

Zastosowanie komputerowego wspomaganie wyznaczania rangi ważności parametrów z funkcją zmienna zastępcza oraz warunkowość na rzeczywistym przykładzie ze zmiennymi zależnymi o podobnej ważności

Agnieszka Tiszbierek

1. Wprowadzenie

Współcześnie istnieje wiele złożonych układów automatyki i sterowania, których parametry są od siebie zależne, a ich zależność w znaczącym stopniu wpływa na jakość lub wydajność pracy danego urządzenia. Interakcyjność zmiennych nie jest więc sprawą obojętną w procesie projektowania i konstruowania maszyn, a odpowiednio położony nacisk na „doszlifowanie” konkretnej zmiennej w procesie ma znaczący wpływ na późniejsze efekty pracy tworzonego układu. Dlatego też tak istotny staje się proces ustalania rangi ważności parametrów badanego urządzenia. Jednak i w tym procesie nie wolno zapominać o zależności zmiennych wynikających np. z praw fizyki czy ułożenia elementów maszyny. Czas poprawiania, testowania i udoskonalania urządzenia powinien iść zatem w parze z poprawnym procesem wyznaczania rangi ważności poszczególnych parametrów, nawet jeśli różni się ona minimalnie. By proces ten był skuteczny i poprawny, należy zabezpieczyć nierozzerwalność zależnych zmiennych. Sam proces był złożony i skomplikowany obliczeniowo, zatem powstał pomysł oprogramowania algorytmu i utworzenia programu komputerowego [1]. Jak się okazało, zastosowanie programu znacznie ułatwiło proces obliczeniowy, jednak zależność zmiennych wymusiła wprowadzenie takich funkcjonalności, jak zmienna zastępcza i warunkowość, które pomogły uchronić przed rozdzieleniem zmiennych zależnych.

2. Nowe funkcjonalności programu komputerowego: zmienna zastępcza i warunkowość

Pierwszą z wymienionych nowych funkcjonalności programu jest możliwość utworzenia jednej lub kilku zmiennych zastępczych. Program sam – automatycznie – zakoduje wybrane zmienne w nową zmienną. Po wybraniu i zaznaczeniu odpowiednich parametrów program sam ustala nową wartościowość zmiennej zastępczej oraz przydziela danej wartości kodowej odpowiednio ułożone wiersze z wartościami parametrów wewnętrznych (np. 000 – 0). Zabieg ten nie tylko skraca czas kodowania, ale także zabezpiecza przed pojawieniem się błędów wynikającego z nieuwagi człowieka lub mnogości zmiennych oraz ich znacznej wielowartościowości. Po wczytaniu wartości parametrów oraz ustaleniu składu zmiennej zastępczej można, klikając odpowiedni przycisk, uruchomić proces

Streszczenie: Artykuł opisuje problem złożoności obliczeniowej na rzeczywistym przykładzie ze zmiennymi zależnymi o podobnej ważności, do rozwiązania którego zastosowano opracowany program komputerowy. Program bazuje na decyzyjnej wersji algorytmu Quine’a-McCluskeya minimalizacji wielowartościowych funkcji logicznych i oblicza rangę ważności parametrów konstrukcyjno-eksploatacyjnych badanego układu. Ponadto program posiada możliwość zastosowania w obliczeniach zmiennej zastępczej, a także nałożenia zależności warunkowej. W badanym przypadku zastosowano obie metody, by uprościć proces obliczeniowy oraz otrzymać bardziej przystępne w analizie wyniki. Dane wykorzystane w artykule to wyniki badań występowania wybranych objawów choroby tarczycy.

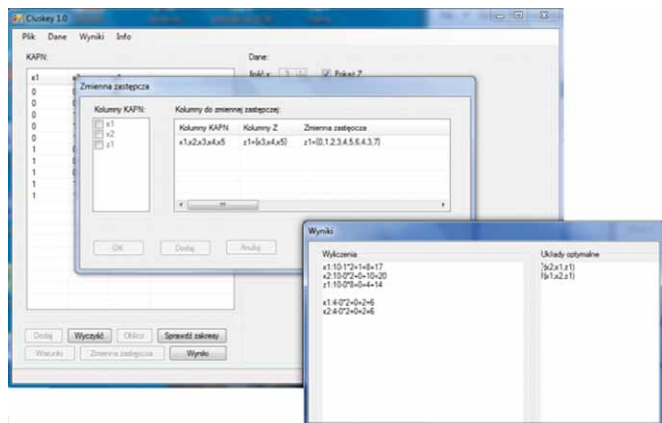
Słowa kluczowe: algorytm Quine’a-McCluskeya minimalizacji indywidualnych funkcji logicznych, ranga ważności zmiennych decyzyjnych, warunkowość zmiennych oraz zmienna zastępcza

APPLICATION OF THE COMPUTATION OF PARAMETER IMPORTANCE RANK WITH FUNCTION SUBSTITUTE VARIABLE AND CONDITIONS TO A REAL EXAMINATION OF DEPENDENT VARIABLES OF RELEVANT VARIABLES

Abstract: Synopsis: The paper describes the calculation complexity using a real life example with dependent variables with similar importance, for solving which was a computer programme developed. The programme is based on the decision version of the Quine’a-McCluskey algorithm for minimizing multivalued logic functions and calculating the importance rank of constructiv-exploativ parameter of the researched system. Furthermore the programme has the ability to use a substitute variable during calculations, and to implement a condition dependency. In the researched instance both methods were implemented as to simplify the calculations and receive more accessible results for analysis. The data used in the example are the results of tests for chosen thyroid disease symptoms.

Key words: algorithm Quine-McCluskey for minimizing individual logic functions, decision variables importance rank, conditionality of variables and substitute variable

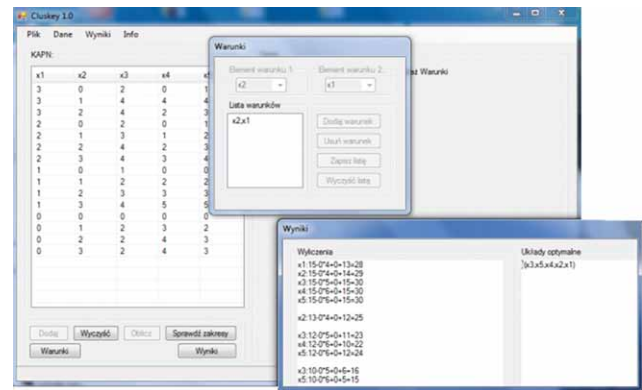
obliczeniowy. Po kliknięciu w przycisk *Oblicz* w kilka sekund otrzymywane jest potwierdzenie pomyślnego zakończenia obliczeń, a po uruchomieniu przycisku *Wyniki* w osobnym oknie pojawiły się odpowiednie dane. Prócz obliczeń pośrednich (etapowych) widnieją na nim także układy optymalne. W programie zmienna zastępcza prócz nazwy „z” otrzymuje kolejny numer (np. z₁), co wynika z możliwości zakodowania większej liczby zmiennych zastępczych. Funkcjonalność *Zmienna zastępcza* posiada możliwość podglądu „wnętrza” zmiennych zastępczych po wykonaniu obliczeń. Wystarczy zaznaczyć kwadracik przy nazwie zmiennej zastępczej, by uruchomić nowe okno, w którym zobrazowany jest nie tylko skład zmiennej zastępczej, ale i sposób jej kodowania, dzięki czemu można skonfrontować wyniki czy też sprawdzić poprawność wybrania parametrów do zakodowania zmiennych zastępczych. Proces obliczeniowy w przypadku tej funkcjonalności bazuje na tym samym algorytmie programu, jak jego wersja podstawowa [1]. Po zakodowaniu zmiennych zastępczych są one traktowane jak „nowe” parametry (zmienne wchodzące w ich skład są kasowane z tabeli pamięci parametrów) i następuje klasyczny proces obliczeniowy z wyświetleniem po zakończeniu ostatecznych wyników [2].



