

Zygmunt KUŚ

MACIERZOWA POSTAĆ OGRANICZEŃ W ZADANIU PROGRAMOWANIA LINIOWEGO DEFINIUJĄCYM FILTR STOSOWY OPTIMALNY ZE WZGLĘDU NA ŚREDNI BŁĄD BEZWZGLĘDNY

Streszczenie. W pracy przedstawiono rozwiązanie zadania projektowania filtru stosowego w oparciu o kryterium średniego błędu bezwzględnego.

Idea filtracji stosowej oparta jest na dekompozycji ciągu wielowartościowego wybranego z macierzy obrazu, odpowiednio do użytej apertury, do ciągów binarnych, filtracji tych ciągów binarnych a następnie powrocie do wyjściowego ciągu wielowartościowego przez dodanie wynikowych ciągów binarnych.

Stosowanie różnych funkcji boolowskich pozwala na uzyskiwanie różnych filtrów.

W pracy sformułowano model zadania optymalizacji. Następnie określono minimalizowany wskaźnik jakości i zaprezentowano przeformułowanie ograniczeń właściwości układania w stos do postaci macierzowej. Kolejny krok stanowiło sformułowanie zadania minimalizacji jako zadania programowania liniowego. Rozwiązanie tego zadania, które zależy od parametrów sygnału i współczynników kosztu, określa poszukiwaną funkcję boolowską.

Na koniec przedstawiono rozwiązania pewnych problemów numerycznych i przykłady macierzy ograniczeń dla szerokości apertury 3 i 5.

MATRIX CONSTRAINTS FOR LINEAR PROGRAMMING PROBLEM DEFINING OPTIMAL STACK FILTER BASED ON THE MEAN ABSOLUTE ERROR CRITERION

Summary. In the paper a solution to the problem of stack filter design based on the mean absolute error criterion is given.

The stack filtering idea is based on decomposition of multivalued sequence taken from the image matrix, according to the used window, into binary sequences, filtering of those binary sequences and then returning to multivalued output sequence by adding resulting binary sequence.

By applying appropriate Boolean function to each binary sequence different types of filtering are defined.

In the paper the model of optimization problem is formulated. Then a quality index to be minimized is defined an reformulation of stacking property constraints to matrix

form is presented. The next step consists in formulation of the minimization problem as a linear problem. Solution to this problem, which depends on signal parameters and cost coefficients, define the searched Boolean function.

Finally, solutions of some numerical problems and examples of constraints matrix for window width 3 and 5 are presented.

Wstęp

Główną motywacją badań związanych z przetwarzaniem stosowym jest dążenie do zwiększenia szybkości przetwarzania obrazów oraz zmniejszenia kosztów produkcji. Szczególnie duże wymagania względem szybkości działania stawia przetwarzanie obrazów w czasie rzeczywistym.

Przetwarzanie stosowe polega na wybraniu z obrazu, zgodnie z ustaloną aperturą, wielowartościowego ciągu liczb (mogą one oznaczać poziomy szarości dla obrazu 2D lub wartości pomiarów dla sygnału 1D), następnie dekompozycji tego M - wartościowego ciągu do $M - 1$ ciągów binarnych. Kolejnym krokiem jest obliczenie dla każdego z tych ciągów wartości odpowiedniej funkcji boolowskiej, przy czym kolejne wartości ciągu odpowiadają kolejnym zmiennym funkcji boolowskiej. Wartość wyjścia filtru stosowego otrzymujemy przez dodanie wartości funkcji boolowskiej ze wszystkich poziomów progowania.

Zmieniając postać użytej funkcji boolowskiej otrzymujemy różne filtry stosowe. O tak zbudowanym przetworniku mówimy, że jest to przetwornik stosowy, jeżeli funkcja boolowska posiada właściwość układania w stos. Funkcja boolowska posiada właściwość układania w stos, jeżeli można ją przedstawić w zminimalizowanej postaci sumacyjnej, nie zawierającej dopełnień zmiennych tej funkcji. Takie funkcje są nazywane dodatnimi funkcjami boolowskimi [3].

Ważną właściwością filtrów stosowych jest możliwość realizowania ich w technologii VLSI - powoduje to zwiększenie szybkości, umożliwiające przetwarzanie obrazów w czasie rzeczywistym.

Podstawowe właściwości filtrów stosowych opisano w [1,9 - 12].

Ograniczenie posiadania właściwości układania w stos nie jest ograniczeniem zbytnio zmniejszającym liczbę dodatnich funkcji boolowskich. W literaturze [14] można znaleźć rozważania, z których wynika, że dla 3 zmiennych jest 20 dodatnich funkcji boolowskich, dla 5 zmiennych jest ich 7581, a dla 7 zmiennych jest ich co najmniej 2^{35} . Zatem pojawia się pytanie, jak wybrać postać funkcji najbardziej odpowiednią. W dalszym toku rozważań przyjmuje się, podobnie jak w [2,4-8,13], że funkcja boolowska ma minimalizować średni błąd bezwzględny pomiędzy wyjściem filtru stosowego a sygnałem (obrazem) idealnym niezakłóconym. Sygnały: idealny, zakłócający i zakłócony będą modelowane za pomocą stacjonarnych, dyskretnych w czasie łańcuchów Markowa. W celu znalezienia optymalnej funkcji boolowskiej przekształca się wskaźnik jakości do postaci Zadania Programowania Liniowego.

Ograniczenia ZPL wynikają z właściwości układania w stos i z dopuszczalnych wartości funkcji boolowskiej.

Szczegółowy model zadania poszukiwania funkcji boolowskiej, minimalizującej średni błąd bezwzględny oraz przekształcenie zadania poszukiwania funkcji boolowskiej, minimalizującej średni błąd bezwzględny do ZPL, przedstawiono w [13]. W pracy zaproponowano metodę przedstawienia w postaci nierówności macierzowej ograniczeń ZPL wynikających z właściwości układania w stos i z dopuszczalnych wartości funkcji boolowskiej.

