

3
INSTYTUT CYBERNETYKI TECHNICZNEJ
POLITECHNIKI WROCŁAWSKIEJ
Komunikat nr 110

ANALIZA PORÓWNAWCZA ALGORYTMÓW
ROZPOZNAWANIA Z UCZENIEM
DLA PROSTYCH ŁANCUCHÓW MARKOWA

Praca doktorska

Marek Kurzyński

Promotor:

Prof. dr hab. inż. Zdzisław Bubnicki

Słowa kluczowe: rozpoznawanie, uczenie, łańcuch Markowa

Wrocław 1974

mgr inż. Marek Kurzyński
Instytut Cybernetyki Technicznej
Politechniki Wrocławskiej
Wrocław, ul. Janiszewskiego 11/17
Budynek C3

Komunikat wpłynął 27.04.1974

SPIS TRESCI

1. WSTĘP	4
2. SFORMUŁOWANIE PROBLEMU I ALGORYTM ROZPOZNAWANIA PRZY PEŁNEJ INFORMACJI PROBABILISTYCZNEJ	9
3. ALGORYTMY ROZPOZNAWANIA Z UCZENIEM DLA PROSTYCH ŁANCUCHÓW MARKOWA	16
3.1. Zmodyfikowany algorytm " $h(N)$ -ty najbliższy sąsiad"	17
3.2. Algorytm rozpoznawania wykorzystujący nieparametry- czny estymator gęstości Parzena	20
3.3. Algorytm rozpoznawania z nieparametrycznym estyma- torem gęstości Murthy'ego	24
3.4. Zmodyfikowany algorytm " najmniejszy przedział "	28
3.5. Zmodyfikowane algorytmy " najbliższa średnia ", "najbliższy sąsiad" i " α najbliższych sąsiadów"	30
4. PODSTAWY I METODYKA ANALIZY PORÓWNAWCZEJ	35
5. ANALIZA PORÓWNAWCZA ALGORYTMÓW ROZPOZNAWANIA Z UCZENIEM DLA PROSTYCH ŁANCUCHÓW MARKOWA	42
5.1. Pewne rezultaty analityczne	42
5.2. Wyniki eksperymentalnego badania algorytmów rozpoznawania	52
6. PRZYKŁAD ZASTOSOWANIA PRAKTYCZNEGO	87
7. WNIOSKI I UWAGI KOŃCOWE	91
LITERATURA	93
DODATEK Biblioteka podprogramów	96

1. WSTĘP

Zadanie rozpoznawania obiektu polega na przyporządkowaniu temu obiektowi, na podstawie pomiarów pewnych wybranych jego fizycznych cech, nazwy lub numeru klasy obiektów, do której on należy. Takie jednolite ujęcie zagadnień rozpoznawania pozwala rozwiązywać je w oparciu o te same metody, których opracowanie jest właśnie zadaniem teorii rozpoznawania obiektów. Rozpoznawanie obiektów jako dyscyplina naukowa zajmuje się więc opracowywaniem teorii i zasad budowy urządzeń technicznych, które byłyby w stanie, w sposób możliwie najdoskonalszy, rozwiązywać te zagadnienia.

Zauważmy, że przyporządkowanie obiektowi numeru klasy nie musi być jednoznaczne, np. w zadaniu klasyfikacji tekstów pisanych konieczne jest uwzględnienie faktu, że daną literę różne osoby piszą różnie, z tego też względu wydaje się, że najlepszym modelem klasyfikacji jest przyjęcie schematu probabilistycznego, w którym parametry opisujące obiekt jak i numery klas traktowane są jako realizacje zmiennych losowych.

W wielu praktycznych sytuacjach związanych z klasyfikacją różnych zjawisk zdarza się, że poszczególne klasy, do których należą pojawiające się losowo obiekty, są od siebie zależne. Sytuacja taka jest szczególnie typowa dla zadań rozpoznawania tekstów literowych oraz dla większości zadań związanych z rozpoznawaniem stanów charakterystycznych procesów technologicznych.

Rozpoznawanie sytuacji technologicznej, polegające na przetworzeniu dużej ilości informacji otrzymanych w wyniku pomiarów charakterystycznych parametrów danego procesu technologicznego, jest jednym z elementów w kompleksowej automatyzacji procesów przemysłowych i związane jest na ogół z zadaniem sterowania danym procesem. W przypadku gdy stan charakterystyczny w pewnej fazie procesu technologicznego zależy od stanu, jaki wystąpił w fazie poprzedniej, dany proces moż-

na traktować jako wieloetapowy, a odpowiednie rozpoznawanie stanów charakterystycznych oraz ewentualne wykorzystanie wyników rozpoznawania do sterowania tym procesem wymaga uwzględnienia zależności pomiędzy poszczególnymi etapami badanego procesu.

Za najprostszy probabilistyczny model takiej zależności uważać można sytuację, w której ciąg klas (stanów charakterystycznych), do których należą pojawiające się obiekty (pomiar) tworzy prosty łańcuch Markowa. Przykładem może tu być reakcja syntezy chemicznej, której produkt zależy od proporcji substratów oraz warunków syntezy. Gdy wymagane proporcje są realizacjami zmiennych losowych i zależą od wyników poprzednich faz (etapów) procesu syntezy, otrzymujemy łańcuch Markowa, w którym wynik kolejnego etapu reakcji chemicznej zależy od poprzedniego, a jednocześnie uzależnia od siebie wynik etapu następnego.

Temat niniejszej pracy sugeruje, że przedmiotem rozważań będą algorytmy rozpoznawania z uczeniem dla przypadku, gdy ciąg klas powiązany jest w prosty łańcuch Markowa oraz ich analiza porównawcza.

Omawiana w pracy problematyka zarówno w literaturze polskiej jak i w dostępnej literaturze światowej jest stosunkowo uboga. Niewielu autorów się nią zajmowało i zajmuje. Pierwszą pozycją, w sensie chronologicznym, która na ten temat się ukazała, to [9], w której na przykładzie rozpoznawania tekstów literowych przedstawiono wyniki eksperymentalnego badania algorytmu uwzględniającego "kontekst", to jest zależności pomiędzy klasyfikowanymi literami. Wyniki te okazały się obiecujące, a więc odpowiedź na postawione w tytule pytanie była twierdząca.

Podstawową pracą w zakresie interesującego nas problemu jest pozycja [29], w której po raz pierwszy został sformułowany matematyczny model zadania oraz uzyskany został na drodze analitycznej optymalny (w sensie minimum ryzyka) algorytm rozpoznawania przy przyjęciu modelu bayesowskiego dla łańcuchów Markowa. Przedstawiono tam również

pewne jego warianty i odmiany oraz dokonano eksperymentalnego ich badania. Jakkolwiek autor spojrział na zagadnienie w sposób bardzo ogólny, to jednak pominał zupełnym milczeniem sytuację niepełnej informacji o klasyfikowanych sytuacjach, dlatego pewnym uzupełnieniem tej pracy jest pozycja [3], w której nie tylko uzyskano bardzo zgrabną, rekurencyjną postać algorytmu rozpoznawania przy pełnej informacji probabilistycznej, ale sformułowano w niej również kilka różnych algorytmów rozpoznawania z uczeniem, które powstały przez uogólnienie na przypadek łańcuchów Markowa rezultatów opracowanych uprzednio dla niezależnego ciągu obrazów.

Z kolei w pracy [13] konstruuje się pewne modyfikacje znanych algorytmów rozpoznawania typu "najbliższy sąsiad", "najbliższa średnia", "α najbliższych sąsiadów" dla przypadku, gdy ciąg klas rozpoznawanych obiektów tworzy prosty łańcuch Markowa oraz dokonuje się ich eksperymentalnego badania. Wymienić również należy opracowanie [6], w którym uzyskano górne oszacowanie prawdopodobieństwa błędnej klasyfikacji bayesowskiego algorytmu rozpoznawania prostych łańcuchów Markowa. Oszacowanie to dotyczy najprostszego przypadku (dychotomia, obrazy jednowymiarowe), dlatego wyniki jakie autor zaprezentował, stanowiące niewatpliwie cenne osiągnięcie, mają raczej charakter fragmentaryczny i ogólnopoznawczy, a w mniejszym stopniu znaczenie praktyczne.

Na tym tle wyniki zaprezentowane w niniejszej pracy przedstawiają się następująco.

W rozdziale drugim przedstawiony został model matematyczny zadania oraz algorytm rozpoznawania prostych łańcuchów Markowa przy pełnej informacji probabilistycznej sformułowany w [3,29]. Zawarto ponadto w nim kilka spostrzeżeń autora dotyczących pewnych własności opisanego algorytmu.

W oparciu o nieparametryczne estymatory gęstości prawdopodobieństwa [18,20,22,27,31], z których niektóre wykorzystano już do sformułowania reguł decyzyjnych dla przypadku niezależnie pojawiających się

obiektów [12], w rozdziale trzecim skonstruowane zostały algorytmy rozpoznawania z uczeniem dla ciągu obiektów, których klasy tworzą proste łańcuchy Markowa. Przedstawione algorytmy (patrz też [14,15]) są analitycznymi algorytmami rozpoznawania z uczeniem i powstały przez zastąpienie w algorytmie wykorzystującym pełną informację probabilistyczną o klasyfikowanych sytuacjach [3], nieznanymi rozkładami ich estymatorami. Nie są to więc intuicyjne algorytmy rozpoznawania takie, jak na przykład "najbliższy sąsiad" czy "najbliższa średnia".

Zagadnieniem istotnym z punktu widzenia praktyki jest porównanie różnych sposobów uczenia rozpoznawania polegającego na wykorzystaniu przy wyznaczeniu nieznanymi rozkładów prawdopodobieństw, informacji statystycznej zawartej w tak zwanym ciągu uczącym utworzonym z pewnej liczby pomiarów i ich poprawnych klasyfikacji.

Taka właśnie analiza porównawcza różnych algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa była zasadniczym celem pracy, a jej opis i rezultaty zawarte zostały w dwóch następnych rozdziałach. Rozdział czwarty o charakterze opisowym zawiera informacje dotyczące podstaw i metodyki analizy porównawczej. Wyniki natomiast zaprezentowane zostały w rozdziale piątym. W pierwszej jego części przedstawiono rezultaty uzyskane na drodze analitycznej, które zawarto w trzech twierdzeniach dotyczących własności asymptotycznych badanych algorytmów. W drugiej natomiast jego części przedstawione zostały wyniki eksperymentalnego badania niektórych algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa.

Cele badań eksperymentalnych były następujące:

1. sprawdzenie celowości uwzględnienia w algorytmach markowskiej zależności pomiędzy klasami obiektów (zauważyć bowiem należy, że algorytmy rozpoznawania uwzględniające zależność typu Markowa są na ogół znacznie bardziej skomplikowane niż odpowiednie algorytmy nie uwzględniające takiej zależności),
2. dokonanie porównania badanych algorytmów,

3. zbadanie wpływu niektórych warunków rozpoznawania na jego jakość.

Na całość przedstawionych w pracy eksperymentalnych rezultatów składają się również pewne wyniki prezentowane już w [13,14].

Za podstawę eksperymentalnego porównania algorytmów przyjęto czas klasyfikacji pojedynczego obrazu oraz częstość błędnych rozpoznań, która - przy zerojedynkowej funkcji strat - stanowi przybliżoną ocenę prawdopodobieństwa błędnej klasyfikacji.

Na koniec, w rozdziale szóstym, przedstawiono pewien poglądowy i ilustracyjny przykład praktycznego zastosowania algorytmów rozpoznawania prostych łańcuchów Markowa oraz w rozdziale siódmym zawarte zostały wnioski i uwagi końcowe.

Programy realizujące zaprezentowane algorytmy rozpoznawania dla m.c. ODRA 1204 oraz kilka pomocniczych procedur wraz z ich opisami przedstawiono w dodatku.

2. SFORMUŁOWANIE PROBLEMU I ALGORYTM ROZPOZNAWANIA PRZY PEŁNEJ INFORMACJI PROBABILISTYCZNEJ

W części tej zostanie rozważony przypadek pełnej informacji probabilistycznej pozwalający posłużyć się przy określeniu reguły decyzyjnej klasycznymi metodami statystyki matematycznej. Wpływ zakłóceń stochastycznych uniemożliwia podjęcie całkowicie poprawnej i jednoznacznej decyzji, w konsekwencji poszukiwane są zatem algorytmy optymalne, gwarantujące ekstremalizację przyjętego kryterium jakości rozpoznawania.

Niech M oznacza liczbę klas, do których należą pojawiające się losowo obiekty. Klasyfikacji obiektów dokonuje się na podstawie wyników pomiarów wyróżnionych ich parametrów tworzących wektory w k -wymiarowej przestrzeni euklidesowej.

Wektor x_n oznacza pomiar n -tego rozpoznawanego obiektu; $x_n \in \mathbb{X}^k$ / k -wymiarowa przestrzeń wektorowa/, a j_n wynik jego poprawnej klasyfikacji / $j_n=1,2,\dots,M$ /.

Ciąg wektorów $\bar{x}_n \triangleq (x_1, x_2, \dots, x_n)$; $\bar{x}_n \in \bar{\mathbb{X}}_n$ / $k \cdot n$ -wymiarowa przestrzeń wektorowa/ oraz ciąg $\bar{j}_n \triangleq (j_1, j_2, \dots, j_n)$ oznaczają zatem odpowiednio ciąg pomiarów kolejno pojawiających się obiektów i ciąg numerów klas, do których te obiekty należą.

Obecnie będziemy rozpatrywać przypadek, gdy ciąg $\bar{j}_n = (j_1, j_2, \dots, j_n)$ jest realizacją prostego łańcucha Markowa *) zadanego wektorem prawdopodobieństw początkowych

$$p = (p_1, p_2, \dots, p_M)^T \quad (2.1)$$

oraz ciągiem macierzy przejścia

$$P_n = [p_{n,ij}] \quad , \quad i, j=1,2,\dots,M, \quad n=1,2,\dots \quad , \quad (2.2)$$

gdzie

$$\begin{aligned} p_i &= P(j_1=i) \quad , \quad i=1,2,\dots,M, \\ p_{n,ij} &= P(j_{n+1}=i/j_n=j) \quad , \quad i, j=1,2,\dots,M. \end{aligned} \quad (2.3)$$

*) Mówimy, że ciąg zmiennych losowych $\{J_n; n=1,2,\dots\}$ powiązany jest w prosty łańcuch Markowa /łańcuch Markowa pierwszego rzędu/, jeśli zachodzi [1] $P(J_n=j_n/J_{n-1}=j_{n-1})=P(J_n=j_n/J_{n-1}=j_{n-1}, \dots, J_1=j_1)$.

Zakładamy ponadto, że istnieją gęstości prawdopodobieństwa $f(x/i)$ w poszczególnych klasach obiektów $/i=1,2,\dots,M/$, a tym samym istnieje łączna warunkowa gęstość prawdopodobieństwa $f_n(\bar{x}_n/\bar{j}_n)$ oraz że ciąg pomiarów $\bar{x}_n=(x_1, x_2, \dots, x_n)$ jest "warunkowo niezależny" [29], to znaczy zachodzi:

$$f_n(\bar{x}_n/\bar{j}_n) = \prod_{s=1}^n f(x_s/j_s) . \quad (2.4)$$

Ustalenie algorytmu rozpoznawania polega na ustaleniu zależności

$$i_n = \Psi(x_1, x_2, \dots, x_n) \stackrel{\Delta}{=} \Psi_n(x_n) , \quad (2.5)$$

$$i=1,2,\dots,M , \quad n=1,2,\dots ,$$

która pomiar x_n zalicza do klasy i_n .

Przy pełnej informacji probabilistycznej, to jest przy znajomości wektora prawdopodobieństw początkowych p , macierzy prawdopodobieństw przejścia P_n i wszystkich warunkowych gęstości prawdopodobieństwa $f(x/i)$ /model bayesowski/, dla wybranej funkcji $L(i,j)$ oceniającej stratę jaką ponosimy z powodu zaklasyfikowania pomiaru z klasy j do klasy i , wyznaczyć można analitycznie [3,29] ciąg optymalnych algorytmów rozpoznawania /ciąg optymalnych reguł decyzyjnych Bayesa/

$$i_n = \Psi_n^*(x_n) \quad (2.6)$$

minimizując średnie ryzyka na kolejnych etapach klasyfikacji $/n=1,2,\dots/$

$$R_n[\Psi_n(x_n)] \stackrel{\Delta}{=} E_{\bar{x}_n, \bar{j}_n} \left\{ L(i_n, j_n) \right\} =$$

$$= \sum_{j_n=1}^M \int_{\bar{x}_n} L[\Psi_n(x_n)=i_n, j_n] \cdot p_{j_n}^{(n)} \cdot \bar{f}_n(\bar{x}_n/j_n) d\bar{x}_n , \quad (2.7)$$

gdzie

$$\delta(j, \bar{x}_n) \stackrel{\Delta}{=} p_{j_n}^{(n)} \cdot \bar{f}_n(\bar{x}_n/j_n) = f(x_n/j_n) \sum_{j_{n-1}=1}^M p_{n-1, j_n j_{n-1}} \cdot p_{j_{n-1}}^{(n-1)} \cdot$$

$$\cdot \bar{f}_{n-1}(\bar{x}_{n-1}/j_{n-1}) , \quad (2.8)$$

$$\delta(j, x_1) \stackrel{\Delta}{=} p_j \cdot f(x_1/j) .$$

W tym celu wystarczy minimizować ryzyka warunkowe na poszczególnych etapach klasyfikacji $/n=1,2,\dots /$

$$r_n(\Psi_n(x_n), \bar{x}_n) = \frac{1}{\delta(\bar{x}_n)} \sum_{j_n=1}^M L[\Psi_n(x_n)=i_n, j_n] \cdot \delta(j_n, \bar{x}_n), \quad (2.9)$$

gdzie

$$\delta(\bar{x}_n) = \sum_{j=1}^M \delta(j, \bar{x}_n) \quad . \quad (2.10)$$

Na podstawie (2.9) i pomijając $\delta(\bar{x}_n)$, która nie wpływa na wynik klasyfikacji, otrzymujemy zatem następujący optymalny algorytm rozpoznawania na n -tym etapie klasyfikacji $/n=1,2,\dots /$:

$$\begin{aligned} \Psi_n^*(x_n) = i_n, \text{ jeśli} \\ r_n(i_n, \bar{x}_n) < r_n(l, \bar{x}_n), \quad l=1,2,\dots, i_{n-1}, i_{n+1}, \dots, M \quad . \end{aligned} \quad (2.11)$$

W szczególności dla zerojedynkowej funkcji strat

$$L(i, j) = \begin{cases} 0 & \text{dla } i=j \\ 1 & \text{dla } i \neq j \end{cases} \quad (2.12)$$

algorytm rozpoznawania minimizujący średnie ryzyko /średnie prawdopodobieństwo błędnej klasyfikacji/ przybiera postać [3]:

$$\Psi_n^*(x_n) = i_n, \text{ jeśli} \quad (2.13)$$

$$\delta(i_n, \bar{x}_n) > \delta(l, \bar{x}_n), \quad l=1,2,\dots, i_{n-1}, i_{n+1}, \dots, M, \quad n=1,2,\dots \quad .$$

Na podstawie (2.8) otrzymujemy

$$\delta(i, \bar{x}_n) = f(x_n/i) \sum_{j=1}^M p_{n-1,ij} \cdot \delta(j, \bar{x}_{n-1}) \quad . \quad (2.14)$$

Stąd wektor

$$\delta_n = (\delta(1, \bar{x}_n), \delta(2, \bar{x}_n), \dots, \delta(M, \bar{x}_n))^T \quad (2.15)$$

spełnia rekurencyjną zależność [3]

$$\delta_n = D(x_n) \cdot P_{n-1} \cdot \delta_{n-1}, \quad (2.16)$$

$$\delta_1 = D(x_1) \cdot p,$$

gdzie $D(x_n)$ jest macierzą diagonalną

$$D(x_n) = \begin{bmatrix} f(x_n/1) & & & \\ & f(x_n/2) & & \\ & 0 & \ddots & \\ & & & f(x_n/M) \end{bmatrix} \quad (2.17)$$

W dalszym ciągu rozważać będziemy wyłącznie stacjonarne, jednorodnie i nieprzywiedlne proste łańcuchy Markowa, ograniczymy się zatem do tej klasy zależności, w której:

1. $p_{ij} = P(j_{n+1}=i/j_n=j)$ dla każdego n /jednorodność/,
2. $p_{ij} > 0$ /nieprzywiedlność/,
3. $p = P \cdot p$ - p jest wektorem własnym macierzy P odpowiadającym wartości własnej 1 /stacjonarność/.

Zauważmy, że w pewnych sytuacjach może się zdarzyć

$$\lim_{n \rightarrow \infty} \|\delta_n\| = 0 \quad (2.19)$$

Przypadek taki będzie miał miejsce, gdy dla pewnej normy $\|\cdot\|$ [3]

$$\bigwedge_{x_n} \|D(x_n) \cdot P\| < 1 \quad (2.20)$$

W szczególności, jeśli wprowadzić normę

$$\|\delta_n\| = \sum_{i=1}^M \delta(i, \bar{x}_n) \quad (2.21)$$

spełnienie warunku

$$\bigwedge_i \max_x f(x/i) \sum_{j=1}^M p_{ij} < 1 \quad (2.22)$$

zapewnia zbieżność $\|\delta_n\|$ do zera [3, 4, 19].

W takim przypadku porównanie poszczególnych składowych wektora δ_n , a więc w konsekwencji podjęcie decyzji, może być trudne dla dużych n . Przykładowo, dla obrazów jednowymiarowych / $k=1$ /, przyjmując we wszystkich klasach rozkłady normalne $f(x/i) = N(m_i, \delta_i)$ / $i=1, 2, \dots, M$ / warunek (2.22) przyjmuje postać:

$$\delta_i > \frac{1}{\sqrt{2\pi}} \sum_{j=1}^M p_{ij}, \quad i=1, 2, \dots, M \quad (2.23)$$

lub szacując prawdopodobieństwa przejścia p_{ij}

$$\delta_i > \frac{M}{\sqrt{2\pi}} \quad , \quad i=1,2,\dots,M \quad . \quad (2.24)$$

Dla przypadku dychotomii spełnienie zatem warunku

$$\delta_1, \delta_2 > 0.8$$

zapewnia zbieżność $\|\delta_n\|$ do zera.

Wykonując podobne obliczenia dla innych rozkładów łatwo się przekonać, że warunek (2.22) jest bardzo łagodny i dla "typowych" postaci różnych konkretnych rozkładów jest on spełniony /przeprowadzono również badania eksperymentalne mające na celu określenie prędkości zbieżności procedury rekurencyjnej (2.16). Przykładowo dla danych $k=1, M=2$, $f(x/1)=N(3,16)$, $f(x/2)=N(12,18)$, $p_{11}=0.2$, $p_{22}=0.6$ po około 620 iteracjach współrzędne wektora δ_n równe były zeru maszynowemu, a więc praktycznie nierozróżnialne/.

Istotną przeto z praktycznego punktu widzenia jest taka konstrukcja algorytmu (2.13), która pozwoliłaby uniknąć niepożądanego zbieżności wektora funkcji decyzyjnych (2.15). W tym celu proponujemy pewną prostą /i zupełnie naturalną - patrz wzory (2.9), (2.10)/modyfikację algorytmu (2.13) polegającą na normowaniu na jeden poszczególnych funkcji decyzyjnych (2.14) na kolejnych etapach klasyfikacji

$$\delta(i, \bar{x}_n) = \frac{1}{\sum_{j=1}^M \delta(j, \bar{x}_n)} f(x_n/i) \cdot \sum_{j=1}^M p_{ij} \cdot \delta(j, \bar{x}_{n-1}) \quad (2.25)$$

$$i=1,2,\dots,M, \quad n=1,2,\dots$$

i teraz

$$\|\delta_n\| = \sum_{j=1}^M \delta(j, \bar{x}_n) = 1 \quad . \quad (2.26)$$

Oczywiście modyfikacja ta nie ma wpływu na wynik klasyfikacji i z tego też względu podstawą do dalszych rozważań teoretycznych będzie algorytm (2.13) z funkcją decyzyjną (2.14) a nie (2.25). Pamiętać jedynie będziemy, że modyfikacja ta ma istotne znaczenie w momencie technicznej realizacji algorytmu (2.13) jak i dalszych algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa.

Zauważmy ponadto [3], że w poszczególnych momentach klasyfikacji

kształt hiperpowierzchni rozdzielających określony przez funkcje $f(x/i)$ nie ulega zmianie. Np., jeśli dla $M=2$ równanie hiperpowierzchni rozdzielającej jest

$$\sum_{j=1}^2 p_{1j} \cdot \delta(j, \bar{x}_{n-1}) \cdot f(x_n/1) = f(x_n/2) \sum_{j=1}^2 p_{2j} \cdot \delta(j, \bar{x}_{n-1}) \quad , \quad (2.27)$$

czyli

$$\alpha_n + F(x_n) = 0 \quad , \quad (2.28)$$

gdzie:

$$\alpha_n = \log \frac{\sum_{j=1}^2 p_{1j} \cdot \delta(j, \bar{x}_{n-1})}{\sum_{j=1}^2 p_{2j} \cdot \delta(j, \bar{x}_{n-1})} \quad , \quad (2.29)$$

$$F(x) = \log \frac{f(x/1)}{f(x/2)} \quad ,$$

to w poszczególnych momentach klasyfikacji ulega zmianie α_n niezależna od x , co powoduje jedynie pewne przesunięcie hiperpowierzchni rozdzielającej z zachowaniem jej kształtu.

Na koniec rozważmy jeszcze pewien interesujący przypadek, w którym macierz przejścia jest następującej, szczególnej postaci:

$$P = \begin{bmatrix} \alpha_1 & \alpha_1 & \dots & \alpha_1 \\ \alpha_2 & \alpha_2 & \dots & \alpha_2 \\ \dots & \dots & \dots & \dots \\ \alpha_M & \alpha_M & \dots & \alpha_M \end{bmatrix} \quad , \quad \sum_{i=1}^M \alpha_i = 1 \quad , \quad (2.30)$$

co ma miejsce, gdy ciąg klas jest niezależny [10].

Dla łańcucha stacjonarnego

$$p_i = \alpha_i \quad , \quad i=1, 2, \dots, M \quad . \quad (2.31)$$

Równanie hiperpowierzchni rozdzielającej klasę i -tą od j -tej ma postać:

$$f(x_n/i) \cdot \sum_{k=1}^M p_{ik} \delta(k, \bar{x}_{n-1}) = f(x_n/j) \cdot \sum_{k=1}^M p_{jk} \delta(k, \bar{x}_{n-1}) \quad , \quad (2.32)$$

$$i, j=1, 2, \dots, M \quad , \quad i \neq j \quad .$$

Z (2.30) i (2.31) otrzymujemy

$$f(x_n/i) \cdot \alpha_i \sum_{k=1}^M \delta(k, \bar{x}_{n-1}) = f(x_n/j) \cdot \alpha_j \sum_{k=1}^M \delta(k, \bar{x}_{n-1}) \quad , \quad (2.33)$$

$$f(x_n/i) \cdot p_i = f(x_n/j) \cdot p_j , \quad (2.34)$$

$$i, j = 1, 2, \dots, M , \quad i \neq j .$$

Dostaliśmy zatem funkcję rozdzielającą bayesowskiego klasyfikatora dla przypadku niezależnie pojawiających się obrazów. Takiego wyniku oczywiście należało się spodziewać.

$$f(x_n/i) \cdot p_i = f(x_n/j) \cdot p_j , \quad (2.34)$$

$$i, j = 1, 2, \dots, M , \quad i \neq j .$$

Dostaliśmy zatem funkcję rozdzielającą bayesowskiego klasyfikatora dla przypadku niezależnie pojawiających się obrazów. Takiego wyniku oczywiście należało się spodziewać.

3. ALGORYTMY ROZPOZNAWANIA Z UCZENIEM DLA PROSTYCH ŁAŃCUCHÓW MARKOWA

W praktycznych zadaniach rozpoznawania najczęściej prawdopodobieństwa p , P oraz gęstości $f(x/i)$ nie są znane, a jedyna dostępna informacja zawarta jest w ciągu uczącym

$$\{S\}_N \stackrel{\Delta}{=} (x_1, j_1), (x_2, j_2), \dots, (x_N, j_N), \quad (3.1)$$

gdzie j_i oznacza wynik poprawnej klasyfikacji obiektu reprezentowanego pomiarem x_i . Z uwagi na ograniczenie naszych rozważań do przypadku stacjonarnych i jednorodnych łańcuchów Markowa, wystarczy rozpatrywać jeden, dostatecznie długi, ciąg uczący, to znaczy jedną tylko realizację dyskretnego procesu stochastycznego $\{X_n, J_n\}$.

Zagadnienie klasyfikacji obrazów polega w tym przypadku na określeniu poprzez zgromadzenie informacji statystycznej o rozkładach w poszczególnych klasach i odpowiednich prawdopodobieństwach uzyskanej w oparciu o ciąg uczący (3.1) /jest to tak zwany proces uczenia/, a następnie na realizacji reguły decyzyjnej

$$i = \Psi(\{S\}_N, x). \quad (3.2)$$

Różne sposoby wykorzystania ciągu uczącego prowadzą do różnych sposobów uczenia, a w konsekwencji do różnych algorytmów rozpoznawania (3.2).

