INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Wybrane problemy

Tom 11

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2009

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Wybrane problemy
Tom 11

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2009

Wykaz opiniodawców artykułów zamieszczonych
w niniejszym tomie:

prof. dr hab. inż. Jerzy HOŁUBIEC
dr inż. Lech KRUŚ
doc. dr hab. inż. Wiesław KRAJEWSKI
doc. dr hab. Jacek MALINOWSKI
dr inż. Edward MICHALEWSKI
prof. dr Adam SKOREK
dr hab. Ryszard SMARZEWSKI
prof. dr hab. inż. Andrzej STRASZAK
dr Dominik ŚLĘZAK
prof. dr hab. inż. Stanisław WALUKIEWICZ
doc. dr hab. Sławomir ZADROŻNY

ISBN 9788389475220

Druk: Zakład Poligraficzny Jerzy Kosiński, Warszawa

OPTYMALIZACJA SYSTEMU KODOWEGO Z UŻYCIEM ODLEGŁOŚCI HAMMINGA PO WPROWADZENIU PROBLEMU ZEROWEGO

Andrzej Michałek

Studia Doktoranckie IBS PAN

Codebook system is one of the fault localization techniques used by the event correlation. This paper introduces the implementation of the "problem zero" P0 and the way of optimizing the codebook system, after implementation of P0. Problem zero P0 describes the balanced state of the system.
The implementation of the problem zero is changing the way of the codebook optimization and allows defining the level of codebook system optimization.

Key words: *event correlation, codebook system, problem zero.*

Niniejsza praca przedstawia, krótki przegląd technik lokalizacji awarii w systemach analizy korelacyjnej oraz przykład optymalizacji metody codebook – systemu kodowego przy wprowadzeniu pojęcia problemu zerowego. Przedstawiona została propozycja wprowadzenie pojęcia problemu zerowego (P0), jego reprezentacji graficznej oraz zastosowanie problemu zerowego dla optymalizacji systemu kodowego. W oparciu o definicję odległości Hamminga wprowadzone zostalo pojęcie stopnia optymalizacji systemu kodowego oraz pojęcie odległości Hamminga dla systemu kodowego.

Praca składa się z czterech części. W części pierwszej zostały przedstawione wybrane zadania analizy korelacyjnej oraz krótki przegląd technik lokalizacji awarii w systemach analizy korelacyjnej. W części drugiej została przedstawiona propozycja wprowadzenie pojęcia problemu zerowego, jego reprezentacji graficznej oraz zastosowanie problemu zerowego dla optymalizacji systemu kodowego. Przedstawiona zostala również definicja odległości Hamminga dla systemu kodowego. W części trzeciej został przedstawiony przykład optymalizacji systemu kodowego, definicja stopnia optymalizacji systemu kodowego oraz definicja zoptymalizowanego systemu kodowego. Przedstawiono również przyklad macierzy systemu kodowego zoptymalizowanej w stopniu trzecim, drugim i pierwszym. Część czwarta stanowi podsumowanie.

Słowa kluczowe: *analiza korelacyjna, system kodowy, problem zerowy.*

1. Zadania analizy korelacyjnej i techniki lokalizacji usterek

W systemach analizy korelacyjnej będziemy stosowali pojęcie symptomu. „Symptomy są zewnętrznymi manifestacjami niepowodzenia. Symptomy są postrzegane jako alarmy – komunikaty o potencjalnych niepowodzeniach. Te komunikaty mogą pochodzić z zewnętrznych źródeł, na przykład agentów systemów zarządzania i mogą być przesyłane za pomocą protokołów takich jak trapy SNMP z systemów monitorowania sieci, korzystających z systemów diagnostyki, z logów systemowych i komunikatów urządzeń" [2].

Jednym z zadań systemów analizy korelacyjnej jest kojarzenie symptomów, jakie wystąpiły w badanym środowisku i na podstawie ich analizy określenie przyczyn, które doprowadziły do ich wygcnerowania. Inaczej mówiąc jednym z zadań analizy korclacyjnej jest określenie źródła obserwowanych symptomów. W tym celu stosuje się różnorodne metody analizy i kojarzenia komunikatów w zależności od celu, jaki zamierzamy osiągnąć, posiadanych środków technicznych, doświadczenia oraz czasu przeznaczonego na budowę systemu.

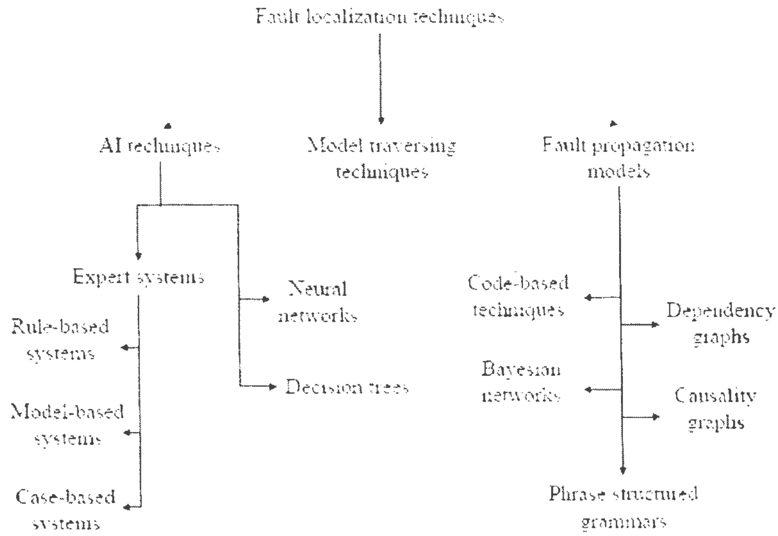
Spośród wielu czynników, jakie mają wpływ na efektywność procesu analizy korelacyjnej niewątpliwie na uwagę zasługują szybkość i dokładność, z jaką przebiega proces [1]. Jeśli zależy nam tym, aby analiza przebiegała w czasie rzeczywistym, to pośrednio zgadzamy się na to, że szybki proces analizy korelacyjnej dostarcza przybliżonych, a przez to niezbyt dokładnych wyników. I odwrotnie dokładny proces analizy korelacyjnej wymaga czasu.