Rys. 1. Widok programu z oknem wynikowym oraz oknem zmiennej zastępczej

Kolejną nową funkcjonalnością jest możliwość narzucenia odpowiednich warunków występowania po sobie parametrów. Funkcjonalność ta jest pod przyciskiem *Warunki*, a po jej włączeniu pojawia się dodatkowe okno z listą parametrów i specjalnym wyborem, za pomocą którego należy zaznaczyć, które parametry i w jakim układzie powinny się pojawić po sobie w układach optymalnych. Również i w tym przypadku występuje możliwość podglądu narzuconych warunków po wykonaniu obliczeń. Algorytm programu, przy funkcji warunków, działa podobnie jak wersja podstawowa programu, z innowacją, którą jest każdorazowe sprawdzenie, czy nowo wyliczone minimum nie wchodzi w skład narzuconego warunku (jest przynajmniej drugim od końca parametrem).

Jeśli tak by się zdarzyło, to należy je odrzucić i znaleźć nowe. Natomiast jeśli minimum okaże się parametr znajdujący się na końcu narzuconego warunku, pozostałe parametry należy ułożyć bez dodatkowych obliczeń w narzuconym ciągu, a dopiero potem przejść do kolejnych etapów obliczeniowych [2].



Rys. 2. Widok programu z oknem wynikowym oraz oknem warunków

3. Problem złożoności obliczeniowej przy zmiennych zależnych o podobnej ważności

Opisanym w artykule problemem jest złożoność obliczeniowa pojawiająca się w procesie wyznaczania rangi ważności parametrów konstrukcyjno-eksploatacyjnych o podobnej ważności. Dodatkowym utrudnieniem jest występująca zależność między parametrami badanego układu. Dobrym przykładem takiej zależności, a także minimalnej różnicy ważności, jest układ objawów występujących przy chorobie tarczycy. Objawy te zostały zebrane w czasie badań i oznaczone odpowiednimi wartościami x. Parametry x są dwuwartościowe (1 oznacza występowanie objawu, a 0 jego brak) i układają się w osiem zmiennych – x₁, x₂, x₃, x₄, x₅, x₆, x₇, x₈ – zapisanych numerycznie w tabeli 1. Wykonano podstawowe obliczenia początkowo za pomocą zbiorów przybliżonych [3], a następnie za pomocą dwuwartościowych decyzyjnych drzew logicznych [4].

Tabela 1. Numeryczny zapis kodowania zmiennych

x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈
0	1	0	0	1	0	0	1
0	0	0	1	1	0	0	1
0	1	0	1	0	1	1	0
1	1	0	1	0	1	1	0
0	1	1	1	0	1	1	0

Obie metody wykazały, że parametry x₁ i x₃ mają najmniejszą rangę ważności w porównaniu do pozostałych parametrów. Wyniki te zostały potwierdzone komputerowymi obliczeniami: x₁: 5 – 1 * 2 + 1 + 3 = 7; x₃: 4 – 0 * 2 + 0 + 3 = 7; x₂, x₄, x₅, x₆, x₇, x₈: 4 – 0 * 2 + 0 + 4 = 8 [5].

Jednak dogłębsza analiza otrzymanych wyników wykazała, że wewnątrz obliczeń etapowych pojawiają się pewne zależności pomiędzy parametrami. Poszczególne wyniki podetapów obliczeniowych dla danych zmiennych różniły się o 1 (w zależności od tego, dla jakiego parametru wykonano obliczenia, a także po jakim parametrze była wykonana redukcja wcześniejszego etapu), jednak analizowany problem zawiera

Tabela 2. Logiczne przekształcenia (znak "–" z lewej strony oznacza iloczyn logiczny, który powstał m.in. z uprzedniego uproszczenia zgodnie z aksjomatem algebry Boole'a)

Logiczne przekształcenia względem x_1									
Etap I					Etap II				
V									
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
t_1	0	1	0	0	1	0	0	1	\underline{V}
t_2	0	0	0	1	1	0	0	1	\underline{V}
t_4	0	1	0	1	0	1	1	0	V
t_3	1	1	0	1	0	1	1	0	V
t_5	0	1	1	1	0	1	1	0	\underline{V}
Logiczne przekształcenia względem x_3									
Etap I					Etap II				
V									
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
t_1	0	1	0	0	1	0	0	1	\underline{V}
t_2	0	0	0	1	1	0	0	1	\underline{V}
t_4	0	1	0	1	0	1	1	0	V
t_3	1	1	0	1	0	1	1	0	\underline{V}
t_5	0	1	1	1	0	1	1	0	V

Tabela 3 a. Etapowe obliczenia dla wybranych układów optymalnych (z układem $\dots x_3 x_1$)

$x1:5-1*2+1+3=7$ $x2:5-0*2+0+5=10$ $x3:5-1*2+1+3=7$ $x4:5-0*2+0+5=10$ $x5:5-0*2+0+5=10$ $x6:5-0*2+0+5=10$ $x7:5-0*2+0+5=10$ $x8:5-0*2+0+5=10$					
$x2:4-0*2+0+4=8$ $x3:4-0*2+0+3=7$ $x4:4-0*2+0+4=8$ $x5:4-0*2+0+4=8$ $x6:4-0*2+0+4=8$ $x7:4-0*2+0+4=8$ $x8:4-0*2+0+4=8$					
$x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$					
$x4:3-0*2+0+2=5$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$	\dots $x2:3-0*2+0+2=5$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x8:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$
$x5:2-0*2+0+2=4$ $x6:2-0*2+0+2=4$ $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x5:2-0*2+0+2=4$ $x6:2-0*2+0+2=4$ $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x7:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$
\dots $x6:2-0*2+0+2=4$ $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x6:2-0*2+0+2=4$ $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x6:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$	\dots $x2:3-0*2+0+3=6$ $x4:3-0*2+0+3=6$ $x5:3-0*2+0+3=6$
\dots $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x7:2-0*2+0+2=4$ $x8:2-0*2+0+2=4$	\dots $x2:3-0*2+0+2=5$ $x6:3-0*2+0+2=5$	\dots $x2:3-0*2+0+2=5$ $x5:3-0*2+0+2=5$	\dots $x2:3-0*2+0+2=5$ $x4:3-0*2+0+2=5$	\dots $x2:3-0*2+0+2=5$ $x4:3-0*2+0+2=5$
$x8x7x6x5x4x2x3x1$	$x7x8x6x5x4x2x3x1$	$x6x2x4x7x8x5x3x1$	$x5x2x4x7x8x6x3x1$	$x4x2x5x6x8x7x3x1$	$x2x4x5x6x7x8x3x1$

Tabela 3 b. Etapowe obliczenia dla wybranych układów optymalnych (z układem x_1x_3)