Wykorzystując skonstruowane na podstawie ciągu uczącego rozmaite nieparametryczne estymatory gęstości $f_N(x/i)$ w miejsce rozkładów rzeczywistych $f(x/i)$, a w miejsce nieznanymi prawdopodobieństw p_i i p_{ij} następujące ich oszacowania:

$$p_{i,N} = \frac{N_i}{N} \quad , \quad p_{ij,N} = \frac{N_{ij}}{N_j} \quad , \quad (3.3)$$

gdzie N oznacza długość ciągu uczącego, N_i - liczbę pomiarów w ciągu uczącym z klasy i , a N_{ij} - liczbę sytuacji, w których pomiar z klasy i -tej występuje w ciągu uczącym bezpośrednio po pomiarze z klasy j -tej, uzyskuje się, dla różnych sposobów estymacji gęstości, różne analitycz-

*) Dokładniej $p_{i,N}$ jest oszacowaniem prawdopodobieństwa całkowitego łańcucha Markowa - c_i , ale ze względu na założoną stacjonarność mamy

$$p_i = c_i, \quad i=1,2,\dots,M.$$

ne algorytmy rozpoznawania z uczeniem uwzględniające zależność typu Markowa pomiędzy poszczególnymi klasami obiektów. Przykładem tak uzyskanego algorytmu jest zmodyfikowany algorytm "najmniejszy przedział" sformułowany w [3].

Rozważania nasze ograniczymy do przypadku zerojedynkowej funkcji strat, dla którego po zastąpieniu rozkładów rzeczywistych ich empirycznymi oszacowaniami, zależność (2.14) przybiera postać:

$$\delta_N(i, \bar{x}_n) = f_N(x_n/i) \cdot \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}), \quad (3.4)$$

zaś algorytm rozpoznawania jest następujący:

$$\Psi(\{s\}_N, \bar{x}_n) \triangleq \Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.5)$$

$$\delta_N(i_n, \bar{x}_n) > \delta_N(r, \bar{x}_n), \quad r=1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M, \quad n=1, 2, \dots$$

W dalszym ciągu sformułujemy różne konkretne postacie tego algorytmu, a następnie krótko przypomnimy znane już algorytmy rozpoznawania [3, 13] prostych łańcuchów Markowa skonstruowane również według powyższego schematu.

3.1. Zmodyfikowany algorytm "h(N)-ty najbliższy sąsiad" (M-h(N)NN).

W przypadku, gdy klasy obiektów są stochastycznie niezależne, algorytm "h(N)-ty najbliższy sąsiad" zalicza pomiar x do tej klasy, w której jego odległość od h(N)-tego najbliższego obrazu uczącego jest najmniejsza [11]. W pracy [12] dla niezależnego ciągu obiektów w oparciu o estymator gęstości prawdopodobieństwa podany przez Loftsgaardena i Quesenberry'ego uzyskano uogólnienie algorytmu h(N)NN na przypadek dowolnej liczby klas i dowolnej postaci funkcji strat oraz wykazano, że uzyskana reguła decyzyjna jest zbieżna do reguły Bayesa według prawdopodobieństwa i średniej z kwadratem, a ryzyko do ryzyka Bayesa.

Estymator gęstości sformułowany przez Loftsgaardena i Quesenberry'ego ma postać [20]

$$f_N(x/i) = \frac{h(N_i)-1}{C \cdot N_i} \frac{1}{\{d[x; h(N_i)]\}^k}, \quad (3.6)$$

gdzie

$$C = \frac{2 \cdot \sqrt{\pi}}{k \cdot \Gamma\left(\frac{k}{2}\right)},$$

$d[x; h(N_i)]$ - odległość euklidesowa pomiędzy klasyfikowanym pomiarem x a $h(N_i)$ -tym najbliższym pomiarem w ciągu uczącym z klasy i .

Estymator ten jest zbieżny według prawdopodobieństwa do $f(x/i)$ w punktach jej ciągłości, jeśli

$$h(N_i) > 0, \quad \lim_{N_i \rightarrow \infty} h(N_i) = \infty, \quad \lim_{N_i \rightarrow \infty} \frac{h(N_i)}{N_i} = 0. \quad (3.7)$$

Warunki (3.7) spełnia przykładowo ciąg $h(N_i) = \sqrt{N_i}$.

Stosując oszacowanie (3.6) do oceny nieznannej gęstości prawdopodobieństwa dla przypadku, gdy klasy obiektów tworzą prosty łańcuch Markowa sformułujemy zmodyfikowany algorytm "h(N)-ty najbliższy sąsiad". Wstawiając powyższy estymator do (3.4) otrzymujemy

$$\delta_N(i, \bar{x}_n) = \frac{h(N_i) - 1}{C \cdot N_i} \frac{1}{\{d[x_n; h(N_i)]\}^k} \sum_{j=1}^M p_{ij, N} \cdot \delta_N(j, \bar{x}_{n-1}). \quad (3.8)$$

Oznaczając z kolei

$$\alpha_{i, n} \triangleq N_i / [h(N_i) - 1] \sum_{j=1}^M p_{ij, N} \cdot \delta_N(j, \bar{x}_{n-1}) \quad (3.9)$$

i pomijając C , które nie wpływa na wynik klasyfikacji, otrzymujemy z (3.8) następującą funkcję decyzyjną:

$$\delta_N(i, \bar{x}_n) = 1 / \alpha_{i, n} \cdot \{d[x_n; h(N_i)]\}^k. \quad (3.10)$$

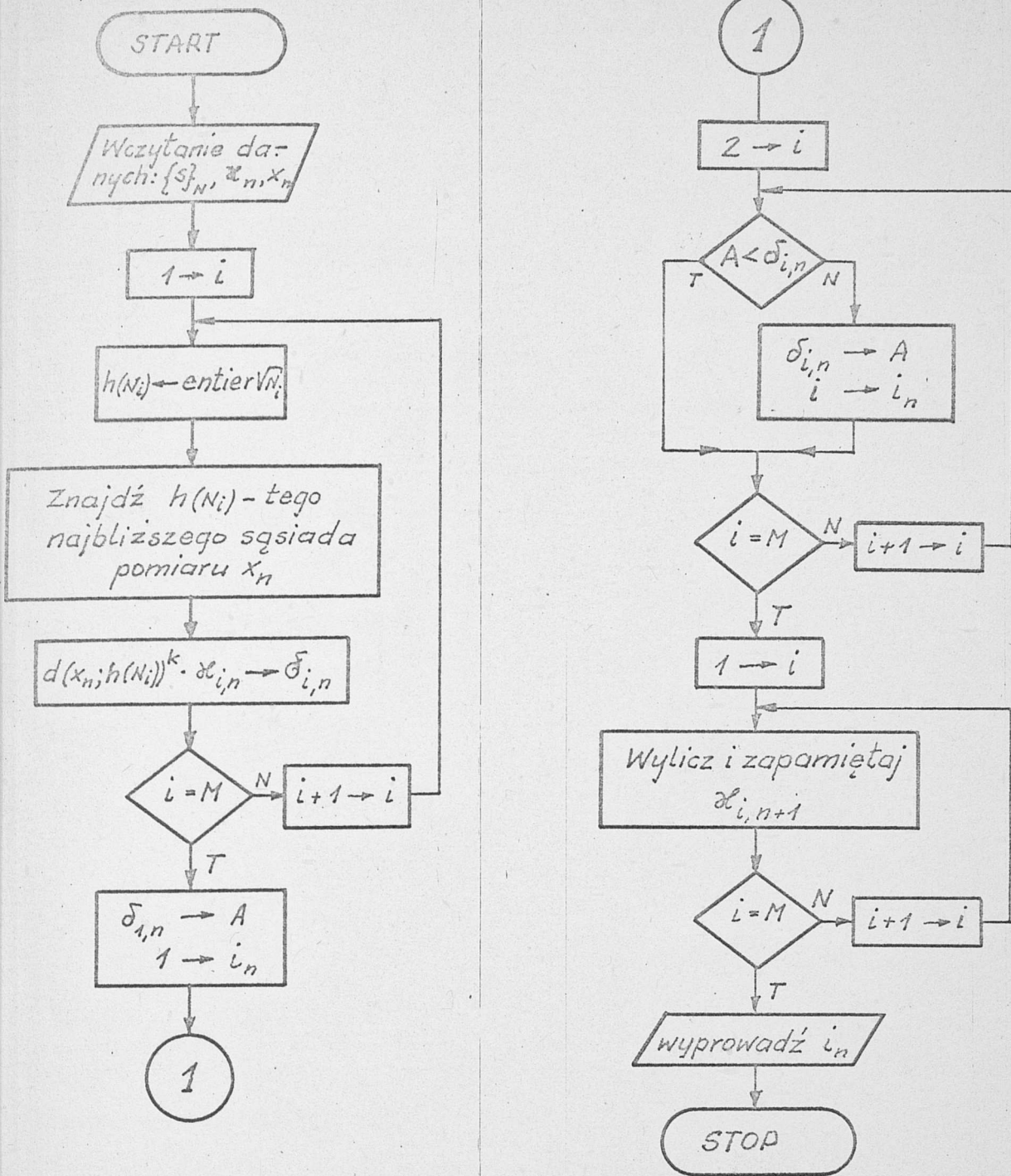
Algorytm rozpoznawania (3.5) ma zatem postać:

$$\Psi_{N, n}(x_n) = i_n, \text{ jeśli}$$

$$\alpha_{i_n, n} \{d[x_n; h(N_{i_n})]\}^k < \alpha_{r, n} \{d[x_n; h(N_r)]\}^k \quad (3.11)$$

$$\text{dla } r=1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M, \quad n=1, 2, \dots$$

Współczynniki wagowe $\alpha_{i, n}$ dla klasy i -tej w kolejnych krokach klasyfikacji / $n=2, \dots$ / wyznaczyć można rekurencyjnie



Rys. 3.1. Sieć działań dla algorytmu M-h(N)NN .

$$x_{i,n} = N_i \left\{ [h(N_i) - 1] \sum_{j=1}^M p_{ij,N} \frac{1}{x_{i,n-1} \{d[x_{n-1}; h(N_j)]\}^k} \right\}^{-1} \quad (3.12)$$

$$x_{i,1} = \frac{N}{h(N_i) - 1} \cdot$$

Uwzględnienie zależności Markowa pomiędzy poszczególnymi klasami obiektów uwidacznia się w podanym algorytmie poprzez współczynniki wa-

gowe, które zmieniają się dla każdej klasy wraz z n , a więc są różne na różnych etapach klasyfikacji.

W dalszym ciągu algorytm (3.11) będziemy nazywać krótko $M-h(N)NN$, a jego wersję niezmodyfikowaną /dla obrazów pojawiających się niezależnie/ $[12]-h(N)NN$.

Uproszczoną sieć działań algorytmu $M-h(N)NN$ przedstawia rysunek 3.1, a odpowiedni program realizujący ten algorytm zamieszczono w dodatku / procedura $MHNNN$ /.

3.2. Algorytm rozpoznawania wykorzystujący nieparametryczny estymator gęstości Parzena.

W pracy [27] dowodzi się, że ciąg

$$f_N(x/i) = \frac{1}{N_i [h(N_i)]^k} \sum_{l=1}^{N_i} K \left[\frac{x-x_{l,i}}{h(N_i)} \right], \quad (3.13)$$

gdzie $x_{1,i}, x_{2,i}, \dots, x_{N_i,i}$ jest ciągiem niezależnych pomiarów uczących z klasy i , jest zbieżny według prawdopodobieństwa do funkcji $f(x/i)$ w punktach jej ciągłości, jeśli

$$\sup_x \|K(x)\| < \infty, \quad \sup_x \|x \cdot K(x)\| < \infty, \quad \int_x K(x) dx = 1, \quad \int_x K^2(x) dx < \infty,$$

oraz jeśli

$$\lim_{N_i \rightarrow \infty} h(N_i) = 0, \quad \lim_{N_i \rightarrow \infty} N_i \cdot h(N_i) = \infty.$$

(3.14)

Wykorzystując estymator gęstości prawdopodobieństwa (3.13) celem sformułowania funkcji decyzyjnej (3.4) otrzymujemy

$$\begin{aligned} \delta_N(i, \bar{x}_n) &= \frac{1}{N_i [h(N_i)]^k} \sum_{l=1}^{N_i} K \left[\frac{x_n - x_{l,i}}{h(N_i)} \right] \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) = \\ &= \mathcal{X}_{i,n} \sum_{l=1}^{N_i} K \left[\frac{x_n - x_{l,i}}{h(N_i)} \right], \end{aligned} \quad (3.15)$$

gdzie

$$\mathcal{X}_{i,n} \triangleq \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) / N_i [h(N_i)]^k \quad (3.16)$$

jest współczynnikiem wagowym dla i -tej klasy wynikającym z uwzglę-

dnienia zależności typu Markowa pomiędzy poszczególnymi klasami obiektów.

Algorytm rozpoznawania (3.5) przybiera, dla przyjętego sposobu estymacji gęstości, następującą postać:

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli}$$

$$\mathcal{X}_{i_n,n} \sum_{l=1}^{N_{i_n}} K \left[\frac{x_n - x_{l,i_n}}{h(N_{i_n})} \right] > \mathcal{X}_{r,n} \sum_{l=1}^{N_r} K \left[\frac{x_n - x_{l,r}}{h(N_r)} \right] \quad (3.17)$$

dla $r = 1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M$.

Współczynniki wagowe, podobnie jak poprzednio, wyznacza się rekurencyjnie

$$\mathcal{X}_{i,n} = \sum_{j=1}^M \left\{ p_{ij,N} \cdot \mathcal{X}_{j,n-1} \sum_{l=1}^{N_j} K \left[\frac{x_{n-1} - x_{l,j}}{h(N_j)} \right] \right\} / N_i \cdot [h(N_i)]^k, \quad (3.18)$$

$$\mathcal{X}_{i,1} = \frac{1}{N \cdot [h(N_i)]^k}.$$

W tym przypadku współczynniki wagowe są również niestacjonarne i wymagają wyliczenia w każdym kroku oddzielnie.

Dla różnych funkcji $K(x)$ i różnych ciągów $h(N_i)$ otrzymuje się różne konkretne postacie algorytmu (3.17). W szczególności dla $k=1$ /przypadek obrazów jednowymiarowych/ przy

$$K \left[\frac{x - x_{l,i}}{h(N_i)} \right] = \frac{1}{\sqrt{2\pi}} \exp \left[- \left(\frac{x - x_{l,i}}{2 h(N_i)} \right)^2 \right] \quad (3.19)$$

otrzymujemy algorytm, dla którego w trakcie klasyfikacji nie zachodzi konieczność pamiętania ciągu uczącego (3.1).

Przekształcając funkcję (3.19) i rozwijając ją w szereg Taylora otrzymujemy [31]

$$\sum_{l=1}^{N_i} K \left[\frac{x - x_{l,i}}{h(N_i)} \right] = \sum_{l=1}^{N_i} \frac{1}{\sqrt{2\pi}} \exp \left[- \frac{x^2}{2 [h(N_i)]^2} \right] \cdot \exp \left[- \frac{x_{l,i}^2}{2 [h(N_i)]^2} \right] \cdot \left[1 + \frac{x \cdot x_{l,i}}{1! [h(N_i)]^2} + \frac{x^2 \cdot x_{l,i}^2}{2! [h(N_i)]^4} + \dots \right] =$$

$$\begin{aligned}
&= \sum_{l=1}^{N_i} \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{x^2}{2[h(N_i)]^2} \right] \exp \left[-\frac{x_{l,i}^2}{2[h(N_i)]^2} \right] \sum_{s=0}^{\infty} \frac{x^s \cdot x_{l,i}^s}{s! [h(N_i)]^{2s}} = \\
&= \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{x^2}{2[h(N_i)]^2} \right] \sum_{s=0}^{\infty} x^s \sum_{l=1}^{N_i} \frac{1}{s! [h(N_i)]^{2s}} \cdot x_{l,i}^s \cdot \\
&\exp \left[-\frac{x_{l,i}^2}{2[h(N_i)]^2} \right] = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{x^2}{2[h(N_i)]^2} \right] \sum_{s=0}^{\infty} c_{s,i} \cdot x^s,
\end{aligned}$$

przy czym

$$c_{s,i} = \frac{1}{s! [h(N_i)]^{2s}} \sum_{l=1}^{N_i} x_{l,i}^s \cdot \exp \left[-\frac{x_{l,i}^2}{2[h(N_i)]^2} \right]. \quad (3.20)$$

Funkcja decyzyjna (3.15) przybiera wówczas postać

$$\delta_{N(i, \bar{x}_n)} = \mathcal{X}_{i,n} \exp \left[-\frac{x_n^2}{2[h(N_i)]^2} \right] \sum_{s=0}^{\infty} c_{s,i} \cdot x_n^s, \quad (3.21)$$

gdzie

$$\mathcal{X}_{i,n} = \sum_{j=1}^M \left\{ p_{ij,N} \cdot \mathcal{X}_{j,n-1} \exp \left[-\frac{x_{n-1}^2}{2[h(N_j)]^2} \right] \sum_{s=0}^{\infty} c_{s,j} \cdot x_{n-1}^s \right\} / N_i \cdot h(N_i), \quad (3.22)$$

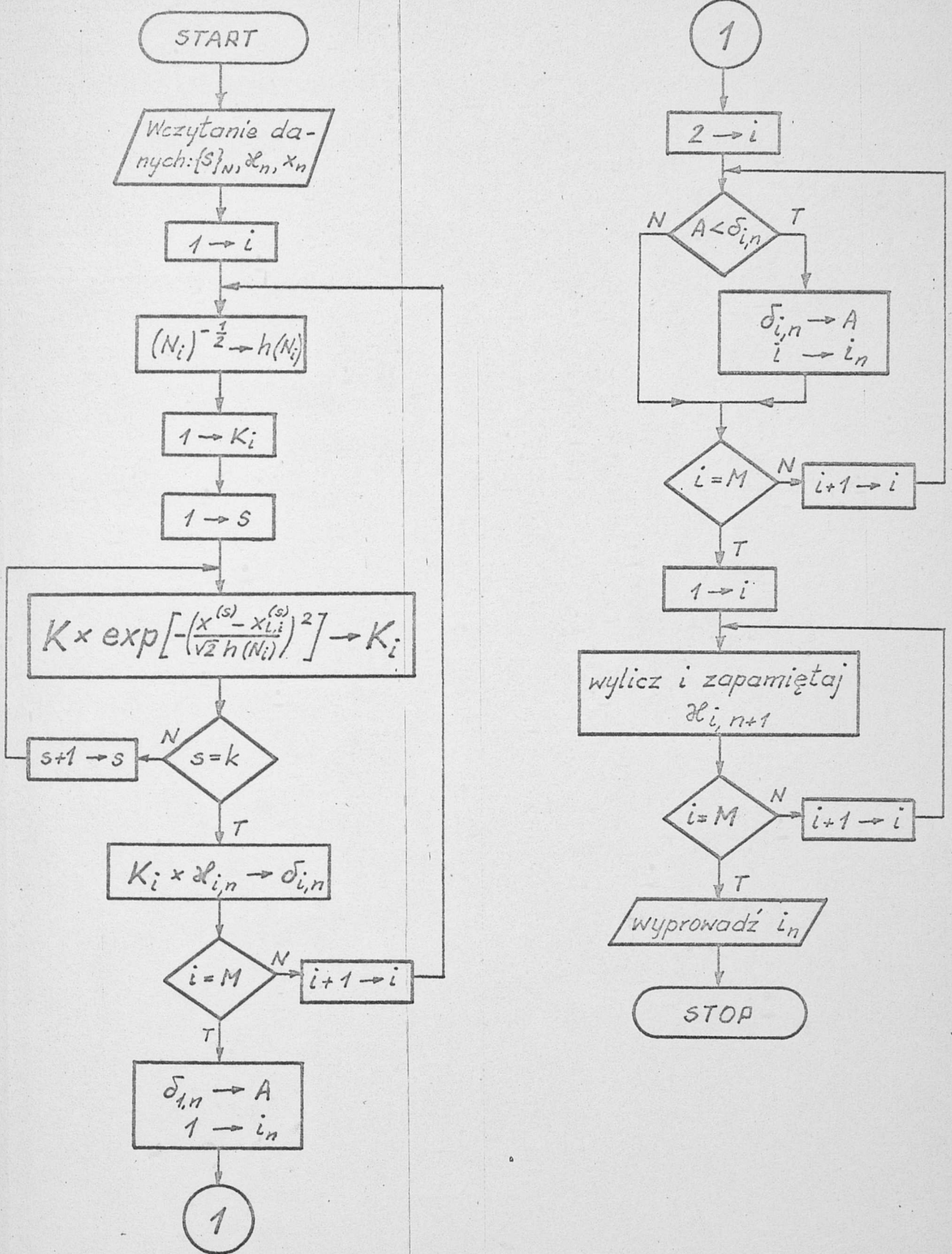
a reguła decyzyjna (3.17) zalicza pomiar x_n do tej klasy i_n , dla której wyrażenie (3.21) jest największe.

W przypadku tego algorytmu zamiast ciągu uczącego (3.1) wystarczy pamiętać współczynniki $c_{s,i}$ (3.20) wyznaczone na podstawie tego ciągu w procesie uczenia. Liczbę tych współczynników dla każdej klasy obiektów/ $i=1,2,\dots,M$ / wybieramy arbitralnie w zależności od założonej dokładności rozwinięcia.

W dalszym ciągu algorytm (3.17) będziemy nazywać krótko M-PAR, a jego wersję niezmodyfikowaną /dla obrazów pojawiających się niezależnie/ [12] - PAR.

Uproszczoną sieć działań algorytmu wykorzystującego estymator gęstości Parzena przedstawia rysunek 3.2. Przyjęto w nim

$$h(N_i) = \binom{N_i}{i}^{-1/2},$$



Rys. 3.2 Sieć działań algorytmu M-PAR.

$$K\left(\frac{x-x_{1,i}}{h(N_i)}\right) = \prod_{s=1}^k \left(\frac{1}{\sqrt{2\pi}}\right)^k \exp\left[-\left(\frac{x^{(s)}-x_{1,i}^{(s)}}{\sqrt{2}h(N_i)}\right)^2\right], \quad (3.23)$$

a odpowiedni program zamieszczono w dodatku /procedura MPAR /.

3.3. Algorytm rozpoznawania z nieparametrycznym estymatorem gęstości Murthy'ego.

Obecnie do sformułowania algorytmu rozpoznawania wykorzystamy następujący estymator gęstości prawdopodobieństwa [22]:

$$f_N(x/i) = \frac{B_N}{N_i} \sum_{l=1}^{N_i} K(B_N(x_{1,i} - x)). \quad (3.24)$$

Jest on zbieżny według prawdopodobieństwa do funkcji $f(x/i)$ w punktach jej ciągłości, jeśli

$$K(x) \geq 0, \quad K(x) = K(-x), \quad \lim_{|x| \rightarrow \infty} x \cdot K(x) = 0, \quad \int_x K(x) dx = 1$$

oraz jeśli

$$\lim_{N \rightarrow \infty} B_N = \infty, \quad \lim_{N \rightarrow \infty} \frac{B_N}{N} = 0. \quad (3.25)$$

Dla estymatora (3.24) funkcja decyzyjna (3.4) ma postać

$$\begin{aligned} \delta_N(i, \bar{x}_n) &= \frac{B_N}{N_i} \sum_{l=1}^{N_i} K(B_N(x_{1,i} - x_n)) \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) = \\ &= \mathcal{K}_{i,n} B_N \sum_{l=1}^{N_i} K(B_N(x_{1,i} - x_n)), \end{aligned} \quad (3.26)$$

gdzie

$$\mathcal{K}_{i,n} = \frac{1}{N_i} \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}), \quad (3.27)$$

a algorytm rozpoznawania jest w tym przypadku następujący:

$$\Psi_{N,n}(x_n) = i_n, \quad \text{jeśli}$$

$$\mathcal{K}_{i_n,n} B_N \sum_{l=1}^{N_{i_n}} K(B_N(x_{1,i_n} - x_n)) > \mathcal{K}_{r,n} B_N \sum_{l=1}^{N_r} K(B_N(x_{1,r} - x_n)) \quad (3.28)$$

dla $r = 1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M$.

Współczynniki wagowe wylicza się, tak jak w przypadkach poprzednich, rekurencyjnie

$$\mathcal{X}_{i,n} = \frac{1}{N_i} \sum_{j=1}^M p_{ij,N} \cdot \mathcal{X}_{i,n-1} \cdot B_N \sum_{l=1}^N K(B_N(x_{l,j} - x_{n-1})), \quad (3.29)$$

$$\mathcal{X}_{i,1} = \frac{1}{N}$$

i wyliczenie to należy powtarzać na wszystkich etapach klasyfikacji, z uwagi na niestacjonarny charakter współczynników $\mathcal{X}_{i,n}$.

Wybierając różne funkcje $K(x)$ i różne ciągi B_N otrzymuje się różne odmiany algorytmu (3.28). Przykładowo kładąc [18]

$$K(B_N(x_{l,i} - x)) = \frac{1}{\pi^k} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x^{(s)})}{B_N^{(s)}(x_{l,i}^{(s)} - x^{(s)})}, \quad (3.30)$$

$$B_N = \prod_{s=1}^k B_N^{(s)},$$

gdzie $x_{1,i}^{(s)}, x_{2,i}^{(s)}, \dots, x_{N_i,i}^{(s)}$ jest ciągiem s -tych składowych $/s=1, 2, \dots, k/$ niezależnych obrazów uczących z klasy $i /i=1, 2, \dots, M/$, $x^{(s)}$ - s -tą składową klasyfikowanego pomiaru, a ciąg wektorów

$$B_N = (B_N^{(1)}, B_N^{(2)}, \dots, B_N^{(s)}, \dots, B_N^{(k)})$$

spełnia warunki

$$\lim_{N \rightarrow \infty} \min_s \{B_N^{(s)}\} = \infty, \quad \lim_{N \rightarrow \infty} \frac{1}{N} \prod_{s=1}^k B_N^{(s)} = 0, \quad (3.31)$$

dostajemy następującą funkcję decyzyjną:

$$\begin{aligned} \delta_N(i, \bar{x}_n) &= \frac{1}{N_i} \sum_{l=1}^{N_i} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x_n^{(s)})}{x_{l,i}^{(s)} - x_n^{(s)}} \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) = \\ &= \mathcal{X}_{i,n} \sum_{l=1}^{N_i} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x_n^{(s)})}{x_{l,i}^{(s)} - x_n^{(s)}}, \end{aligned} \quad (3.32)$$

dla $r = 1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M$.

Współczynniki wagowe wylicza się, tak jak w przypadkach poprzednich, rekurencyjnie

$$\mathcal{H}_{i,n} = \frac{1}{N_i} \sum_{j=1}^M p_{ij,N} \cdot \mathcal{H}_{i,n-1} \cdot B_N \sum_{l=1}^{N_j} K(B_N(x_{l,j} - x_{n-1})) , \quad (3.29)$$

$$\mathcal{H}_{i,1} = \frac{1}{N}$$

i wyliczenie to należy powtarzać na wszystkich etapach klasyfikacji, z uwagi na niestacjonarny charakter współczynników $\mathcal{H}_{i,n}$.