Istnieje wiele technik, będących podstawą budowy mechanizmu analizy korelacyjnej. Małgorzata Steinder i Adarshapal S.Sethi w pracy „The present and future of the event correlation" [2], zamieszczają następujący rysunek przedstawiający klasyfikację technik lokalizowania awarii:

Klasyfikacja przedstawiona na Rys. 1 jest podstawą dla wielu opracowań dotyczących technik analizy korelacyjnej.

2. Wprowadzenie pojęcia problemu zerowego i jego reprezentacji graficznej P0. Wprowadzenie pojęcia odległości Hamminga dla systemu kodowego.

Codebook Systems – książka kodowa - system kodowy. Jest to system zbudowany w oparciu o znaczną wiedzę ckspercką z danej dziedziny i dobrą znajomość diagnozowanego środowiska. Na podstawie tej wiedzy określamy, jakie zjawiska – zmiany stanu systemu mogą wystąpić w systemic oraz jakie są wymierne następstwa tych zjawisk. W oparciu o korelacje symptomów i problemów budowana jest macierz, służąca do diagnozy badanego systemu.



Rys. 1. Klasyfikacja techniki lokalizowania awarii [2].

Wyjściem do budowy książki kodowej jest graf skierowany acykliczny [3].

W dalszej części zmiany stanu systemu będziemy określali jako problemy oznaczane literą P. Każde wystąpienie problemu owocuje wystąpieniem pewnej liczby symptomów, które możemy rejestrować. Symptomy będziemy oznaczali literą S.

Na podstawie posiadanej wiedzy oraz grafu budujemy macierz grupującą problemy i symptomy. Dobrze zbudowana macierz pozwoli nam w oparciu o wiedzę dotyczącą symptomów, jakie pojawiły się w systemie określić drogą dedukcji ich źródło, czyli wskazać na problem, zmianę stanu systemu, jaka miała miejsce. Idea systemu kodowego polega na takim pogrupowaniu symptomów, aby na ich podstawie jednoznacznie określić, co było ich przyczyną, to znaczy jednoznacznie określić, jaka zmiana stanu systemu miała miejsce. Bazująca na zestawie symptomów, jakie wystąpiły i jakie udało się nam zarejestrować określamy, jaki problem miał miejsce. System kodowy zawsze na wyjściu podaje wynik diagnozy, czyli jednoznacznie określa, jaki problem wystąpił. Zakłada się, że stosując metodę książki kodowej nie prowadzi się diagnozy pojedynczych symptomów, ale zawsze pod uwagę jest brany cały, określony wcześniej w macierzy zbiór symptomów. Suma wartości symptomów tworzy wektor będący reprezentantem stanu bieżącego badanego środowiska [3]. W celu przyśpieszenia procesu diagnozy wprowadzana jest optymalizacja zbudowanej macierzy. Najczęściej optymalizacja tworzonej macierzy polega na wyznaczeniu unikalnych kombinacji – wektorów, które jednoznacznie wskazują na przyczynę pojawienia się komunikatów wejściowych. Różnice pomiędzy wektorami są mierzone po przez odległość Hamminga.

Wikipedia [4] definiuje odległość Hamminga, którą będziemy oznaczali przez DH, jako miarę odmienności dwóch ciągów o takiej samej długości wyrażającą liczbę miejsc, na których te dwa ciągi się różnią. Przy czym nie bez znaczenia jest wielkość odległości Hamminga. Ma ona wpływ na możliwość wykrycia i korekty błędów jakie mogą się pojawić przy zapisie porównywanych ciągów. I tak dla odległości Hamminga:

DH = 1 - możliwe jest rozróżnienie ciągów, nie jest możliwe wykrycie błędów.

DH = 2 - jest możliwe wykrycie błędu, nie jest możliwa korekcja.

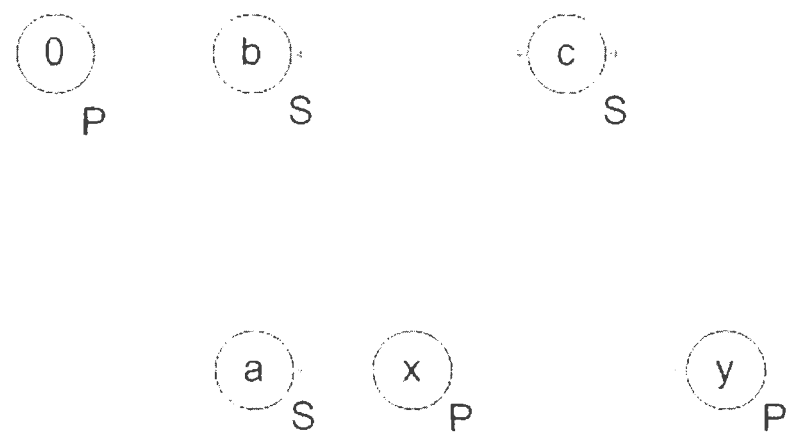
DH = 3 - możliwa jest korekcja błędu.