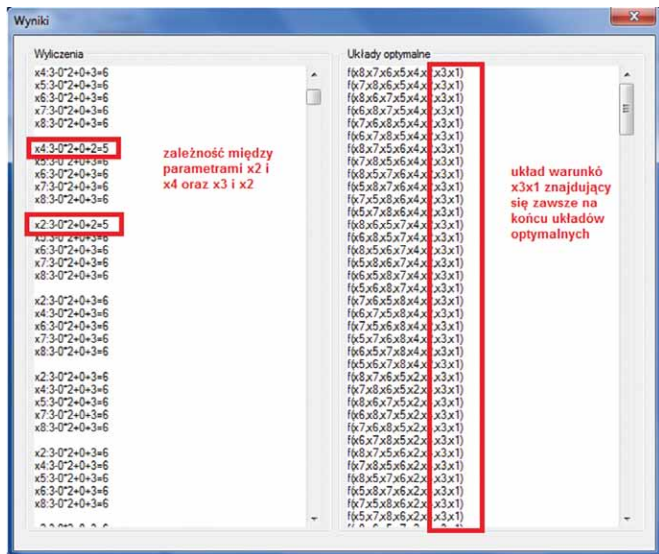
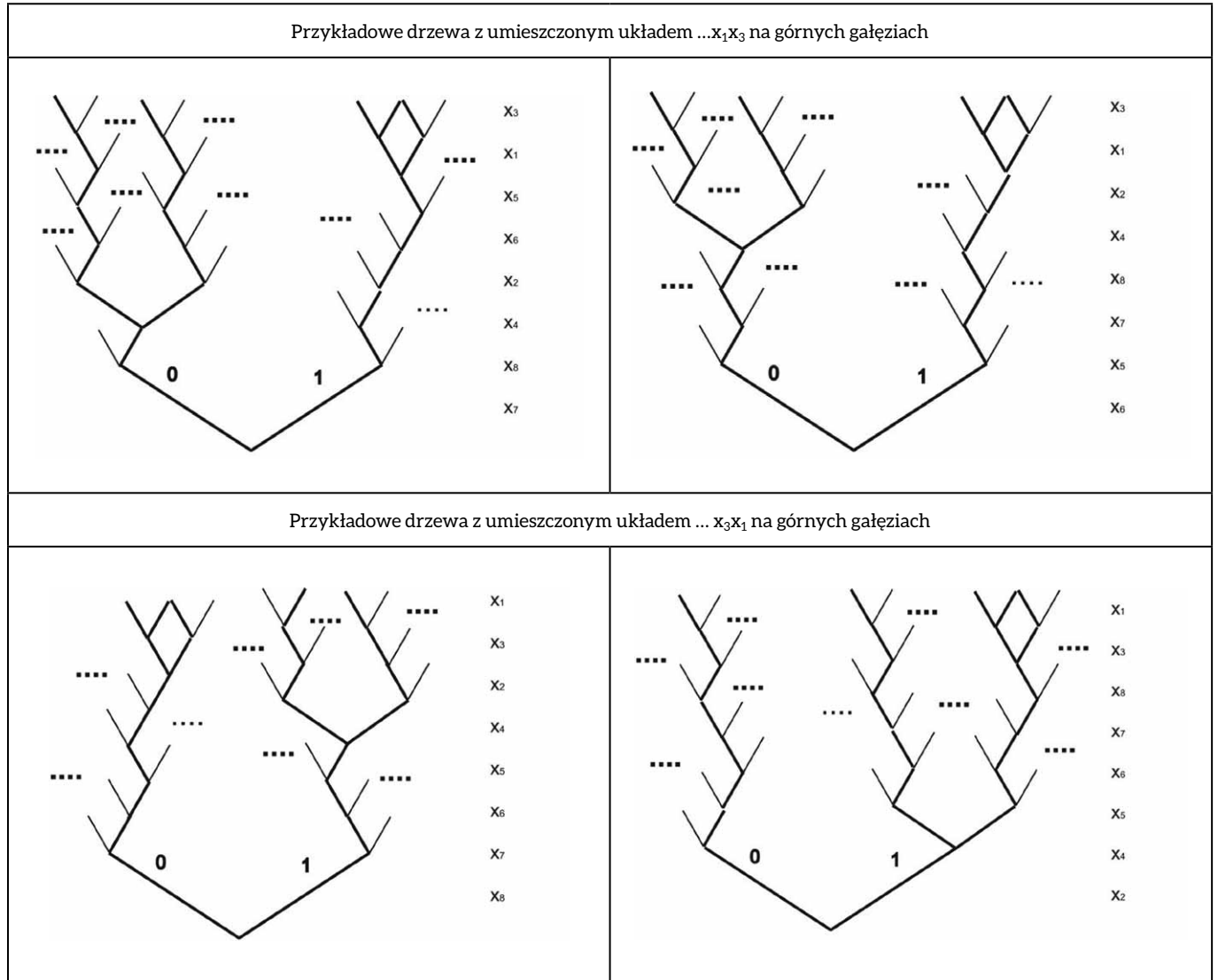
$x_1:5-1^*2+1+3=7$ $x_2:5-0^*2+0+5=10$ $x_3:5-1^*2+1+3=7$ $x_4:5-0^*2+0+5=10$ $x_5:5-0^*2+0+5=10$ $x_6:5-0^*2+0+5=10$ $x_7:5-0^*2+0+5=10$ $x_8:5-0^*2+0+5=10$					
$x_1:4-0^*2+0+3=7$ $x_2:4-0^*2+0+4=8$ $x_4:4-0^*2+0+4=8$ $x_5:4-0^*2+0+4=8$ $x_6:4-0^*2+0+4=8$ $x_7:4-0^*2+0+4=8$ $x_8:4-0^*2+0+4=8$					
$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$					
$x_4:3-0^*2+0+2=5$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+2=5$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_5:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$
$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_7:2-0^*2+0+2=4$ $x_8:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_7:2-0^*2+0+2=4$ $x_8:2-0^*2+0+2=4$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+3=6$ $x_4:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+2=5$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_2:3-0^*2+0+2=5$ $x_5:3-0^*2+0+3=6$ $x_6:3-0^*2+0+3=6$ $x_7:3-0^*2+0+3=6$
$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_7:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_7:2-0^*2+0+2=4$	$x_4:3-0^*2+0+2=5$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_4:3-0^*2+0+2=5$ $x_7:3-0^*2+0+3=6$ $x_8:3-0^*2+0+3=6$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_8:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$ $x_7:2-0^*2+0+2=4$
$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$	$x_7:2-0^*2+0+2=4$ $x_8:2-0^*2+0+2=4$	$x_7:2-0^*2+0+2=4$ $x_8:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$	$x_5:2-0^*2+0+2=4$ $x_6:2-0^*2+0+2=4$
x6x5x7x8x4x2x1x3	x5x6x7x8x2x4x1x3	x7x8x4x2x6x5x1x3	x8x7x4x2x5x6x1x3	x5x6x8x2x4x7x1x3	x6x5x7x2x4x8x1x3

Tabela 4. Tabela z układami optymalnymi dla badanego przykładu bez zastosowania warunków i zmiennych zastępczych