Wybierając różne funkcje $K(x)$ i różne ciągi B_N otrzymuje się różne odmiany algorytmu (3.28). Przykładowo kładąc [18]

$$K(B_N(x_{l,i} - x)) = \frac{1}{\prod_{s=1}^k} \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x^{(s)})}{B_N^{(s)}(x_{l,i}^{(s)} - x^{(s)})} , \quad (3.30)$$

$$B_N = \prod_{s=1}^k B_N^{(s)} ,$$

gdzie $x_{1,i}^{(s)}, x_{2,i}^{(s)}, \dots, x_{N_i,i}^{(s)}$ jest ciągiem s -tych składowych / $s=1, 2, \dots, k$ / niezależnych obrazów uczących z klasy i / $i=1, 2, \dots, M$ /, $x^{(s)}$ - s -tą składową klasyfikowanego pomiaru, a ciąg wektorów

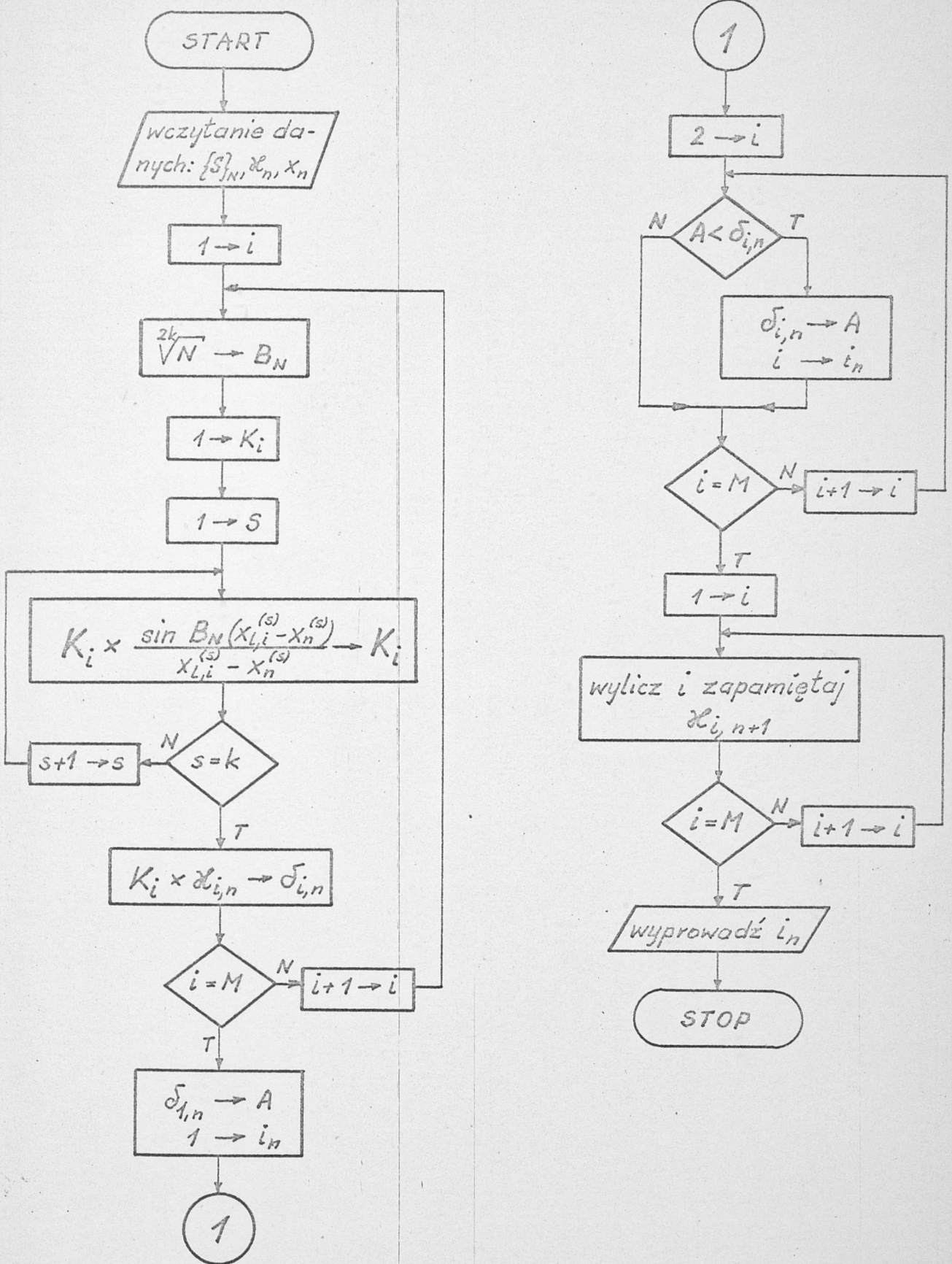
$$B_N = (B_N^{(1)}, B_N^{(2)}, \dots, B_N^{(s)}, \dots, B_N^{(k)})$$

spełnia warunki

$$\lim_{N \rightarrow \infty} \min_s \{B_N^{(s)}\} = \infty , \quad \lim_{N \rightarrow \infty} \frac{1}{N} \prod_{s=1}^k B_N^{(s)} = 0 , \quad (3.31)$$

dostajemy następującą funkcję decyzyjną:

$$\begin{aligned} \delta_N(i, \bar{x}_n) &= \frac{1}{N_i} \sum_{l=1}^{N_i} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x_n^{(s)})}{x_{l,i}^{(s)} - x_n^{(s)}} \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) = \\ &= \mathcal{H}_{i,n} \sum_{l=1}^{N_i} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i}^{(s)} - x_n^{(s)})}{x_{l,i}^{(s)} - x_n^{(s)}} , \end{aligned} \quad (3.32)$$



Rys. 3.3 Sieć działań dla algorytmu MMURTH

gdzie

$$\mathcal{X}_{i,n} = \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) / N_i, \quad (3.33)$$

zaś algorytm rozpoznawania (3.28) ma postać:

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.34)$$

$$\mathcal{X}_{i_n,n} \sum_{l=1}^{N_{i_n}} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,i_n}^{(s)} - x_n^{(s)})}{x_{l,i_n}^{(s)} - x_n^{(s)}} > \mathcal{X}_{r,n} \sum_{l=1}^{N_r} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,r}^{(s)} - x_n^{(s)})}{x_{l,r}^{(s)} - x_n^{(s)}}$$

dla $r = 1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M$.

Współczynniki wagowe wylicza się według następującej procedury rekurencyjnej:

$$\mathcal{X}_{i,n} = \sum_{j=1}^M p_{ij,N} \cdot \mathcal{X}_{j,n-1} \sum_{l=1}^{N_j} \prod_{s=1}^k \frac{\sin B_N^{(s)}(x_{l,j}^{(s)} - x_{n-1}^{(s)})}{x_{l,j}^{(s)} - x_{n-1}^{(s)}} / N_i, \quad (3.35)$$

$$\mathcal{X}_{i,1} = \frac{1}{N}.$$

W dalszym ciągu algorytm (3.28) będziemy nazywać krótko M-MURTH, a jego wersję niezmodyfikowaną /dla obrazów pojawiających się niezależnie/ - MURTH.

Uproszczoną sieć działań algorytmu (3.34) wykorzystującego estymator gęstości Murthy'ego przedstawia rysunek 3.3. Przyjęto w nim

$$B_N^{(1)} = B_N^{(2)} = \dots = B_N^{(k)} = \frac{2k}{\sqrt{N}}.$$

Program realizujący powyższy algorytm zamieszczono w dodatku /procedura MMURTH/.

Obecnie przedstawimy algorytmy rozpoznawania z uczeniem dla prostych łańcuchów Markowa sformułowane w [3,13], które powstały przez uogólnienie na przypadek łańcuchów Markowa rezultatów opracowanych uprzednio dla niezależnego ciągu obrazów [5,7,11,24].

3.4. Zmodyfikowany algorytm "najmniejszy przedział" /M-II/ [3].

Niech $x_{1,i}^{(s)}, x_{2,i}^{(s)}, \dots, x_{N_i,i}^{(s)}$ oznacza rosnąco uporządkowany ciąg s -tych składowych niezależnych pomiarów uczących z klasy i , $x^{(s)}$ - s -tą składową klasyfikowanego pomiaru x . W pracy [3] przyjmuje się - przy założeniu, że poszczególne składowe wektora x są niezależne - następujące empiryczne oszacowanie gęstości prawdopodobieństwa /z uzupełnieniem prawostronnym/:

$$f_N(x/i) = \prod_{s=1}^k f_N^{(s)}(x^{(s)}/i) = \prod_{s=1}^k \frac{1}{N_i \cdot \Delta_{i,N}^{(s)}(x^{(s)})}, \quad (3.36)$$

gdzie

$$\Delta_{i,N}^{(s)}(x^{(s)}) = \begin{cases} x_{l+1,i}^{(s)} - x_{l,i}^{(s)} & \text{gd } x_{l,i}^{(s)} < x^{(s)} \leq x_{l+1,i}^{(s)}, \\ & l=1, 2, \dots, N_i+1, \end{cases} \quad (3.37)$$

w przeciwnym razie,

przy czym

$$x_{N_i+1,i}^{(s)} = 2x_{N_i,i}^{(s)} - x_{N_i-1,i}^{(s)}.$$

Oznaczając z kolei

$$V_{i,N}(x) \triangleq \prod_{s=1}^k \Delta_{i,N}^{(s)}(x^{(s)}) \quad (3.38)$$

otrzymujemy z (3.4) następującą funkcję decyzyjną:

$$\delta_N(i, \bar{x}_n) = \frac{1}{N_i^k \cdot V_{i,N}(x_n)} \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) = \frac{1}{\mathcal{X}_{i,N} \cdot V_{i,N}(x_n)}, \quad (3.39)$$

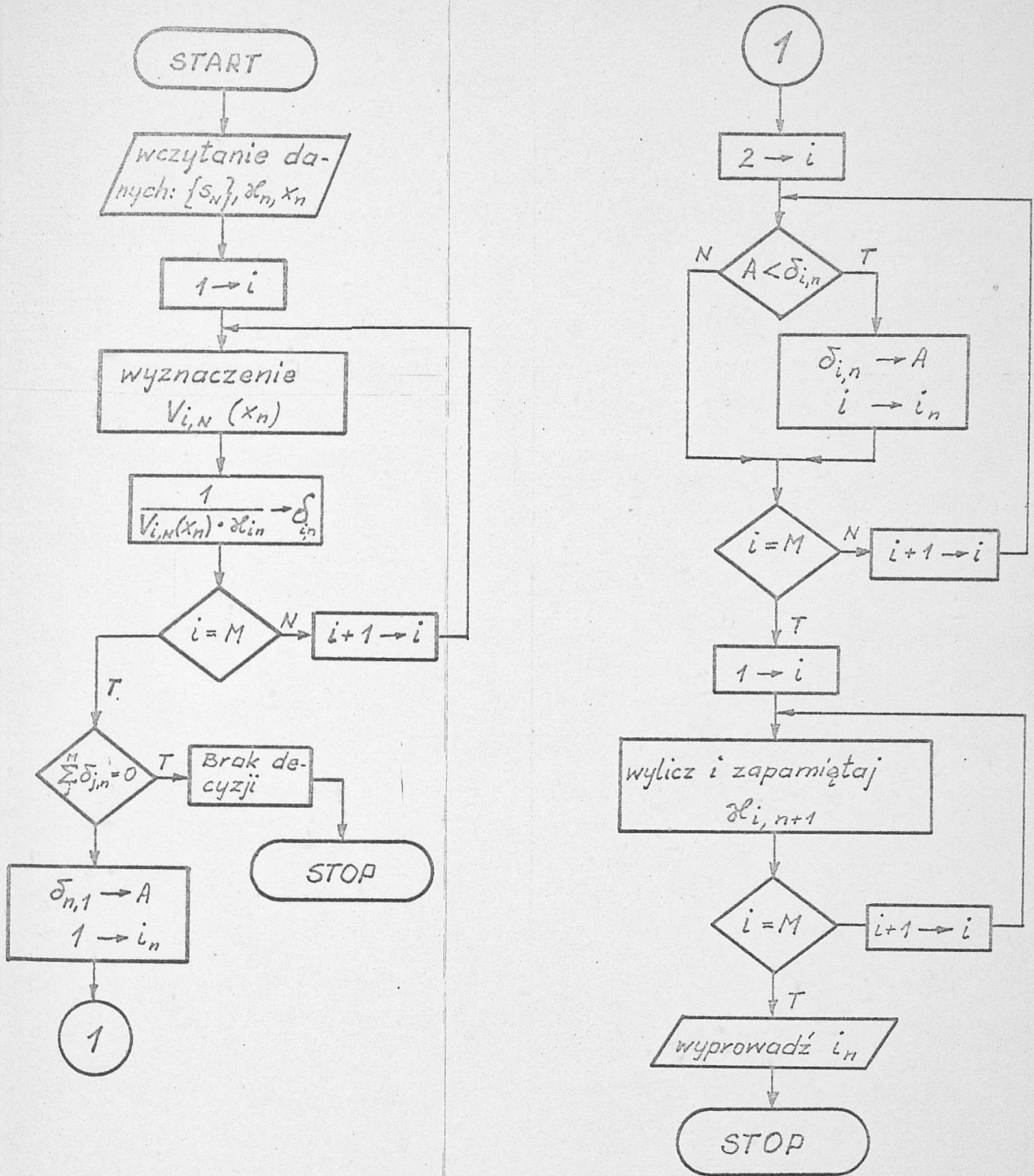
gdzie

$$\mathcal{X}_{i,n} \triangleq \frac{N_i^k}{\sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1})}, \quad (3.40)$$

a algorytm rozpoznawania (3.5) przybiera postać

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.41)$$

$$\mathcal{X}_{i_n,n} \cdot V_{i_n,N}(x_n) < \mathcal{X}_{r,n} \cdot V_{r,N}(x_n), \quad r=1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M.$$



Rys.3.4 Sieć działań dla algorytmu M-LI

Współczynniki wagowe $\mathcal{X}_{i,n}$ obliczane są zgodnie z regułą:

$$\mathcal{X}_{i,n} = \frac{N_i^k}{\sum_{j=1}^M p_{ij,N} / \mathcal{X}_{j,n-1} \cdot V_{j,N}(x_{n-1})}, \quad \mathcal{X}_{i,1} = 1. \quad (3.42)$$

Jeśli pomiar x_n leży na zewnątrz zestawu obrazów uczących, to znaczy gdy:

$$V_{i,N}(x_n) = \infty \quad \text{dla } i=1,2,\dots,M,$$

wówczas algorytm M-LI nie podejmuje decyzji i należy go zastąpić inną regułą. Prawdopodobieństwo takiej sytuacji maleje do zera, gdy wzrasta długość ciągu uczącego.

Uproszczoną sieć działań algorytmu M-LI przedstawia rysunek 3.4, a odpowiedni program zamieszczono w dodatku /procedura MLI/.

3.5. Zmodyfikowane algorytmy "najbliższa średnia" /M-NM/, "najbliższy sąsiad" /M-NN/ i " α najbliższych sąsiadów" /M- α NN/.

Istota zaproponowanych w [13] algorytmów rozpoznawania z uczeniem, które uwzględniają zależność typu Markowa, polega na tym, że:

1. $1/d(x, \tilde{x}^{(i)})$ - odwrotność odległości pomiaru x od średniej arytmetycznej $\tilde{x}^{(i)}$ pomiarów w ciągu uczącym sklasyfikowanych w klasie i ,
2. $1/d(x, D^{(i)}) = \min_{1 \leq l \leq N_i} d(x, x_{l,i})$ - odwrotność odległości pomiędzy pomiarem x a najbliższym pomiarem uczącym z klasy i ,
3. $n_i(x)$ - liczbę sąsiadów pomiaru x z i -tej klasy wśród α najbliższych

traktuje się jako różne oszacowania wartości $p_i \cdot f(x/i)$. W rezultacie otrzymano trzy różne algorytmy stanowiące pewne modyfikacje algorytmów NM, NN i α NN.

Przekształćmy (3.4) do postaci

$$\delta_N(i, x_n) = \frac{p_{i,N} \cdot f_N(x_n/i)}{p_{i,N}} \sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1}) \quad (3.43)$$

3.5.1. Zmodyfikowany algorytm "najbliższa średnia"

Z (3.43) otrzymujemy

$$\delta_N(i, \bar{x}_n) = \frac{1}{d(x, \tilde{x}^{(i)}) \cdot \mathcal{X}_{i,n}} \quad (3.44)$$

gdzie

$$\mathcal{X}_{i,n} \triangleq \frac{p_{i,N}}{\sum_{j=1}^M p_{ij,N} \cdot \delta_N(j, \bar{x}_{n-1})} \quad (3.45)$$

Algorytm rozpoznawania (3.5) przybiera zatem postać:

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.46)$$

$$\mathcal{X}_{i_n,n} \cdot d(x_n, \tilde{x}^{(i_n)}) < \mathcal{X}_{r,n} \cdot d(x_n, \tilde{x}^{(r)}), \quad r=1,2,\dots,i_{n-1},i_{n+1},\dots,M.$$

Współczynniki wagowe wylicza się rekurencyjnie

$$\mathcal{X}_{i,n} = \frac{p_{i,N}}{\sum_{j=1}^M \frac{p_{i,j,N}}{\mathcal{X}_{j,n-1} \cdot d(x_{n-1}, \tilde{x}^{(j)})}} \quad \mathcal{X}_{i,1} = 1. \quad (3.47)$$

3.5.2. Z m o d y f i k o w a n y a l g o r y t m " n a j - b l i ż s z y s ą s i a d "

Uzyskuje się go analogicznie do poprzedniego zastępując jedynie oszacowanie wielkości $p_i \cdot f(x/i)$.

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.48)$$

$$\mathcal{X}_{i_n,n} \cdot d(x_n, D^{(i_n)}) < \mathcal{X}_{r,n} \cdot d(x_n, D^{(r)}),$$

$$r = 1, 2, \dots, i_{n-1}, i_{n+1}, \dots, M.$$

Współczynniki wagowe $\mathcal{X}_{i,n}$ wyznacza się według (3.47) zastępując odległość $d(x_n, \tilde{x}^{(j)})$ odległością $d(x_n, D^{(j)})$.

3.5.3. Z m o d y f i k o w a n y a l g o r y t m " \alpha - n a j - b l i ż s z y c h s ą s i a d ó w "

Z (3.43) mamy

$$\delta_N(i, \bar{x}_n) = n_i(x_n) \cdot \mathcal{X}_{i,n}, \quad (3.49)$$

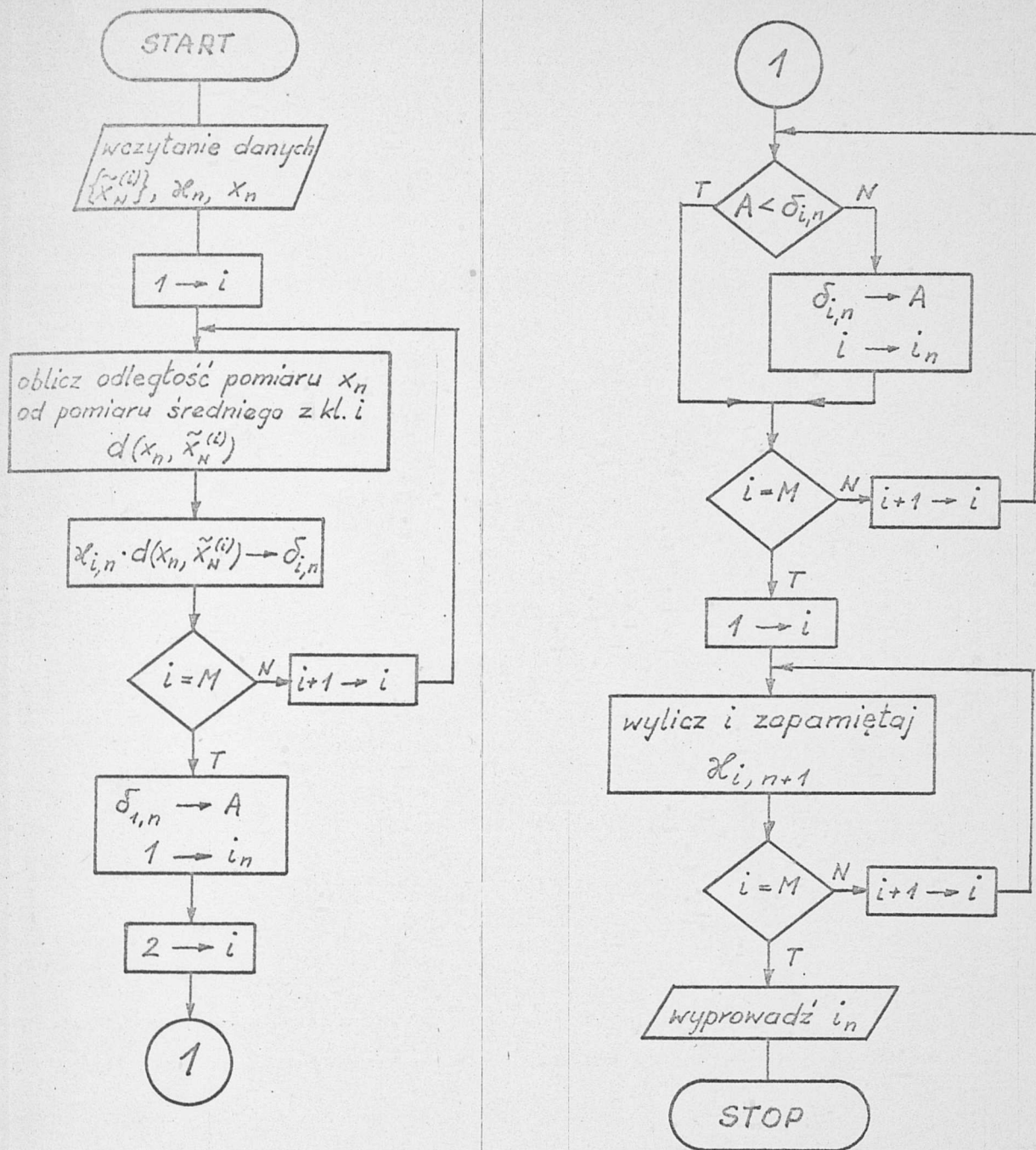
gdzie

$$\mathcal{X}_{i,n} = \frac{1}{p_{i,N}} \sum_{j=1}^M p_{i,j,N} \cdot \delta_N(j, \bar{x}_{n-1}). \quad (3.50)$$

Z (3.5) wynika następująca postać algorytmu:

$$\Psi_{N,n}(x_n) = i_n, \text{ jeśli} \quad (3.51)$$

$$\mathcal{X}_{i_n,n} \cdot n_{i_n}(x_n) > \mathcal{X}_{r,n} \cdot n_r(x_n), \quad r=1,2,\dots,i_{n-1},i_{n+1},\dots,M.$$



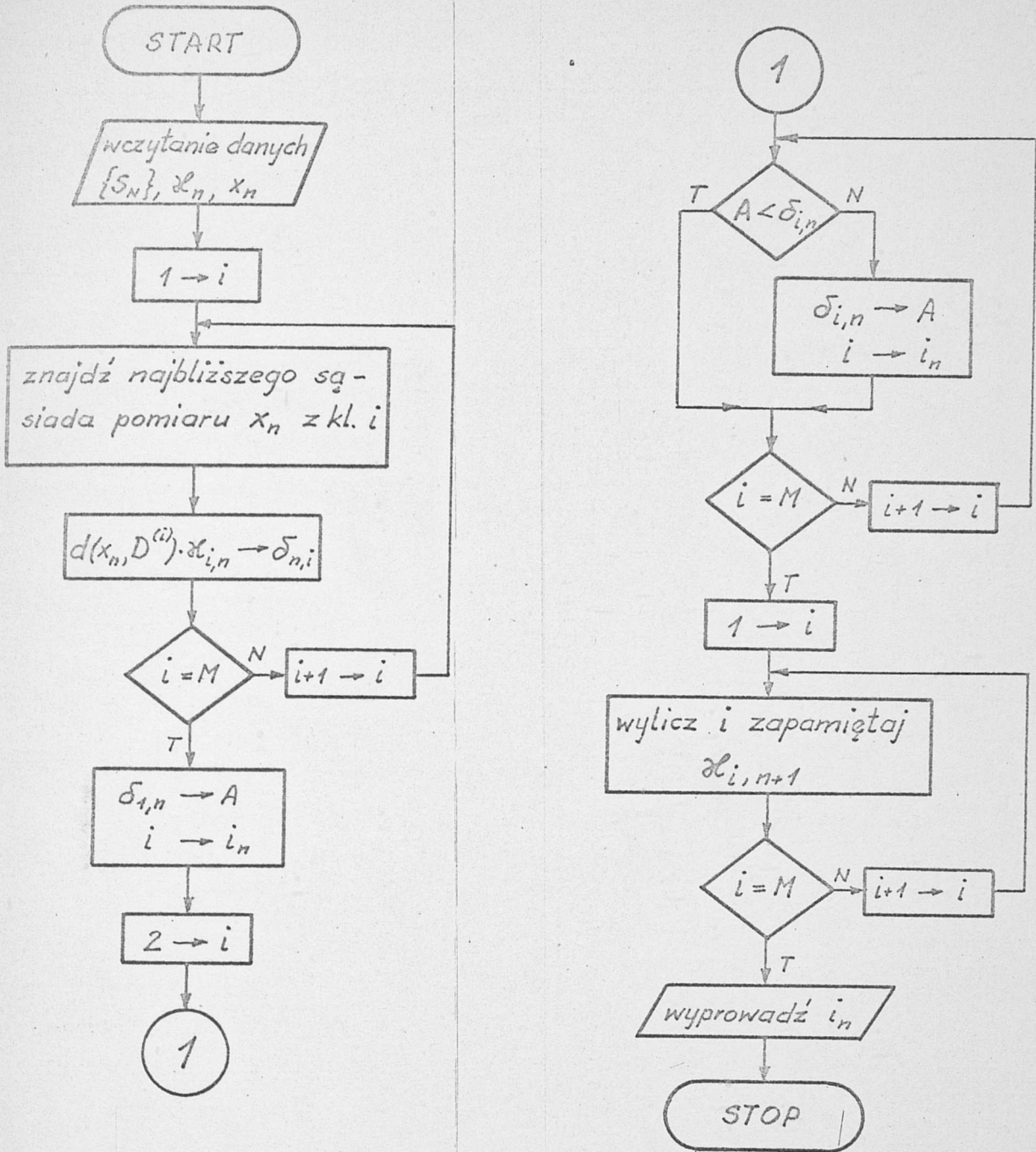
Rys. 3.5 Sieć działań dla algorytmu M-NM

Współczynniki wagowe wylicza się również rekurencyjnie:

$$\alpha_{i,n} = \frac{1}{p_{i,N}} \sum_{j=1}^M p_{ij,N} \cdot \alpha_{j,n-1} \cdot n_j(x_{n-1}), \quad \alpha_{i,1} = 1 \quad (3.52)$$

Jeśli uzależnić liczbę najbliższych pomiarów α od N i wybrać ją tak aby spełnione były warunki

$$\lim_{N \rightarrow \infty} \alpha(N) = \infty, \quad \lim_{N \rightarrow \infty} N^{-1} \cdot \alpha(N) = 0 \quad (3.53)$$



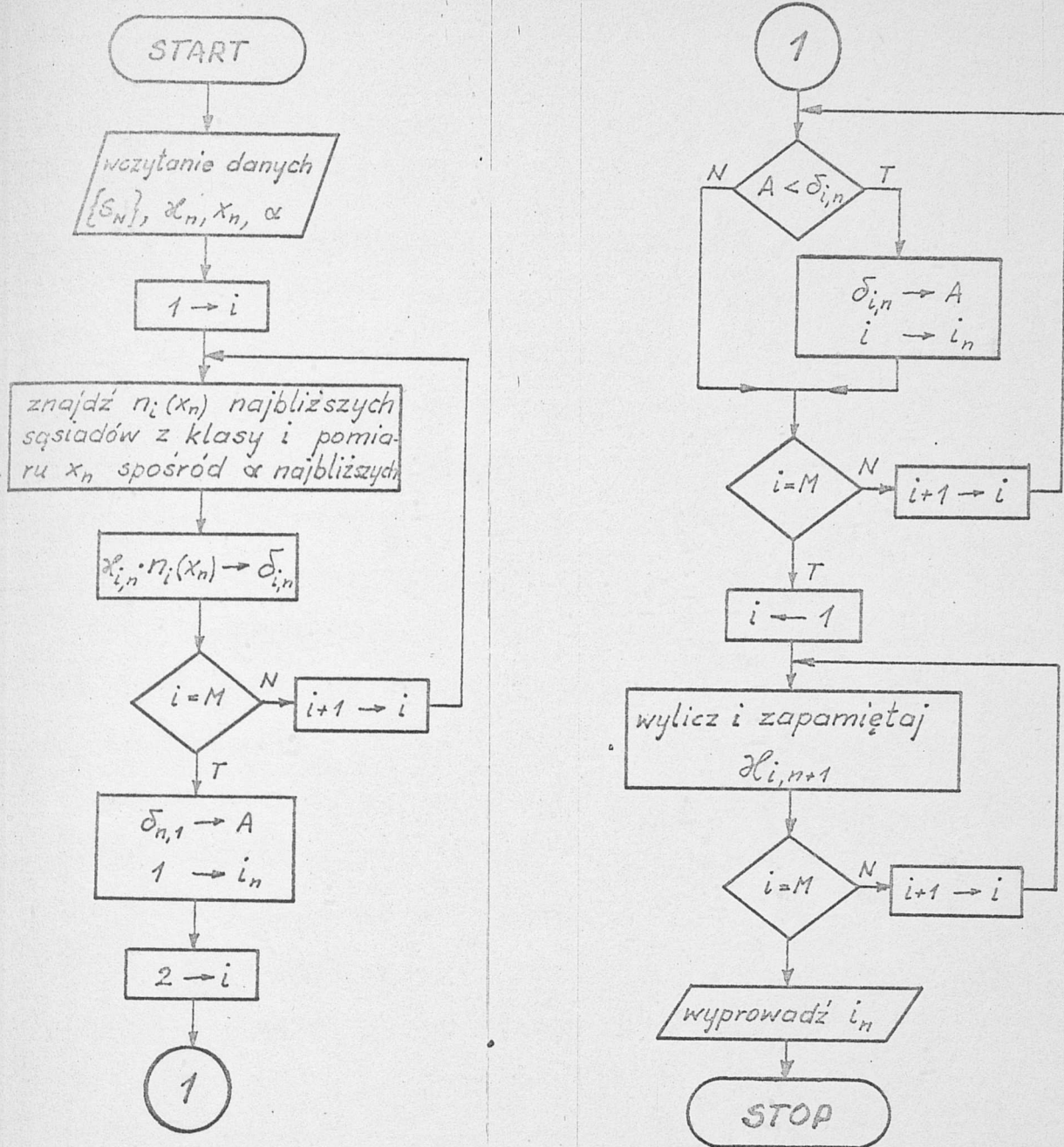
Rys. 3.6 Sieć działań dla algorytmu M-NN

wówczas dla przypadku dychotomii $M=2$ wyrażenie

$$\frac{n_i(x)}{N_i}$$

jest zgodnym estymatorem warunkowej gęstości $f(x/i)$ [32].

Uprozczone schematy blokowe algorytmów M-NN, M- α NN przedstawiono na rysunkach 3.5, 3.6, 3.7, a odpowiednie programy zamieszczono w dodatku /procedury MNM, MNN, Malfann /.

