kódolás a következőképpen történt:

Amennyiben az adatsor értéke

- a figyelmeztetési határokon belül van 0-val,
- a figyelmeztetési és alarm határok között van 1, illetve -1-el,
- az alarm határokon kívül van 2, illetve -2-vel jelöljük.

A pozitív előjelű kódokat értelemszerűen akkor használjuk, ha egy adatpont eltérése a stacionerhez képest pozitív.

Ezen technikát alkalmazva minden egyes változó minden adatpontját kódoltuk, majd következő lépésként létrehoztuk belőlük az eseménynaplót, melynek létrehozásához az egyes állapotok azonosítóit, illetve kezdő- és végpontjait kellett meghatároznunk.

Az állapotazonosítókat a következő képlettel határoztam meg:

$$e_{id} = \text{változó száma} \cdot 10 + x$$

,ahol a *változó száma* az adattáblában annak az oszlopnak a száma, amelyben elhelyezkedik, *x* pedig az esemény kódja, mely ha az adatpont:

- pozitív alarm (2), akkor 3,
- pozitív figyelmeztetés (1), akkor 1,
- negatív figyelmeztetés (-1), akkor 2,
- negatív alarm (-2), akkor 4 értéket vesz fel.

Az események kezdő-, illetve végpontjait az alapján határoztam meg, hogy az adattáblában mely sorban kezdődik, valamint végződik egy összefüggő eseménysor.

Miután minden állapot meghatározásra került, mindössze meg kell számlálni az előfordulásait, és ez alapján az eseménynaplóban sorszámozni azokat.

Fontos megjegyezni, hogy a változók stacioner közeli alakulása nem minősül eseménynek, mivel a feladat szempontjából információtartalma nincs, csak lassítaná a feltáró algoritmust, és hosszabb szekvenciákat eredményezne!

Az esettanulmány eseménynaplója a 8 hiba 1000 perces szimulációjának összefűzéséből jött létre, így 154 esemény idéz elő 48 egyedi állapotot. A következő címen megtekinthető:

<https://copy.com/mcOvCsWQZCpw7j8A>

Az eredmények

A javasolt eszköz a létrehozott eseménynaplón a következő paraméterekkel futtattuk: gyakoriság küszöb, $\alpha = 0,19$; esemény távolság tolerancia, $window = 9$ [perc].

5. Táblázat: A kapott kimeneti fájl:

Sequence	Supp.	Conf.	Effective Time Spans			
23	0,375	1	[14 117]	[3025 3059]	[5024 5025]	[5027 5028]
			[5030 5033]	[5043 5054]		
32	0,625	1	[1016 1018]	[1082 1106]	[3085 3130]	[4020 4021]
			[4032 4032]	[5019 5020]	[5038 5038]	[5066 5074]
			[7067 7072]	[7110 7130]		
33	0,375	1	[19 77]	[3015 3017]	[3034 3058]	[4037 4045]
			[5024 5035]	[7036 7056]		
34	0,25	1	[1019 1081]	[4022 4031]	[5039 5065]	[7073 7109]
43	0,3125	1	[14 116]	[3024 3059]	[5024 5025]	[5027 5034]
			[5042 5054]			
52	0,5	1	[16 21]	[4028 4029]	[4036 4037]	[5018 5018]
			[5029 5029]	[5047 5049]	[5060 5075]	[7071 7093]
53	0,375	1	[3017 3020]	[3033 3043]	[4043 4049]	[5030 5032]
			[5040 5044]	[7045 7058]		
63	0,3125	1	[1014 1027]	[3017 3021]	[3032 3056]	[5029 5030]
			[5040 5045]			
72	0,25	1	[14 19]	[4031 4032]	[5019 5019]	[5044 5053]
82	0,375	1	[17 31]	[4038 4040]	[5021 5022]	[5032 5032]
			[5052 5054]	[5066 5074]		
83	0,25	1	[1020 1029]	[3023 3042]	[5034 5035]	[5045 5049]
92	0,25	1	[3019 3035]	[4038 4039]	[7027 7029]	[7045 7046]
93	0,25	1	[3048 3080]	[4016 4018]	[4048 4053]	[7052 7097]
102	1	1	[11 21]	[92 129]	[1012 1012]	[1157 1269]
			[2013 2049]	[2090 3003]	[3014 3021]	[3056 3484]
			[4015 4029]	[4042 4111]	[5016 5016]	[5025 5025]
			[5051 5058]	[6017 6053]	[7018 7031]	[7043 7077]
113	0,25	1	[1014 1095]	[3026 3052]	[5030 5031]	[5044 5053]
[23 D 43]	0,25	0,8	[3024 3059]	[5027 5034]	[5027 5034]	[5042 5054]
[32 B 34]	0,25	0,4	[1016 1081]	[4020 4031]	[5038 5065]	[7067 7109]
[32 B 52]	0,25	0,4	[4020 4029]	[4032 4037]	[5019 5029]	[5038 5049]
[34 B 32]	0,25	1	[1019 1106]	[4022 4032]	[5039 5074]	[7073 7130]
[52 B 82]	0,3125	0,625	[4028 4040]	[4036 4040]	[5018 5022]	[5029 5032]
			[5047 5054]			
[102 B 23]	0,25	0,25	[3014 3059]	[5016 5025]	[5025 5028]	[5025 5033]
[102 B 32]	0,25	0,25	[1012 1018]	[4015 4032]	[5016 5020]	[5051 5074]
[102 B 52]	0,25	0,25	[4015 4037]	[5016 5018]	[5025 5029]	[5051 5075]
[102 B 82]	0,25	0,25	[4015 4040]	[5016 5022]	[5025 5032]	[5051 5074]
[32 B 34 B 32]	0,25	0,4	[1016 1106]	[4020 4032]	[5038 5074]	[7067 7130]
[32 B 52 B 82]	0,25	0,4	[4020 4040]	[4032 4040]	[5019 5032]	[5038 5054]
[52 B 82 B 53]	0,25	0,5	[4028 4049]	[4036 4049]	[5018 5032]	[5029 5044]
[102 B 23 O 43]	0,25	0,25	[3014 3059]	[5016 5025]	[5025 5034]	[5025 5034]
[102 B 23 O 43 B 23]	0,25	0,25	[5016 5028]	[5016 5033]	[5025 5054]	[5025 5054]
[102 B 23 O 43 B 23 O 43]	0,25	0,25	[5016 5034]	[5016 5034]	[5025 5054]	[5025 5054]