Układy optymalne z ... x_3x_1					
z ... $x_2x_3x_1$	z ... $x_4x_3x_1$	z ... $x_5x_3x_1$	z ... $x_6x_3x_1$	z ... $x_7x_3x_1$	z ... $x_8x_3x_1$
f(x8,x7,x6,x5,x4,x2,x3,x1)	f(x8,x7,x6,x5,x2,x4,x3,x1)	f(x8,x7,x6,x4,x2,x5,x3,x1)	f(x8,x7,x5,x4,x2,x6,x3,x1)	f(x8,x6,x5,x4,x2,x7,x3,x1)	f(x7,x6,x5,x4,x2,x8,x3,x1)
f(x7,x8,x6,x5,x4,x2,x3,x1)	f(x7,x8,x6,x5,x2,x4,x3,x1)	f(x7,x8,x6,x4,x2,x5,x3,x1)	f(x7,x8,x5,x4,x2,x6,x3,x1)	f(x6,x8,x5,x4,x2,x7,x3,x1)	f(x6,x7,x5,x4,x2,x8,x3,x1)
f(x8,x6,x7,x5,x4,x2,x3,x1)	f(x8,x6,x7,x5,x2,x4,x3,x1)	f(x8,x6,x7,x4,x2,x5,x3,x1)	f(x8,x5,x7,x4,x2,x6,x3,x1)	f(x8,x5,x6,x4,x2,x7,x3,x1)	f(x7,x5,x6,x4,x2,x8,x3,x1)
f(x6,x8,x7,x5,x4,x2,x3,x1)	f(x6,x8,x7,x5,x2,x4,x3,x1)	f(x6,x8,x7,x4,x2,x5,x3,x1)	f(x5,x8,x7,x4,x2,x6,x3,x1)	f(x5,x8,x6,x4,x2,x7,x3,x1)	f(x5,x7,x6,x4,x2,x8,x3,x1)
f(x7,x6,x8,x5,x4,x2,x3,x1)	f(x7,x6,x8,x5,x2,x4,x3,x1)	f(x7,x6,x8,x4,x2,x5,x3,x1)	f(x7,x5,x8,x4,x2,x6,x3,x1)	f(x6,x5,x8,x4,x2,x7,x3,x1)	f(x6,x5,x7,x4,x2,x8,x3,x1)
f(x6,x7,x8,x5,x4,x2,x3,x1)	f(x6,x7,x8,x5,x2,x4,x3,x1)	f(x6,x7,x8,x4,x2,x5,x3,x1)	f(x5,x7,x8,x4,x2,x6,x3,x1)	f(x5,x6,x8,x4,x2,x7,x3,x1)	f(x5,x6,x7,x4,x2,x8,x3,x1)
f(x8,x7,x5,x6,x4,x2,x3,x1)	f(x8,x7,x5,x6,x2,x4,x3,x1)	f(x8,x7,x6,x2,x4,x5,x3,x1)	f(x8,x7,x5,x2,x4,x6,x3,x1)	f(x8,x6,x5,x2,x4,x7,x3,x1)	f(x7,x6,x5,x2,x4,x8,x3,x1)
f(x7,x8,x5,x6,x4,x2,x3,x1)	f(x7,x8,x5,x6,x2,x4,x3,x1)	f(x7,x8,x6,x2,x4,x5,x3,x1)	f(x7,x8,x5,x2,x4,x6,x3,x1)	f(x6,x8,x5,x2,x4,x7,x3,x1)	f(x6,x7,x5,x2,x4,x8,x3,x1)
f(x8,x5,x7,x6,x4,x2,x3,x1)	f(x8,x5,x7,x6,x2,x4,x3,x1)	f(x8,x6,x7,x2,x4,x5,x3,x1)	f(x8,x5,x7,x2,x4,x6,x3,x1)	f(x8,x5,x6,x2,x4,x7,x3,x1)	f(x7,x5,x6,x2,x4,x8,x3,x1)
f(x5,x8,x7,x6,x4,x2,x3,x1)	f(x5,x8,x7,x6,x2,x4,x3,x1)	f(x6,x8,x7,x2,x4,x5,x3,x1)	f(x5,x8,x7,x2,x4,x6,x3,x1)	f(x5,x8,x6,x2,x4,x7,x3,x1)	f(x5,x7,x6,x2,x4,x8,x3,x1)
f(x7,x5,x8,x6,x4,x2,x3,x1)	f(x7,x5,x8,x6,x2,x4,x3,x1)	f(x7,x6,x8,x2,x4,x5,x3,x1)	f(x7,x5,x8,x2,x4,x6,x3,x1)	f(x6,x5,x8,x2,x4,x7,x3,x1)	f(x6,x5,x7,x2,x4,x8,x3,x1)
f(x5,x7,x8,x6,x4,x2,x3,x1)	f(x5,x7,x8,x6,x2,x4,x3,x1)	f(x6,x7,x8,x2,x4,x5,x3,x1)	f(x5,x7,x8,x2,x4,x6,x3,x1)	f(x5,x6,x8,x2,x4,x7,x3,x1)	f(x5,x6,x7,x2,x4,x8,x3,x1)
f(x8,x6,x5,x7,x4,x2,x3,x1)	f(x8,x6,x5,x7,x2,x4,x3,x1)	f(x8,x7,x4,x2,x6,x5,x3,x1)	f(x8,x7,x4,x2,x5,x6,x3,x1)	f(x8,x6,x4,x2,x5,x7,x3,x1)	f(x7,x6,x4,x2,x5,x8,x3,x1)
f(x6,x8,x5,x7,x4,x2,x3,x1)	f(x6,x8,x5,x7,x2,x4,x3,x1)	f(x7,x8,x4,x2,x6,x5,x3,x1)	f(x7,x8,x4,x2,x5,x6,x3,x1)	f(x6,x8,x4,x2,x5,x7,x3,x1)	f(x6,x7,x4,x2,x5,x8,x3,x1)
f(x8,x5,x6,x7,x4,x2,x3,x1)	f(x8,x5,x6,x7,x2,x4,x3,x1)	f(x8,x7,x2,x4,x6,x5,x3,x1)	f(x8,x7,x2,x4,x5,x6,x3,x1)	f(x8,x6,x2,x4,x5,x7,x3,x1)	f(x7,x6,x2,x4,x5,x8,x3,x1)
f(x5,x8,x6,x7,x4,x2,x3,x1)	f(x5,x8,x6,x7,x2,x4,x3,x1)	f(x7,x8,x2,x4,x6,x5,x3,x1)	f(x7,x8,x2,x4,x5,x6,x3,x1)	f(x6,x8,x2,x4,x5,x7,x3,x1)	f(x6,x7,x2,x4,x5,x8,x3,x1)
f(x6,x5,x8,x7,x4,x2,x3,x1)	f(x6,x5,x8,x7,x2,x4,x3,x1)	f(x8,x4,x2,x7,x6,x5,x3,x1)	f(x8,x4,x2,x7,x5,x6,x3,x1)	f(x8,x4,x2,x6,x5,x7,x3,x1)	f(x7,x4,x2,x6,x5,x8,x3,x1)