Rys.3.7 Sieć działań dla algorytmu M- α NN

Jak wynika z przedstawionych algorytmów, uwzględnienie zależności Markowa prowadzi do skomplikowanych na ogół reguł decyzyjnych. We wszystkich podanych algorytmach wyróżnić można pewne współczynniki wagowe związane z charakterem zależności pomiędzy poszczególnymi klasami obiektów. Postać tych współczynników, jak i sam algorytm rozpoznawania zależą w istotny sposób od przyjętej metody estymacji nieznannej gęstości prawdopodobieństwa, we wszystkich jednak przypadkach współczynniki te mają charakter niestacjonarny, a w związku z tym niestacjonarny jest również algorytm rozpoznawania.

4. PODSTAWY I METODYKA ANALIZY PORÓWNAWCZEJ

Zagadnieniem istotnym z punktu widzenia praktyki jest porównanie różnych algorytmów rozpoznawania z uczeniem w przypadku niepełnej informacji apriorycznej o rozpoznawanych obiektach. Jako podstawę porównania własności różnych algorytmów rozpoznawania z uczeniem, a tym samym różnych sposobów uczenia, uważać można ryzyko średnie, będące w przypadku zerojedynekowej funkcji strat, uśrednieniem prawdopodobieństwa błędnej klasyfikacji po wszystkich możliwych ciągach uczących. Jednakże, jak do tej pory, analityczne rozwiązanie zagadnienia oceny jakości algorytmu rozpoznawania dla prostych łańcuchów Markowa w oparciu o dany sposób uczenia, a więc wyznaczenie średniego ryzyka na drodze analitycznej, napotyka na ogromne trudności i z tego też powodu wydaje się, że zasadniczą możliwością jest tutaj bądź eksperymentalne porównanie danych algorytmów rozpoznawania, bądź też dokonanie oszacowania ryzyka średniego i przyjęcie go jako liczbowego wskaźnika jakości algorytmu [6].

Ryzyko średnie może być również podstawą do sformułowania i badania własności asymptotycznych algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa zgodnie z następującymi definicjami [3.5]:

DEFINICJA 1

Algorytm rozpoznawania $\Psi_{N,n}(x_n)$ jest zbieżny, jeśli dla każdego naturalnego n zbieżny jest ciąg $R_n[\Psi_{N,n}(x_n)]$, to znaczy gdy istnieje granica

$$\bigwedge_n \lim_{N \rightarrow \infty} R_n[\Psi_{N,n}(x_n)] = R_n.$$

DEFINICJA 2

Algorytm rozpoznawania $\Psi_{N,n}(x_n)$ jest asymptotycznie optymalny, gdy dla pewnych rozkładów (p.p. $f(x/i)$)

$$\bigwedge_n \lim_{N \rightarrow \infty} R_n[\Psi_{N,n}(x_n)] = R_n^*.$$

gdzie $R_n^* \triangleq R_n[\Psi_n^*(x_n)]$ oznacza średnie ryzyko na n -tym etapie klasyfikacji przy zastosowaniu optymalnej reguły bayesowskiej.

DEFINICJA 3

Algorytm rozpoznawania $\gamma_{N,n}(x_n)$ jest bezwzględnie asymptotycznie optymalny, jeśli jest on asymptotycznie optymalny dla dowolnych rozkładów prawdopodobieństwa.

W następnym rozdziale przedstawiono pewne analityczne rezultaty dotyczące asymptotycznych własności algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa uzyskane w oparciu o powyższe definicje, a które to rezultaty zostały zawarte w trzech twierdzeniach. W pierwszym z nich, podobnie jak to zrobiono dla niezależnych klas obiektów [30], określono pewne wystarczające warunki asymptotycznej optymalności algorytmów rozpoznawania prostych łańcuchów Markowa. W następnym twierdzeniu wykazano, że algorytmy rozpoznawania (3.5), w których przy formułowaniu reguł decyzyjnych wykorzystuje się estymatory nieznanymi gęstości zbieżne według prawdopodobieństwa, są dla dowolnej, skończonej funkcji strat - asymptotycznie optymalne. Na koniec, w ostatnim twierdzeniu dowodzi się zbieżności zmodyfikowanego algorytmu typu "najbliższa średnia".

W dalszej części opisane zostały wyniki eksperymentów, jakie przeprowadzono celem sprawdzenia przydatności przedstawionych w rozdziale 3. algorytmów rozpoznawania z uczeniem dla klasyfikacji prostych łańcuchów Markowa oraz celem zbadania ich pewnych własności.

Ogólne, eksperymentalne badanie algorytmu rozpoznawania ułatwiające jego ocenę w przypadku, gdy analityczne obliczenie ryzyka jest skomplikowane lub wręcz niemożliwe, polega na tym, że przy jego pomocy dokonuje się klasyfikacji pewnego ciągu obiektów z określonymi, lecz nieznanymi rozkładami i na wyliczeniu następnie częstości błędnej klasyfikacji w zależności od liczebności ciągu uczącego, przy czym klasyfikowane obiekty mogą mieć dwojaki charakter:

1. mogą to być pewne obrazy rzeczywiste (n.p. litery^[17]) - wtedy do zadania klasyfikacji dochodzi jeszcze bardzo istotna w praktycznych zastosowaniach problematyka dotycząca sposobu wyboru i przygotowania pomiarów, ich wstępnej obróbki, wprowadzenia na maszynę cyfrową itp.,

2. mogą to być liczby losowe o odpowiednich rozkładach losowane z określonymi prawdopodobieństwami z populacji generalnych odpowiadających poszczególnym klasom. Liczby te traktować można jako gotowe pomiary odpowiadające bliżej nie określonym obiektom.

Oczywiście w obu przypadkach prawdziwa przynależność tych obiektów do klas musi być znana.

W badaniach eksperymentalnych, jakie przeprowadzono celem określenia własności niektórych przedstawionych w pracy algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa, przyjęto drugi sposób, co pozwoliło nie tylko uprościć zagadnienie, ale również wyeliminować wpływ etapu obróbki danych na jakość klasyfikacji.

Podstawą oceny poszczególnych algorytmów jest w pracy częstość błędnych klasyfikacji

$$c = \frac{l}{L} ,$$

gdzie

L - liczba klasyfikowanych pomiarów,

l - liczba błędnie sklasyfikowanych pomiarów w sekwencji o długości L,

oraz czas trwania klasyfikacji pojedynczego obrazu.

Odpowiednie wyniki liczbowe charakteryzujące jakość rozpoznawania będą zależały oczywiście od warunków klasyfikacji oraz od parametrów rozkładu prawdopodobieństwa składających się na zestaw informacji apriorycznej. Dlatego też badania przeprowadzono w różnych warunkach i dla różnych wartości tych parametrów.

Zauważmy ponadto, że algorytmy uwzględniające zależność typu Markowa pomiędzy poszczególnymi klasami obiektów są na ogół znacznie bardziej skomplikowane niż odpowiednie algorytmy nie uwzględniające takiej zależności. Istotne z punktu widzenia praktyki jest przeto sprawdzenie o ile uwzględnianie zależności i stosowanie w konsekwencji bardziej skomplikowanych algorytmów zmniejsza częstość błędnych klasyfikacji oraz o ile wydłuża czas klasyfikacji. W tym celu

przeprowadzono również eksperymentalne porównanie algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa z niezmodyfikowanymi wersjami tych algorytmów. Pozwoliło to zbadać nie tylko własności przedstawionych modyfikacji badanych algorytmów, ale również sprawdzić w jakich przypadkach modyfikacja taka - przy wybranych w eksperymentach długościach ciągów uczących - daje pozytywny wynik. Dokładnie zakres i cele eksperymentalnych badań algorytmów rozpoznawania prostych łańcuchów Markowa przedstawiono schematycznie w tabelach 1 i 2. Jak z nich wynika, w przeprowadzonych badaniach eksperymentalnych algorytmów rozpoznawania dla prostych łańcuchów Markowa wyróżnia się dwie części.

PRZEDMIOT BADAŃ (ALGORYTMY)	OKREŚLANO WPLYW:					PORÓWNYWANO Z:							
	DEŁGOCÍ CIĄGU UCZĄCEGO	IŁOCÍ KLAS	IŁOCÍ WYMIARÓW OBRZÓW	PARAMETRÓW ŁĄCUCZA MARKOWA	PARAMETRÓW ROZKŁADÓW WARUNKOWYCH	WERSJĄ NIEZMODYFIKOWANĄ	ALGORYTMEM M-h(N) NN	ALGORYTMEM M-PAR	ALGORYTMEM M-MURTH	ALGORYTMEM M-LI	ALGORYTMEM M-NN	ALGORYTMEM M-NM	ALGORYTMEM M- α NN
M-h(N) NN	X	X	X	X	X	X		X	X	X	X	X	X
M-PAR	X	X	X	X	X	X	X		X	X	X	X	X
M-MURTH	X	X	X	X	X	X	X	X		X	X	X	X
M-LI	X	X	X	X	X	X	X	X	X		X	X	X
M-NN	X	X	X	X	X	X	X	X	X	X		X	X
M-NM	X	X	X	X	X	X	X	X	X	X	X		X
M- α NN	X	X	X	X	X	X	X	X	X	X	X	X	

Tab. 1 Cele i zakres badań eksperymentalnych algorytmów rozpoznawania pod względem czasów klasyfikacji

W pierwszej części polegającej na określeniu czasów klasyfikacji badanych algorytmów założono rozkłady normalne we wszystkich klasach i sprawdzono kolejno jak się kształtuje czas rozpoznawania pojedynczego obrazu w zależności od:

1. długości ciągu uczącego (N),
2. ilości klas (M),
3. ilości wymiarów przestrzeni obrazów (k),
4. parametrów łańcucha Markowa,
5. parametrów rozkładów normalnych.

Aby oddzielnie określić wpływ poszczególnych rodzajów parametrów na czas klasyfikacji w trakcie eksperymentów zmieniano kolejno jeden z nich ustalając wszystkie pozostałe.

PRZEDMIOT BADAŃ (ALGORYTM)	OKREŚLANO WPŁYW:			PORÓWNYWANO Z:						
	CELE I ZAKRES BADAŃ	DLUGOŚĆ CIĄGU UCZĄCEGO	PARAMETRÓW ŁAŃCUCHA MARKOWA	PARAMETRÓW ROZKŁADÓW WARUNKOWYCH	WERSJĄ NIEZMODYFIKOWANĄ	ALGORYTMEM M-MURTH	ALGORYTMEM M-LI	ALGORYTMEM M-NN	ALGORYTMEM M-NM	ALGORYTMEM M- α NN
M - MURTH		X	X	X	X		X	X	X	X
M - LI		X	X	X	X	X		X	X	X
M - NN		X	X	X	X	X	X		X	X
M - NM		X	X	X	X	X	X	X		X
M - α NN		X	X	X	X	X	X	X	X	

Tab. 2 Cele i zakres badań eksperymentalnych algorytmów rozpoznawania pod względem częstości błędnych klasyfikacji

Realizacja tego eksperymentu przeprowadzona całkowicie na m.c. ODRA 1204 polegała na generacji różnych ciągów losowych składających się z 100 + 1000 pomiarów x tworzących prosty łańcuch Markowa wraz z ich poprawnymi klasyfikacjami korzystając z opracowanych w bibliotece procedur ALGOLU 1204 [35] generatorów liczb losowych o rozkładzie równomiernym (procedura "unif") z przedziału $[0,1)$ oraz liczb losowych o rozkładzie normalnym $N(0,1)$ (procedura "gennorm"), a następnie dokonywano klasyfikacji jednego z obrazów losowych (pozostałe tworzyły ciąg uczący) według badanych algorytmów mierząc czasy rozpoznawania przy użyciu procedury "timing" [37].

Druga część badań eksperymentalnych polegała na określeniu częstości błędnych klasyfikacji badanych algorytmów rozpoznawania. Przyjęto tutaj $k = 1$ (obrazy jednowymiarowe) oraz założono dwie klasy obiektów ($M = 2$), w których warunkowe rozkłady prawdopodobieństwa są normalne

$$f(x/1) = N(m_1, \sigma_1), \quad f(x/2) = N(m_2, \sigma_2).$$

a klasy obiektów tworzą prosty, stacjonarny i jednorodny łańcuch Markowa o macierzy przejścia:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

i prawdopodobieństwach początkowych p_1 i p_2 . Badanie własności algorytmów rozpoznawania pod względem częstości błędnych klasyfikacji przeprowadzono w ten sposób, że dla różnych długości ciągu uczącego:

1. ustalono probabilistyczne charakterystyki łańcucha Markowa (to znaczy macierz P oraz p_1 i p_2) zmieniając kolejno parametry rozkładów warunkowych $f(x/1)$ i $f(x/2)$,
2. ustalono parametry rozkładów warunkowych $f(x/1)$ i $f(x/2)$ zmieniając natomiast kolejno parametry łańcucha Markowa, czyli jego prawdopodobieństwa przejścia i prawdopodobieństwa początkowe.

W pierwszym etapie na m.c. ODRA 1003 wygenerowano 12 różnych ciągów losowych, z których każdy składał się z 1000 pomiarów x powiązanych w prosty łańcuch Markowa wraz ze znajomością ich poprawnych klasyfikacji. Etap ten zrealizowano korzystając z generatora liczb losowych o rozkładzie jednostajnym z przedziału $[0,1]$ (III generator podprogramu bibliotecznego 93 m.c. ODRA 1003) charakteryzującego się małym skorelowaniem generowanych liczb oraz tablic dystrybuanty rozkładu normalnego $N(0,1)$.

W następnym etapie na m.c. ODRA 1003 i ODRA 1204 dokonywano klasyfikacji 500 pomiarów x według każdego z badanych algorytmów rozpoznawania przy różnej liczbie obrazów uczących (różnych długościach N ciągu uczącego).

Porównując dla kolejnych pomiarów x uzyskane rezultaty ich klasyfikacji z klasyfikacją poprawną, wyznaczono liczbę błędnych decyzji a a następnie częstość c błędnych klasyfikacji dla badanych algorytmów.

Ponieważ w przeprowadzonych eksperymentach częstość błędnych klasyfikacji przy ustalonym N wyznaczana była tylko dla jednego ciągu uczącego, dlatego nie można jej nadawać interpretacji średniego ryzyka, co wymagałoby bowiem uśrednienia wartości c dla wszystkich możliwych ciągów uczących.

5. ANALIZA PORÓWNAWCZA ALGORYTMÓW ROZPOZNAWANIA Z UCZENIEM DLA PROSTYCH ŁANCUCHÓW MARKOWA

5.1. Pewne rezultaty analityczne.

Załóżmy, że na n -tym etapie klasyfikacji istnieje optymalna bayesowska reguła decyzyjna

$$\Psi_n^*(x_n) \triangleq \Psi_n^*(x_n / \bar{x}_{n-1}) \quad (5.1)$$

minimizująca ryzyko warunkowe

$$\begin{aligned} r_n(\Psi_n^*(x_n), \bar{x}_n) &= \sum_{j_n=1}^M L(\Psi_n^*(x_n), j_n) \cdot p_{j_n} \cdot \bar{f}_n(\bar{x}_n / j_n) = \\ &= \min_{\Psi_n} r_n(\Psi_n(x_n), \bar{x}_n). \end{aligned} \quad (5.2)$$

Stąd dla dowolnej funkcji decyzyjnej $\Psi_n(x_n)$

$$R_n[\Psi_n^*(x_n)] \triangleq \int_{\bar{x}_n} \min_{\Psi_n} r_n(\Psi_n(x_n), \bar{x}_n) d\bar{x}_n \leq R_n[\Psi_n(x_n)], \quad (5.3)$$

gdzie $R_n[\cdot]$ oznacza ryzyko średnie na n -tym etapie klasyfikacji.

Niech $\Psi_{N,n}(x_n) \triangleq \Psi(x_n, \bar{x}_{n-1}, \{S\}_N)$ oznacza algorytm rozpoznawania z uczeniem na n -tym etapie klasyfikacji skonstruowany w oparciu o ciąg uczący $\{S\}_N$. Ryzyko średnie dla tej reguły wynosi:

$$R_n[\Psi_{N,n}(x_n)] \triangleq \int_{\bar{x}_n} E r_n(\Psi_{N,n}(x_n), \bar{x}_n) d\bar{x}_n, \quad (5.4)$$

gdzie E oznacza uśrednienie po wszystkich możliwych ciągach uczących.

Oczywiście

$$R_n[\Psi_n^*(x_n)] \leq R_n[\Psi_{N,n}(x_n)]. \quad (5.5)$$

Na to, aby algorytm rozpoznawania $\Psi_{N,n}(x_n)$ był asymptotycznie optymalny potrzeba i wystarcza z definicji 2, aby \bigwedge_n

$$\lim_{N \rightarrow \infty} \int_{\bar{x}_n} E r_n(\Psi_{N,n}(x_n), \bar{x}_n) d\bar{x}_n = \int_{\bar{x}_n} r_n(\Psi_n^*(x_n), \bar{x}_n) d\bar{x}_n \quad (5.6)$$

Z twierdzenia Lebesgue'a [33] wynika, że na to, aby zachodziło (5.6)

*) W dalszym ciągu dla uproszczenia zapisu kwantyfikator \bigwedge_n opuścimy.

Wystarczy, aby

$$\lim_{N \rightarrow \infty} E r_n(\Psi_{N,n}(x_n), \bar{x}_n) = r_n(\Psi_n^*(x_n), \bar{x}_n) \text{ pr. wszędzie na } \bar{X}_n \quad (5.7)$$

oraz

$$\bigwedge_N E r_n(\Psi_{N,n}(x_n), \bar{x}_n) \leq H(\bar{x}_n), \quad (5.8)$$

gdzie

$$\int H(\bar{x}_n) d\bar{x}_n < \infty. \quad (5.9)$$

Niech $L(i, j) \geq 0$ będzie funkcją strat, taką że

$$\infty > L(j_n) \stackrel{\Delta}{=} \max_i L(i, j_n), \quad j_n, i=1, 2, \dots, M. \quad (5.10)$$

Kładąc

$$H(\bar{x}_n) \stackrel{\Delta}{=} \sum_{j_n=1}^M L(j_n) \cdot \bar{f}_n(\bar{x}_n/j_n) \cdot p_{j_n} \geq 0 \quad (5.11)$$

otrzymujemy

$$H(\bar{x}_n) = \max_{i_n} \sum_{j_n=1}^M L(i_n, j_n) \cdot \bar{f}_n(\bar{x}_n/j_n) \cdot p_{j_n} = \max_{i_n} r_n(i_n, \bar{x}_n). \quad (5.12)$$

A więc

$$\bigwedge_N r_n(\Psi_{N,n}(x_n), \bar{x}_n) \leq H(\bar{x}_n). \quad (5.13)$$

Stąd oraz z twierdzenia o wartości średniej [21] wynika (5.8).

Z (5.10) otrzymujemy natomiast

$$\begin{aligned} \int_{\bar{X}_n} H(\bar{x}_n) d\bar{x}_n &= \int_{\bar{X}_n} \sum_{j_n=1}^M L(j_n) \cdot \bar{f}_n(\bar{x}_n/j_n) \cdot p_{j_n} d\bar{x}_n = \\ &= \sum_{j_n=1}^M L(j_n) \left\{ \int_{\bar{X}_n} \bar{f}_n(\bar{x}_n/j_n) d\bar{x}_n \right\} \cdot p_{j_n} < \infty. \end{aligned} \quad (5.14)$$

Zauważmy ponadto, że

$$H(\bar{x}_n) < \infty \text{ pr. wszędzie na } \bar{X}_n. \quad (5.15)$$

Stąd aby zachodziło (5.7) wystarczy (5.10) oraz aby [23,30]

$$\lim_{N \rightarrow \infty} P \left\{ \left| r_n(\Psi_{N,n}(x_n), \bar{x}_n) - r_n(\Psi_n^*(x_n), \bar{x}_n) \right| < \varepsilon \right\} = 1 \quad (5.16)$$

prawie wszędzie na \bar{X}_n .

Wynika więc, że (5.16) oraz (5.10) wystarczą aby $\Psi_{N,n}(x_n)$ było asymptotycznie optymalne.

Założmy dalej, że można znaleźć ciąg funkcji

$$r_{N,n}(i, \bar{x}_n) \triangleq r_n(\{S\}_{N,n}^i, \bar{x}_n), \quad (5.17)$$

taki że dla każdego $i=1, 2, \dots, M$

$$\lim_{N \rightarrow \infty} P \left\{ \left| r_{N,n}(i, \bar{x}_n) - r_n(i, \bar{x}_n) \right| < \varepsilon \right\} = 1 \quad \text{pr. wsz. na } \bar{X}_n \quad (5.18)$$

oraz

$$r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) = \min_i r_{N,n}(i, \bar{x}_n). \quad (5.19)$$

Z (5.2), (5.18) i (5.19) wnioskujemy, że

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N > \bar{N}} P \left\{ \left| r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) - r_{N,n}(\Psi_n^*(x_n), \bar{x}_n) \right| < \varepsilon \right\} \geq 1 - \delta$$

prawie wszędzie na \bar{X}_n . (5.20)

Z póładdytywności miary mamy

$$1 - P(A \wedge B \wedge C) = P(\bar{A} \vee \bar{B} \vee \bar{C}) \leq P(\bar{A}) + P(\bar{B}) + P(\bar{C}) = 1 - P(A) + 1 - P(B) + 1 - P(C) = 3 - [P(A) + P(B) + P(C)],$$

a stąd

$$P(A \wedge B \wedge C) \geq P(A) + P(B) + P(C) - 2. \quad (5.21)$$

Z (5.18), (5.20) i (5.21) wynika więc, że:

$$\begin{aligned} & \bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N > \bar{N}} P \left\{ \left| r_n(\Psi_{N,n}(x_n), \bar{x}_n) - r_n(\Psi_n^*(x_n), \bar{x}_n) \right| < 3 \cdot \varepsilon = \right. \\ & = P \left\{ \left| \left[r_n(\Psi_{N,n}(x_n), \bar{x}_n) - r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) \right] + \left[r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) + \right. \right. \right. \\ & \left. \left. - r_{N,n}(\Psi_n^*(x_n), \bar{x}_n) \right] + \left[r_{N,n}(\Psi_n^*(x_n), \bar{x}_n) - r_n(\Psi_n^*(x_n), \bar{x}_n) \right] \right| < 3 \cdot \varepsilon \right\} \geq \\ & \geq P \left\{ \left| r_n(\Psi_{N,n}(x_n), \bar{x}_n) - r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) \right| < \varepsilon, \left| r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) + \right. \right. \\ & \left. \left. - r_{N,n}(\Psi_n^*(x_n), \bar{x}_n) \right| < \varepsilon, \left| r_{N,n}(\Psi_n^*(x_n), \bar{x}_n) - r_n(\Psi_n^*(x_n), \bar{x}_n) \right| < \varepsilon \right\} \geq \end{aligned}$$

$$\geq (1 - \delta) + (1 - \delta) + (1 - \delta) - 2 = 1 - 3\delta \quad \text{pr. wsz. na } \bar{X}_n. \quad (5.22)$$

Udowodniliśmy zatem następujące:

TWIERDZENIE 1.

Niech $\max_{i,j} L(i,j) < \infty$, $i, j = 1, 2, \dots, M$ oraz dla każdego naturalnego n :

1. $r_{N,n}(i, \bar{x}_n)$ będzie ciągiem funkcji, takich że

$$\lim_{N \rightarrow \infty} P \left\{ |r_{N,n}(i, \bar{x}_n) - r_n(i, \bar{x}_n)| < \xi \right\} = 1$$

dla $i = 1, 2, \dots, M$ oraz pr. wsz. na \bar{X}_n ,

2. $\Psi_{N,n}(x_n)$ będzie algorytmem rozpoznawania, takim że

$$r_{N,n}(\Psi_{N,n}(x_n), \bar{x}_n) = \min_i r_{N,n}(i, \bar{x}_n). \quad (5.23)$$

Wtedy $\Psi_{N,n}(x_n)$ jest asymptotycznie optymalny.

Powyższe twierdzenie określa pewne wystarczające warunki na to, aby dany algorytm rozpoznawania z uczeniem dla prostych łańcuchów Markowa był asymptotycznie optymalny. Jest to więc pewne kryterium asymptotycznej optymalności.

Oczywiście, jeśli dla reguły decyzyjnej z uczeniem $\Psi_{N,n}(x_n)$ spełnione są założenia twierdzenia 1 bez żadnych ograniczeń nałożonych na odpowiednie rozkłady i prawdopodobieństwa, to jest ona bezwzględnie asymptotycznie optymalna.

Zbadamy obecnie pewne asymptotyczne własności algorytmów /dla $N \rightarrow \infty$ / rozpoznawania prostych łańcuchów Markowa, w których przy formułowaniu reguł decyzyjnych wykorzystuje się estymatory nieznanymi rozkładów i prawdopodobieństw zbieżne według prawdopodobieństwa. Dla niezależnie pojawiających się klas obiektów własności takie zbadano w pracy [12].

Korzystając z twierdzenia 1 udowodnimy, że jeśli estymator $f_N(x/i)$ nieznannej gęstości $f(x/i)$ oraz oszacowania $p_{i,N}$, $p_{ij,N}$ prawdopodobieństw początkowych i przejścia jednorodnego łańcucha Markowa są zbieżne według prawdopodobieństwa, wówczas odpowiednie ryzyko jest zbieżne do ryzyka Bayesa, a więc algorytm rozpoznawania (3.5) jest asymptotycznie optymalny.

TWIERDZENIE 2.

Niech $p_{i,N}$ oraz $p_{ij,N}$ będą oszacowaniami prawdopodobieństw początkowych p_i i prawdopodobieństw przejścia p_{ij} jednorodnego, prostego łańcucha Markowa, takimi że

$$\bigwedge_i p_{i,N} \rightarrow p_i \quad \text{według prawdopodobieństwa,} \quad (5.24)$$

$$\bigwedge_{i,j} p_{ij,N} \rightarrow p_{ij} \quad \text{według prawdopodobieństwa.} \quad (5.25)$$

Niech $f_N(x/i)$ będzie estymatorem nieznannej warunkowej gęstości $f(x/i)$ takim że

$$\bigwedge_i f_N(x/i) \rightarrow f(x/i) \quad \text{według prawdopodobieństwa i} \quad (5.26)$$

prawie wszędzie na X .

Wtedy dla dowolnej, skończonej i nieujemnej funkcji strat algorytm rozpoznawania z uczeniem dla prostych łańcuchów Markowa określony w sposób następujący:

$$\psi_{N,n}(x_n) = i_n, \quad \text{jeśli} \quad (5.27)$$

$$\sum_{j_n=1}^M L(i_n, j_n) \cdot \delta_N(j_n, \bar{x}_n) < \sum_{j_n=1}^M L(r, j_n) \cdot \delta_N(j_n, \bar{x}_n),$$

$$r = 1, 2, \dots, M, \quad r \neq i_n, \quad n=1, 2, \dots$$

gdzie $\delta_N(\cdot)$ wyraża się jak w (3.4), jest asymptotycznie optymalny.

Dowód.

Położmy

$$r_{N,n}(i, \bar{x}_n) \triangleq \sum_{j_n=1}^M L(i, j_n) \cdot \delta_N(j_n, \bar{x}_n), \quad n=1, 2, \dots \quad (5.28)$$

Metodą indukcji udowodnimy obecnie, że dla każdego naturalnego n

$$\bigwedge_i \lim_{N \rightarrow \infty} P \left\{ |r_{N,n}(i, \bar{x}_n) - r_n(i, \bar{x}_n)| < \varepsilon \right\} = 1 \quad \text{pr. wsz. na } \bar{X}_n. \quad (5.29)$$

Pokażemy prawdziwość (5.29) dla $n=1$. Z (5.24), (5.26) oraz ze znanych własności zbieżności zmiennych losowych według prawdopodobieństwa [33] wynika, że dla każdego i

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N > \bar{N}} P \left\{ |r_{N,1}(i, \bar{x}_1) - r_1(i, \bar{x}_1)| < \varepsilon \right\} =$$

$$= P \left\{ \left| \sum_{j=1}^M L(i, j) \cdot p_{j, N} f_N(x/j) - \sum_{j=1}^M L(i, j) \cdot p_j \cdot f(x/j) \right| < \varepsilon \right\} \geq 1 - \delta$$

prawie wszędzie na X .

(5.30)

Zakładamy teraz, że dla dowolnego $i / i=1, 2, \dots, M /$ zachodzi

$$\lim_{N \rightarrow \infty} P \left\{ |r_{N, n}(i, \bar{x}_n) - r_n(i, \bar{x}_n)| < \varepsilon \right\} = 1 \text{ pr. wsz. na } \bar{X}_n, \quad (5.31)$$

czyli

$$\bigwedge_i \sum_{j=1}^M L(i, j) \cdot \delta_N(j, \bar{x}_n) \rightarrow \sum_{j=1}^M L(i, j) \cdot \delta(j, \bar{x}_n) \quad (5.32)$$

według prawdopodobieństwa i prawie wszędzie na \bar{X}_n .