Xin Hu [3] oraz S.Yemini, S. Kliger, E. Mozes Y. Yemini D. Ohsie [5] w swoich pracach obliczają odległość między problemami posługując się pojęciem promienia książki kodowej, który jest określany jako połowa odległości Hamminga pomiędzy kodami problemów. Takie podejście powoduje, że odległość Hamminga równa 1, jest rozpatrywana jako promień o wartości 0,5. Ponieważ taki zapis jest nieczytelny przy budowie macierzy odległości (tabela nr 4) w dalszej części będziemy używać odległości Hamminga zamiast promienia książki kodowej.

Zanim przejdziemy do analizy przykładu konstrukcji i optymalizacji systemu kodowego wprowadzimy pojęcie problemu „zerowego" i jego reprezentacji graficznej P0.

Załóżmy, że mamy do czynienie z następującą sytuacją przestawioną na rysunku 2. Przez P oznaczmy problemy a przez S oznaczymy symptomy. Na rysunku 2 widzimy problemy Px, Py i P0 oraz symptomy Sa, Sb i Sc.

Pojawienie się problemu Px powoduje wygenerowanie symptomów Sa, Sb i Sc. Pojawienie się problemu Py powoduje wygenerowanie symptomu Sc. Możemy zapisać wartość symptomów dla każdego problemu, nadając im wartości ze zbioru dwuelementowego {0,1}. W sytuacji, kiedy wystąpienie problemu powoduje wystąpienie symptomu nadajemy symptomowi wartość 1. W sytuacji przeciwnej, kiedy wystąpienie problemu nie powoduje wystąpienia symptomu, nadajemy symptomowi wartość 0.



Rys. 2. Wprowadzenie symbolu graficznego dla problemu zerowego P0.

W rezultacie otrzymujemy następujący zapis:

Px(SA=1, Sb=1, Sc=1)
Py(Sa=0, Sb=0, Sc=1).
Możemy ten zapis przedstawić w postaci tabeli:

	Sa	Sb	Sc
Px	1	1	1
Py	0	0	1

Tabela 1.

Wartości z tabeli 1 możemy przedstawić również w postaci łańcucha znaków, otrzymując następujące łańcuchy znakowe - wektory reprezentujące problemy Px i Py:
Px = 111
Py = 001

Bazując na takim zapisie jesteśmy w stanie obliczyć odległość Hamminga pomiędzy Py i Px.

Px = **11**1
Py = **00**1
DH(Px,Py) = 2

Formalna analiza pozwala wyliczyć odległość Hamminga pomiędzy Px i Py, DH(Px,Py) = 2. Zwróćmy jednak uwagę, że problem Py ma przypisany zaledwie jeden symptom. Błąd, który może spowodować brak możliwości jego odczytania spowoduje, że problem Py zostanie niezauważony.

Stan poprawnej pracy systemu, kiedy nie występuje żaden z badanych problemów i tym samy nie odnotowujemy żadnego z badanych symptomów określamy jako problem zerowy, który oznaczamy przez P0. Możemy również powiedzieć, że jest to stan zbalansowany, poprawny badanego systemu.

W systemach analizy korelacyjnej możemy mieć do czynienia z sytuacją, kiedy nie zrejestrujemy wystąpienia żadnego z badanych symptomów. W naszym przykładzie z rysunku nr 2 oznacza to brak symptomów Sa, Sb, Sc. Może to być sytuacja stanu poprawnej pracy badanego systemu, którą oznaczamy przez P0. Dla P0 wartość wszystkich symptomów Sa, Sb, Sc wynosi 0. Możemy mieć jednak do czynienie z sytuacją, kiedy na skutek wystąpienia błędów nie odnotowaliśmy żadnego z symptomów Sa, Sb, Sc. Czy i kiedy możemy odróżnić od siebie te dwa stany systemu?

Zmierzmy odległości Hamminga pomiędzy stanem pracy poprawnej a pozostałymi stanami pracy systemu, czyli odległości pomiędzy P0 a Px i Py. Na tej podstawie możemy określić czy jesteśmy w stanie wykryć i ewentualnie skorygować błąd pomiaru symptomów świadczących o niepoprawnej pracy systemu. To znaczy, moglibyśmy określić czy i kiedy potrafimy odróżnić stan poprawnej pracy systemu (P0) od stanu, kiedy nie udało się nam zarejestrować symptomów świadczących o problemach, jakie się pojawiły.

Wprowadzenie P0 pozwala na wyliczenie dystansu Hamminga pomiędzy problemami Px i Py a P0. Pomiar odległości Hamminga pomiędzy P0 a pozostałymi problemami ma znaczenie przy optymalizacji systemu kodowego i pozwala na bieżącą kontrolę wprowadzonych zmian.

Przedstawmy zapis z tabeli nr 1 wzbogacony o wprowadzenie problemu P0: tabela 2.

	Sa	Sb	Sc
Px	1	1	1
Py	0	0	1
P0	0	0	0

Tabela 2.

Wprowadzenie P0 spowoduje, że będziemy mogli wyliczyć odległość pomiędzy P0 i Px oraz pomiędzy Py i P0. Przedstawmy zapis z tabeli nr 2 w postaci wektorów dla każdego z problemów Px, Py i P0.