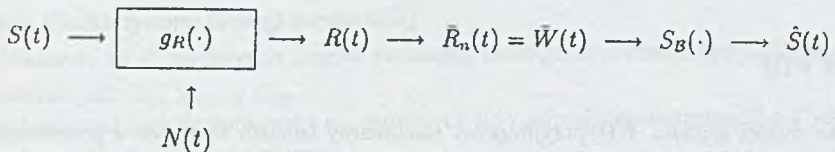
Przedstawiono również przykłady algorytmów w Matlabie pozwalających otrzymywać odpowiednie macierze ograniczeń oraz przykłady macierzy ograniczeń dla funkcji boolowskich 3 i 5 zmiennych.

1. Zadanie filtracji optymalnej

W rozdziale tym zostaną przedstawione: model zadania optymalizacji, wskaźnik jakości oraz zadanie minimalizacji średniego błędu bezwzględnego jako zadanie programowania liniowego.

1.1. Model zadania optymalizacji

Na rys.1 pokazany jest model zadania filtracji optymalnej. Sygnał oryginalny (niezakłócony) $S(t)$ jest zaszumiany sygnałem $N(t)$. Sposób tego zaszumiania opisuje statyczna funkcja $g_R(\cdot)$, wartość $R(t)$ w chwili t zależy tylko od wartości $S(t)$ i $N(t)$ w chwili t . Z sygnału $R(t)$ wybierany jest ciąg $\bar{R}_n(t)$ o długości n równej szerokości apertury filtru $S_B(\cdot)$. Ciąg ten nazywany będzie wektorem aperturowym i oznaczany jako $\bar{W}(t)$. W wyniku przetwarzania przez filtr stosowy $S_B(\cdot)$ sygnału wektora aperturowego uzyskujemy ocenę sygnału oryginalnego $\hat{S}(t)$.



$S(t)$ - sygnał idealny

$N(t)$ - sygnał zakłócający

$R(t)$ - sygnał obserwowany przez filtr

n - szerokość apertury filtru

$\bar{W}(t)$ - wektor aperturowy

$\hat{S}(t)$ - sygnał wyjściowy z filtru

$S_B(\cdot)$ - filtr stosowy

Rys.1. Model zadania filtracji optymalnej

Fig.1. Structure of the optimal filtering problem

Model S (t) i N (t)

Jako model sygnału S (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_S = \{S_1, \dots, S_{M_S}\}$, gdzie M_S jest liczbą stanów przyjmowanych przez S (t) i macierzą przejść P_S .

Prawdopodobieństwa graniczne stanów S_1, \dots, S_{M_S} oznaczamy przez h_1, \dots, h_{M_S} , gdzie

$$h_j = \lim_{t \rightarrow \infty} P(S(t) = S_j) \quad (1)$$

Jako model sygnału N (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_N = \{N_1, \dots, N_{M_N}\}$, gdzie M_N jest liczbą stanów przyjmowanych przez N (t) i macierzą przejść P_N .

Prawdopodobieństwa graniczne stanów N_1, \dots, N_{M_N} oznaczamy przez v_1, \dots, v_{M_N} , gdzie

$$v_j = \lim_{t \rightarrow \infty} P(N(t) = N_j) \quad (2)$$

Model R (t)

Jako model sygnału R (t) przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_R = \{r_1, \dots, r_{M_R}\}$, gdzie M_R jest liczbą stanów przyjmowanych przez R (t) i macierzą przejść P_R .

$$M_R = M_S \cdot M_N \quad (3)$$

$$Q_R = Q_S \times Q_N \quad (4)$$

$$P_R = P_S \otimes P_N \quad (5)$$

Prawdopodobieństwo graniczne stanu $r_k = (S_i, N_j)$ przy założeniu niezależności S_i oraz N_j obliczamy jako:

$$g_k = \lim_{t \rightarrow \infty} P(R(t) = r_k) = h_i \cdot v_j \quad (6)$$

Model $\bar{W}(t)$

Jako model sygnału $\bar{W}(t)$ przyjmujemy stacjonarny łańcuch Markowa z przestrzenią stanów $Q_W = \{\bar{W}_1, \dots, \bar{W}_{M_W}\}$, gdzie M_W jest liczbą stanów przyjmowanych przez $\bar{W}(t)$ i macierzą przejść P_W .

Sygnal $\bar{W}(t)$ powstaje przez nakładanie apertury o szerokości n na sygnał R (t) i przesuwaniu jej kolejno o jedną próbkę.

Zagadnienia dotyczące znajdowania Q_W i P_W opisano w [13].

Przez $\pi(\bar{W}_k)$, $k = 1, \dots, M_W$ oznaczmy prawdopodobieństwa graniczne

$$\pi(\bar{W}_k) = \lim_{t \rightarrow \infty} P(\bar{W}(t) = \bar{W}_k) \quad (7)$$

1.2. Wskaźnik jakości

Jako minimalizowany wskaźnik jakości przyjmujemy średni błąd bezwzględny pomiędzy sygnałem niezakłóconym a wyjściem filtru.

$$C(S_B) = E[|S(t) - S_B(\bar{W}(t))|] \quad (8)$$

Właściwość układania w stos pozwala nam przekształcić średni błąd bezwzględny dla sygnałów wielowartościowych do sumy średnich błędów bezwzględnych dla sygnałów binarnych, występujących na poszczególnych poziomach progowania:

$$C(S_B) = \sum_{i=1}^M E[|s_i(t) - B(\bar{w}_i(t))|], \quad (9)$$

gdzie M jest liczbą poziomów progowania, $s_i(t)$ - sygnałem idealnym sprogowanym na poziomie i , a $\bar{w}_i(t)$ jest wektorem aperturowym $\bar{W}(t)$ sprogowanym na poziomie i .