Układy optymalne z ... x3x1 (cd.)					
z ...x2x3x1	z ...x4x3x1	z ...x5x3x1	z ...x6x3x1	z ...x7x3x1	z ...x8x3x1
f(x5,x6,x8,x7,x4,x2,x3,x1)	f(x5,x6,x8,x7,x2,x4,x3,x1)	f(x4,x8,x2,x7,x6,x5,x3,x1)	f(x4,x8,x2,x7,x5,x6,x3,x1)	f(x4,x8,x2,x6,x5,x7,x3,x1)	f(x4,x7,x2,x6,x5,x8,x3,x1)
f(x7,x6,x5,x8,x4,x2,x3,x1)	f(x7,x6,x5,x8,x2,x4,x3,x1)	f(x8,x2,x4,x7,x6,x5,x3,x1)	f(x8,x2,x4,x7,x5,x6,x3,x1)	f(x8,x2,x4,x6,x5,x7,x3,x1)	f(x7,x2,x4,x6,x5,x8,x3,x1)
f(x6,x7,x5,x8,x4,x2,x3,x1)	f(x6,x7,x5,x8,x2,x4,x3,x1)	f(x2,x8,x4,x7,x6,x5,x3,x1)	f(x2,x8,x4,x7,x5,x6,x3,x1)	f(x2,x8,x4,x6,x5,x7,x3,x1)	f(x2,x7,x4,x6,x5,x8,x3,x1)
f(x7,x5,x6,x8,x4,x2,x3,x1)	f(x7,x5,x6,x8,x2,x4,x3,x1)	f(x4,x2,x8,x7,x6,x5,x3,x1)	f(x4,x2,x8,x7,x5,x6,x3,x1)	f(x4,x2,x8,x6,x5,x7,x3,x1)	f(x4,x2,x7,x6,x5,x8,x3,x1)
f(x5,x7,x6,x8,x4,x2,x3,x1)	f(x5,x7,x6,x8,x2,x4,x3,x1)	f(x2,x4,x8,x7,x6,x5,x3,x1)	f(x2,x4,x8,x7,x5,x6,x3,x1)	f(x2,x4,x8,x6,x5,x7,x3,x1)	f(x2,x4,x7,x6,x5,x8,x3,x1)
f(x6,x5,x7,x8,x4,x2,x3,x1)	f(x6,x5,x7,x8,x2,x4,x3,x1)	f(x7,x4,x2,x8,x6,x5,x3,x1)	f(x7,x4,x2,x8,x5,x6,x3,x1)	f(x6,x4,x2,x8,x5,x7,x3,x1)	f(x6,x4,x2,x7,x5,x8,x3,x1)
f(x5,x6,x7,x8,x4,x2,x3,x1)	f(x5,x6,x7,x8,x2,x4,x3,x1)	f(x4,x7,x2,x8,x6,x5,x3,x1)	f(x4,x7,x2,x8,x5,x6,x3,x1)	f(x4,x6,x2,x8,x5,x7,x3,x1)	f(x4,x6,x2,x7,x5,x8,x3,x1)
		f(x7,x2,x4,x8,x6,x5,x3,x1)	f(x7,x2,x4,x8,x5,x6,x3,x1)	f(x6,x2,x4,x8,x5,x7,x3,x1)	f(x6,x2,x4,x7,x5,x8,x3,x1)
		f(x2,x7,x4,x8,x6,x5,x3,x1)	f(x2,x7,x4,x8,x5,x6,x3,x1)	f(x2,x6,x4,x8,x5,x7,x3,x1)	f(x2,x6,x4,x7,x5,x8,x3,x1)
		f(x4,x2,x7,x8,x6,x5,x3,x1)	f(x4,x2,x7,x8,x5,x6,x3,x1)	f(x4,x2,x6,x8,x5,x7,x3,x1)	f(x4,x2,x6,x7,x5,x8,x3,x1)
		f(x2,x4,x7,x8,x6,x5,x3,x1)	f(x2,x4,x7,x8,x5,x6,x3,x1)	f(x2,x4,x6,x8,x5,x7,x3,x1)	f(x2,x4,x6,x7,x5,x8,x3,x1)
		f(x8,x6,x4,x2,x7,x5,x3,x1)	f(x8,x5,x4,x2,x7,x6,x3,x1)	f(x8,x5,x4,x2,x6,x7,x3,x1)	f(x7,x5,x4,x2,x6,x8,x3,x1)
		f(x6,x8,x4,x2,x7,x5,x3,x1)	f(x5,x8,x4,x2,x7,x6,x3,x1)	f(x5,x8,x4,x2,x6,x7,x3,x1)	f(x5,x7,x4,x2,x6,x8,x3,x1)
		f(x8,x6,x2,x4,x7,x5,x3,x1)	f(x8,x5,x2,x4,x7,x6,x3,x1)	f(x8,x5,x2,x4,x6,x7,x3,x1)	f(x7,x5,x2,x4,x6,x8,x3,x1)
		f(x6,x8,x2,x4,x7,x5,x3,x1)	f(x5,x8,x2,x4,x7,x6,x3,x1)	f(x5,x8,x2,x4,x6,x7,x3,x1)	f(x5,x7,x2,x4,x6,x8,x3,x1)
		f(x8,x4,x2,x6,x7,x5,x3,x1)	f(x8,x4,x2,x5,x7,x6,x3,x1)	f(x8,x4,x2,x5,x6,x7,x3,x1)	f(x7,x4,x2,x5,x6,x8,x3,x1)
		f(x4,x8,x2,x6,x7,x5,x3,x1)	f(x4,x8,x2,x5,x7,x6,x3,x1)	f(x4,x8,x2,x5,x6,x7,x3,x1)	f(x4,x7,x2,x5,x6,x8,x3,x1)
		f(x8,x2,x4,x6,x7,x5,x3,x1)	f(x8,x2,x4,x5,x7,x6,x3,x1)	f(x8,x2,x4,x5,x6,x7,x3,x1)	f(x7,x2,x4,x5,x6,x8,x3,x1)
		f(x2,x8,x4,x6,x7,x5,x3,x1)	f(x2,x8,x4,x5,x7,x6,x3,x1)	f(x2,x8,x4,x5,x6,x7,x3,x1)	f(x2,x7,x4,x5,x6,x8,x3,x1)
		f(x4,x2,x8,x6,x7,x5,x3,x1)	f(x4,x2,x8,x5,x7,x6,x3,x1)	f(x4,x2,x8,x5,x6,x7,x3,x1)	f(x4,x2,x7,x5,x6,x8,x3,x1)
		f(x2,x4,x8,x6,x7,x5,x3,x1)	f(x2,x4,x8,x5,x7,x6,x3,x1)	f(x2,x4,x8,x5,x6,x7,x3,x1)	f(x2,x4,x7,x5,x6,x8,x3,x1)
		f(x6,x4,x2,x8,x7,x5,x3,x1)	f(x5,x4,x2,x8,x7,x6,x3,x1)	f(x5,x4,x2,x8,x6,x7,x3,x1)	f(x5,x4,x2,x7,x6,x8,x3,x1)
		f(x4,x6,x2,x8,x7,x5,x3,x1)	f(x4,x5,x2,x8,x7,x6,x3,x1)	f(x4,x5,x2,x8,x6,x7,x3,x1)	f(x4,x5,x2,x7,x6,x8,x3,x1)
		f(x6,x2,x4,x8,x7,x5,x3,x1)	f(x5,x2,x4,x8,x7,x6,x3,x1)	f(x5,x2,x4,x8,x6,x7,x3,x1)	f(x5,x2,x4,x7,x6,x8,x3,x1)
		f(x2,x6,x4,x8,x7,x5,x3,x1)	f(x2,x5,x4,x8,x7,x6,x3,x1)	f(x2,x5,x4,x8,x6,x7,x3,x1)	f(x2,x5,x4,x7,x6,x8,x3,x1)
		f(x4,x2,x6,x8,x7,x5,x3,x1)	f(x4,x2,x5,x8,x7,x6,x3,x1)	f(x4,x2,x5,x8,x6,x7,x3,x1)	f(x4,x2,x5,x7,x6,x8,x3,x1)
		f(x2,x4,x6,x8,x7,x5,x3,x1)	f(x2,x4,x5,x8,x7,x6,x3,x1)	f(x2,x4,x5,x8,x6,x7,x3,x1)	f(x2,x4,x5,x7,x6,x8,x3,x1)
		f(x7,x6,x4,x2,x8,x5,x3,x1)	f(x7,x5,x4,x2,x8,x6,x3,x1)	f(x6,x5,x4,x2,x8,x7,x3,x1)	f(x6,x5,x4,x2,x7,x8,x3,x1)
		f(x6,x7,x4,x2,x8,x5,x3,x1)	f(x5,x7,x4,x2,x8,x6,x3,x1)	f(x5,x6,x4,x2,x8,x7,x3,x1)	f(x5,x6,x4,x2,x7,x8,x3,x1)
		f(x7,x4,x2,x6,x8,x5,x3,x1)	f(x7,x4,x2,x5,x8,x6,x3,x1)	f(x6,x4,x2,x5,x8,x7,x3,x1)	f(x6,x4,x2,x5,x7,x8,x3,x1)
		f(x4,x7,x2,x6,x8,x5,x3,x1)	f(x4,x7,x2,x5,x8,x6,x3,x1)	f(x4,x6,x2,x5,x8,x7,x3,x1)	f(x4,x6,x2,x5,x7,x8,x3,x1)
		f(x7,x2,x4,x6,x8,x5,x3,x1)	f(x7,x2,x4,x5,x8,x6,x3,x1)	f(x6,x2,x4,x5,x8,x7,x3,x1)	f(x6,x2,x4,x5,x7,x8,x3,x1)
		f(x2,x7,x4,x6,x8,x5,x3,x1)	f(x2,x7,x4,x5,x8,x6,x3,x1)	f(x2,x6,x4,x5,x8,x7,x3,x1)	f(x2,x6,x4,x5,x7,x8,x3,x1)
		f(x4,x2,x7,x6,x8,x5,x3,x1)	f(x4,x2,x7,x5,x8,x6,x3,x1)	f(x4,x2,x6,x5,x8,x7,x3,x1)	f(x4,x2,x6,x5,x7,x8,x3,x1)
		f(x2,x4,x7,x6,x8,x5,x3,x1)	f(x2,x4,x7,x5,x8,x6,x3,x1)	f(x2,x4,x6,x5,x8,x7,x3,x1)	f(x2,x4,x6,x5,x7,x8,x3,x1)
		f(x6,x4,x2,x7,x8,x5,x3,x1)	f(x5,x4,x2,x7,x8,x6,x3,x1)	f(x5,x4,x2,x6,x8,x7,x3,x1)	f(x5,x4,x2,x6,x7,x8,x3,x1)
		f(x4,x6,x2,x7,x8,x5,x3,x1)	f(x4,x5,x2,x7,x8,x6,x3,x1)	f(x4,x5,x2,x6,x8,x7,x3,x1)	f(x4,x5,x2,x6,x7,x8,x3,x1)
		f(x6,x2,x4,x7,x8,x5,x3,x1)	f(x5,x2,x4,x7,x8,x6,x3,x1)	f(x5,x2,x4,x6,x8,x7,x3,x1)	f(x5,x2,x4,x6,x7,x8,x3,x1)
		f(x2,x6,x4,x7,x8,x5,x3,x1)	f(x2,x5,x4,x7,x8,x6,x3,x1)	f(x2,x5,x4,x6,x8,x7,x3,x1)	f(x2,x5,x4,x6,x7,x8,x3,x1)
		f(x4,x2,x6,x7,x8,x5,x3,x1)	f(x4,x2,x5,x7,x8,x6,x3,x1)	f(x4,x2,x5,x6,x8,x7,x3,x1)	f(x4,x2,x5,x6,x7,x8,x3,x1)
		f(x2,x4,x6,x7,x8,x5,x3,x1)	f(x2,x4,x5,x7,x8,x6,x3,x1)	f(x2,x4,x5,x6,x8,x7,x3,x1)	f(x2,x4,x5,x6,x7,x8,x3,x1)
Układy optymalne z ... x1x3					
z ...x2x1x3	z ...x4x1x3	z ...x5x1x3	z ...x6x1x3	z ...x7x1x3	z ...x8x1x3
f(x8,x7,x6,x5,x4,x2,x1,x3)	f(x8,x7,x6,x5,x2,x4,x1,x3)	f(x8,x7,x6,x4,x2,x5,x1,x3)	f(x8,x7,x5,x4,x2,x6,x1,x3)	f(x8,x6,x5,x4,x2,x7,x1,x3)	f(x7,x6,x5,x4,x2,x8,x1,x3)
f(x7,x8,x6,x5,x4,x2,x1,x3)	f(x7,x8,x6,x5,x2,x4,x1,x3)	f(x7,x8,x6,x4,x2,x5,x1,x3)	f(x7,x8,x5,x4,x2,x6,x1,x3)	f(x6,x8,x5,x4,x2,x7,x1,x3)	f(x6,x7,x5,x4,x2,x8,x1,x3)
f(x8,x6,x7,x5,x4,x2,x1,x3)	f(x8,x6,x7,x5,x2,x4,x1,x3)	f(x8,x6,x7,x4,x2,x5,x1,x3)	f(x8,x5,x7,x4,x2,x6,x1,x3)	f(x8,x5,x6,x4,x2,x7,x1,x3)	f(x7,x5,x6,x4,x2,x8,x1,x3)
f(x6,x8,x7,x5,x4,x2,x1,x3)	f(x6,x8,x7,x5,x2,x4,x1,x3)	f(x6,x8,x7,x4,x2,x5,x1,x3)	f(x5,x8,x7,x4,x2,x6,x1,x3)	f(x5,x8,x6,x4,x2,x7,x1,x3)	f(x5,x7,x6,x4,x2,x8,x1,x3)
f(x7,x6,x8,x5,x4,x2,x1,x3)	f(x7,x6,x8,x5,x2,x4,x1,x3)	f(x7,x6,x8,x4,x2,x5,x1,x3)	f(x7,x5,x8,x4,x2,x6,x1,x3)	f(x6,x5,x8,x4,x2,x7,x1,x3)	f(x6,x5,x7,x4,x2,x8,x1,x3)
f(x6,x7,x8,x5,x4,x2,x1,x3)	f(x6,x7,x8,x5,x2,x4,x1,x3)	f(x6,x7,x8,x4,x2,x5,x1,x3)	f(x5,x7,x8,x4,x2,x6,x1,x3)	f(x5,x6,x8,x4,x2,x7,x1,x3)	f(x5,x6,x7,x4,x2,x8,x1,x3)
f(x8,x7,x5,x6,x4,x2,x1,x3)	f(x8,x7,x5,x6,x2,x4,x1,x3)	f(x8,x7,x6,x2,x4,x5,x1,x3)	f(x8,x7,x5,x2,x4,x6,x1,x3)	f(x8,x6,x5,x2,x4,x7,x1,x3)	f(x7,x6,x5,x2,x4,x8,x1,x3)
f(x7,x8,x5,x6,x4,x2,x1,x3)	f(x7,x8,x5,x6,x2,x4,x1,x3)	f(x7,x8,x6,x2,x4,x5,x1,x3)	f(x7,x8,x5,x2,x4,x6,x1,x3)	f(x6,x8,x5,x2,x4,x7,x1,x3)	f(x6,x7,x5,x2,x4,x8,x1,x3)