4.3. Az eredmények kiértékelése

Tekintsük a [102 B 23 O 43 B 23 O 43] szekvenciát. Részszekvenciáján ([102 B 23 O 43 B 23]) megfigyelhető, hogy támogatottságuk, illetve megbízhatóságuk is megegyezik. Az effektív időtartamokat is figyelembe véve kijelenthetjük, hogy amennyiben [102 B 23 O 43 B 23] állapot szekvencia bekövetkezik, akkor biztos, hogy a technológia következő jelzése a 4-es folyamatváltozó pozitív alarmjelzés lesz, még mielőtt a 2-es változó pozitív alarmjelzése véget érne. Megfigyelhető az is, hogy ez a szekvencia meghatározza azt is, hogy mely hiba következett be. Mivel tudjuk, hogy a szimuláció 5015 percében következik be a 6-os azonosítóval rendelkező hiba, kijelenthetjük, hogy a [102 B 23 O 43 B 23] szekvencia egyértelműen azonosítja ezt a hibát, és a 43-as állapot alarmjelzését elnyomhatjuk.

Hasonlóakat tudunk megfogalmazni az [52 B 82 B 53] szekvenciáról is. Látható, hogy amennyiben [52 B 82] bekövetkezik, akkor az 5-ös folyamatváltozó nagy valószínűséggel pozitív alarmot fog generálni. Ez előrejelzés szempontjából rendkívül informatív. Az effektív időtartamokat összevetve látható, hogy [52 B 82] esemény első két előfordulása után 3 perccel következik 53-as vészjelzés. Ez az idő – amennyiben az operátori képeken az így feltárt összefüggések egy előrejelző rendszerben feltüntetésre kerülnek – rendkívül hasznos lehet es esetleges alarm flood elkerüléséhez, illetve katasztrófa elkerüléséhez.

4.4. További lehetőség: tanító minták feltárása

Amennyiben az egy hiba eseménynaplóját képesek vagyunk izoláltan elemezni (ahogy azt meg tudjuk tenni jelen esetünkben az egyes különálló eseménynaplók felhasználásával), minimális támogatottsági küszöb mellett, akkor képesek vagyunk feltárni azokat a szekvenciákat, melyek „leírják” a hiba lefolyását. Ezeket a mintákat tanító mintáknak nevezzük, és segítségükkel képesek lehetünk a folyamatokat korábban visszajuttatni a normál állapotukba, a megfelelő vezérlés kialakítása mellett. A tanító minták feltárásának problémája a rendkívül nagy számítási igény: egy x db eseményből álló eseménynaplóból feltárt tanító minta akár x hosszúságú szekvenciát jelenthet, miközben a futási idő exponenciális összefüggésben áll a hosszúsággal.