Ponieważ (5.32) zachodzi dla dowolnej /byłe skończonej/ funkcji strat, wynika więc że:

$$\bigwedge_j \delta_N(j, \bar{x}_n) \rightarrow \delta(j, \bar{x}_n) \text{ według pr. i pr. wsz. na } \bar{X}_n. \quad (5.33)$$

Zbadamy obecnie prawdziwość (5.29) dla $n+1$. Z (5.25), (5.26) i (5.33) wnioskujemy, że dla każdego i

$$\begin{aligned} & \bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N > \bar{N}} P \left\{ |r_{N, n+1}(i, \bar{x}_{n+1}) - r_{n+1}(i, \bar{x}_{n+1})| < \varepsilon \right\} = \\ & = P \left\{ \left| \sum_{j=1}^M L(i, j) \cdot \delta_N(j, \bar{x}_{n+1}) - \sum_{j=1}^M L(i, j) \cdot \delta(j, \bar{x}_{n+1}) \right| < \varepsilon \right\} \geq \\ & \geq P \left\{ M \cdot L \sum_{j=1}^M \delta_N(j, \bar{x}_{n+1}) - \sum_{j=1}^M \delta(j, \bar{x}_{n+1}) \right\} = \\ & = P \left\{ M \cdot L \left[\sum_{j=1}^M f_N(\bar{x}_{n+1}/j) \sum_{k=1}^M p_{jk, N} \cdot \delta_N(k, \bar{x}_n) - \sum_{j=1}^M f(x_{n+1}/j) \cdot \right. \right. \\ & \left. \left. \sum_{k=1}^M p_{jk} \cdot \delta(k, \bar{x}_n) \right] < \varepsilon \right\} \geq 1 - \delta \end{aligned} \quad (5.34)$$

prawie wszędzie na \bar{X}_n i dla prawie wszystkich x_{n+1} , a więc prawie wszędzie na \bar{X}_{n+1} .

Wykazaliśmy zatem (5.29), co kończy indukcyjny dowód.

Są więc spełnione założenia twierdzenia 1.

c.b.d.o.

Z twierdzenia Bernoulliego [26] wynika, że dla łańcuchów nieprzywiedlnych oszacowania (3.3) są zbieżne według prawdopodobieństwa do odpowiednich prawdopodobieństw początkowych i przejścia prostego łańcucha Markowa. Stąd oraz z powyższego twierdzenia wnioskujemy, że w przypadku, gdy estymator $f_N(x/i)$ nieznannej gęstości $f(x/i)$ jest zbieżny prawie wszędzie według prawdopodobieństwa, to ryzyko zdąża do ryzyka Bayesa, a więc algorytmy rozpoznawania z uczeniem dla prostych, nieprzywiedlnych, stacjonarnych, jednorodnych łańcuchów Markowa: (3.11), (3.17), (3.28), (3.34) oraz w pewnych sytuacjach (3.51) są - dla przypadku warunkowych gęstości ciągłych prawie wszędzie - asymptotycznie optymalne.

Zbadamy obecnie asymptotyczne własności algorytmu "najbliższa średnia" (3.46).

TWIERDZENIE 3.

Niech $p_{i,N}$ i $p_{ij,N}$ będą oszacowaniami prawdopodobieństw początkowych p_i oraz prawdopodobieństw przejścia p_{ij} jednorodnego, prostego łańcucha Markowa, takimi że:

$$\bigwedge_i p_{i,N} \rightarrow p_i \quad \text{według prawdopodobieństwa,} \quad (5.35)$$

$$\bigwedge_{i,j} p_{ij,N} \rightarrow p_{ij} \quad \text{według prawdopodobieństwa.} \quad (5.36)$$

Niech poszczególne składowe wektora pomiarów $x = (x^{(1)}, \dots, x^{(k)})$ mają jednakowe w danej klasie i rozkłady prawdopodobieństwa o skończonych wartościach średnich $m_i^{(s)} / i=1,2,\dots,M, s=1,2,\dots,k$.

Wtedy dla zerojedynkowej funkcji strat zmodyfikowany algorytm "najbliższa średnia" (3.46) jest zbieżny.

Dowód.

Oznaczmy

$$\tilde{x}_{N,i}^{(s)} = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{j,i}^{(s)}, \quad i=1,2,\dots,M, \quad s=1,2,\dots,k \quad (5.37)$$

- średnia arytmetyczna s -tych składowych pomiarów w ciągu uczącym o długości N z klasy i .

Z twierdzenia Chinczyna [10] mamy

$$\tilde{x}_{N,i}^{(s)} \rightarrow m_i^{(s)} \quad \text{według pr., } i=1,2,\dots,M, \quad s=1,\dots,k. \quad (5.38)$$

Niech $d(\cdot)$ będzie odległością euklidesową w k -wymiarowej przestrzeni wektorowej X

$$d(x_1, x_2) = \sqrt{\sum_{s=1}^k (x_1^{(s)} - x_2^{(s)})^2} \quad (5.39)$$

Z nierówności trójkąta

$$\begin{aligned} |d(x_1, x_2) - d(x_1, x_3)| &\leq |d(x_1, x_3) + d(x_2, x_3) - d(x_1, x_3)| = \\ &= d(x_2, x_3) \quad (5.40) \end{aligned}$$

Z (5.38) oraz (5.40) wynika, że

$$\bigwedge_i \bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \left| d(x, \tilde{x}_{N_1, i}) - d(x, \tilde{x}_{N_2, i}) \right| < \delta \right\} \geq 1 - \delta, \quad (5.41)$$

gdzie

$$\tilde{x}_{N, i} = (\tilde{x}_{N, i}^{(1)}, \dots, \tilde{x}_{N, i}^{(k)})$$

Mamy pokazać, że dla algorytmu M-NM

$$\bigwedge_n \lim_{N \rightarrow \infty} R_n[\Psi_{N, n}(x_n)] = R_n \quad (5.42)$$

Z twierdzenia Lebesgue'a / i z tego, że funkcja strat jest zerojedynkowa / wynika, że aby zachodziło (5.42) wystarcza aby

$$\bigwedge_n \lim_{N \rightarrow \infty} E r_n(\Psi_{N, n}(x_n), \bar{x}_n) = r_n(\bar{x}_n) \quad \text{pr. wsz. na } \bar{X}_n \quad (5.43)$$

Skorzystamy z kryterium Cauchy'ego [33] zbieżności według średniej i pokażemy, że

$$\bigwedge_n \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} E \left| r_n(\Psi_{N_1, n}(x_n), \bar{x}_n) - r_n(\Psi_{N_2, n}(x_n), \bar{x}_n) \right|^2 < \xi \quad (5.44)$$

pr. wsz. na \bar{X}_n .

Z nierówności Schwarz'a otrzymujemy

$$\begin{aligned} E \left| r_n(\Psi_{N_1, n}(x_n), \bar{x}_n) - r_n(\Psi_{N_2, n}(x_n), \bar{x}_n) \right|^2 &= \\ &= E \left| \sum_{j=1}^M [L(\Psi_{N_1, n}(x_n), j) - L(\Psi_{N_2, n}(x_n), j)] \cdot \delta(j, \bar{x}_n) \right|^2 \leq \\ &\leq E \left\{ \sum_{j=1}^M |L(\Psi_{N_1, n}(x_n), j) - L(\Psi_{N_2, n}(x_n), j)| \cdot \delta(j, \bar{x}_n) \right\}^2 \leq \end{aligned}$$

$$\leq M \cdot E \left\{ \sum_{j=1}^M \left| L(\Psi_{N_1, n}(x_n), j) - L(\Psi_{N_2, n}(x_n), j) \right|^2 \cdot \delta^2(j, \bar{x}_n) \right\}. \quad (5.45)$$

Dla zerojedynkowej funkcji strat

$$E \left| L(\Psi_{N_1, n}(x_n), j) - L(\Psi_{N_2, n}(x_n), j) \right| \leq 1 \cdot P \left\{ \Psi_{N_1, n}(x_n) \neq \Psi_{N_2, n}(x_n) \right\}.$$

Zatem

$$E \left| r_n(\Psi_{N_1, n}(x_n), \bar{x}_n) - r_n(\Psi_{N_2, n}(x_n), \bar{x}_n) \right|^2 \leq M^3 \cdot P \left\{ \Psi_{N_1, n}(x_n) \neq \Psi_{N_2, n}(x_n) \right\}. \quad (5.46)$$

Metodą indukcji udowodnimy, że dla każdego i i dowolnego n zachodzi

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \left| \delta_{N_1}(i, \bar{x}_n) - \delta_{N_2}(i, \bar{x}_n) \right| < \varepsilon \right\} \geq 1 - \delta \quad (5.47)$$

prawie wszędzie na \bar{X}_n .

Dla $n=1$ z (5.41) mamy dla każdego i

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \left| \frac{1}{d(x, \tilde{x}_{N_1, i})} - \frac{1}{d(x, \tilde{x}_{N_2, i})} \right| < \varepsilon \right\} \geq 1 - \delta \quad (5.48)$$

dla każdego $x \neq \tilde{x}_{N_1, i}, \tilde{x}_{N_2, i}$, a więc pr. wsz. na X .

Zakładamy teraz, że zachodzi

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \left| \delta_{N_1, n}(i, \bar{x}_n) - \delta_{N_2, n}(i, \bar{x}_n) \right| < \varepsilon \right\} \geq 1 - \delta \quad (5.49)$$

dla każdego i i prawie wszędzie na \bar{X}_n

i badamy prawdziwość (5.47) dla $n+1$. Z (5.35), (5.36), (5.41) i

(5.49) wnioskujemy, że dla każdego i

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \left| \delta_{N_1, n+1}(i, \bar{x}_{n+1}) - \delta_{N_2, n+1}(i, \bar{x}_{n+1}) \right| < \varepsilon \right\} =$$

$$= P \left\{ \left| \frac{\sum_{j=1}^M p_{ij, N_1} \cdot \delta_{N_1}(j, \bar{x}_n)}{d(x_{n+1}, \tilde{x}_{N_1, i}) \cdot p_{i, N_1}} - \frac{\sum_{j=1}^M p_{ij, N_2} \cdot \delta_{N_2}(j, \bar{x}_n)}{d(x_{n+1}, \tilde{x}_{N_2, i}) \cdot p_{i, N_1}} \right| < \varepsilon \right\} \geq 1 - \delta \quad (5.50)$$

pr. wszędzie na \bar{X}_n i dla każdego $x_{n+1} \neq \tilde{x}_{N_1, i}, \tilde{x}_{N_2, i}$, a więc prawie wszędzie na \bar{X}_{n+1} ,

co kończy indukcyjny dowód (5.47).

Niech teraz dla pewnego $N_2 > \bar{N}$ istnieje taki wskaźnik i_0 , że

$$\bigwedge_{i \neq i_0} \delta_{N_2}(i_0, \bar{x}_n) - \delta_{N_2}(i, \bar{x}_n) > 3 \cdot \varepsilon. \quad (5.51)$$

Z (5.21), (5.47) i ostatniej nierówności wynika, że

$$\begin{aligned} & \bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1 > \bar{N}} P \left\{ \delta_{N_1}(i_0, \bar{x}_n) - \delta_{N_1}(i, \bar{x}_n) > \varepsilon \right\} = \\ & = P \left\{ \left[\delta_{N_1}(i_0, \bar{x}_n) - \delta_{N_2}(i_0, \bar{x}_n) \right] + \left[\delta_{N_2}(i_0, \bar{x}_n) - \delta_{N_2}(i, \bar{x}_n) \right] + \right. \\ & \left. + \left[\delta_{N_2}(i, \bar{x}_n) - \delta_{N_1}(i, \bar{x}_n) \right] > \varepsilon \right\} \geq P \left\{ \left| \delta_{N_1}(i_0, \bar{x}_n) - \delta_{N_2}(i_0, \bar{x}_n) \right| < \varepsilon, \right. \end{aligned}$$

$$\left. \delta_{N_2}(i_0, \bar{x}_n) - \delta_{N_2}(i, \bar{x}_n) > 3 \cdot \varepsilon, \left| \delta_{N_2}(i, \bar{x}_n) - \delta_{N_1}(i, \bar{x}_n) \right| < \varepsilon \right\} \geq$$

$$\geq (1 - \delta) + (1 - \delta) + (1 - \delta) - 2 = 1 - \delta \cdot 3. \quad (5.52)$$

Wykazaliśmy więc, że dla każdego n

$$\bigwedge_{\delta > 0} \bigvee_{\bar{N}} \bigwedge_{N_1, N_2 > \bar{N}} P \left\{ \Psi_{N_1, n}(x_n) \neq \Psi_{N_2, n}(x_n) \right\} < \delta \quad (5.53)$$

prawie wszędzie na \bar{X}_n .

Stąd poprzez (5.46) wynika (5.44), co kończy dowód.

c.b.d.o.

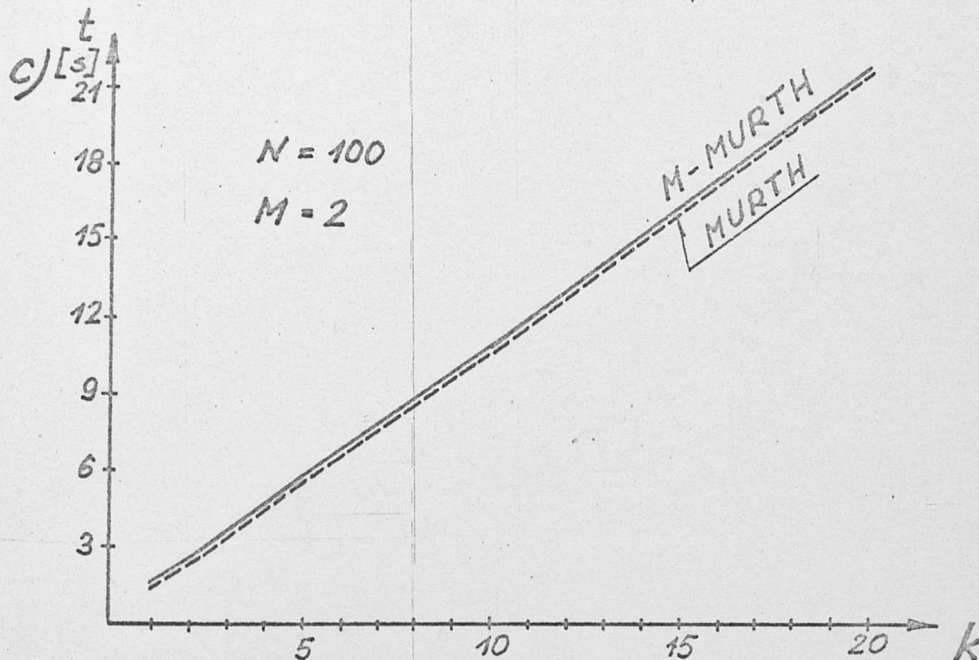
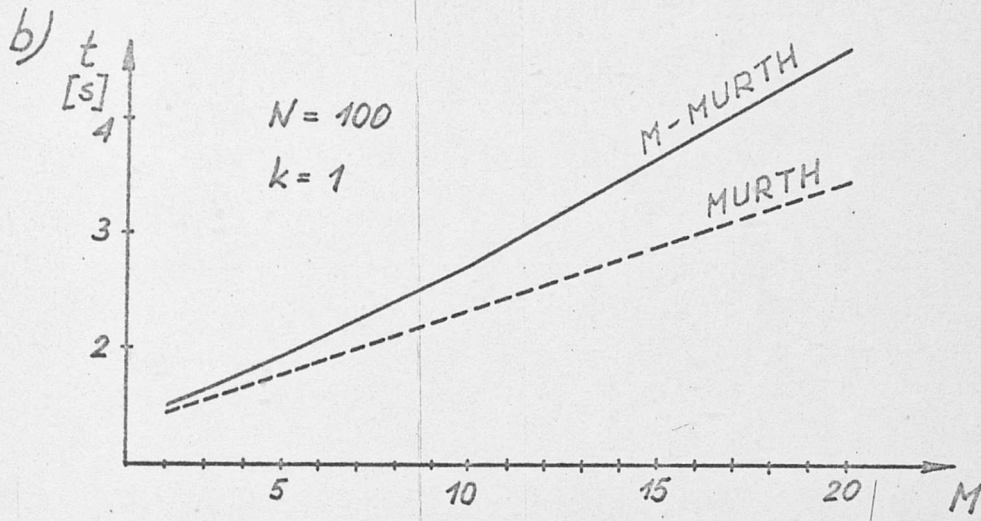
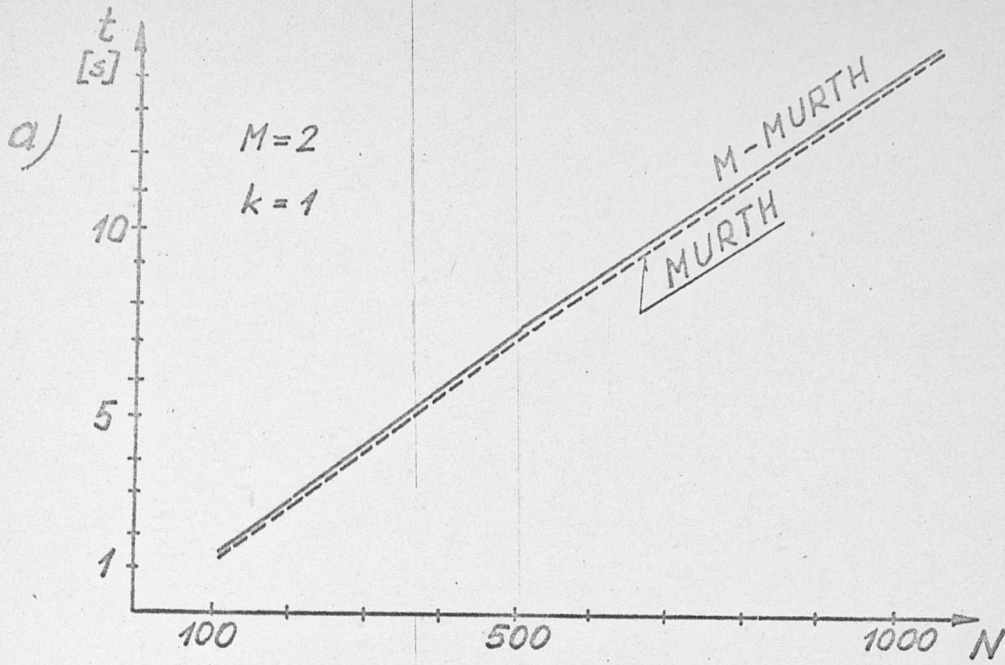
5.2. Wyniki eksperymentalnego badania algorytmów rozpoznawania.

W pierwszej części eksperymentu określono czasy klasyfikacji po jedyńczego obrazu dla badanych algorytmów. Na podstawie wyników przeprowadzonych obliczeń numerycznych na m.c. ODRA 1204 sporządzone zostały wykresy czasu klasyfikacji w funkcji długości ciągu uczącego, ilości klas i ilości wymiarów przestrzeni obrazów. (Rysunki 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7) Wynika z nich, że wyżej wymienione parametry mają istotny wpływ (z wyjątkiem, co jest zrozumiałe, czasów klasyfikacji algorytmu typu "najbliższa średnia" w funkcji długości ciągu uczącego) na czasy klasyfikacji badanych algorytmów.

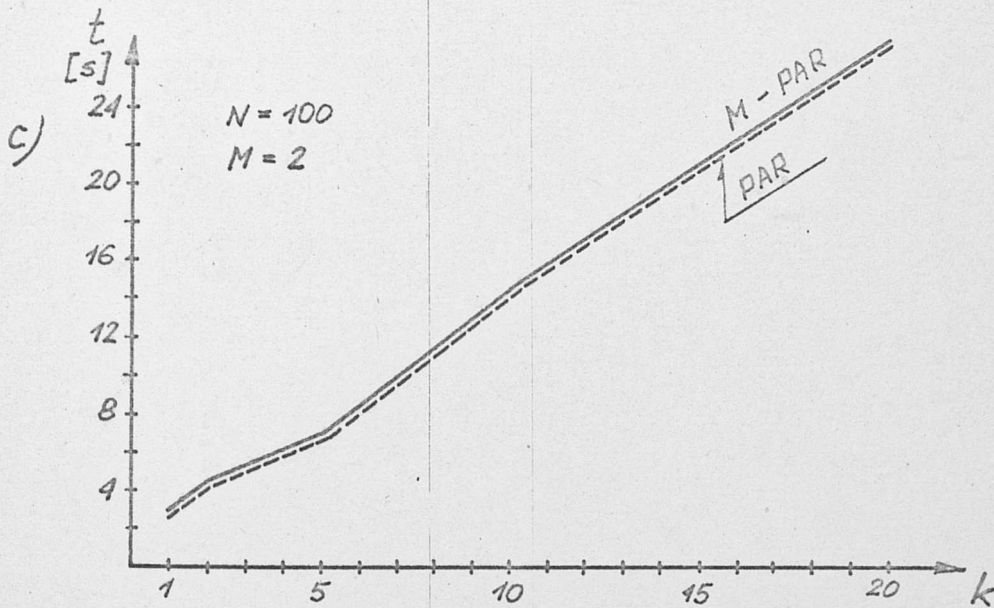
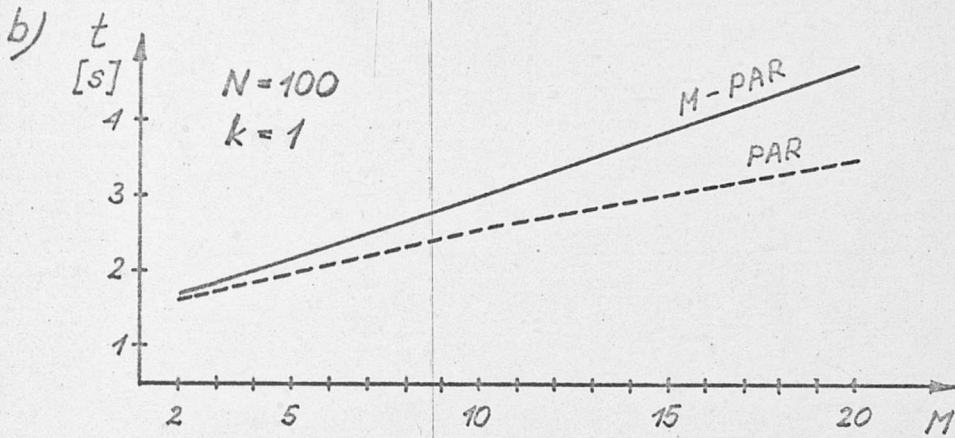
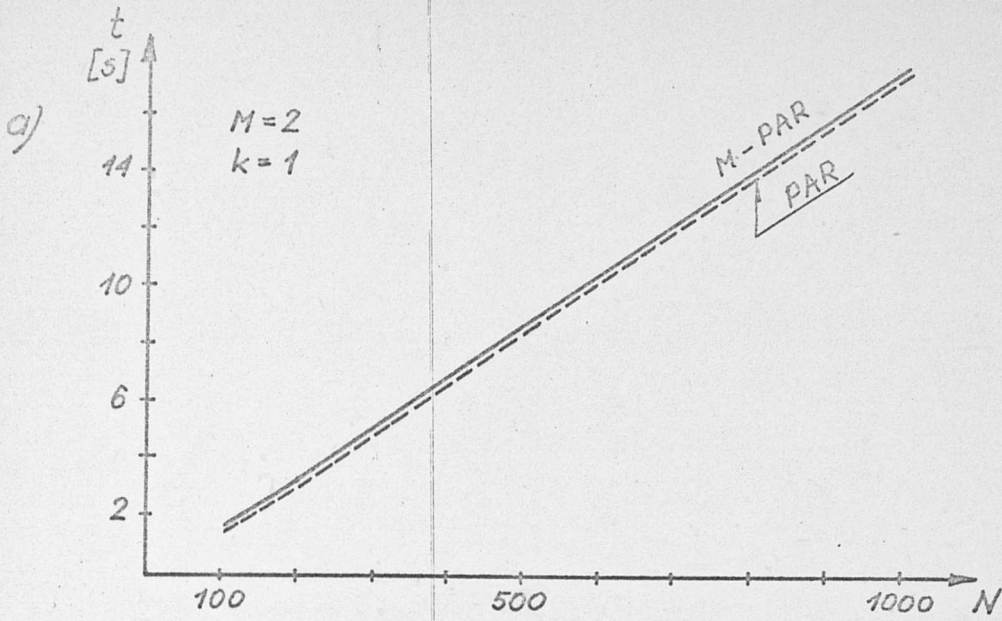
Widać ponadto, że zdecydowanie najszybszym algorytmem rozpoznawania prostych łańcuchów Markowa jest M-NM. W drugiej kolejności wymienić należy algorytm M-LI, natomiast w przypadku pozostałych czasy klasyfikacji są mniej więcej tego samego rzędu i wahają się w granicach od kilku do 30 sekund, a więc są stosunkowo długie.

Na szczególną uwagę zasługuje minimalna różnica między czasami klasyfikacji przy pomocy algorytmów uwzględniających zależność typu Markowa a ich wersjami niezmodyfikowanymi i rośnie ona jedynie w przypadku wzrostu liczby klas. Wynika to stąd, że czas modyfikacji algorytmu (obliczenie odpowiednich współczynników wagowych i wymnożenie przez nie funkcji decyzyjnych) jest znikomo mały w porównaniu z czasem "rzeczywistej" klasyfikacji, a wzrasta on wraz ze wzrostem ilości klas. Wyjątkiem jest tutaj algorytm typu "najbliższa średnia" (rys. 5.6), w którym czas modyfikacji jest nie tylko tego samego rzędu, co czas "rzeczywistej" klasyfikacji, a nawet w pewnych warunkach go przewyższa.