Tabela 5. Tabela z przykładowymi układami z tabeli 4 przedstawionymi na skróconych drzewach logicznych dla badanego przykładu



Rys. 3. Okno programu z prezentacją wyników, zarówno obliczeń etapów pośrednich, jak i układów optymalnych, wraz z zaznaczeniami zależności oraz układami warunkowymi

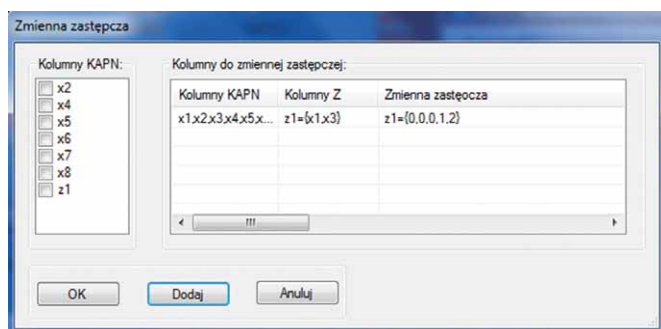
parametry, których nawet niewielka różnica wartości liczbowych może mieć ogromne znaczenie. By dokonać głębszej analizy, pierwotnie nałożono warunek: x_3x_1 , a potem wykonano obliczenia. Pozwoliło to na zauważenie zachodzących zależności w dalszych podetapach obliczeniowych, a także redukcję 576 układów optymalnych do połowy.

Oczywiście analiza rozrysowanych poszczególnych drzew logicznych, będąca wyczerpaniem gałęzi prawdziwych (pogrubionych na drzewach) wykaże, że wśród układów optymalnych wyczerpanych przez program, istnieją różnice w ilości tych gałęzi. Przykładowo układ $x_6x_5x_7x_2x_4x_8x_1x_3$ (drzewo z prawej górnej strony w tabeli 5) posiada 21 nieredukowalnych gałęzi prawdziwych, gdy układ $x_7x_8x_4x_2x_6x_5x_1x_3$ (drzewo z lewej górnej strony tabeli 5.) ma ich już 23. Oczywiście do pewnego poziomu drzewa oba układy są na identycznym poziomie – posiadając tyle samo, bo 13 gałęzi prawdziwych (czwarte piętro drzewa, licząc od góry). Zatem wybór x_7 po x_8 w pierwszym układzie był korzystny (na tym poziomie drzewa istnieje 15 gałęzi prawdziwych), natomiast wybór x_2 po x_6 już niekoniecznie (16 gałęzi prawdziwych). Możliwe, że zastosowanie w drugim układzie

innej zmiennej z pozostałych parametrów po x_6 pozwoliłoby uzyskać tę samą ilość gałęzi prawdziwych na obu rozrysowanych drzewach. Metoda sprawdzania nieredukowalnych gałęzi prawdziwych na wielowartościowych drzewach logicznych pozwala na znalezienie wśród korzystnych układów parametrów tych najlepszych, nawet jeśli różnice między nimi są minimalne, tak jak w analizowanym przykładzie. Dlatego też kolejną wartą rozważenia funkcjonalnością programu może być moduł liczący ilość gałęzi prawdziwych wielowartościowych, logicznych drzew decyzyjnych, na których ułożone są wybrane przez program optymalne układy. Moduł ten pomógłby zawęzić grono korzystnych układów parametrów do tych najlepszych.

4. Zastosowanie zmiennej zastępczej oraz nałożenie warunków na zmienne interakcyjne

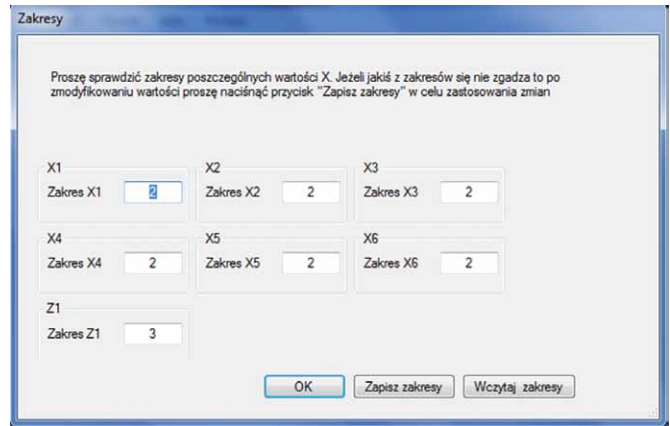
Po dokładniejszej analizie podetapów obliczeń pośrednich zauważono zależność między x_2 oraz x_4 . W każdym podetapie zredukowanym po x_2 wśród wszystkich parametrów parametr x_4 zawsze posiadał najmniejszą wartość (różną od pozostałych o 1), natomiast w podetapach zredukowanych po x_4 sytuacja się powtarzała. Tam minimum różniącym się od pozostałych parametrów o wartość 1 było zawsze x_2 . Założono zatem, że parametry x_2 i x_4 łączy jakaś zależność warunkowa. Założenie to było zaczątkiem pomysłu zastosowania na badanym przykładzie zarówno zmiennej zastępczej, jak i warunku. Zmienna zastępcza łączyłaby w sobie parametry najmniej ważne, czyli x_1 oraz x_3 , co zapewniłoby odpowiednie ułożenie parametrów o najniższej randze ważności, równocześnie nie zaciemniając układów pozostałych parametrów. Natomiast parametry x_2 oraz x_4 połączono by warunkiem, który nie pozwoliłby na ich rozdzielenie, mając na uwadze ich zależność względem siebie.