5. Összefoglalás

A folyamatirányító rendszerek nem csak a technológia szabályozásában működnek közre, hanem a nem kívánt üzemállapotok elkerülése érdekében olyan rendszereket is tartalmaznak, melyek amennyiben a folyamatváltozók nem kívánt tartományba jutása esetén figyelmeztető és vészjelzéseket (warningokat és alarmokat) adnak. Az elmúlt időszakban a legtöbb tragédia és üzemi baleset a kezelhetetlen mennyiségű alarmjelzéshez köthető. Kutatásunk célja az alarmok racionalizálására alkalmas eszköz, illetve ezen eszköz alapjául szolgáló algoritmus fejlesztése volt. Az általunk javasolt megközelítés alapgondolata, hogy a folyamatirányító rendszerek eseménynaplói alapján javaslatok tehetők az üzemek alarm menedzsment rendszereinek optimalizálására, az alarmok számának csökkentésére, prioritásuk hangolására.

A kidolgozott megközelítésmód kulcsa, hogy a vészjelzések eseményeknek tekinthetők és ezen események szekvenciákba szervezhetők, így meghatározható, hogy mely események függenek egymástól, mely események előfordulása jelzi előre a további vészjelzések kialakulását. A követelményeknek való megfelelés érdekében arra a következtetésre jutottunk, hogy olyan eszközre van szükségünk, mely alkalmas több változó együttes kezelésére, valamint az események közti kapcsolatok/átfedések feltárására. Választásunk egy olyan algoritmus fejlesztésére esett, mely képes olyan több-időbeli (ún. multi-temporal) minták kezelésére, melyekben az események között egyszerre több reláció is fennállhat. Megközelítésünk előnye, hogy az algoritmus információt ad az alarmok gyakoriságára, a feltárt szekvencia javaslatot ad a felesleges alarmok elnyomásának logikájának meghatározására.

A fejlesztett eszközünk alkalmazhatóságát egy vinil-acetát előállítására alkalmas technológia alarm jelzéseinek racionalizálása során illusztráltuk, mely példa alkalmas volt arra, hogy a multi-temporal szekvencia elemzés legnagyobb előnyét - a több változón értelmezett alarmjelzések szimultán kezelését – bemutassa.

A feltárt *gyakori* és *megbízható* szekvenciák alapján a folyamatirányítási rendszerek jelzéseinek számát képesek vagyunk redukálni, a fennmaradó jelzések prioritását optimalizálni, valamint kialakítani predikciós sémákat.

6. Irodalomjegyzék

- [1] Larry O'Brien, Dave Woll – Alarm management strategies –2004
- [2] Mika Klemettien, Heikki Mannila, Hannu Toivonen – Rule Discovery in Telecommunication Alarm Data –Journal of Network and Systems Management, Vol. 7. No. 4. – 1999
- [3] Saeed Salah, Gabriel Maciá-Fernández, Jesús E., Díaz-Verdejo A model-based survey of alert correlation techniques — Computer Networks 57 – 2013
- [4] Tim Southam – Alarm management – An introduction – prezentáció, 18. oldal – 2013
- [5] Sandeep R. Kondaveeti, Iman Izadi, Sirish L. Shah, Tim Black, Tongwen Chen – Graphical tools for routine assessment of industrial alarm systems –Computers and Chemical Engineering 46 39-47 – 2012
- [6] Tongyan Li, Xingming Li – Novel alarm correlation analysis system based on association rules mining in telecommunication networks –Information Sciences 180 2960-2978 – 2010
- [7] Xiaoxiao Kong, Qiang Wei, Guoqing Chen – An approach to discovering multi-temporal patterns and its application to financial databases –Information Sciences Vol. 180, Issue 6 873-885 – 2010
- [8] Zheng Chen, Susan Lu, Sarah Lam – A hybrid system for SPC concurrent pattern recognition –Advanced Engineering Informatics Vol.21, Issue 3 303-310 – 2007

7. Köszönetnyilvánítás

Köszönöm dr. Abonyi Jánosnak a rengeteg segítséget, amit kaptam, hogy ez a dolgozat létrejöhessen.