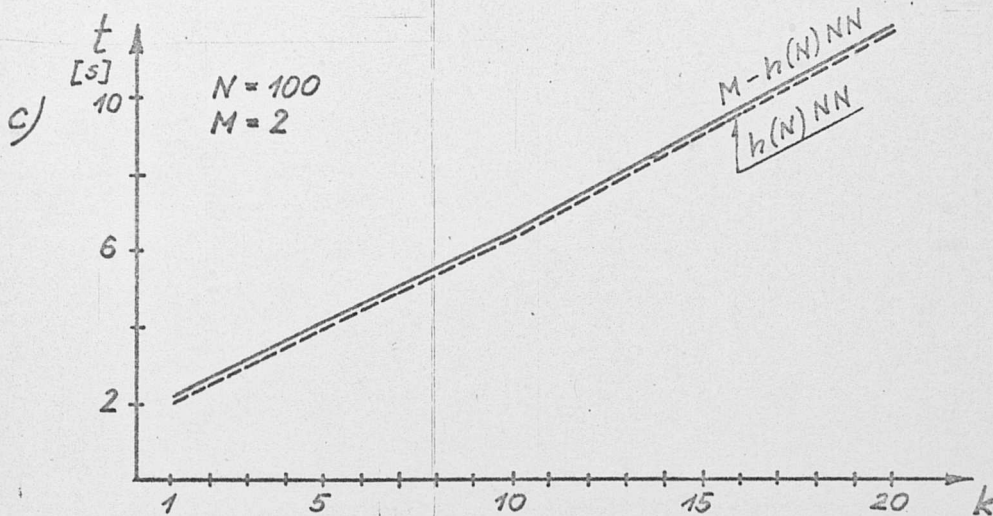
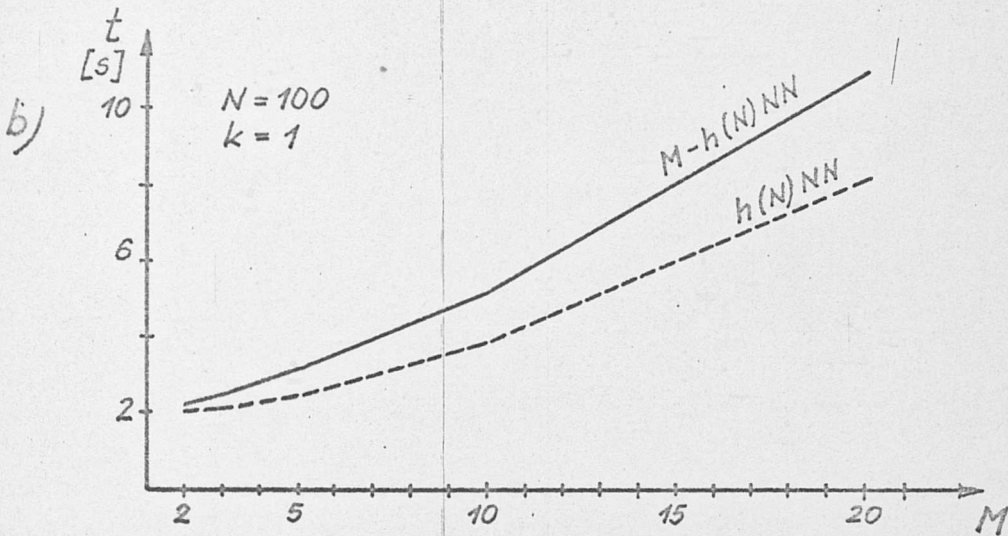
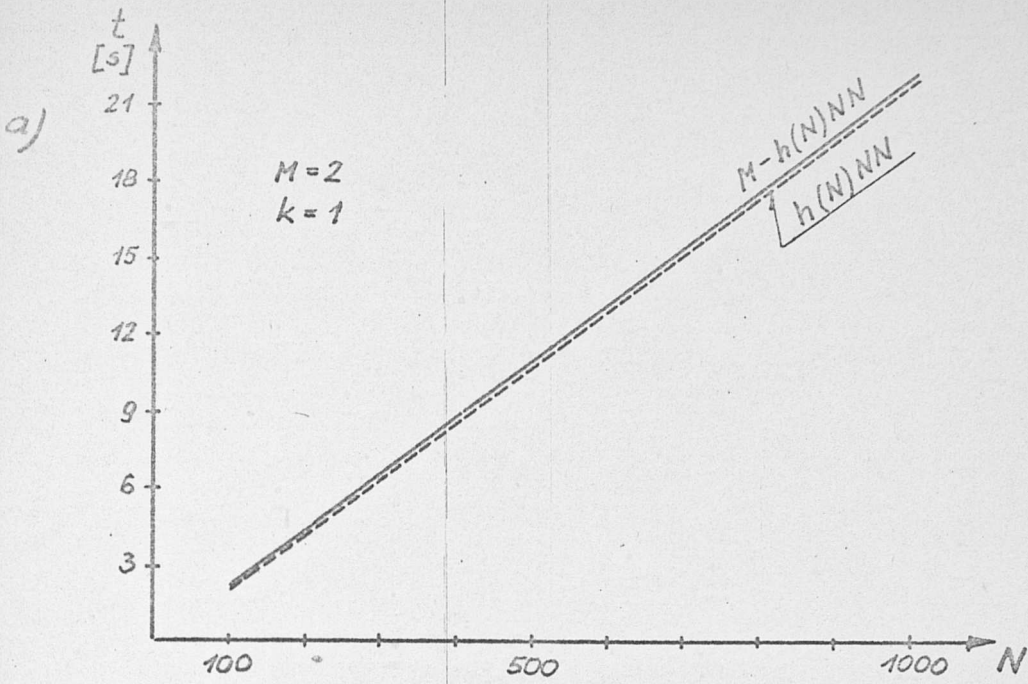
Podkreślić należy, że podane czasy dotyczą tylko klasyfikacji a nie uwzględniają czasu jednorazowego procesu uczenia (obliczanie obrazów średnich w przypadku algorytmu M-NM, odpowiednie uszeregowanie pomiarów uczących dla algorytmu M-LI itp.) poprzedzającego rozpo-



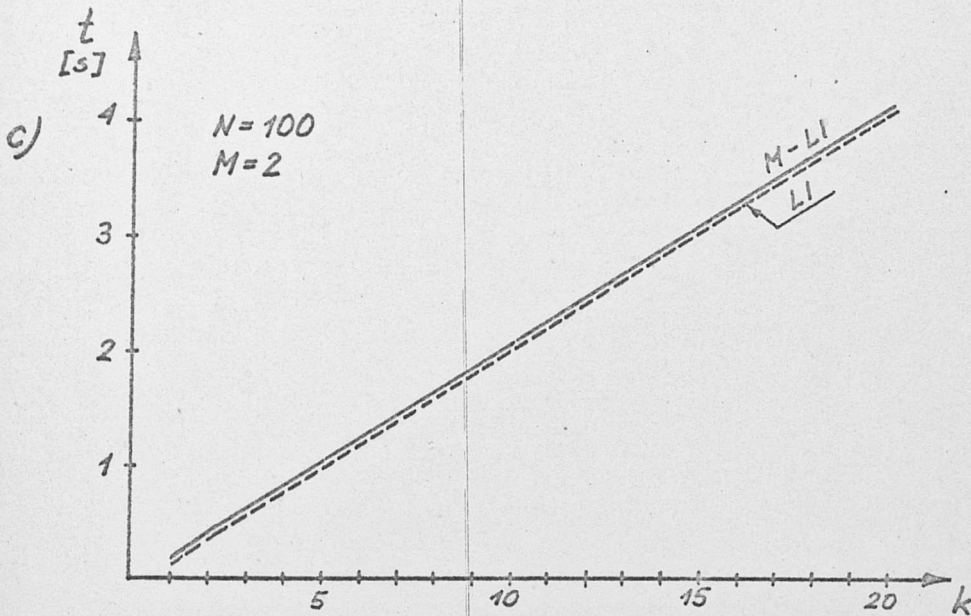
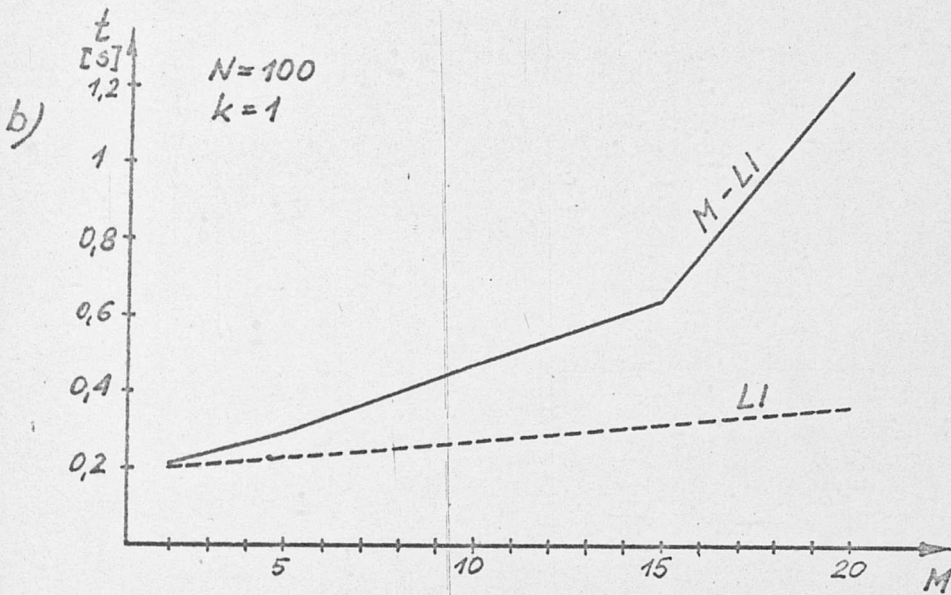
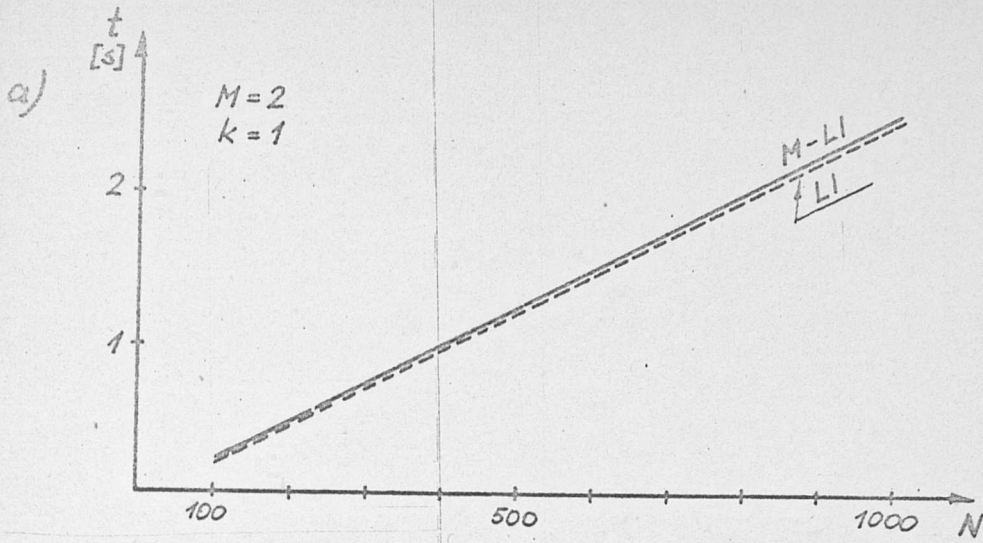
Rys. 5.1. Czasy klasyfikacji algorytmów M-MURTH i MURTH w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c)



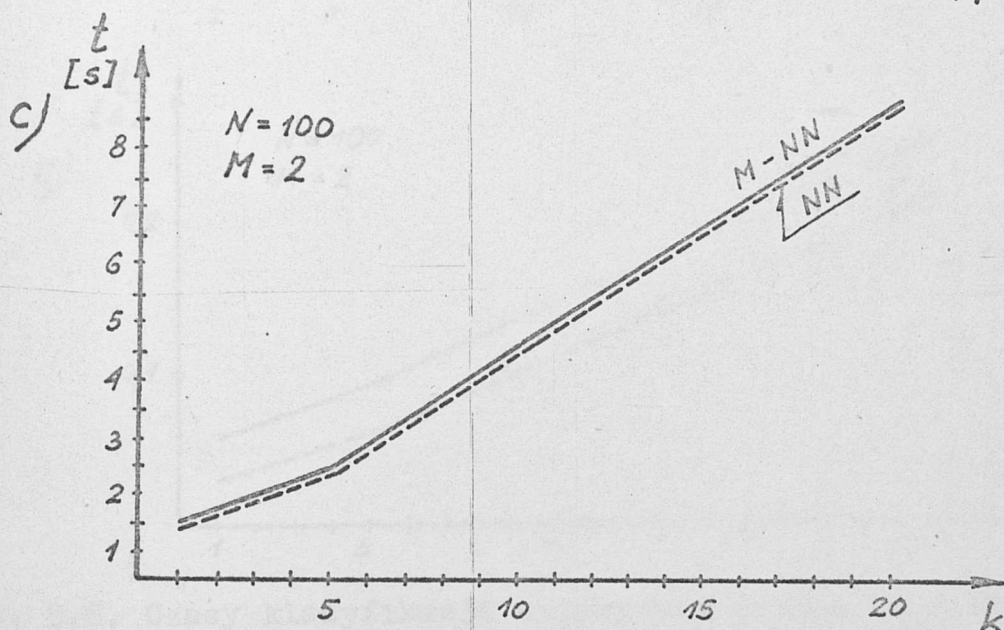
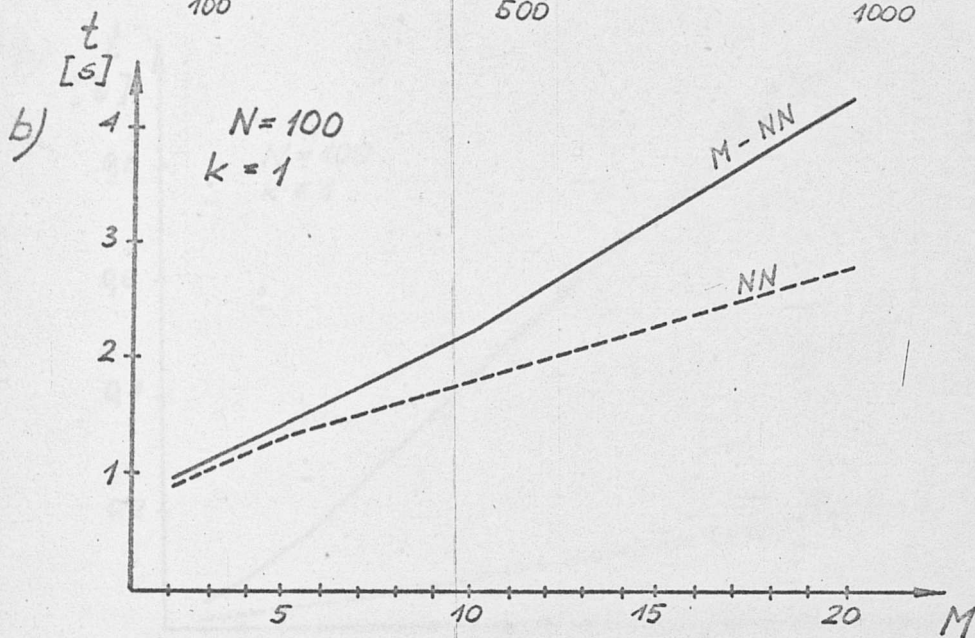
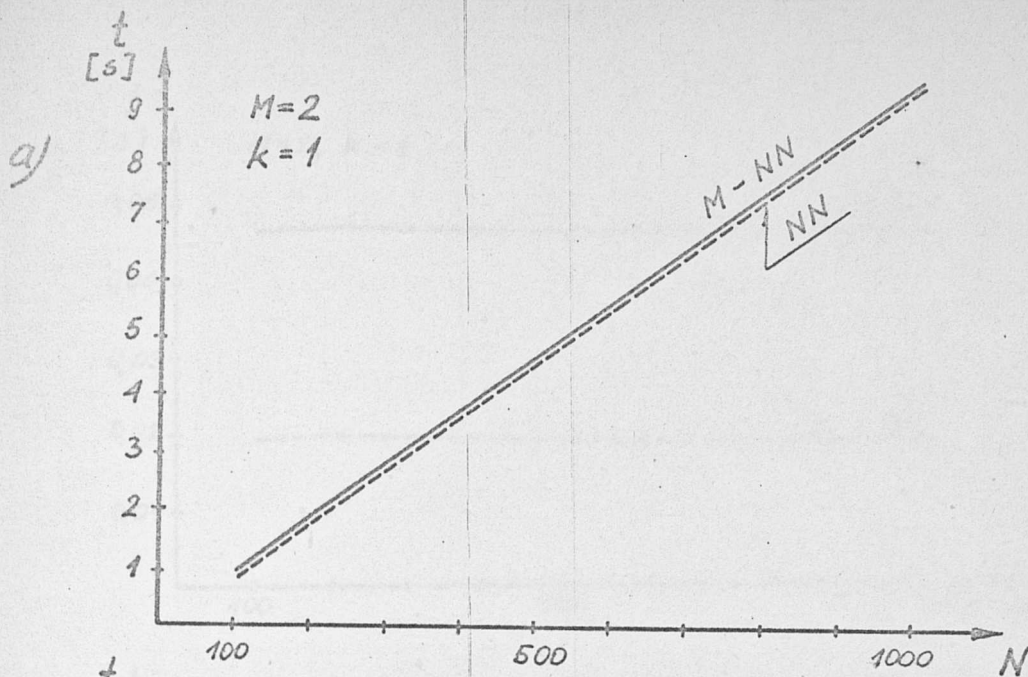
Rys. 5.2. Czasy klasyfikacji algorytmów M-PAR i PAR w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c)



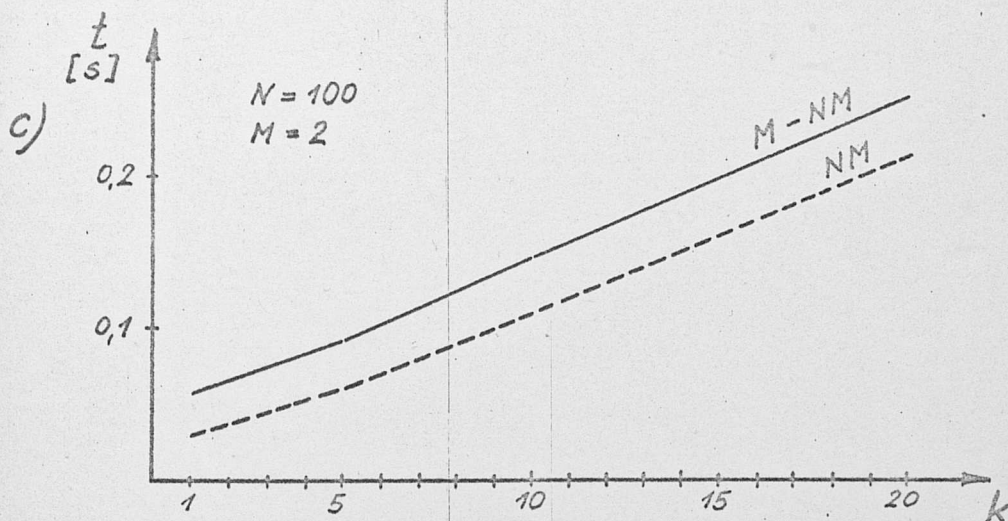
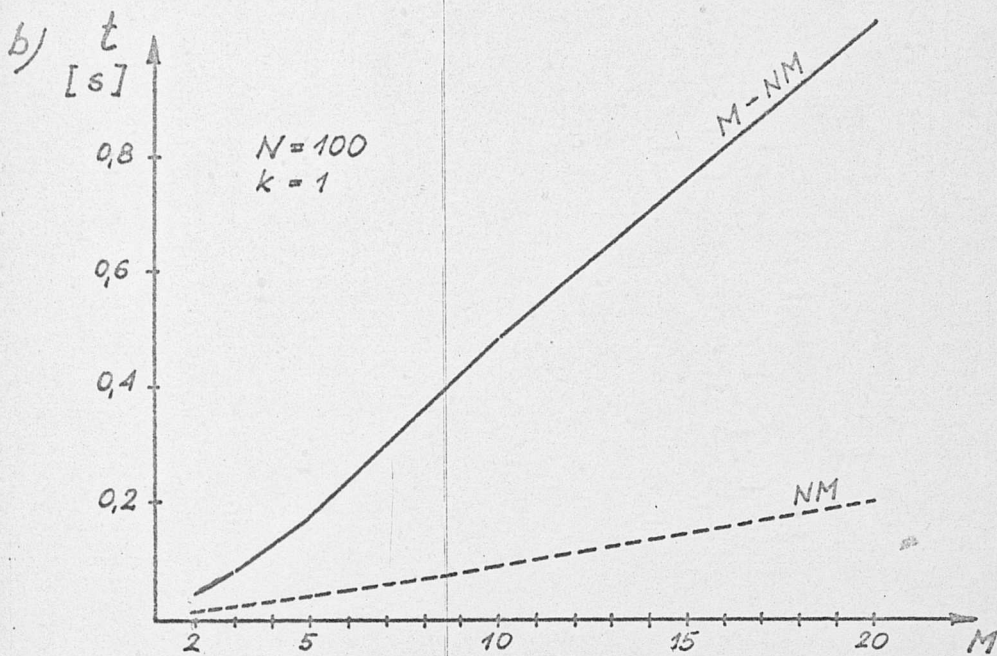
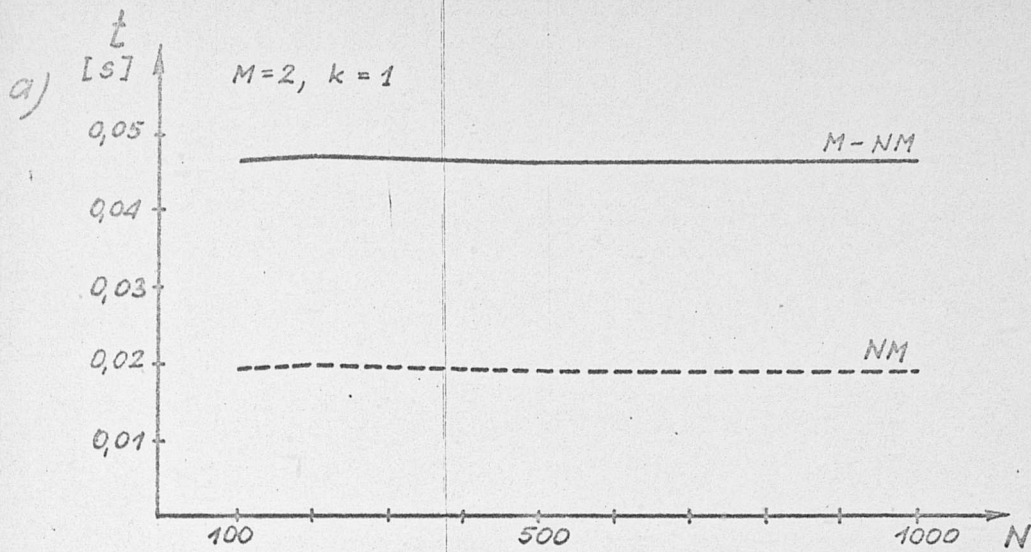
Rys. 5.3. Czasy klasyfikacji algorytmów $M-h(N)NN$ i $h(N)NN$ w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c).



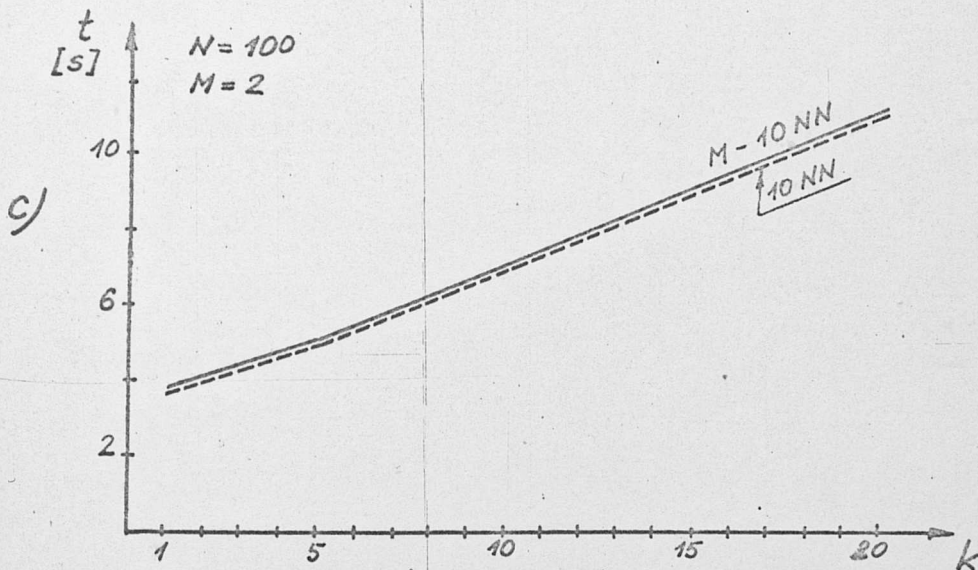
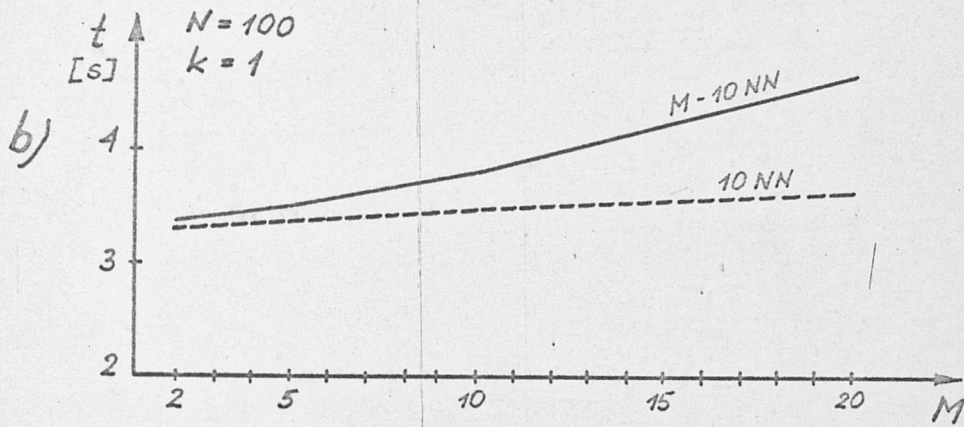
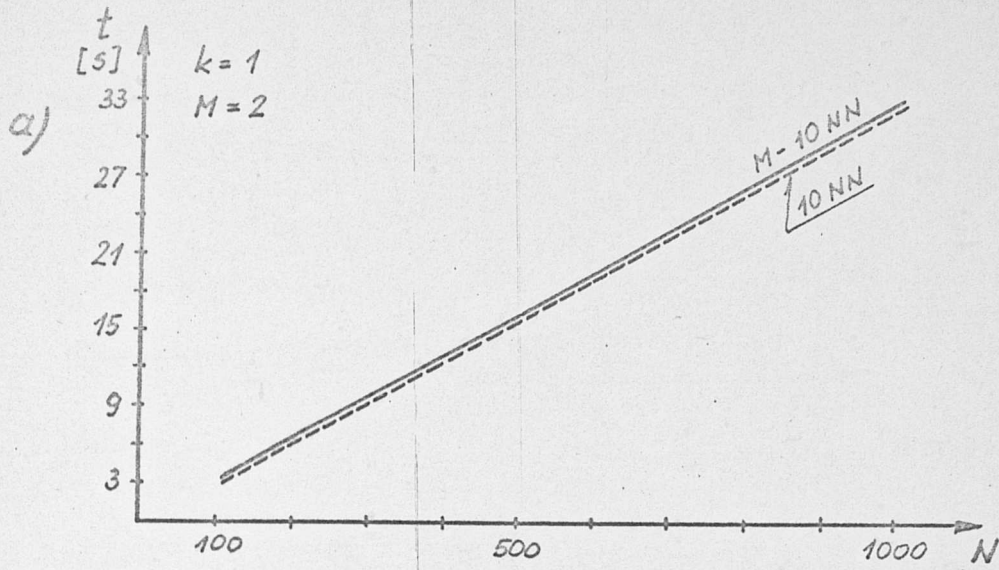
Rys. 5.4. Czasy klasyfikacji algorytmów M-LI i LI w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c)



Rys. 5.5. Czasy klasyfikacji algorytmów M-NN i NN w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c)



Rys. 5.6. Czasy klasyfikacji algorytmów M-NM i NM w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów (c)



Rys. 5.7. Czasy klasyfikacji algorytmów M-10NN i 10 NN w zależności od długości ciągu uczącego (a), ilości klas (b), ilości wymiarów obrazów. (c)

znawanie, a które to czasy są różne dla różnych algorytmów.

Badano również, jaki wpływ na czasy klasyfikacji algorytmów mają parametry łańcucha Markowa oraz parametry rozkładów warunkowych w poszczególnych klasach, które zadano normalne. Na podstawie szeregu przeprowadzonych eksperymentów stwierdzono, że parametry te nie wpływają na czasy klasyfikacji badanych algorytmów ani też na różnicę między nimi a czasami klasyfikacji wersji niezmodyfikowanych tych algorytmów.

W drugiej części badań eksperymentalnych określono wpływ parametrów warunkowych rozkładów prawdopodobieństwa na zachowanie się częstości błędnych klasyfikacji badanych algorytmów.

Sprawdzano przy tym kolejno:

1. wpływ różnicy wartości średnich rozkładów przy ustalonych ich dyspersjach na częstość błędnych decyzji,
2. wpływ wielkości dyspersji przy stałych wartościach średnich rozkładów.