Rys. 4. Zakodowanie zmiennej zastępczej z_1 składającej się z parametrów x_1 oraz x_3

Zakodowanie zmiennej zastępczej Z_1 (taką bowiem nazwę otrzymał nowopowstały parametr), w skład której weszły x_1 oraz x_3 , zostało wykonane automatycznie przez program. Nowa zmienna stała się trójwartościowa, bowiem w zbiorze numerycznie zapisanych układów pojawiły się trzy różne układy występowania wspomnianych parametrów: 0..0.. (trzy razy), 1..0.., 0..1... Każdy z nich otrzymał kolejną wartość tj. 0, 1, 2, stąd wartościowość nowej zmiennej.

Zestawienie kolejnych wartości dla każdego układu wartości parametrów nowej zmiennej zostało przedstawione w tabeli 6, a samo pojawienie się z_1 zredukowało ilość parametrów do

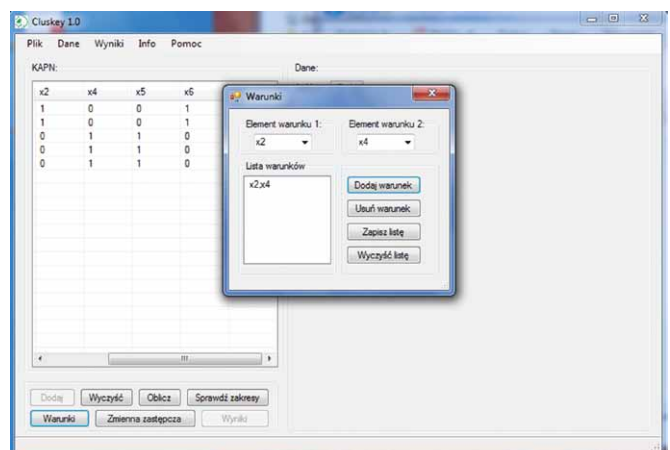


Rys. 5. Zakresy nowych zmiennych biorących udział w procesie obliczeniowym

Tabela 6. Ostateczny zapis kodowania zmiennych

Nowa numeracja	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Stara numeracja	x_2	x_4	x_5	x_6	x_7	x_8	$z(x_1x_3)$
	1	0	1	0	0	1	0
	0	1	1	0	0	1	0
	1	1	0	1	1	0	0
	1	1	0	1	1	0	1
	1	1	0	1	1	0	2

siedmiu. Nowy zestaw parametrów został zapisany do pliku .txt, z którego można było pobrać go przy kolejnym etapie obliczeń. W następnym kroku na nowy zestaw parametrów nałożono warunek, który zapewnia łączne występowanie koło siebie parametrów x_2x_4 (stare nazewnictwo), których zależność została wcześniej zauważona. Warunek zapewniał, że parametry nie zostaną rozdzielone i w każdym optymalnym rozwiązaniu będą ustawione koło siebie w kombinacji : x_2x_4 .



Rys. 6. Utworzenie warunku występowania parametrów: x_2x_4

Tabela 7. Tabela z korzystnym sposobem układu parametrów przy zastosowaniu warunku i zmiennej zastępczej

Układy optymalne				
f(x6,x5,x4,x3,x1,x2,x7)	f(x6,x5,x4,x1,x2,x3,x7)	f(x6,x5,x3,x1,x2,x4,x7)	f(x6,x4,x3,x1,x2,x5,x7)	f(x5,x4,x3,x1,x2,x6,x7)
f(x5,x6,x4,x3,x1,x2,x7)	f(x5,x6,x4,x1,x2,x3,x7)	f(x5,x6,x3,x1,x2,x4,x7)	f(x4,x6,x3,x1,x2,x5,x7)	f(x4,x5,x3,x1,x2,x6,x7)
f(x6,x4,x5,x3,x1,x2,x7)	f(x6,x4,x5,x1,x2,x3,x7)	f(x6,x3,x5,x1,x2,x4,x7)	f(x6,x3,x4,x1,x2,x5,x7)	f(x5,x3,x4,x1,x2,x6,x7)
f(x4,x6,x5,x3,x1,x2,x7)	f(x4,x6,x5,x1,x2,x3,x7)	f(x3,x6,x5,x1,x2,x4,x7)	f(x3,x6,x4,x1,x2,x5,x7)	f(x3,x5,x4,x1,x2,x6,x7)
f(x5,x4,x6,x3,x1,x2,x7)	f(x5,x4,x6,x1,x2,x3,x7)	f(x5,x3,x6,x1,x2,x4,x7)	f(x4,x3,x6,x1,x2,x5,x7)	f(x4,x3,x5,x1,x2,x6,x7)
f(x4,x5,x6,x3,x1,x2,x7)	f(x4,x5,x6,x1,x2,x3,x7)	f(x3,x5,x6,x1,x2,x4,x7)	f(x3,x4,x6,x1,x2,x5,x7)	f(x3,x4,x5,x1,x2,x6,x7)
f(x6,x5,x3,x4,x1,x2,x7)	f(x6,x5,x1,x2,x4,x3,x7)	f(x6,x5,x1,x2,x3,x4,x7)	f(x6,x4,x1,x2,x3,x5,x7)	f(x5,x4,x1,x2,x3,x6,x7)
f(x5,x6,x3,x4,x1,x2,x7)	f(x5,x6,x1,x2,x4,x3,x7)	f(x5,x6,x1,x2,x3,x4,x7)	f(x4,x6,x1,x2,x3,x5,x7)	f(x4,x5,x1,x2,x3,x6,x7)
f(x6,x3,x5,x4,x1,x2,x7)	f(x6,x1,x2,x5,x4,x3,x7)	f(x6,x1,x2,x5,x3,x4,x7)	f(x6,x1,x2,x4,x3,x5,x7)	f(x5,x1,x2,x4,x3,x6,x7)
f(x3,x6,x5,x4,x1,x2,x7)	f(x1,x2,x6,x5,x4,x3,x7)	f(x1,x2,x6,x5,x3,x4,x7)	f(x1,x2,x6,x4,x3,x5,x7)	f(x1,x2,x5,x4,x3,x6,x7)
f(x5,x3,x6,x4,x1,x2,x7)	f(x5,x1,x2,x6,x4,x3,x7)	f(x5,x1,x2,x6,x3,x4,x7)	f(x4,x1,x2,x6,x3,x5,x7)	f(x4,x1,x2,x5,x3,x6,x7)
f(x3,x5,x6,x4,x1,x2,x7)	f(x1,x2,x5,x6,x4,x3,x7)	f(x1,x2,x5,x6,x3,x4,x7)	f(x1,x2,x4,x6,x3,x5,x7)	f(x1,x2,x4,x5,x3,x6,x7)
f(x6,x4,x3,x5,x1,x2,x7)	f(x6,x4,x1,x2,x5,x3,x7)	f(x6,x3,x1,x2,x5,x4,x7)	f(x6,x3,x1,x2,x4,x5,x7)	f(x5,x3,x1,x2,x4,x6,x7)
f(x4,x6,x3,x5,x1,x2,x7)	f(x4,x6,x1,x2,x5,x3,x7)	f(x3,x6,x1,x2,x5,x4,x7)	f(x3,x6,x1,x2,x4,x5,x7)	f(x3,x5,x1,x2,x4,x6,x7)
f(x6,x3,x4,x5,x1,x2,x7)	f(x6,x1,x2,x4,x5,x3,x7)	f(x6,x1,x2,x3,x5,x4,x7)	f(x6,x1,x2,x3,x4,x5,x7)	f(x5,x1,x2,x3,x4,x6,x7)
f(x3,x6,x4,x5,x1,x2,x7)	f(x1,x2,x6,x4,x5,x3,x7)	f(x1,x2,x6,x3,x5,x4,x7)	f(x1,x2,x6,x3,x4,x5,x7)	f(x1,x2,x5,x3,x4,x6,x7)
f(x4,x3,x6,x5,x1,x2,x7)	f(x4,x1,x2,x6,x5,x3,x7)	f(x3,x1,x2,x6,x5,x4,x7)	f(x3,x1,x2,x6,x4,x5,x7)	f(x3,x1,x2,x5,x4,x6,x7)
f(x3,x4,x6,x5,x1,x2,x7)	f(x1,x2,x4,x6,x5,x3,x7)	f(x1,x2,x3,x6,x5,x4,x7)	f(x1,x2,x3,x6,x4,x5,x7)	f(x1,x2,x3,x5,x4,x6,x7)
f(x5,x4,x3,x6,x1,x2,x7)	f(x5,x4,x1,x2,x6,x3,x7)	f(x5,x3,x1,x2,x6,x4,x7)	f(x4,x3,x1,x2,x6,x5,x7)	f(x4,x3,x1,x2,x5,x6,x7)
f(x4,x5,x3,x6,x1,x2,x7)	f(x4,x5,x1,x2,x6,x3,x7)	f(x3,x5,x1,x2,x6,x4,x7)	f(x3,x4,x1,x2,x6,x5,x7)	f(x3,x4,x1,x2,x5,x6,x7)
f(x5,x3,x4,x6,x1,x2,x7)	f(x5,x1,x2,x4,x6,x3,x7)	f(x5,x1,x2,x3,x6,x4,x7)	f(x4,x1,x2,x3,x6,x5,x7)	f(x4,x1,x2,x3,x5,x6,x7)
f(x3,x5,x4,x6,x1,x2,x7)	f(x1,x2,x5,x4,x6,x3,x7)	f(x1,x2,x5,x3,x6,x4,x7)	f(x1,x2,x4,x3,x6,x5,x7)	f(x1,x2,x4,x3,x5,x6,x7)
f(x4,x3,x5,x6,x1,x2,x7)	f(x4,x1,x2,x5,x6,x3,x7)	f(x3,x1,x2,x5,x6,x4,x7)	f(x3,x1,x2,x4,x6,x5,x7)	f(x3,x1,x2,x4,x5,x6,x7)
f(x3,x4,x5,x6,x1,x2,x7)	f(x1,x2,x4,x5,x6,x3,x7)	f(x1,x2,x3,x5,x6,x4,x7)	f(x1,x2,x3,x4,x6,x5,x7)	f(x1,x2,x3,x4,x5,x6,x7)