Px = 110

Py = 001

P0 = 000

Odległość pomiędzy Px i P0 wynosi 2, odległość pomiędzy Px i P0 wynosi 3 a pomiędzy Py i P0 wynosi 1.

DH(Px,Py) = 2

DH(P0,Py) = 1

DH(P0,Px) = 3

Dokonując optymalizacji systemu kodowego należy zwrócić uwagę na to, aby zbadać odległość Hamminga pomiędzy wszystkimi problemami, łącznie z problemem P0.

Odległość Hamminga dla systemu kodowego jest równa najmniejszej odległości Hamminga pomiędzy dwoma problemami tego systemu.

W przykładzie z rysunku nr 2 odległość Hamminga dla systemu kodowego wynosi 1.

3. Przykład optymalizacji systemu kodowego przy wykorzystaniu pojęcia problemu zerowego P0.

W pracach dotyczących mechanizmów analizy korelacyjnej książkę kodową tworzy się na podstawie grafu skierowanego acyklicznego.

Spróbujmy prześledzić cały proces przygotowania i optymalizacji macierzy systemu kodowego na podstawie przykładu zamieszczonego poniżej.

Założenie:

Znamy dobrze zarządzane środowisko i na podstawie posiadanej wiedzy jesteśmy w stanie określić zachowanie tego systemu w różnych sytuacjach. Możemy narysować graf acykliczny skierowany, który pomagałby odzwierciedlić stan naszej wiedzy. Symptomy możemy traktować jako komunikaty na wejściu systemu. Każdy problem powoduje wystąpienie jednego bądź określonej grupy symptomów.

Pojawienie się problemu oraz wartości korespondującej grupy symptomów są zapisywane w postaci tablicy, macierzy, która podlega optymalizacji. Kombinacja symptomów wejściowych jest traktowana jako wektor reprezentujący dane na wejściu systemu. W celu polepszenia wydajności systemu projektant stara się zmniejszyć ilość możliwych kombinacji, minimalizując ilość symptomów, jakie są brane pod uwagę i jakie są niezbędne dla określenia poprawnej diagnozy, czyli do jednoznacznego określenia problemu. Problem nie oznacza wyłącznie wystąpienia sytuacji niepożądanej i może oznaczać po prostu zmianę stanu systemu. Wyjdźmy od następującej sytuacji przedstawionej na rysunku nr 3 w postaci grafu.

Rysunek 3 obrazuje stan naszej wiedzy, dotyczącej występowania problemów i ich symptomów w badanym systemie, przedstawionej w postaci grafu skierowanego acyklicznego.

Problemy i ich symptomy są ponumerowane i powiązane ze sobą poprzez kolejność, w jakiej się pojawiły i zależności pomiędzy nimi.

Przykładowo wystąpienie problemu oznaczonego na rysunku przez P1 w dalszej kolejności prowadzi do wystąpienia symptomu S8, którego wystąpienie powoduje pojawienie się z kolei symptomu S9. Jednocześnie wystąpienie problemu P1 powoduje równolegle pojawienie się symptomu S11, który z kolei powoduje pojawienie się symptomu S7 i S14, przy czym jak wynika z rysunku symptom S11 może również pojawić się w wyniku wystąpienia problemu P2. Sytuacja wygląda na dosyć zagmatwaną i przedstawia układ nie optymalizowany.

Dla każdego problemu oraz symptomu, jakie mogą się pojawić w systemie budujemy matrycę ich możliwych kombinacji przedstawioną w tabeli 3.

	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
P1	1	0	1	1	1	1	1	0	0	1	1
P2	1	0	1	0	0	1	1	1	1	1	1
P3	1	0	0	0	1	1	0	0	0	1	0
P4	1	1	0	0	0	1	0	1	1	1	0
P0	0	0	0	0	0	0	0	0	0	0	0

Tabela 3.

Tabela 3 przedstawia nie zoptymalizowany system kodowy.
Na podstawie wartości zapisanych w tabeli 3 możemy zbudować, w postaci łańcucha znaków o jednakowej długości, wektory reprezentujące stan danych na wejściu systemu. Wektory te są unikalne dla każdego z problemów z tabeli 3.