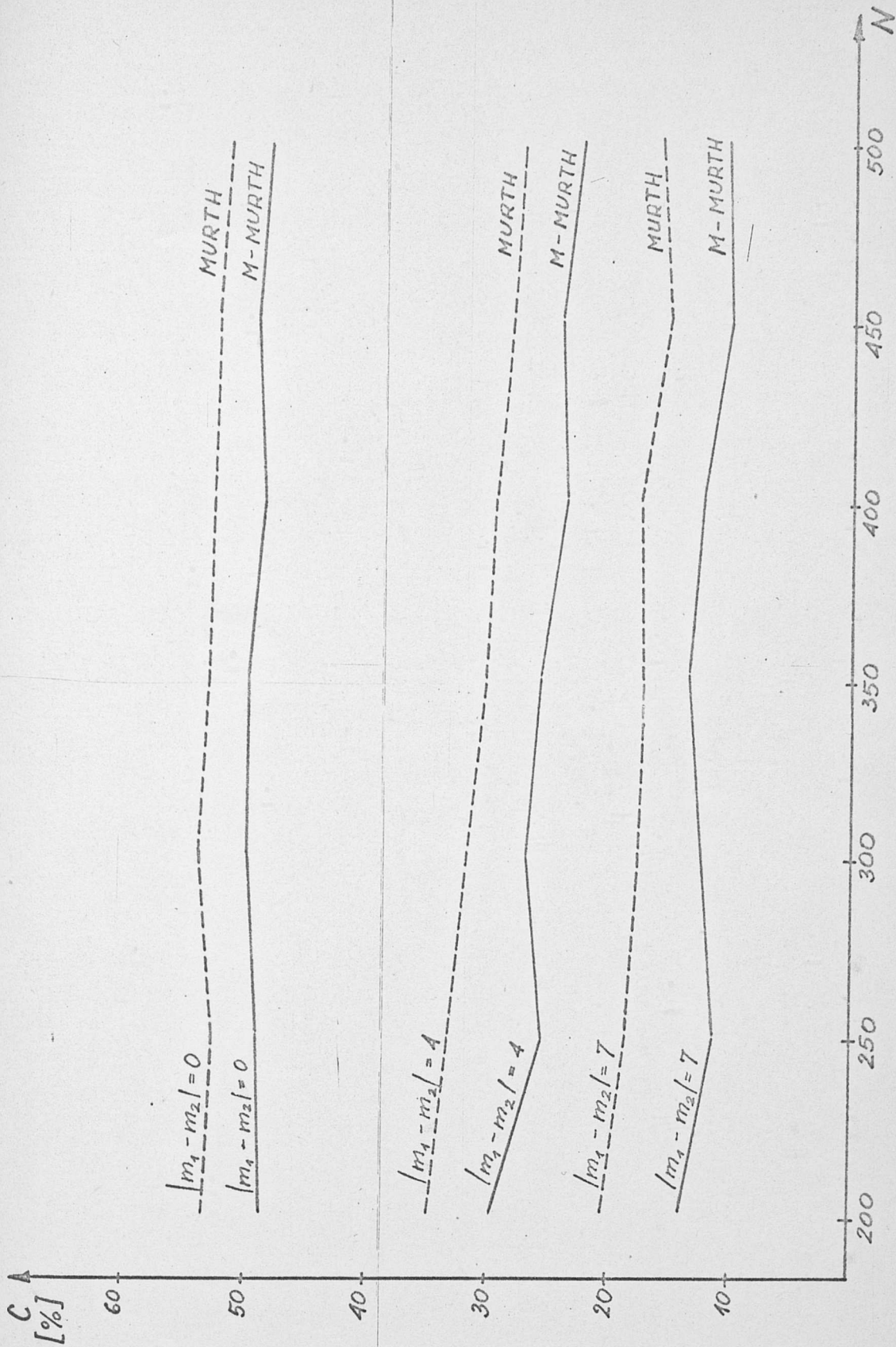
W punktach 1 - 2 przyjęto jednakowe charakterystyki probabilistyczne łańcucha Markowa:

$$p_1 = 0.33, \quad p_2 = 0.67, \quad P = \begin{bmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \end{bmatrix}.$$

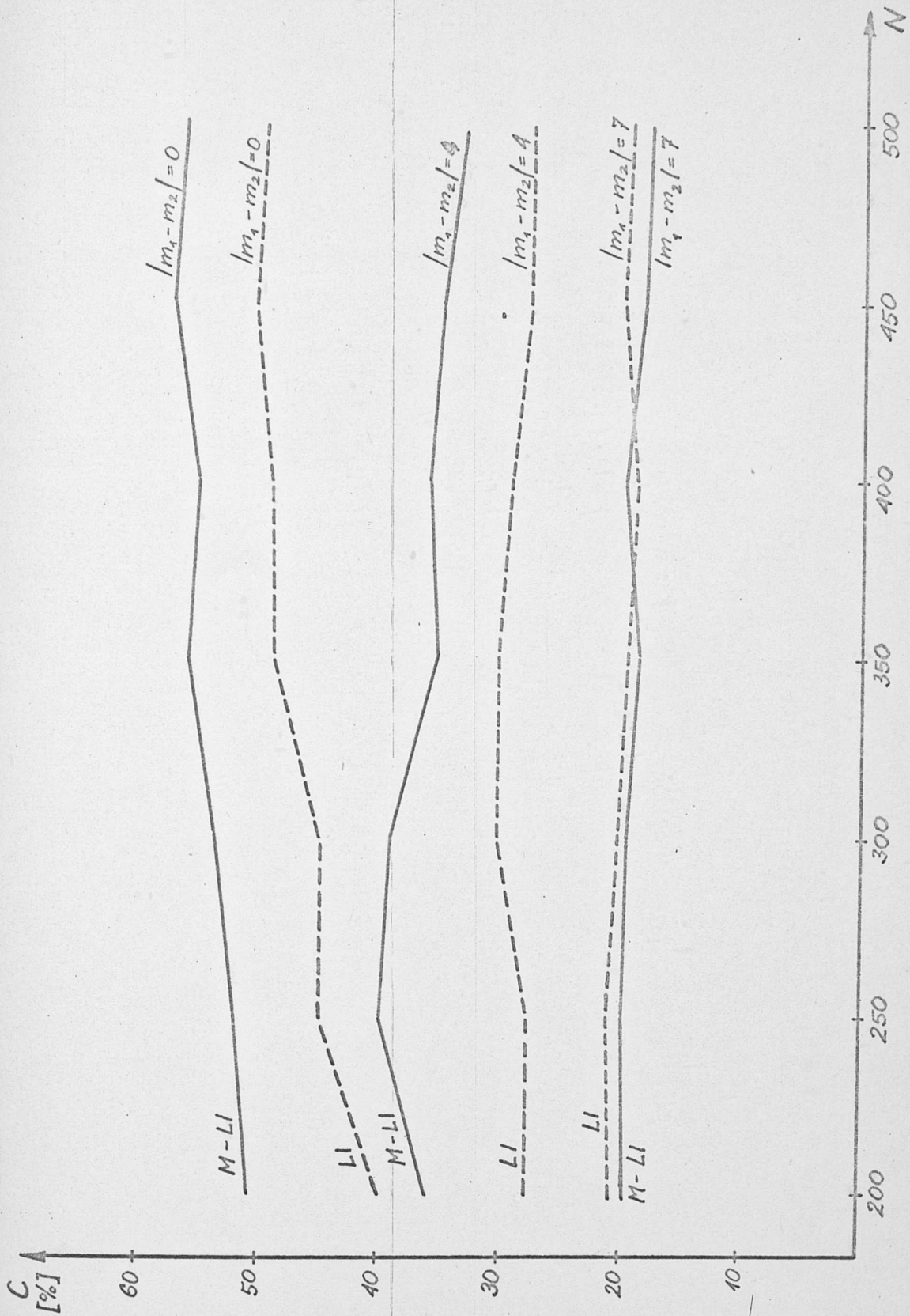
Rozpoznawano przy tym ciąg 500 obrazów przy różnych długościach ciągu uczącego ($N = 200, 250, 300, 350, 400, 450, 500$) i różnych parametrach warunkowych rozkładów prawdopodobieństwa. Uzyskane rezultaty przedstawione są na rysunkach 5.8 ÷ 5.17.

Wykresy 5.8 ÷ 5.12 ilustrują zachowanie się częstości c błędnych decyzji dla badanych algorytmów w przypadku, gdy przy ustalonych dyspersjach rozkładów prawdopodobieństwa zmienia się różnica wartości średnich tych rozkładów.

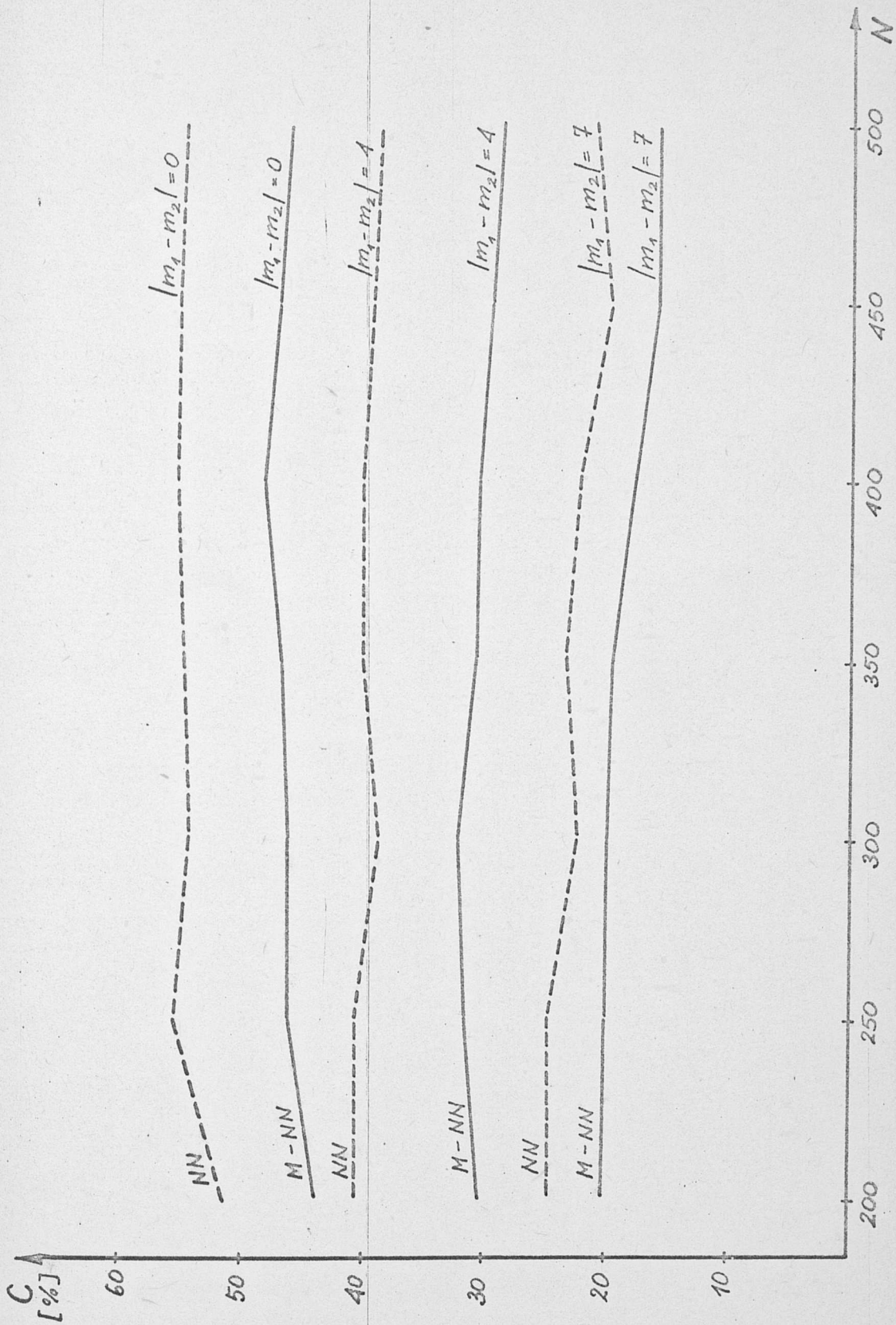
Wynika z nich, że przy ustalonych dyspersjach, częstości błędnych rozpoznań według wszystkich badanych algorytmów maleją ze wzrostem różnicy wartości średnich. Ponadto uzyskane wyniki eksperymentalne



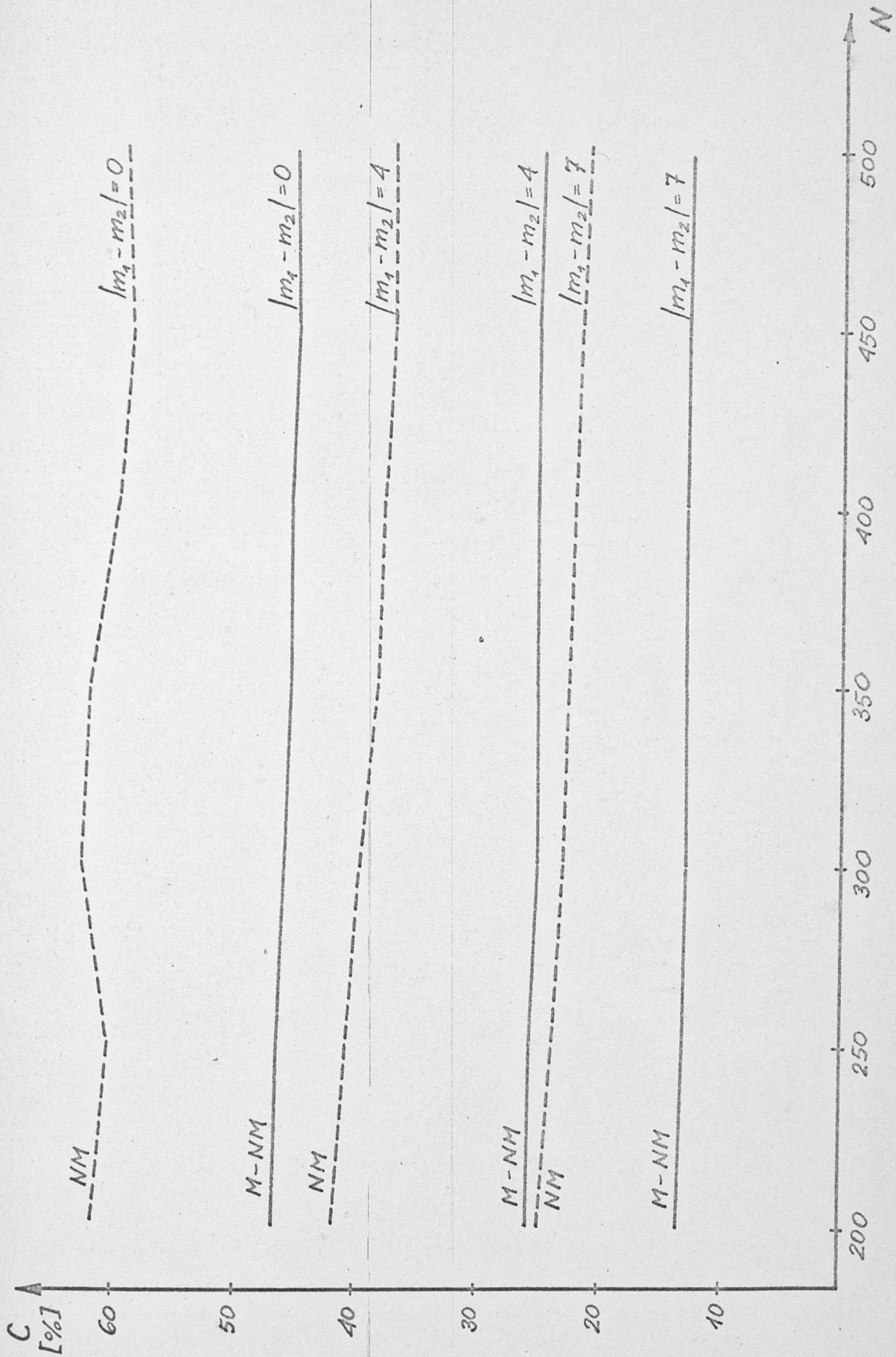
Rys. 5.8. Częstość błędnych klasyfikacji według algorytmów MURTH i M-MURTH przy różnych $|m_1 - m_2|$ i $\delta_1 = \delta_2 = 3$



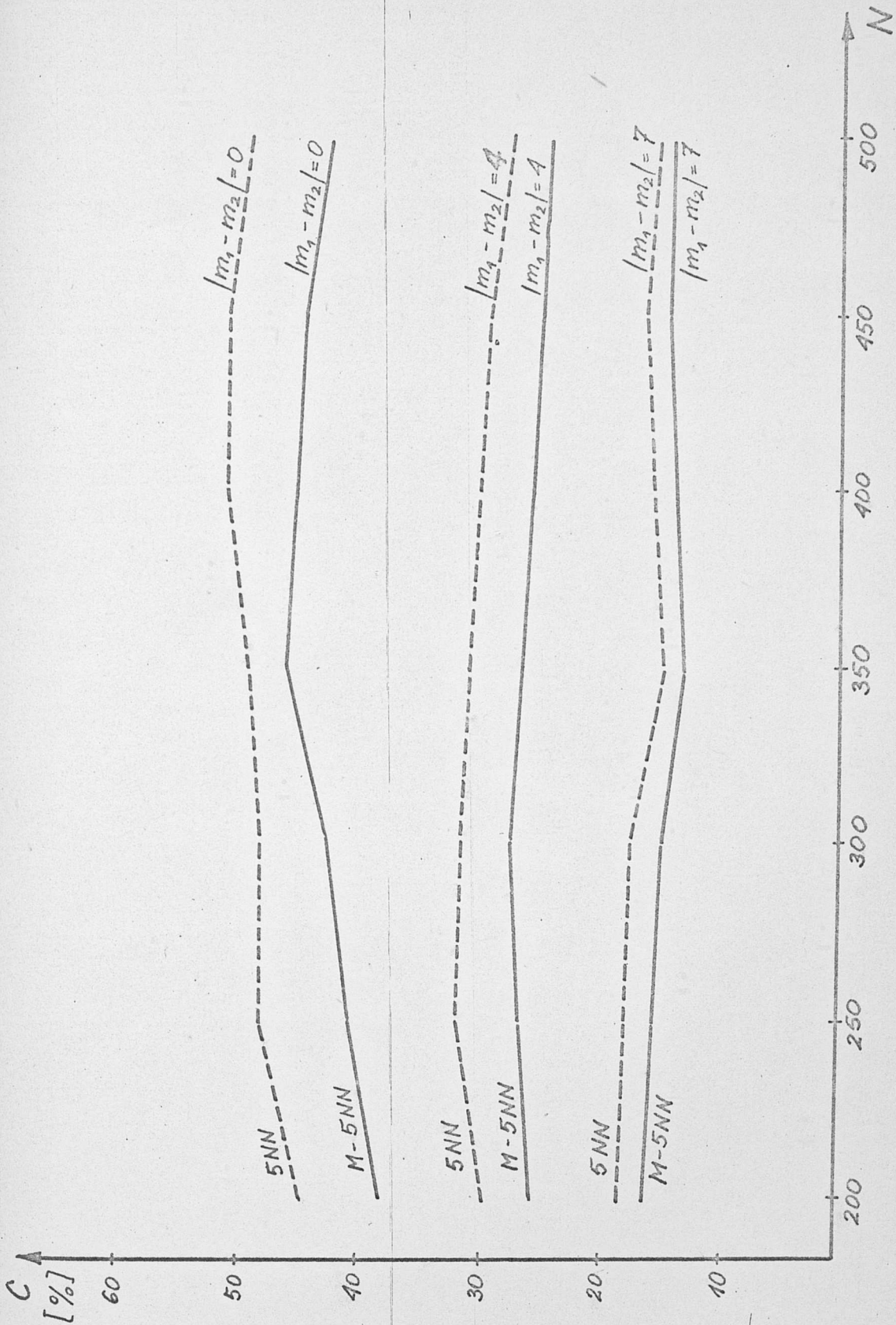
Rys. 5.9. Częstości błędnych klasyfikacji według algorytmów LI i M-LI przy różnych $|m_1 - m_2|$ i $\delta_1 = \delta_2 = 3$



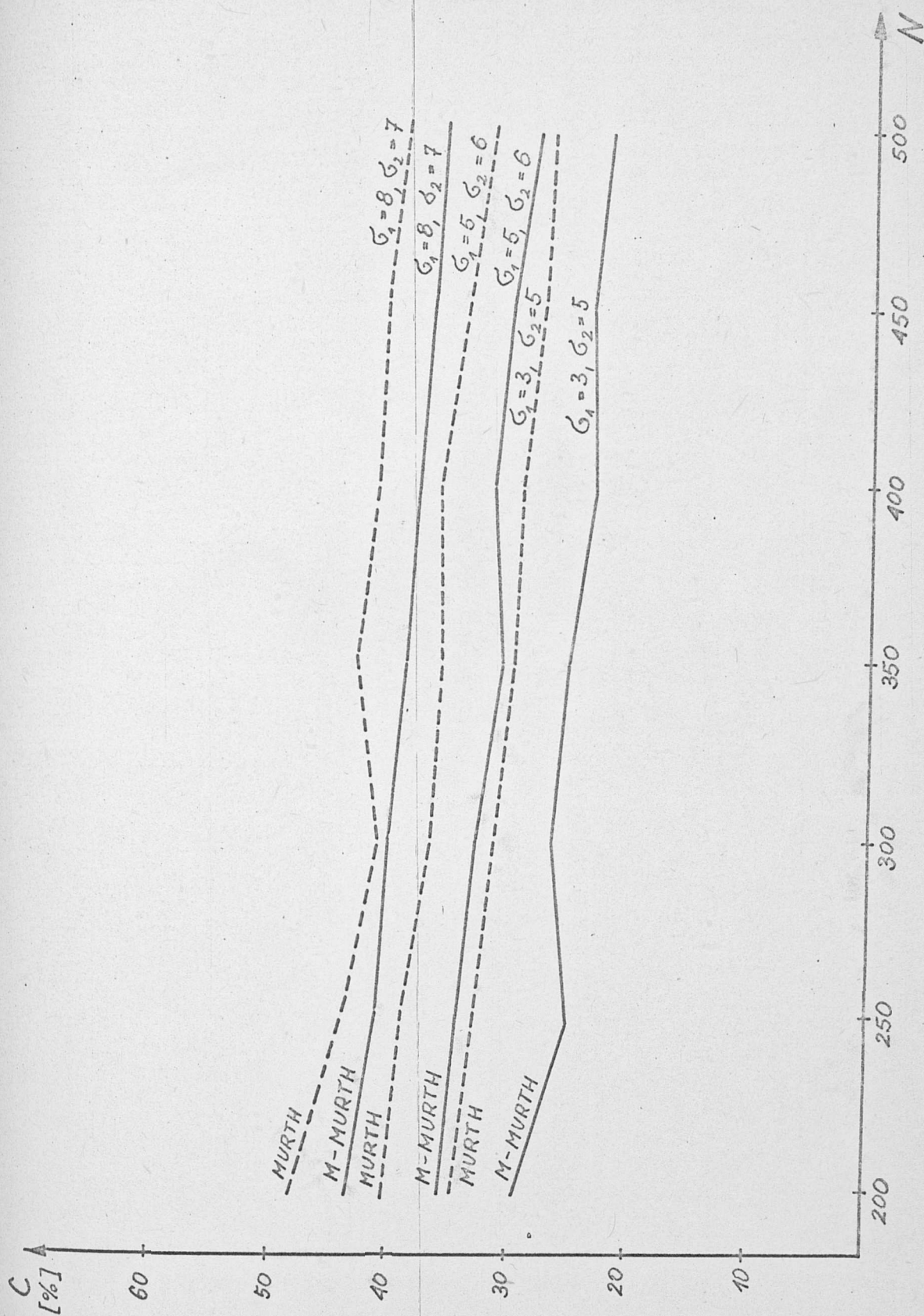
Rys. 5.10. Częstości błędnych klasyfikacji według algorytmów NN i M-NN przy różnych $|m_1 - m_2|$ i $\delta_1 = \delta_2 = 3$



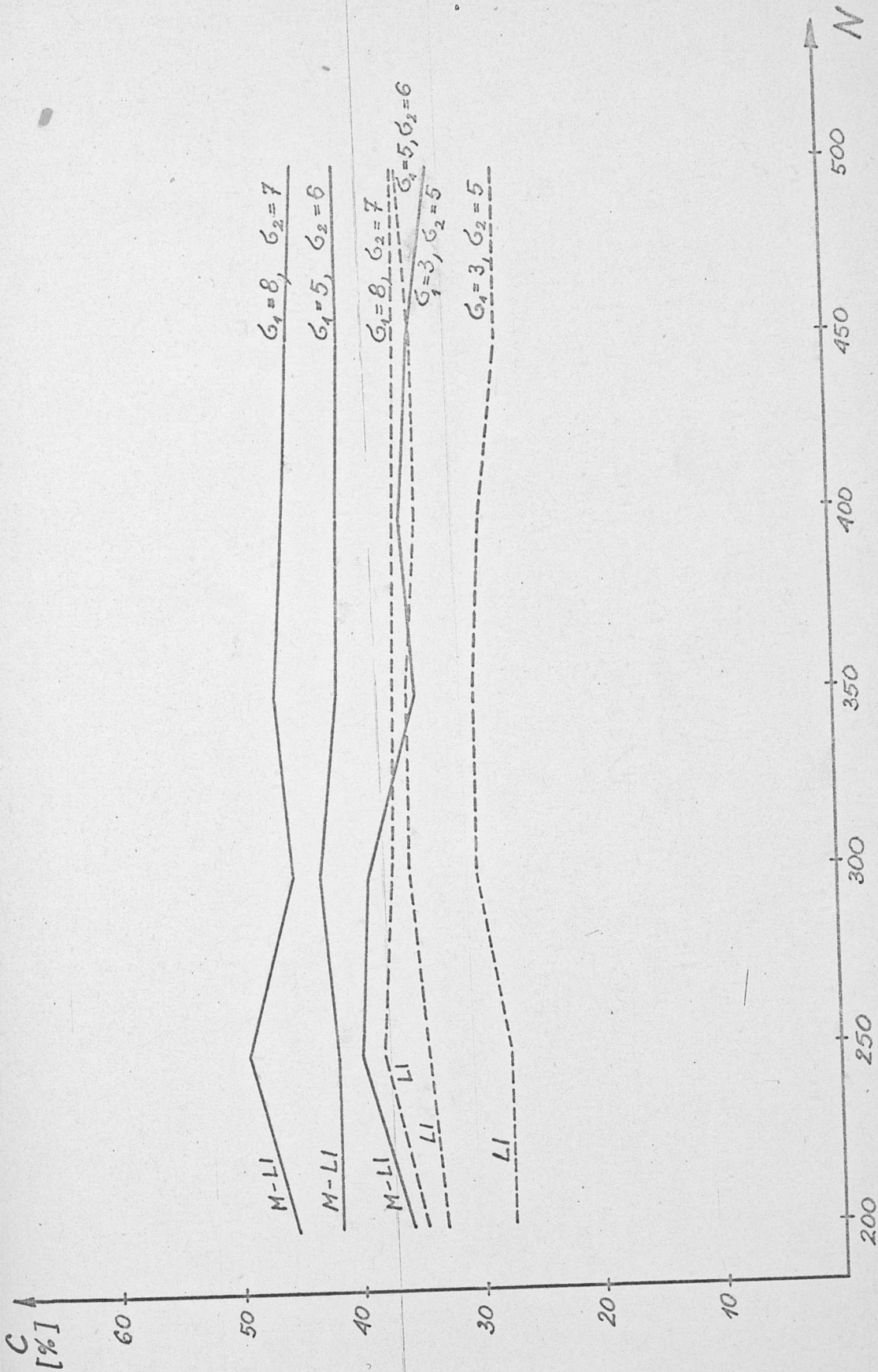
Rys. 5.11. Częstości błędnych klasyfikacji według algorytmów NM i M-NM przy różnych $|m_1 - m_2|$ i $\delta_1 = \delta_2 = 3$



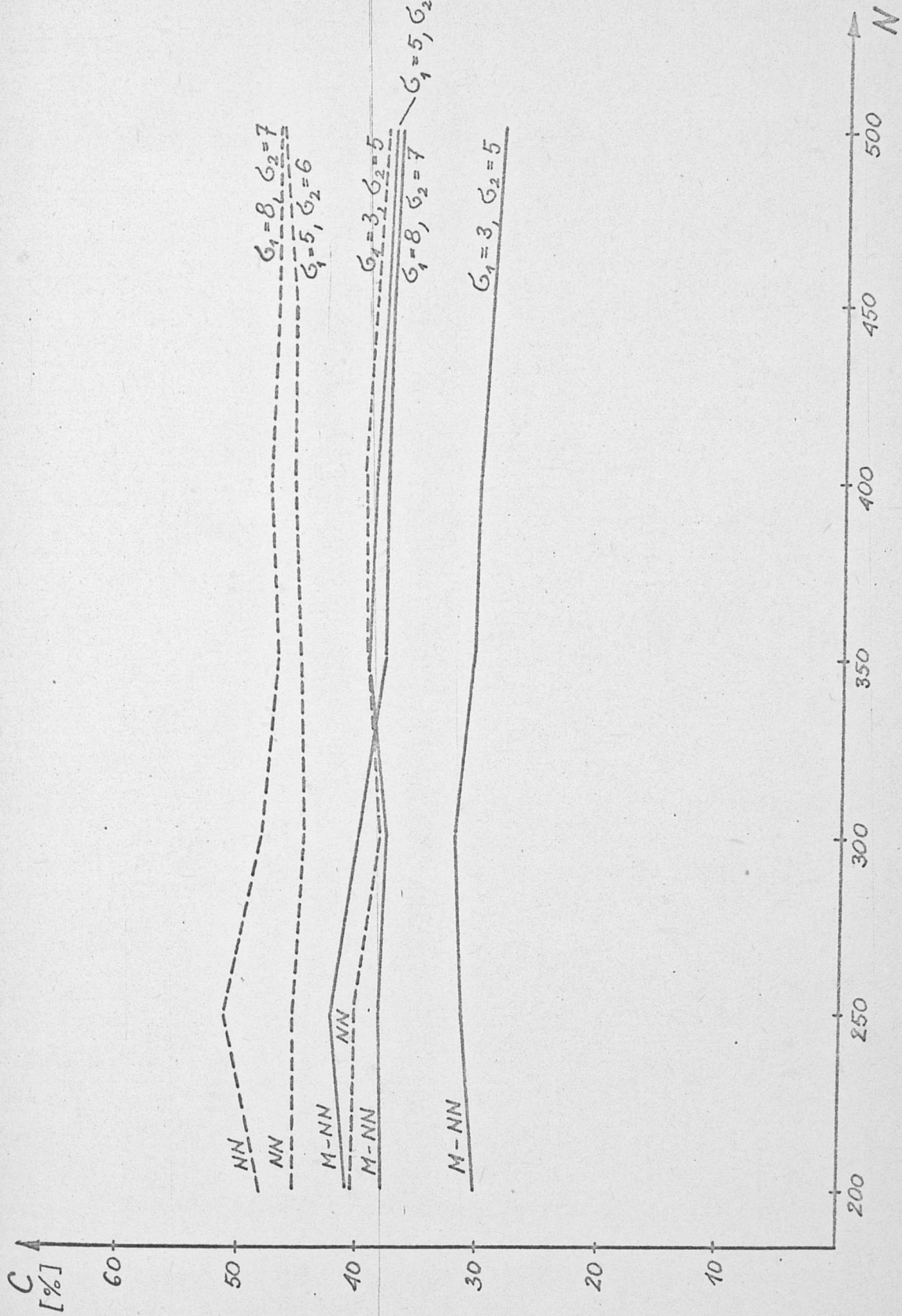
Rys. 5.12. Częstości błędnych klasyfikacji według algorytmów 5NN i M-5NN przy różnych $|m_1 - m_2|$ i $\delta_1 = \delta_2 = 3$