1.3. Sformułowanie zadania minimalizacji średniego błędu bezwzględnego jako zadania programowania liniowego

Do przekształcenia wskaźnika jakości zostanie użyte pojęcie stochastycznej funkcji boolowskiej i statystyczna interpretacja właściwości układania w stos.

Definicja stochastycznej funkcji boolowskiej

Zakładamy, że 2^n możliwych ciągów binarnych o długości n mamy uporządkowanych i oznaczonych jako $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{2^n}$.

Stochastyczną funkcję boolowską n -zmiennych $B(\cdot)$ definiujemy wektorem P z 2^n elementami, w którym i -ty element oznacza:

$$P_B(1 | \bar{x}_i) = P(B \text{ ma wartość } 1 | \bar{x}_i \text{ jest argumentem } B) \quad (10)$$

Jednocześnie

$$P_B(0 | \bar{x}_i) = 1 - P_B(1 | \bar{x}_i) \quad (11)$$

Definicja stochastycznej właściwości układania w stos

Stochastyczna funkcja boolowska posiada stochastyczną właściwość układania w stos jeżeli:

$$E(B(\bar{x})) \geq E(B(\bar{y})) \Leftrightarrow \bar{x} \geq \bar{y}, \quad (12)$$

gdzie E jest operatorem wartości oczekiwanej.

Przyjęty wskaźnik jakości zostanie przekształcony do postaci funkcji liniowej prawdopodobieństw opisujących stochastyczną funkcję boolowską. Minimalizacja tej funkcji liniowej metodą rozwiązania ZPL pozwala znaleźć stochastyczną funkcję boolowską, minimalizującą średni błąd bezwzględny.

Dzięki sformułowaniu oceny sygnału oryginalnego przez filtr na podstawie sygnału zsumowanego jako problemu decyzyjnego [2] możemy przedstawić wskaźnik kosztów ponoszonych na poziomie progowania 1 przy użyciu funkcji boolowskiej B w następującej postaci:

$C(B|1) =$

$$\sum_{j=1}^{2^n} \left[P_B(0|\bar{x}_j) \cdot \left(\sum_{\bar{W} \in Q_w: \bar{w}_1 = \bar{x}_j} [C_1(\bar{W}, 0, 0)\pi(0|\bar{W}, 1) + C_1(\bar{W}, 1, 0)\pi(1|\bar{W}, 1)] \cdot \pi(\bar{W}) \right) + P_B(1|\bar{x}_j) \cdot \left(\sum_{\bar{W} \in Q_w: \bar{w}_1 = \bar{x}_j} [C_1(\bar{W}, 0, 1)\pi(0|\bar{W}, 1) + C_1(\bar{W}, 1, 1)\pi(1|\bar{W}, 1)] \cdot \pi(\bar{W}) \right) \right],$$

gdzie:

$C_1(\bar{W}, 1, 0)$ - koszt ponoszony na poziomie 1, gdy filtr ma wyjście równe 0 a w oryginalnym sygnale jest 1,

$C_1(\bar{W}, 0, 0)$ - wyjściem filtru jest 0, powinno być 0,

$C_1(\bar{W}, 0, 1)$ - wyjściem filtru jest 1, powinno być 0,

$C_1(\bar{W}, 1, 1)$ - wyjściem filtru jest 1, powinno być 1,

$\pi(0|\bar{W}, 1) = 1 - \pi(1|\bar{W}, 1)$ = prawdopodobieństwo (poprawne wyjście filtru = 1 | na wejściu jest \bar{W}).

Całkowity koszt ponoszony przy użyciu w filtrze stosowym funkcji boolowskiej B otrzymujemy przez sumowanie kosztów ponoszonych na poszczególnych poziomach progowania.

$$C(B) = \sum_{i=1}^M C(B|i) \quad (13)$$

Przyjmujemy, że koszty ponoszone przy podjęciu przez filtr poprawnej decyzji są równe 0

$$C_1(\bar{W}, 0, 0) = C_1(\bar{W}, 1, 1) = 0 \quad (14)$$

i podstawiając

$$\pi(1 | \bar{W}, 1) = 1 - \pi(0 | \bar{W}, 1) \quad (15)$$

otrzymujemy:

$$C'(B | 1) =$$

$$\sum_{j=1}^{2^n} P_B(1 | \bar{x}_j) \cdot \left[\sum_{\bar{W} \in Q_w: \bar{w}_1 = \bar{x}_j} \pi(\bar{W}) \cdot \left\{ \pi(0 | \bar{W}, 1) \cdot [C_1(\bar{W}, 0, 1) + C_1(\bar{W}, 1, 0)] - C_1(\bar{W}, 1, 0) \right\} \right] \quad (16)$$

$$C'(B) = \sum_{i=1}^M C'(B | 1) \quad (17)$$

Przyjmując: koszty $C_1(\bar{W}, 0, 1) = CW01$ i $C_1(\bar{W}, 1, 0) = CW10$ niezależne od poziomu progowania i wektora aperturowego oraz binarny sygnał $R(t)$ na wejściu ($l = 1$), otrzymujemy:

$$C'(B) = \sum_{j=1}^{2^n} P_B(1 | \bar{x}_j) \cdot \left[\pi(\bar{x}_j) \cdot \left\{ \pi(0 | \bar{x}_j, 1) \cdot [CW01 + CW10] - CW10 \right\} \right], \quad (18)$$

gdzie:

$$\pi(\bar{x}_j) = \sum_{\bar{W} \in Q_w: \bar{w}_1 = \bar{x}_j} \pi(\bar{W}) \quad (19)$$

$$\pi(0 | \bar{x}_j) = \sum_{\bar{W} \in Q_w: \bar{w}_1 = \bar{x}_j} \pi(0 | \bar{W}) \quad (20)$$

Zauważmy, że uzyskany wskaźnik $C'(B)$ jest liniową kombinacją prawdopodobieństw określających poszukiwaną funkcję boolowską i pewnych współczynników, stałych dla przyjętego modelu sygnałów $S(t)$ i $N(t)$ i dla ustalonych kosztów $CW01$ i $CW10$.