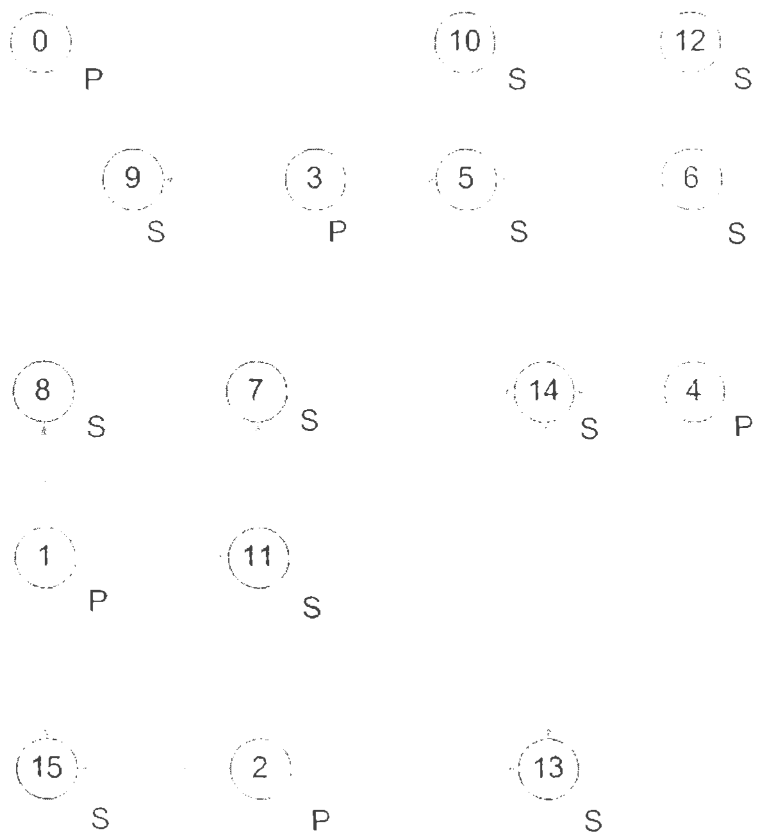
P1 = 1011110011

P2 = 1010011111

P3 = 1000110010

P4 = 1100011110

P0 = 0000000000



Rys. 3. Graf skierowany acykliczny obrazujący badany system.

Kombinacja symptomów wejściowych jest traktowana jako wektor reprezentujący dane na wejściu systemu.

Na podstawie porównania łańcuchów znaków możemy obliczyć odległość Hamminga pomiędzy każdymi dwoma problemami. Przykładowo odległość Hamminga pomiędzy P1 i P2 wynosi 4.

P1 = 101**11100**11
P2 = 101**00111**11
DH(P1,P2) = 4

Rezultaty obliczeń odległości Hamminga dla każdej pary problemów z tabeli 3 zamieścimy w tabeli 4. Tabela 4 jest macierzą odległości pomiędzy problemami.

	P1	P14	P11	P2	P0
P1	X	4	4	8	8
P14	4	X	6	4	8
P11	4	6	X	4	4
P2	8	4	4	X	6
P0	8	8	4	6	X

Tabela 4.

Tabela 4 przestawia odległość Hamminga dla każdej pary problemów ze zbioru {P0,P1,P2,P3,P4}.

Przystępując do optymalizacji układu musimy wiedzieć, jaka odległość Hamminga będzie spełniała nasze założcnia, do jakiego stopnia optymalizacji chcemy doprowadzić. To znaczy musimy wiedzieć, czy zadowala nas tylko możliwość rozróżnienia problemów, czy też chcemy mieć możliwość wykrycia ewentualnych błędów dotyczących pomiaru obserwowanych symptomów, czy też chcemy również mieć możliwość korekty błędnych pomiarów.

Przypomnijmy:

DH = 1 - możliwe jest rozróżnienic ciągów, nie jest możliwe wykrycie błędów.
DH = 2 – jest możliwe wykrycie błędu, nie jest możliwa korekcja.
DH = 3 - możliwa jest korekcja błędu.

Odległość Hamminga dla systemu kodowego jest równa najmniejszej odległości Hamminga pomiędzy dwoma problemami tego systemu.

Możemy również określić stopień optymalizacji systemu kodowego oraz pojęcie zoptymalizowanej macierzy systemu kodowego.

Stopniem optymalizacji systemu kodowego będziemy nazywali odległość Hamminga dla systemu kodowego w zoptymalizowanej macierzy systemu kodowego. Stopień optymalizacji systemu kodowego będziemy oznaczali jako DH(Pn).

Zoptymalizowana macierz systemu kodowego, to macierz gdzie każda próba optymalizacji, czyli zmniejszenia odległość pomiędzy dwoma dowolnymi problemami prowadziłaby do zmiany stopnia optymalizacji systemu kodowego.

Załóżmy, że chcemy doprowadzić system kodowy do trzeciego stopnia optymalizacji, to znaczy zachowujemy odległość Hamminga nie mniejszą niż 3 dla każdej pary problemów. DH(Pn) = 3

Trzeci stopień optymalizacji systemu kodowego. DH(Pn) = 3

Zwróćmy uwagę na symptomy S5,S10 i S14. Mają one wartość 1 dla każdego z problemów P1, P2,P3 i P4. W badanym systemie wystąpienie symptomów S5, S10 i S14 nie pozwala nam na rozstrzygnięcie, który z problemów P1, P2, P3 czy P4 miał miejsce. Agregacja S5 i S10 i zastąpienie tej pary przez S5 nie zmieni odległości Hamminga pomiędzy problemami P1, P2, P3 i P4. Zmieni się natomiast odległość Hamminga pomiędzy tymi problemami a P0 o jeden, a więc o wartość dopuszczalną z naszego punktu widzenia. Tym samym minimalna odległość Hamminga dla całego układu zmieni się na 3. Rysunek 4 oraz tabela 5 i 6 przedstawiają nasz układ po agregacji S10.

System kodowy po agregacji symptomu S10.

	S5	S6	S7	S8	S9	S11	S12	S13	S14	S15
P1	1	0	1	1	1	1	0	0	1	1
P2	1	0	1	0	0	1	1	1	1	1
P3	1	0	0	0	1	0	0	0	1	0
P4	1	1	0	0	0	0	1	1	1	0
P0	0	0	0	0	0	0	0	0	0	0

Tabela 5.

Macierz odległości pomiędzy problemami po agregacji symptomu S10.

	P1	P2	P3	P4	P0
P1	x	4	4	8	7
P2	4	x	6	4	7
P3	4	6	x	4	3
P4	8	4	4	x	5
P0	7	7	3	5	x

Tabela 6.

Rys. 4. Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S10.