Dzięki zastosowaniu obu zabiegów dokonano redukcji czasu procesu obliczeniowego, a także ilości układów optymalnych do 120. Tak zredukowana liczba układów optymalnych pozwala w prostszy sposób dokonać dalszej analizy, skupiając się jedynie na pozostałych zmiennych (x_5, x_6, x_7, x_8).

Jak można zauważyć w tabeli 7, pozostałe parametry posiadają tę samą rangę ważności, gdyż w każdym etapie (redukowanym po danym x) pozostałe zmienne układają się w układach optymalnych na kolejnej pozycji za zredukowaną zmienną, czyli można założyć, że przyjmują tę samą wartość w obliczeniach podetapowych. Taka wiedza z kolei pozwala założyć, że pozostałe parametry posiadają względem siebie tę samą rangę ważności. Można zatem zaryzykować stwierdzenie, że są one równoważne i niezależne.

Wnioski

Przedstawiony w powyższych rozdziałach przykład zastosowania programu komputerowego (wraz z jego dodatkowymi funkcjonalnościami) w procesie wyznaczania rangi ważności pokazuje, jak przydatnym narzędziem jest program. Złożony i czasochłonny proces, pierwotnie pochłaniający czas i uwagę użytkownika, trwa obecnie od kilku do kilkudziesięciu sekund, nie absorbuje uwagi użytkownika i eliminuje

wiele potencjalnych ludzkich błędów. Dzięki pomysłowi zapisu otrzymanych wyników i wyliczeń pośrednich do pliku rezultaty pracy programu można w prosty sposób wykorzystać w dalszych badaniach i analizach.

Jak zostało już wspomniane, współczesne urządzenia posiadają interakcyjne parametry, o których nie wolno zapominać, gdyż powodują one znaczny wpływ danych zmiennych na siebie, a co za tym idzie – na jakość pracy maszyny. Dlatego też dodatkowe funkcjonalności programu (możliwość narzucenia warunkowości układu parametrów oraz utworzenie zmiennych zastępczych) są bardzo wartościowymi elementami przedstawionego programu. Nie tylko ograniczają czas obliczeniowy, dbają o nierozzerwalność zależnych parametrów, ale także ograniczają ilość układów optymalnych, usuwając te, które rozdzielają zmienne interakcyjne.


Opisany przykład pokazuje, że program może być wykorzystywany także w innych dziedzinach życia, w których potrzebna jest solidna wiedza na temat ważności poszczególnych cech, danych, parametrów, opisów lub zmiennych. Wystarczy, że istnieje możliwość odpowiedniego zakodowania danych zebranych w czasie badań czy pomiarów. Jak zostało przedstawione, algorytm sprawdza się nawet w przypadku niewielkiej różnicy wagi poszczególnych rang ważności.

Algorytm logiczny wielowartościowy, będący podstawą opisanego programu komputerowego, stanowi pomocne narzędzie w podejmowaniu decyzji w sytuacjach ustalania wagi pobranych i modyfikowanych danych dotyczących ważnego w danym momencie zagadnienia. Dlatego też wykazana przydatność samego programu, jak i jego dodatkowych funkcji w procesie wyznaczania rangi ważności parametrów konstrukcyjno-eksploatacyjnych zachęca do dalszych prac nad modernizacją, testowaniem, usprawnianiem, a także rozbudową programu, tak aby stał się on jeszcze bardziej wartościowy dla wspomnianego procesu.

Jak zostało wspomniane w punkcie trzecim, ciekawą funkcjonalnością programu może być moduł liczący ilość gałęzi prawdziwych wielowartościowych logicznych drzew decyzyjnych przedstawiających wybrane przez program optymalne układy. Warto pomyśleć nad tym rozwiązaniem, by ograniczyć (szczególnie w obszernych przykładach) ogromną liczbę układów optymalnych do zbioru kilku tych najkorzystniejszych. Kolejnym pomysłem udoskonalenia programu jest możliwość wprowadzenia do procesu obliczeniowego zestawu parametrów z jedną lub kilkoma zmiennymi jednowartościowymi. Bardzo często można bowiem spotkać przypadki zestawów parametrów z jednym przyjmującym tylko wartość 0. Taka zmienna nie powinna podlegać standardowemu procesowi redukcyjnemu. Jej ranga ważności zawsze powinna być wyliczana w specjalny sposób, taki, który nie zakłama dalszych obliczeń. Dalsza praca nad programem pozwoli na wprowadzenie do procesu obliczeniowego także tych specyficznych przypadków i wyliczenie także dla nich układów optymalnych.

Literatura

- [1] PARTYKA M.A., TISZBIEREK A.: *Zastosowanie logicznych algorytmów minimalizacyjnych do komputerowego wspomaganie wyznaczania rangi ważności parametrów w układach automatyki i sterowania*. „Napędy i Sterowanie” 9/2015.
- [2] TISZBIEREK A.: *Wyznaczanie rangi ważności parametrów konstrukcyjno-eksploatacyjnych z uwzględnieniem interakcji i warunkowości na przykładzie CAD układów automatyki i sterowania*. „Zeszyty Naukowe Politechniki Opolskiej. Elektryka” z. 74, nr 362/2016, Opole 2016.
- [3] PAWŁAK Z.: *Systemy informacyjne*. WNT, Warszawa 1983.
- [4] PARTYKA M.A.: *Podobieństwa i różnice przybliżonej klasyfikacji obiektów w ujęciu logiki i systemów informacyjnych dla CAD procesów decyzyjnych*. XXIV Konferencja Zastosowań Matematyki, Instytut Matematyki PAN, Zakopane 1995.
- [5] TISZBIEREK A., PARTYKA M.A.: *Zastosowanie komputerowego wspomaganie procesu wyznaczania optymalnych logicznych wielowartościowych drzew decyzyjnych na rzeczywistym przykładzie zmiennych warunkowych o podobnej ważności*. XLV Ogólnopolska Konferencja Naukowa Zastosowań Matematyki PAN, Zakopane-Kościelisko 2016, Instytut Matematyki PAN, Warszawa 2016.

 dr inż. Agnieszka Tiszbierek jest adiunktem na Wydziale Inżynierii Produkcji i Logistyki Politechniki Opolskiej; e-mail: a.tiszbierek@po.opole.pl

artykuł recenzowany

reklama

Darmowa e-prenumerata!

www.nis.com.pl





napędy i sterowanie miesięcznik naukowo-techniczny