Oznaczając wektor prawdopodobieństw definiujących funkcję boolowską jako

$$P = [P_1, P_2, \dots, P_{2^n}], \quad (21)$$

gdzie

$$P_j = P_B(1 | \bar{x}_j) \quad (22)$$

oraz wektor współczynników jako

$$c = [c_1, c_2, \dots, c_{2^n}], \quad (23)$$

gdzie

$$c_j = \pi(\bar{x}_j) \cdot \{\pi(0 | \bar{x}_j, 1) \cdot [CW01 + CW10] - CW10\} \quad (24)$$

możemy sformułować problem poszukiwania funkcji boolowskiej minimalizującej średni błąd bezwzględny jako Zadanie Programowania Liniowego.

$$\min_P c P^T \quad (25)$$

przy ograniczeniach

$$\bar{x}_j \leq \bar{x}_i \Rightarrow P_B(1 | \bar{x}_j) \leq P_B(1 | \bar{x}_i) \quad (26)$$

$$0 \leq P_B(1 | \bar{x}_i) \leq 1 \quad i = 1, \dots, 2^n \quad (27)$$

Pierwsze równanie ograniczeń wynika z konieczności spełnienia przez funkcję boolowską stochastycznej właściwości układania w stos, drugie natomiast związane jest z dopuszczalnymi wartościami, jakie może przyjmować stochastyczna funkcja boolowska.

Ze względu na charakter ograniczeń ZPL i właściwości rozwiązania ZPL prawdopodobieństwa definiujące optymalną funkcję boolowską przyjmują zawsze wartości 0 lub 1. Zatem uzyskiwana stochastyczna funkcja boolowska równoważna jest deterministycznej funkcji boolowskiej, którą wykorzystujemy do stworzenia filtru stosowego.

2. Sformułowanie ograniczeń ZPL w postaci nierówności macierzowych

Rozważając ograniczenia, które musi spełniać funkcja boolowska, aby mogła być użyta do stworzenia filtru stosowego, chcielibyśmy rozwiązać dwa związane z tymi ograniczeniami problemy. Właściwość układania w stos narzuca konieczność spełniania warunku $\bar{x}_i \leq \bar{x}_j \Rightarrow B(\bar{x}_i) \leq B(\bar{x}_j)$. A zatem z jednej strony chcielibyśmy zminimalizować liczbę porównań koniecznych do sprawdzenia, czy powyższy warunek zawsze zachodzi, a z drugiej strony chcielibyśmy zapisać te porównania w postaci nierówności macierzowej, którą można by wykorzystać jako nierówność ograniczeń ZPL.

Wprowadźmy następujące oznaczenia:

$(r0s)$ - ciąg n binarnych zmiennych, r - podciąg o długości l_r , s - podciąg o długości l_s , $l_r + l_s + 1 = n$

$(r0s, r1s)$ - para dwóch ciągów binarnych różniących się na jednej pozycji

$P_B(r0s,1)$ - prawdopodobieństwo zdarzenia: $B(r0s) = 1$

Definicja stochastycznej lokalnej właściwości układania w stos

O stochastycznej funkcji boolowskiej $B(\cdot)$ mówimy, że posiada lokalnie (dla danej pary $(r0s, r1s)$) właściwość układania w stos, jeżeli

$$P_B(r0s,1) \leq P_B(r1s,1) \quad (28)$$

Stochastyczna funkcja boolowska $B(\cdot)$ posiada stochastyczną właściwość układania w stos, jeżeli posiada stochastyczną lokalną właściwość układania w stos dla wszystkich możliwych par $(r0s, r1s)$.

Spróbujmy zatem zapisać w postaci nierówności macierzowej warunek posiadania lokalnej właściwości układania w stos dla wszystkich możliwych par $(r0s, r1s)$.

Przyjmijmy, że ustaliliśmy pewną kolejność w zbiorze par $(r0s, r1s)$. Zatem dla pary o numerze q możemy zapisać:

$$P_B(r0s,1) \leq P_B(r1s,1) \quad \text{dla pary } q \quad (29)$$

Przechodząc na oznaczenia związane z definicją stochastycznej funkcji boolowskiej i przyjmując, że $\bar{x}_i = r0s$ oraz $\bar{x}_j = r1s$, możemy zapisać powyższą nierówność jako:

$$P_B(1 | \bar{x}_i) \leq P_B(1 | \bar{x}_j) \quad \text{dla pary } q, \quad (30)$$

zatem

$$P_i \leq P_j \quad \text{dla pary } q \quad (31)$$

$$P_i - P_j \leq 0 \quad \text{dla pary } q \quad (32)$$

Przyjmując, że numer kolumny macierzy odpowiada numerowi \bar{x}_k , $k = 1, \dots, 2^n$, a numer wiersza numerowi pary $(r0s, r1s)$, przedstawimy ideę tworzenia macierzy ograniczeń w następujący sposób:

$$\begin{array}{c}
 \begin{matrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_j \\ \vdots \\ P_{2^n} \end{matrix} \\
 1 \dots i \dots j \dots 2^n \\
 \begin{matrix} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{matrix} \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \vdots \\ P_i - P_j \\ \vdots \end{bmatrix} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{matrix}
 \end{array}$$