Należy zwrócić uwagą na odległość Hamminga DH pomiędzy problemem P3 a P0, który wynosi 3, DH=3. Dalsza optymalizacja musi przebiegać w taki sposób, aby ten dystans nie uległ zmniejszeniu. Co skutkuje tym, że dla dalszej optymalizacji nie możemy dokonywać redukcji czy agregacji, na skutek, której zredukowalibyśmy układ o symptomy wskazujące na możliwość wystąpienia problemu P3. W naszym wypadku są to symptomy S5, S9 i S14. W tej sytuacji dobrym kandydatem do agregacji jest symptom S6. Jego wystąpienie zawsze wyzwala wystąpienie S12. Jego agregacja zmieni o 1 odległość pomiędzy P4 a P0, P1, P2 i P3. Odległości pomiędzy P0, P1, P2 i P3 pozostaną bez zmiany. Rezultat tej zmiany przedstawimy w postaci tabeli 7 i 8.

System kodowy po agregacji symptomu S6.

	S5	S7	S8	S9	S11	S12	S13	S14	S15
P1	1	1	1	1	1	0	0	1	1
P2	1	1	0	0	1	1	1	1	1
P3	1	0	0	1	0	0	0	1	0
P4	1	0	0	0	0	1	1	1	0
P0	0	0	0	0	0	0	0	0	0

Tabela 7.

Macierz odległości pomiędzy problemami po agregacji symptomu S6.

	P1	P2	P3	P4	P0
P1	X	4	4	7	7
P2	4	X	6	3	7
P3	4	6	X	3	3
P4	7	3	3	X	4
P0	7	7	3	4	X

Tabela 8.

Po agregacji symptomu S6 widzimy, że odległość DH=3 osiągnęły dodatkowo odległości pomiędzy P4 a P2 i P4 a P3.

Kolejnym kandydatem do agregacji jest S8. Rezultat tej zmiany jest przedstawiony w tabeli 9 i tabeli 10.

System kodowy po agregacji symptomu S8.

	S5	S7	S9	S11	S12	S13	S14	S15
P1	1	1	1	1	0	0	1	1
P2	1	1	0	1	1	1	1	1
P3	1	0	1	0	0	0	1	0
P4	1	0	0	0	1	1	1	0
P0	0	0	0	0	0	0	0	0

Tabela 9.

Macierz odległości pomiędzy problemami po agregacji symptomu S8.

	P1	P2	P3	P4	P0
P1	λ	3	3	6	6
P2	3	λ	6	3	7
P3	3	6	λ	3	3
P4	6	3	3	λ	4
P0	6	7	3	4	λ

Tabela 10.

Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S6 i S8 przedstawiony jest na rysunku 5.
Jest to najbardziej optymalna organizacja przy zachowaniu minimum DH=3. Dalsza agregacja i redukcja symptomów prowadzi nieuchronnie do przekroczenia tej granicy.
W naszym przykładzie dla zachowania minimum DH=3 dla całego układu (pięciu problemów) zachowujemy 8 symptomów.

Drugi stopień optymalizacji systemu kodowego. DH(Pn) = 2

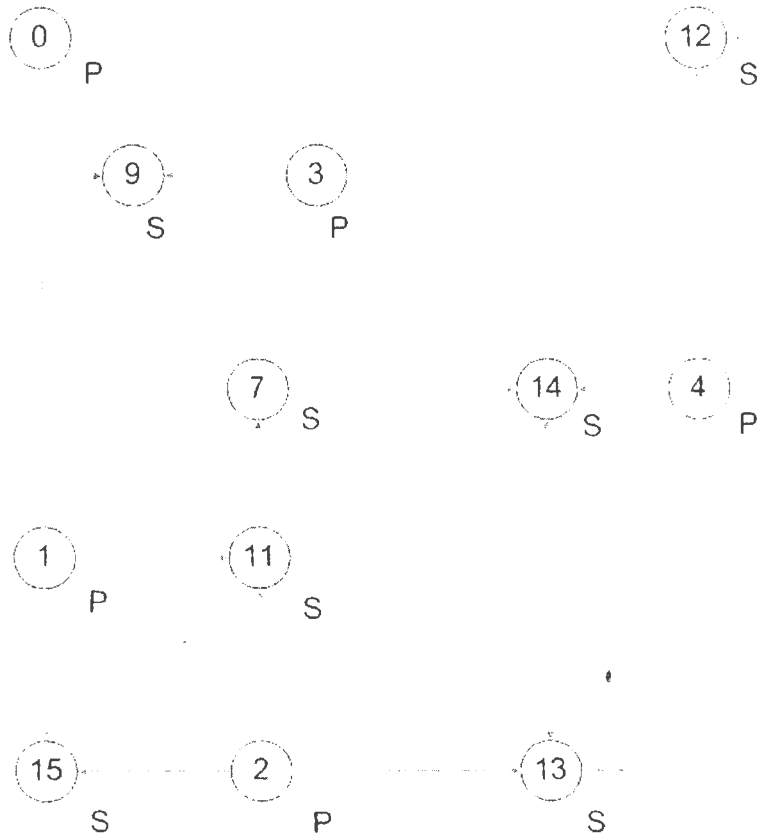
System kodowy w drugim stopniu optymalizacji.

Przeprowadzimy modelowanie dla DH(Pn) = 2, które pozwoli na znaczne zmiany w organizacji grafu. Jako pierwszy do redukcji możemy wytypować symptom S5 lub S14, które nie zmieniają odległości pomiędzy P1, P2, P3 i P4 zmieniają natomiast odległość tych problemów do P0. Ponieważ przyjęliśmy założenie, że DH=2 to możemy zagregować jedynie jeden z tych symptomów. Wykonujemy tę operację dla S5.

System kodowy po agregacji symptomu S5.

	S7	S9	S11	S12	S13	S14	S15
P1	1	1	1	0	0	1	1
P2	1	0	1	1	1	1	1
P3	0	1	0	0	0	1	0
P4	0	0	0	1	1	1	0
P0	0	0	0	0	0	0	0

Tabela 11.

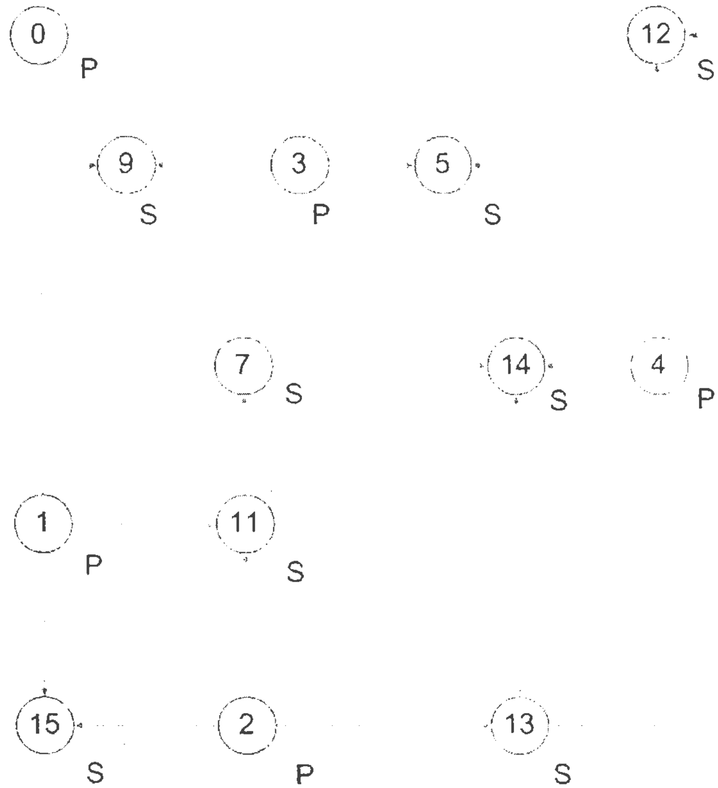


Rys. 5. Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S8 i S6.

Macierz odległości pomiędzy problemami po agregacji symptomu S5.

	P1	P2	P3	P4	P0
P1	λ	3	3	6	5
P2	3	λ	6	3	6
P3	3	6	λ	3	2
P4	6	3	3	λ	3
P0	5	6	2	3	λ

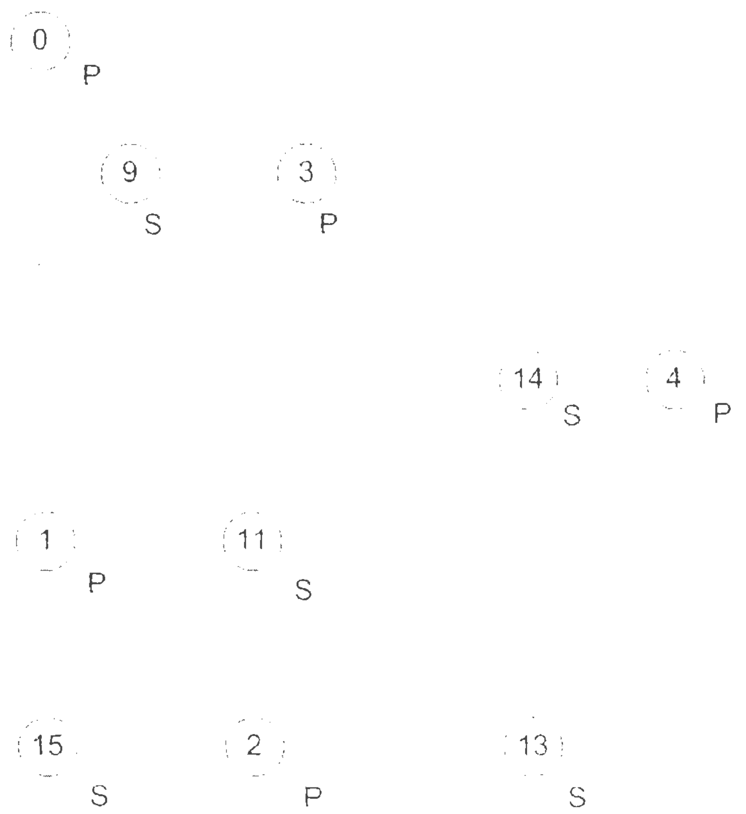
Tabela 12.



Rys. 6. Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S5.

Odległość pomiędzy P0 a P3 wynosi 2. Dalsza optymalizacja musi przebiegać w taki sposób, aby ten dystans nie uległ zmniejszeniu. Skutkuje to tym, że dla dalszej optymalizacji nie możemy dokonywać redukcji czy agregacji, na skutek, której zredukowalibyśmy układ o symptomy wskazujące na możliwość wystąpienia problemu P3. W naszym wypadku są to symptomy S9 i S14.

W kolejnym kroku dokonujemy agregacji symptomu S12, który ma dokładnie taką samą charakterystykę jak S13. Agregacji może również ulec S7, które jest powiązane z S11 na podobnej zasadzie jak S12 i S13. Po agregacji S12 i S7 obraz będzie wyglądał następująco: rysunek 7 i tabele 13 i 14.