Z przedstawionego schematu mnożenia macierzy wynika sposób zapisu w postaci nierówności macierzowej warunku: $P_i - P_j \leq 0$ dla pary q . Oznaczając natomiast poszczególne macierze składowe jako:

$$A_1 = \begin{matrix} 1 & \dots & i & \dots & j & \dots & 2^n \\ \begin{matrix} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{matrix} \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \end{bmatrix} \end{matrix} \quad (33)$$

$$P^T = \begin{bmatrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_j \\ \vdots \\ P_{2^n} \end{bmatrix} \quad b_1 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ q \\ \vdots \\ n2^{n-1} \end{matrix} \quad (34)$$

możemy powyższe ograniczenia, zapewniające posiadanie przez uzyskaną funkcję boolowską właściwości układania w stos, zapisać w następującej postaci:

$$A_1 P^T \leq b_1 \quad (35)$$

Pozostaje jeszcze uwzględnić w macierzy ograniczeń dopuszczalny zakres wartości dla P_i . Mianowicie

$$0 \leq P_i \leq 1 \quad (36)$$

Jeżeli przyjmiemy $\bar{x}_1 = [0 \dots 0]$ oraz $\bar{x}_{2^n} = [1 \dots 1]$, to dzięki właściwości układania w stos warunek $(0 \leq P_i \leq 1)$ jest spełniony, gdy zachodzi $P_1 \geq 0$ i $P_{2^n} \leq 1$, a to zapewnia nam następującą modyfikacja macierzy A_1 i wektora b_1 :

$$A = \begin{bmatrix} 1 & \dots & 2^n \\ -10 & \dots & 0 \\ \vdots & A_1 & \vdots \\ 1 & \dots & 0 \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ n2^{n-1} + 2 \end{matrix} \quad (37)$$

$$b = \begin{bmatrix} 0 \\ \vdots \\ b_1 \\ 1 \end{bmatrix} \begin{matrix} 1 \\ \vdots \\ n2^{n-1} + 2 \end{matrix} \quad (38)$$

Reasumując, stwierdzamy, że funkcję boolowską minimalizującą średni błąd bezwzględny otrzymujemy w wyniku rozwiązania następującego ZPL:

$$\begin{aligned} \min_P & c P^T \\ & AP^T \leq b \end{aligned} \quad (39)$$

3. Numeryczne rozwiązanie problemu obliczania macierzy ograniczeń A i wektora b dla zadanej szerokości apertury n

Obecnie zostanie przedstawiony algorytm pozwalający obliczać A i b dla dowolnego n .

3.1. Ogólna idea numerycznego algorytmu obliczania A i b

Z przedstawionych w poprzednim rozdziale rozważań wynika, że utworzenie macierzy A_1 polega na utworzeniu dla każdej pary $(r0s, r1s)$ wiersza zawierającego w odpowiednich kolumnach „1” i „-1” oraz zera w pozostałych kolumnach. Obliczenia pozwalające uzyskać macierz A_1 wykonujemy w dwóch etapach.

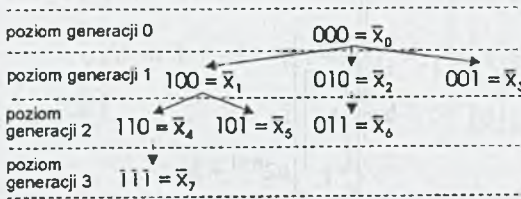
W pierwszym etapie tworzymy tablicę argumentów zawierającą wszystkie możliwe argumenty $\bar{x}_k, k = 1, \dots, 2^n$ funkcji boolowskiej - wszystkie możliwe kombinacje zmiennych funkcji boolowskiej. W drugim etapie na podstawie tej tablicy tworzymy macierz A_1 .

Uporządkowanie \bar{x}_k w tablicy otrzymujemy przez zastosowanie szczególnego sposobu generowania kolejnych \bar{x}_k . Cała tablica podzielona jest na poziomy generacji. Na poziomie p

występują wszystkie możliwe \bar{x}_k , zawierające p jedynek. Tak więc, przykładowo dla $n = 3$, na poziomie $p = 0$ jest jeden argument $\bar{x}_0 = 000$ (same zera). Na poziomie $p = 1$ są trzy argumenty funkcji boolowskiej: $\bar{x}_1 = 100$, $\bar{x}_2 = 010$, $\bar{x}_3 = 001$.

Argument \bar{x}_k z poziomu $p+1$ jest generowany w następujący sposób.

Wybieramy pierwszy argument z poziomu p . Kolejne argumenty z poziomu $p+1$ powstają przez zmianę kolejno zer leżących na prawo od najbardziej na prawo wysuniętej jedynek, na jedynek. Ilustruje to rys.2.



Rys.2. Algorytm generacji argumentów \bar{x}_k

Fig.2. Algorithm of generating \bar{x}_k

Ten sposób zapewnia uwzględnienie wszystkich argumentów \bar{x}_k , niepowtarzanie ich i uporządkowanie w sposób pozwalający tworzyć kolejne wiersze macierzy A_1 .

W zapisie wygenerowanych \bar{x}_k w tablicy argumentów będziemy korzystać z następującego systemu kodowania.

Tablica argumentów zawiera $n - 1$ kolumn i $2^n - 2$ wierszy. Argument zawierający same zera - x_0 i argument zawierający same jedynek - x_{2^n} nie będą wpisywane do tablicy argumentów. Przejście pomiędzy $p = 0$ i $p = 1$ oraz pomiędzy $p = n$ i $p = n$ będzie zapisane w macierzy A_1 bez korzystania z tablicy argumentów. Jeden wiersz tablicy argumentów opisuje jeden argument \bar{x}_k . Niezerowe elementy w tym wierszu oznaczają pozycje w \bar{x}_k , na których występują w \bar{x}_k jedynek. Tak więc w wierszach opisujących argumenty na poziomie p występuje $(n - 1 - p)$ elementów równych zero. Tablicę argumentów dla $n = 3$ przedstawia rys.3.

poziom generacji 1	1 0	\bar{x}_1
	2 0	\bar{x}_2
	3 0	\bar{x}_3
poziom generacji 2	1 2	\bar{x}_4
	1 3	\bar{x}_5
	2 3	\bar{x}_6

Rys.3. Tablica argumentów dla $n = 3$

Fig.3. Generation table for $n = 3$

Jeżeli założymy, że kolumny macierzy A_1 są ponumerowane podobnie jak \bar{x}_k (od 0 do $n-1$), to tworzenie macierzy A_1 przebiega następująco.

Pierwsze n wierszy opisuje pary:

$$\begin{array}{l} \bar{x}_0 \rightarrow \bar{x}_1 \\ \bar{x}_0 \rightarrow \bar{x}_2 \\ \vdots \\ \bar{x}_0 \rightarrow \bar{x}_n \end{array}$$

Wiersze od $n+1$ do $n2^{n-1} - n$ są tworzone na podstawie tablic argumentów.

Najpierw ustalamy, że pierwszy \bar{x}_i z poziomu $p=1$ będzie miał numer jeden - ($\bar{x}_i = \bar{x}_1$).

Następnie bierzemy pierwszy \bar{x}_j z poziomu generacji $p=2$ (dla $n=3$ $\bar{x}_j = \bar{x}_4$).

Przez Z_1 oznaczamy zbiór niezerowych elementów z wiersza opisującego \bar{x}_i , ($Z_1 = \{1\}$).

Przez Z_2 oznaczamy zbiór niezerowych elementów z wiersza opisującego \bar{x}_j , ($Z_2 = \{1,2\}$).

Jeżeli $Z_1 \in Z_2$, to tworzymy nowy wiersz zawierający „1” w i -tej kolumnie i „-1” w j -tej kolumnie w macierzy A_1 .

Następnie porównujemy w ten sposób pierwszy \bar{x}_k z poziomu $p=1$ z następnym \bar{x}_k z poziomu $p=2$ aż do przejścia przez wszystkie \bar{x}_k z poziomu $p=2$. Wtedy bierzemy drugi \bar{x}_k z poziomu $p=1$ i porównujemy z wszystkimi \bar{x}_k z poziomu $p=2$. Powtarzamy te porównania aż do przejścia przez wszystkie \bar{x}_k z poziomu $p=1$.

W ten sam sposób porównujemy kolejno argumenty z poziomu $p=2$ z argumentami z poziomu $p=3$ itd. Procedura zostaje zakończona, gdy $p=n-2$.

Wiersze od $n2^{n-1} - n + 1$ do $n2^{n-1}$ opisują pary:

$$\begin{array}{l} \bar{x}_2^{n-n} \rightarrow \bar{x}_2^n \\ \bar{x}_2^{n-n+1} \rightarrow \bar{x}_2^n \\ \vdots \\ \bar{x}_2^{n-1} \rightarrow \bar{x}_2^n \end{array}$$

Dla $n=3$ otrzymujemy następującą postać A_1

$$A_1 = \begin{array}{cccccccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{array} \quad b_1 = \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}$$

Po uwzględnieniu ograniczeń wynikających z dopuszczalnych wartości prawdopodobieństw określających stochastyczną funkcję boolowską otrzymujemy:

$$A = \begin{pmatrix}
 -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}
 \quad
 b = \begin{pmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 1
 \end{pmatrix}$$

Z przedstawionych rozważań wynika, że wymiar macierzy A oraz wartość jej elementów nie zależą od przyjętych w modelowaniu sygnałów: Q_S, P_S, Q_N, P_N . Zależą one jedynie od rozmiaru apertury, a tym samym od liczby zmiennych w funkcji boolowskiej.

3.2. Algorytmy obliczania A i b za pomocą programu Matlab

Prześledźmy teraz przykładowy algorytm obliczania macierzy A i wektora b , opisujących ograniczenia rozważanego ZPL. Algorytm zapisano zgodnie z notacją dla programu Matlab.

Oto najważniejsze występujące w programie zmienne:

n - rozmiar apertury = liczba zmiennych funkcji boolowskiej

NPOK - tabela wartości symbolu Newtona $NPOK(i, j) = \binom{j}{i}$

JED - tablica argumentów

OFF - tablica zawierająca wartości przesunięć numerów argumentów w tablicy JED - numer argumentu w tablicy JED jest równy numerowi argumentu na poziomie generacji p plus OFF(p)

STAN - tabela przedstawiająca argumenty funkcji boolowskiej w tej samej kolejności co w JED, tyle że w postaci zer i jedynek oraz z uwzględnieniem argumentów zawierających same zera i same jedynki.

Obliczenia można podzielić na dwie części. Pierwsza część służy wyliczeniu pomocniczych zmiennych i tablicy argumentów noszącej nazwę JED. Druga część polega na obliczeniu, na podstawie tablicy JED, macierzy A i wektora b .

Pierwszą część obliczeń realizują następujące algorytmy:

```

%*****
%          OBLICZANIE MACIERZY OGRANICZEN ,,A'' I WEKTORA ,,b''
%
% WEJSCIE: n      - rozmiar apertury,
%          NPOK   - tabela wartosci symbolu Newtona NPOK(i,j) = ,,i po j''
% WYJSCIE: STAN  - macierz definiujaca kolejnosc argumentow w przestrzeni
%                wszystkich mozliwych argumentow
%                rozmiar STAN = (2^n) x (n), zawiera ona zera i jedynki
%          A      - macierz ograniczen dla zadania PL
%          b      - wektor dla ograniczen: Ax <= b
%*****

%-----
% OBLICZANIE POMOCNICZEJ MACIERZY ,,OFF'' ZAWIERAJACEJ PRZESUNIECIA NUMERACJI
%   W ,,JED'' WZGLEDEM NUMERACJI NA POSZCZEGOLNYCH POZIOMACH ,,p''
%-----
OFF = zeros(n-1,1);
for p=2 : (n-1)
    for j=2 : p
        OFF(p,1)=OFF(p,1) + NPOK(n,j-1);
    end
end

%-----
% OBLICZANIE ,,JED'' OPISUJACEJ GDZIE W ARGUMENTACH F.BOOL. SA ZERA I JEDYNKI
%-----
JED = zeros(2^n-2,n-1) ;
for i=1 : n
    JED(i,1)=i ;
end

for p=2 : n-1
    k=0 ;
    r=1 ;

    while (r < NPOK(n,p-1)) | (r == NPOK(n,p-1))
        KZR=0 ;
        ROB=0 ;
        if JED(r+OFF(p-1,1),p-1) < n
            ROB=1 ;
        end

        while ROB == 1
            k=k+1 ; KZR=KZR+1 ;
            for l=1 : p-1
                JED(k+OFF(p,1),l) = JED(r+OFF(p-1,1),l);
            end
            JED(k+OFF(p,1),p) = JED(r+OFF(p-1,1),p-1) + KZR ;
            if JED(k+OFF(p,1),p) == n
                ROB=0 ;
            end
        end
        r=r+1 ;
    end
end
end