Rys. 7 Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S7 i S12.

System kodowy po agregacji symptomu S7 i S12.
Macierz odległości pomiędzy problemami po agregacji symptomu S7 i S12.

	S9	S11	S13	S14	S15
P1	1	1	0	1	1
P2	0	1	1	1	1
P3	1	0	0	1	0
P4	0	0	1	1	0
P0	0	0	0	0	0

Tabela 13.

	P1	P2	P3	P4	P0
P1	x	2	2	4	4
P2	2	x	4	2	4
P3	2	4	x	2	2
P4	4	2	2	x	2
P0	4	4	2	2	x

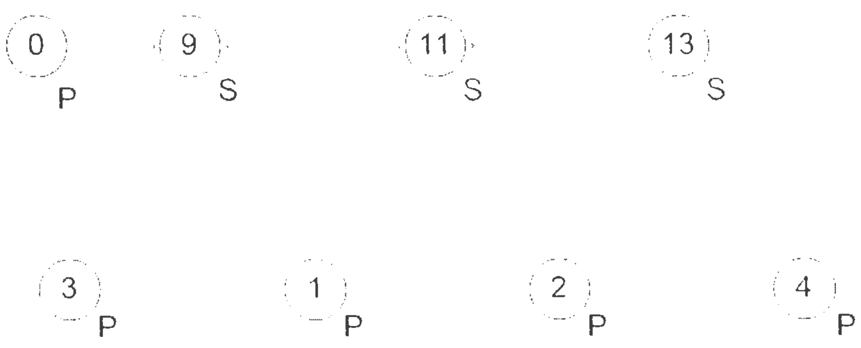
Tabela 14.

Jest to najbardziej optymalna organizacja przy zachowaniu minimum DH=2. W naszym przykładzie dla zachowania minimum DH=2 dla całego układu (pięciu problemów) zachowujemy 5 symptomów.

Przeprowadzimy dalsze uproszczenie i optymalizację decydując się na zmniejszenie odległości Hamminga dla macierzy DH(Pn) = 1.

Pierwszy stopień optymalizacji systemu kodowego. DH(Pn) = 1

Jako pierwszy do redukcji możemy wytypować symptom S14, który nie zmienia odległości pomiędzy P1, P2, P3 i P4 zmienia natomiast odległość tych problemów do P0. Możemy też zredukować S15, który przy DH=1 niesie informację nadmiarową o wystąpieniu P1 lub P2. Tą samą informację niesie S11. Końcowy obraz przedstawia rysunck nr 8, który w uporządkowanej formie przedstawia problemy i symptomy. Tabela nr 15 zawiera zoptymalizowany system kodowy dla DH(PN)=1 a tabela nr 16 przedstawia DH pomiędzy poszczególnymi problemami.



Rys. 8. Graf skierowany acykliczny obrazujący badany system po agregacji symptomu S14 i S15.

System kodowy po agregacji symptomu S14 i S15.

	S9	S11	S13
P1	1	1	0
P2	0	1	1
P3	1	0	0
P4	0	0	1
P0	0	0	0

Tabela 15.

Macierz odległości pomiędzy problemami po agregacji symptomu S14 i S15.

	P1	P2	P3	P4	P0
P1	x	2	1	3	2
P2	2	x	3	1	2
P3	1	3	x	2	1
P4	3	1	2	x	1
P0	2	2	1	1	x

Tabela 16.

Dla zachowania minimum DH(Pn) = 1 dla całego układu (pięciu problemów), w naszym przykładzie zachowujemy 3 symptomy.

Zakończenie

Dokonując optymalizacji systemu kodowego należy zwrócić uwagę na istotną rolę, jaką pełni problem zerowy P0. Jego obecność w macierzy systemu kodowego pozwala na obliczanie odległości Hamminga pomiędzy wszystkimi problemami. Pozwala również na określenie odległości Hamming dla całego systemu kodowego a także stopnia jego optymalizacji - optymalizacji systemu kodowego. Stosując różne stopnie optymalizacji systemu kodowego decydujemy o możliwości detekcji i korekcji błędów.

Niniejsze opracowanie wprowadza trzy definicje przydatne przy optymalizacji systemu kodowego.

Odległość Hamminga dla systemu kodowego jest równa najmniejszej odległości Hamminga pomiędzy dwoma problemami tego systemu.

Stopniem optymalizacji systemu kodowego nazywamy odległość Hamminga dla systemu kodowego w zoptymalizowanej macierzy systemu kodowego. Stopień optymalizacji systemu kodowego oznaczamy jako DH(Pn).

Zoptymalizowana macierz systemu kodowego, to macierz gdzie każda próba optymalizacji, czyli zmniejszenia odległość pomiędzy dwoma dowolnymi problemami prowadzilaby do zmiany stopnia optymalizacji systemu kodowego.

Literatura

[1]. Michael Tiffany(2002): A Survey of Event Correlation Techniques and Related Topics.

[2]. Małgorzata Steinder, Adarshapal S.Sethi: The present and future of event correlation: A need for end-to-end service fault localization .

[3]. Xin Hu (2007): Intelligent Fault Diagnosis in Computer Networks - Kongens Lyngby 2007IMM-THESIS-2007-49.

[4]. Wikipedia.

[5]. S.Yemini, S. Kliger, E. Mozes Y. Yemini, D. Ohsie: High Speed & Robust Event Correlation

ISBN 9788389475220