```

```

%*****
%  OBLICZANIE MACIERZY  ,,STAN''  ZAWIERAJACEJ WSZYSTKIE MOZLIWE  2^n
%  KOMBINACJE  n  ZMIENNYCH  WEJSCIOWYCH;  KAZDY  WIERSZ  ZAWIERA  n  ZER/JEDYNEK
%  I  STANOWI  JEDEN  ARGUMENT  F.BOOL.,  NUMER  WIERSZA  =  NUMER  ARGUMENTU  F.BOOL.
%*****

```

```
STAN = zeros(2^n,n) ;
```

```

for i=1 : 2^n-2
    for j=1 : n-1
        if JED(i,j)>0
            STAN(i+1,JED(i,j)) = 1 ;
        end
    end
end

```

```
STAN(2^n,:) = ones(1,n) ;
```

Obliczanie macierzy A przebiega w następujących etapach. Najpierw tworzona jest macierz zer o wymiarach odpowiadających macierzy A . W etapie I zapisywane są ograniczenia wynikające z przejścia pomiędzy poziomem zerowym a pierwszym. W etapie II zapisywane są ograniczenia dla przejścia z poziomu $n-1$ do poziomu n . Etapy III i IV odpowiadają ograniczeniom na dopuszczalną wartość prawdopodobieństw definiujących stochastyczną funkcję boolowską. Odpowiada to wierszom dodawanym przy przejściu z A_1 na A . Etap V generuje wiersze wynikające z porównania kolejnych stanów zapisanych w tabeli JED.

```

%*****
%  OBLICZANIE MACIERZY OGRANICZEN  ,,A''
%*****

```

```
A = zeros(n*2^(n-1)+2, 2^n) ;
```

```

%..ETAP.IV.....
A(1,1)=-1 ;

```

```

%..ETAP.I.....
for aw=2 : (n+1)
    A(aw,1)=1;
end
for aw=2 : (n+1)
    A(aw,aw)=-1;
end

```

```

%.ETAP.II.....
for aw=(n*2^(n-1)+2-n) : (n*2^(n-1)+1)
    A(aw,2^n)=-1;
end
ak=2^n-n ;
for aw=(n*2^(n-1)+2-n) : (n*2^(n-1)+1)
    A(aw,ak)=1;
    ak=ak+1;
end

%.ETAP.III.....
A(n*2^(n-1)+2,2^n) = 1 ;

%.ETAP.V.....OBLICZENIA..NA..PODSTAWIE...,,JED''.....
% DLA aw=(n+2) : (n*2^(n-1)+1-n)

aw=1+1+n ;

for p=1 : (n-2)
    r=1 ;
    while (x < NPOK(n,p)) | (x == NPOK(n,p))
        k=1 ;
        while (k < NPOK(n,p+1)) | (k == NPOK(n,p+1))
            LOG=zeros(p,1);
            for i=1 : p
                for j=1 : (p+1)
                    if JED(x+OFF(p,1),i) == JED(k+OFF(p+1,1),j)
                        LOG(i,1)=1 ;
                    end
                end
            end
            LOGWAR=0 ;
            for i=1 : p
                LOGWAR=LOGWAR+LOG(i) ;
            end
            if LOGWAR==p
                rglob=r+OFF(p,1)+1 ;
                kglob=k+OFF(p+1,1)+1 ;
                A(aw,rglob) = 1 ;
                A(aw,kglob) = -1 ;
                aw=aw+1 ;
            end
            k=k+1 ;
        end %while-k
        r=r+1 ;
    end %while-r
end %for-p

%*****
%      OBLICZANIE WEKTORA ,,b'' DLA NIEROWNOSCI OGRANICZEN: Ax <= b
%*****

b = zeros(n*2^(n-1)+2, 1) ;
b(n*2^(n-1)+2, 1) = 1 ;

```

Utworzenie wektora b jest ostatnim etapem obliczeń.

3.3. Przykład tablicy argumentów JED, macierzy ograniczeń A i wektora b dla $n = 5$

Obliczone dla $n = 5$ JED i STAN mają następującą postać:

		0 0 0 0 0
	1 0 0 0	1 0 0 0 0
	2 0 0 0	0 1 0 0 0
	3 0 0 0	0 0 1 0 0
	4 0 0 0	0 0 0 1 0
	5 0 0 0	0 0 0 0 1
	1 2 0 0	1 1 0 0 0
	1 3 0 0	1 0 1 0 0
	1 4 0 0	1 0 0 1 0
	1 5 0 0	1 0 0 0 1
	2 3 0 0	0 1 1 0 0
	2 4 0 0	0 1 0 1 0
	2 5 0 0	0 1 0 0 1
	3 4 0 0	0 0 1 1 0
	3 5 0 0	0 0 1 0 1
JED =	4 5 0 0	0 0 0 1 1
	1 2 3 0	1 1 1 0 0
	1 2 4 0	1 1 0 1 0
	1 2 5 0	1 1 0 0 1
	1 3 4 0	1 0 1 1 0
	1 3 5 0	1 0 1 0 1
	1 4 5 0	1 0 0 1 1
	2 3 4 0	0 1 1 1 0
	2 3 5 0	0 1 1 0 1
	2 4 5 0	0 1 0 1 1
	3 4 5 0	0 0 1 1 1
	1 2 3 4	1 1 1 1 0
	1 2 3 5	1 1 1 0 1
	1 2 4 5	1 1 0 1 1
	1 3 4 5	1 0 1 1 1
	2 3 4 5	0 1 1 1 1
		1 1 1 1 1

b jest wektorem kolumnowym o 82 elementach. Pierwsze 81 to zera, a ostatni 82 element to 1. Macierz ograniczeń A ma postać przedstawioną na stronie 144.

4. Zakończenie

W pracy zaproponowano rozwiązanie problemu sformułowania ograniczeń wynikających z właściwości układania w stos w postaci nierówności macierzowej. Takie przedstawienie ograniczeń jest niezbędne do rozwiązania zadania filtracji optymalnej, sformułowanego jako zadanie programowania liniowego. Zaproponowana metoda pozwala na automatyczne znajdowanie macierzy ograniczeń dla dowolnej szerokości apertury filtru. Jest to szczególnie istotne dla zastosowań adaptacyjnych filtrów stosowych. W takim przypadku szerokość apertury jest jednym z parametrów, który może się zmieniać zarówno ze względu na zmiany postaci sygnału idealnego, jak i zakłócenia. Ponieważ szerokość apertury wpływa na postać ograniczeń wynikających z właściwości układania w stos, dlatego przedstawiona w pracy metoda jest kolejnym, niezbędnym krokiem do rozszerzenia możliwości aplikacji filtrów stosowych.

LITERATURA

1. Coyle Edward, Gallagher Neal, Wendt Peter (August 1986) Stack Filters, IEEE Transactions On Acoustics, Speech And Processing, vol. ASS-34, pp. 898-911.
2. Coyle Edward, Lin Jean-Hsang (august 1988) Stack Filters and the Mean Absolute Error Criterion, IEEE Transactions On Acoustics, Speech And Signal Processing, vol. 36, No. 8.
3. Keping Chen, Bit-Serial Realizations of a Class of Nonlinear Filters Based on Positive Boolean Functions, IEEE Transactions On Circuits and Systems, vol.36, No.6, Jun. 1989, pp. 785-794.
4. M. Gabbouj, (December 1986) Optimal stack filter examples and positive Boolean functions, M.S. thesis, School Elec. Eng. Purdue Univ., West Lafayette, IN.
5. Coyle Edward, Jean-Hsang Lin, Babbouj Moncef (december 1989) Optimal Stack Filtering and the Estimation and Structural Approaches to Image Processing, IEEE Transactions On Acoustics, Speech And Signal Processing, vol. 37, No.12, pp. 2037-2066.
6. Gabbouj Moncef, Coyle Edward, (June 1990) Minimum Mean Absolute Error stack Filtering with Structural Constraints and Goals, IEEE Transactions On Acoustics, Speech And Signal Processing, vol.38, No.6, pp. 955-968.
7. Gabbouj Moncef, Coyle Edward, (November 1991) On the LP Which Finds a MMAE Stack Filter, IEEE Transactions On Signal Processing, vol. 39, No.11, pp. 2419-2424.
8. Bing Zeng, Gabbouj Moncef, Yrjo Neuvo, (September 1991) A Unified Design Method for Rank Order, Stack, and Generalized Stack Filters Based on Classical Bayes Decision, IEEE Transactions On Circuits And Systems, vol.38, No.9, pp. 1003-1020.
9. Kuś Z., Wojciechowski K.: Przetwarzanie stosowe. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z.113, Gliwice 1994.

10. Kuś Z., Wojciechowski K.: Deterministyczne właściwości przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z. 113, Gliwice 1994.
11. Kuś Z., Wojciechowski K.: Statystyczne właściwości przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z. 113
12. Kuś Z., Wojciechowski K.: Przetwarzanie medianowe jako szczególny przypadek przetwarzania stosowego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka, z. 113, Gliwice 1994.
13. Kuś Z.: Projektowanie filtrów stosowych w oparciu o kryterium minimalizacji wartości średniej błędu bezwzględnego. Zeszyty Naukowe Politechniki Śląskiej, ser. Automatyka - (w druku).
14. Gilbert E.N., (April 1954) Lattice-theoretic properties of frontal switching functions, J. Math. Phys., vol.33, pp. 57-67.

Recenzent: Prof.dr hab. Mariusz Nieniewski

Wpłynęło do Redakcji 18.01.1996 r.

Abstract

In the paper a solution to the problem of stack filter design based on the mean absolute error criterion is given.

The stack filtering idea is based on decomposition of multivalued sequence taken from the image matrix, according to the used window, into binary sequences, filtering of those binary sequences and then returning to multivalued output sequence by adding resulting binary sequence.

By applying appropriate Boolean function to each binary sequence different types of filtering are defined.

We can talk about stack filtering when the Boolean function has the stacking property i.e. its minimum sum of products representation excludes complementary input variables. These functions are called positive Boolean functions.

The stack filter is fully determined by selection of appropriate Boolean function. As a criteria for selection of this function the minimization of the mean absolute error between required signal and the filter output is taken.

In the paper the model of optimization problem is formulated. Then a quality index to be minimized is defined an reformulation of stacking property constraints to matrix form is presented. The next step consists in formulation of the minimization problem as a linear problem.

Solution to this problem, which depends on signal parameters and cost coefficients, define the searched Boolean function.

Finally, solutions of some numerical problems and examples of constraints matrix for window width 3 and 5 are presented.