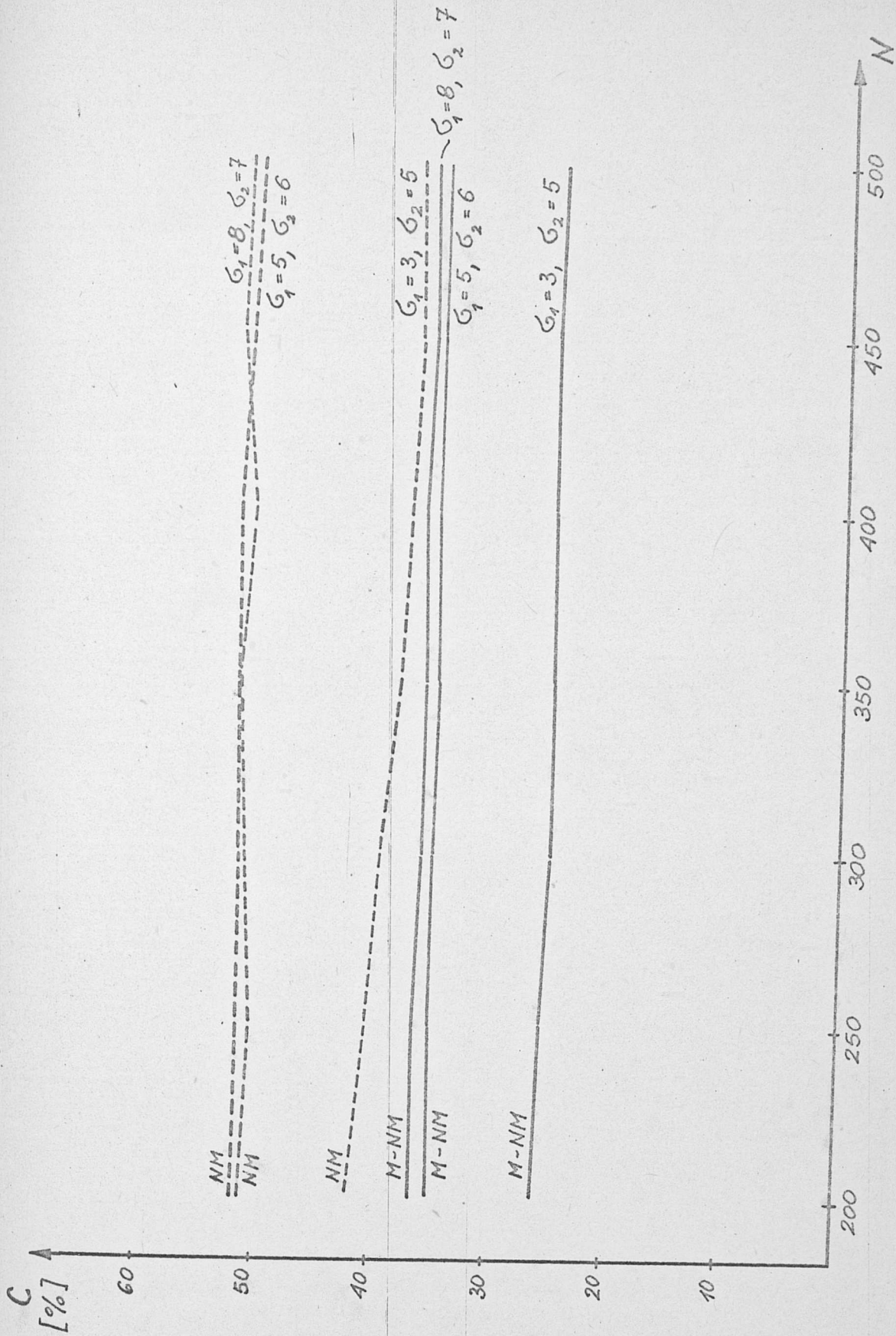
Rys. 5.13. Częstości błędnych klasyfikacji według algorytmów MURTH i M-MURTH przy różnych σ_1 , σ_2 i $m_1=7$, $m_2=3$



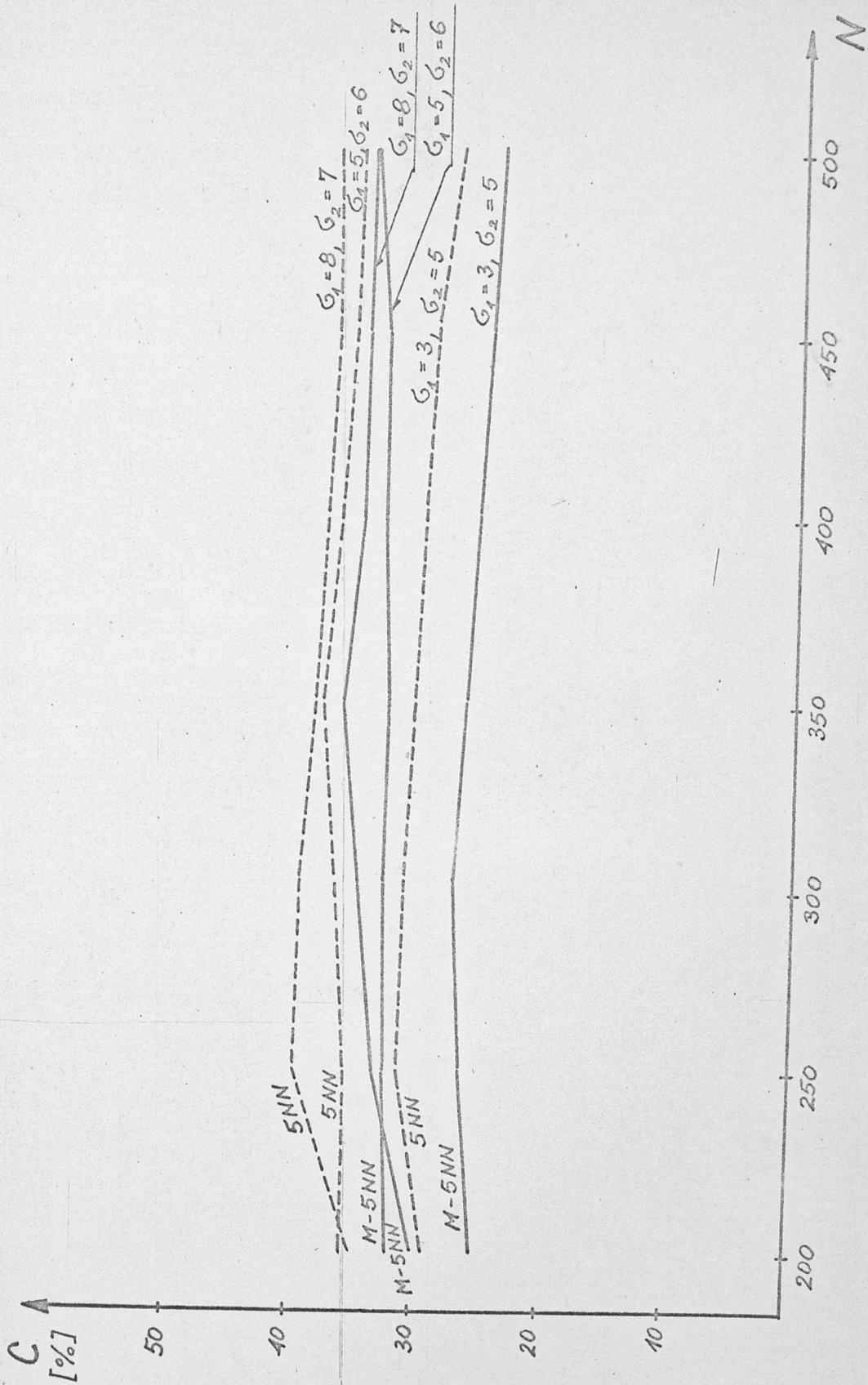
Rys. 5.14. Częstości błędnych klasyfikacji według algorytmów M-LI i LI przy różnych σ_1 , σ_2 i $m_1=7$ i $m_2=3$



Rys. 5.15. Częstości błędnych klasyfikacji według algorytmów M-NN i NN przy różnych σ_1 , σ_2 i $m_1=7$, $m_2=3$



Rys. 5.16. Częstości błędnych klasyfikacji według algorytmów NM i M-NM przy różnych σ_1 , σ_2 i $m_1=7$, $m_2=3$



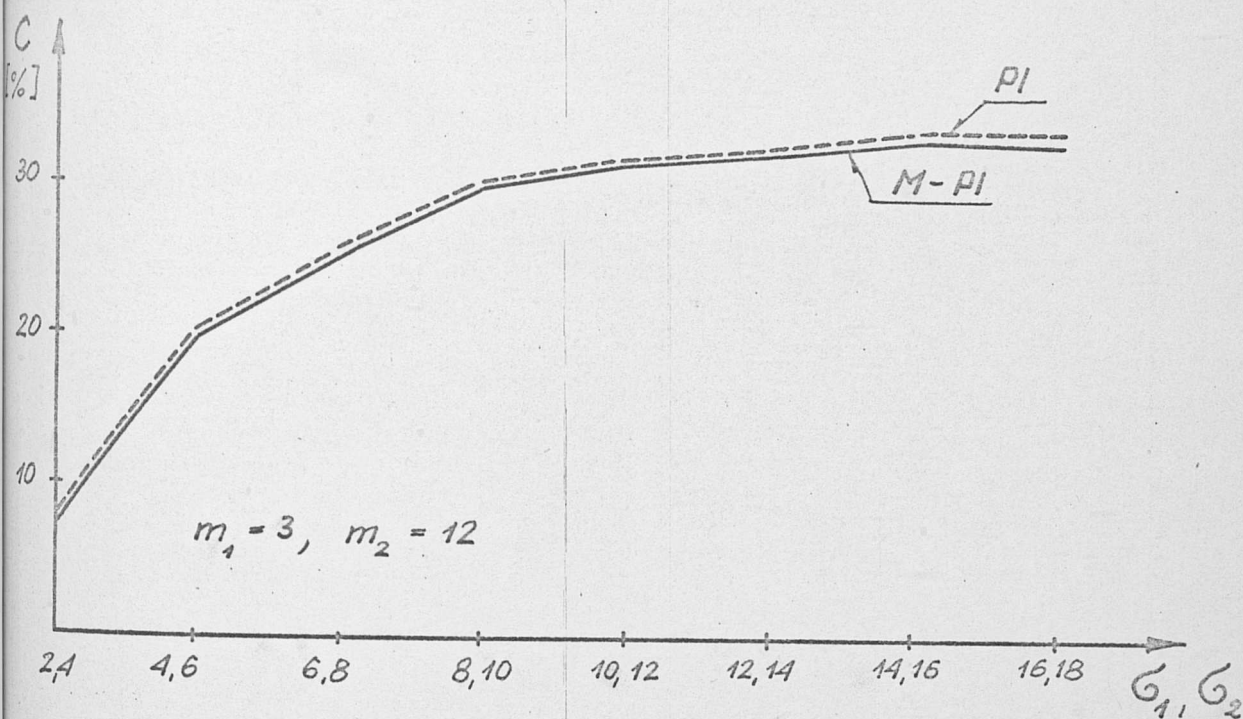
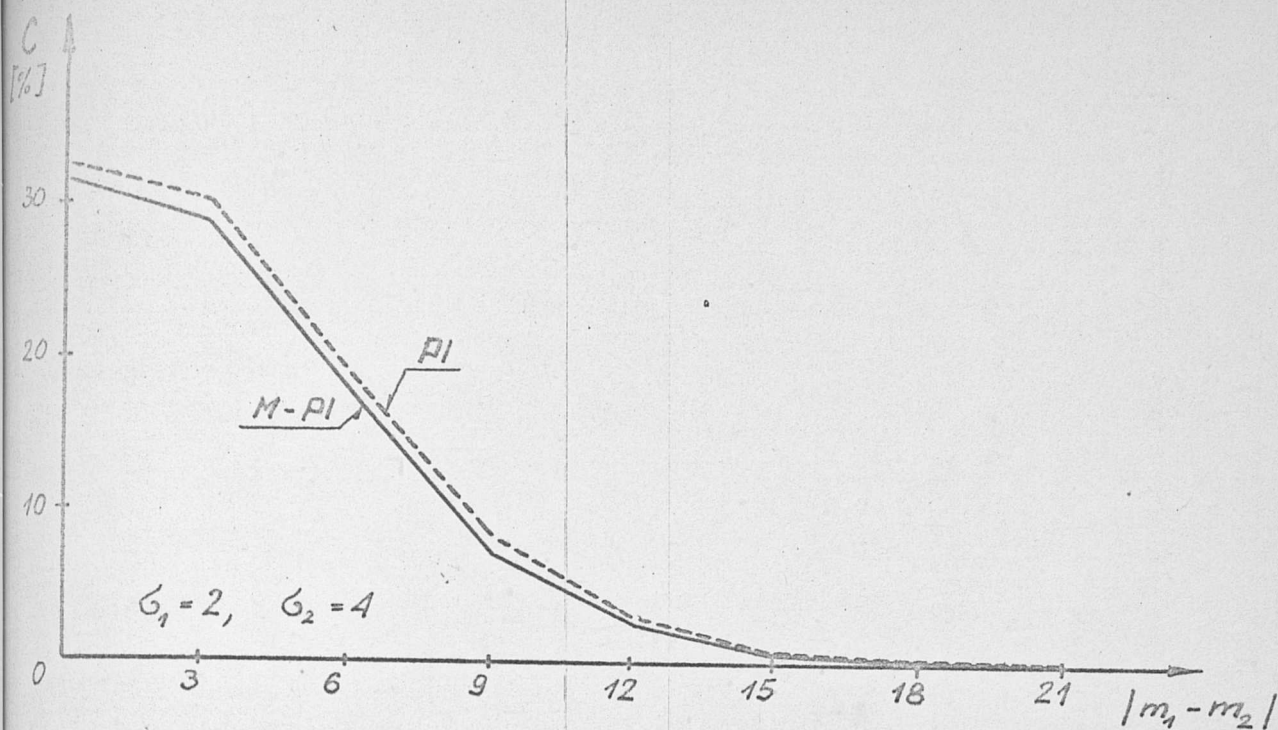
Rys. 5.17. Częstości błędnych klasyfikacji według algorytmów 5NN i M-5NN przy różnych σ_1 i σ_2 i $m_1=7$, $m_2=3$

wskazują na celowość modyfikacji algorytmów NN, NM, α NN i MURTH do postaci M-NN, M-NM, M- α NN i M-MURTH w przypadku klasyfikacji prostego łańcucha Markowa. Modyfikacje te korzystnie wpływają na jakość klasyfikacji zmniejszając w widoczny sposób częstości błędnych rozpoznań. W efekcie jakościowego porównania badanych algorytmów okazało się, że modyfikacja taka jest najkorzystniejsza w przypadku algorytmu M-NM zmniejsza bowiem częstość błędnych decyzji w granicach 10 ÷ 15 % oraz algorytmu M-NN (poprawa w granicach 5 % ÷ 10 %). Dla algorytmów M-5NN i M-MURTH poprawa ta jest mniejsza i wynosi około 3 % ÷ 5 %. W przypadku natomiast algorytmu M-LI wynika, że dla małych różnic wartości średnich modyfikacja taka jest niecelowa, gdyż wyraźnie pogarsza jakość klasyfikacji.

Kształtowanie się częstości c dla badanych algorytmów ze wzrostem dyspersji przy zachowaniu stałych wartości średnich rozkładów przedstawiają rysunki 5.13 ÷ 5.18. Uzyskane rezultaty wskazują, że przy ustalonych wartościach średnich, częstości błędnych rozpoznań według wszystkich badanych algorytmów rosną ze wzrostem dyspersji rozkładów warunkowych. Również jak poprzednio tak i tutaj wyniki eksperymentalne wskazują, że modyfikacja badanych algorytmów korzystnie wpływa na jakość klasyfikacji, a ilościowe zmniejszenie się częstości błędnych rozpoznań kształtuje się bardzo podobnie. Od tej reguły odbiega jedynie algorytm M-LI, dla którego również w tym przypadku modyfikacja okazuje się niecelowa.

Na podstawie przeprowadzonych eksperymentów stwierdzono, że w przypadku rozkładu normalnego, parametry tego rozkładu nie wpływają w istotny sposób na wynik jakościowego porównania badanych algorytmów z ich wersjami niezmodyfikowanymi.

Nie zauważa się natomiast wpływu długości ciągu uczącego na wartość c , w szczególności nie uwidaczniają się asymptotyczne własności algorytmów M-MURTH i MURTH. Świadczy to o zbyt małej jeszcze długości wykorzystywanych ciągów uczących.



Rys. 5.18. Częstości błędnych klasyfikacji według algorytmów przy pełnej informacji probabilistycznej

Wnioski dotyczące zachowania się częstości błędnych klasyfikacji ze wzrostem dyspersji przy zachowaniu stałych wartości średnich oraz w funkcji różnicy wartości średnich rozkładów przy zachowaniu stałych ich dyspersji znajdują swoje wyraźne potwierdzenie na wykresie 5.18 przedstawiającym częstość c dla algorytmów przy pełnej informacji probabilistycznej w funkcji parametrów rozkładów warunkowych.

W trzeciej części badań eksperymentalnych określono wpływ parametrów łańcucha Markowa na zachowanie się częstości błędnych klasyfikacji badanych algorytmów przy zachowaniu stałych rozkładów warunkowych. Przyjęto tutaj jako charakterystyki probabilistyczne dwustanowego łańcucha Markowa następujące parametry [28]:

1. $\gamma = p_{12} + p_{21}$ - parametr określający stopień zależności pomiędzy stanami łańcucha Markowa ($\gamma = 1$ oznacza stany (klasy) niezależne, a im bliższy jest on zeru lub dwójki tym zależność pomiędzy stanami jest silniejsza),

2. $\pi = p_1$ - prawdopodobieństwo początkowe klasy pierwszej i badano kolejno:

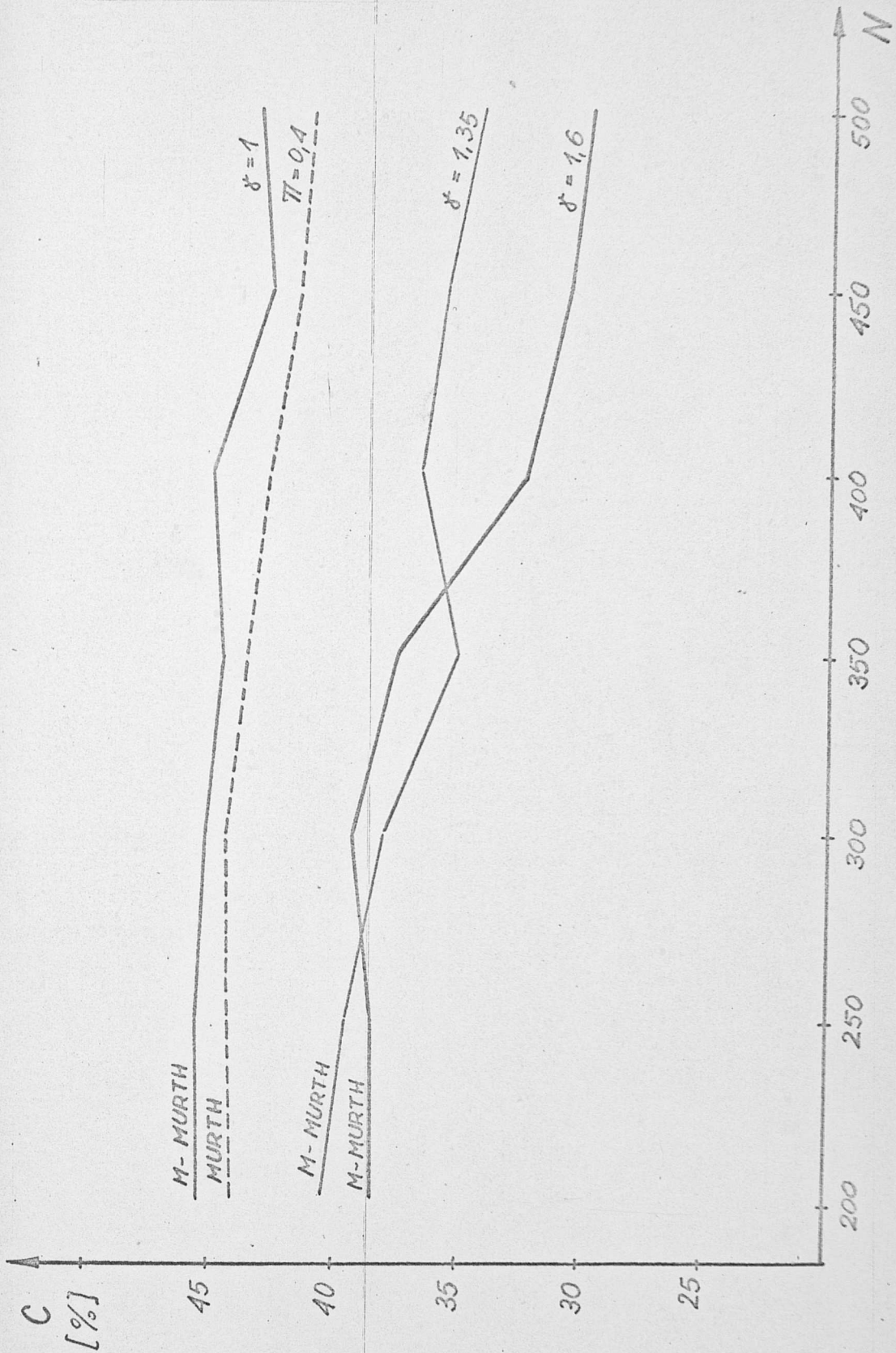
1. wpływ zależności pomiędzy klasami obiektów przy ustalonych prawdopodobieństwach początkowych łańcucha Markowa na częstość błędnych decyzji,

2. wpływ prawdopodobieństw początkowych przy stałym γ .

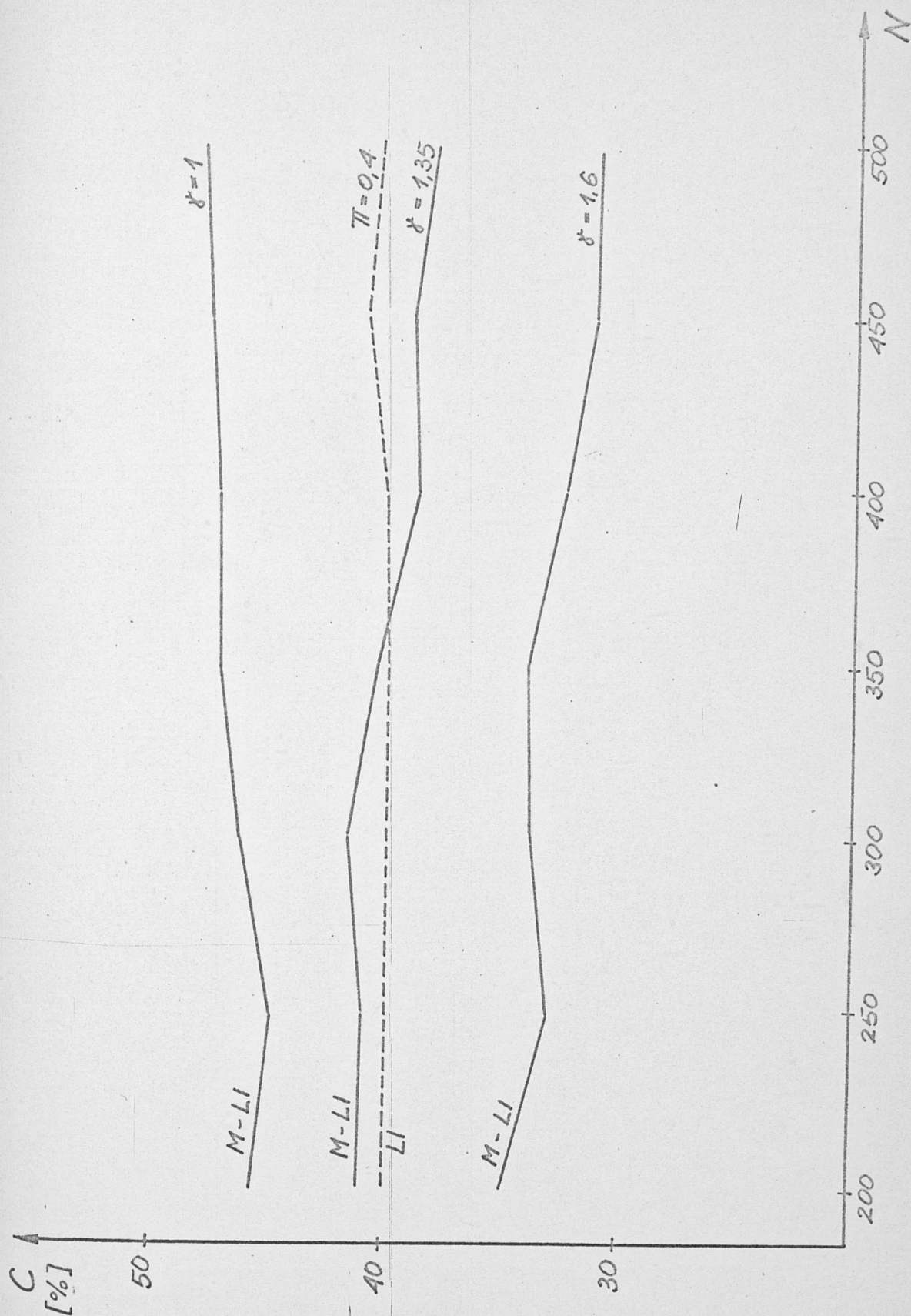
We wszystkich eksperymentach zadano jednakowe parametry rozkładów:

$$f(x/1) = N(10, 11) \quad f(x/2) = N(14, 16)$$

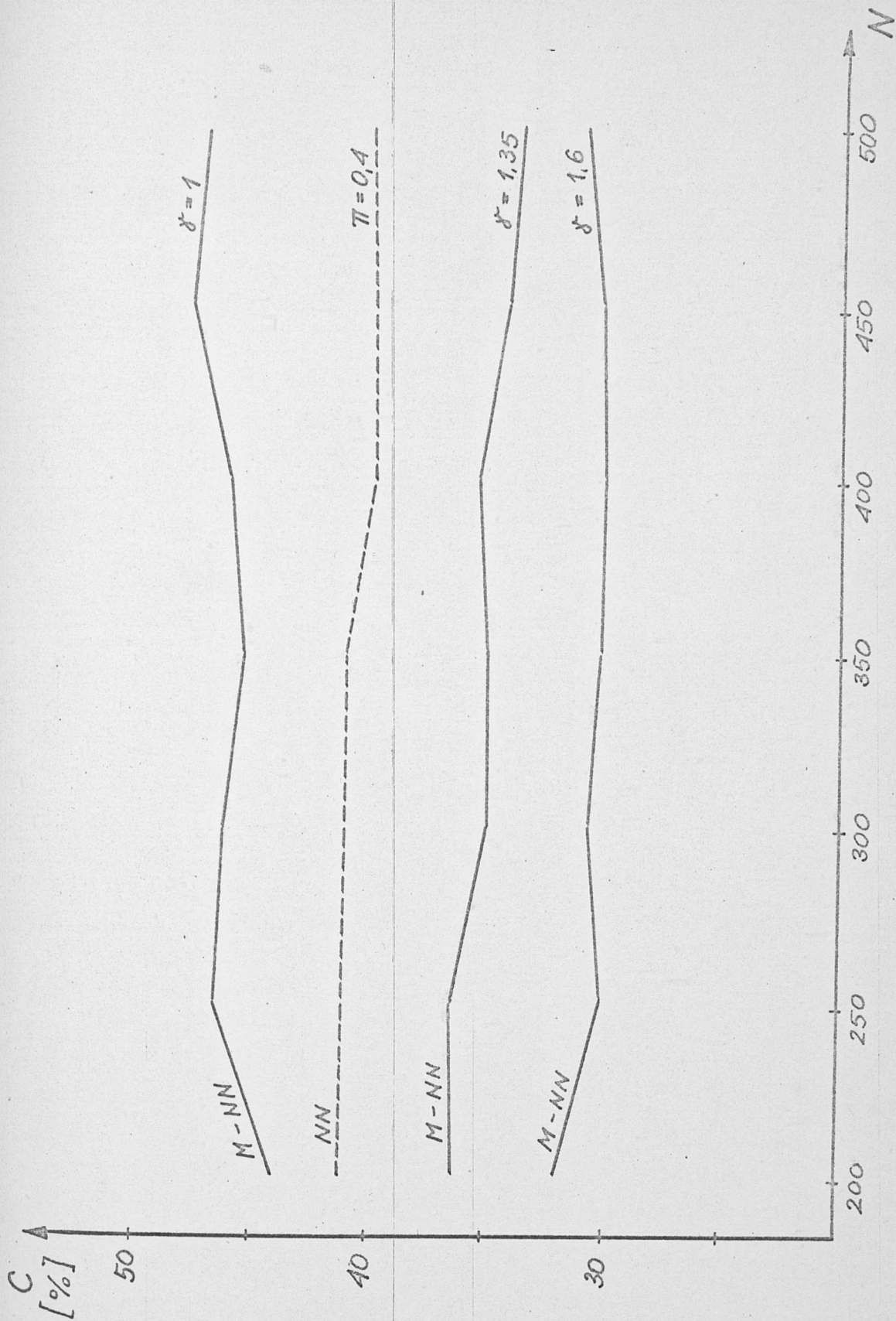
i podobnie jak poprzednio rozpoznawano ciąg 500 pomiarów przy różnych długościach ciągu uczącego ($N = 200, 250, 300, 350, 400, 450, 500$). Uzyskane rezultaty przedstawione zostały na rysunkach 5.19 ÷ 5.28. Wykresy 5.19 ÷ 5.23 obrazują zachowanie się częstości błędnych decyzji dla badanych algorytmów w przypadku, gdy przy ustalonych prawdopodobieństwach początkowych zmienia się suma γ prawdopodobieństw przejścia łańcucha Markowa. Wynika z nich, że stopień zależności pomiędzy klasami ma istotny wpływ na wynik jakościowego porównania badanych algorytmów z ich wersjami niezmodyfikowanymi. Okazuje się mianowicie, że nie zawsze uwzględnienie zależności typu Markowa i stosowanie w konsekwencji bardziej skomplikowanych algorytmów polepsza jakość klasyfikacji. Wynika, że w badanych sytuacjach, to jest przy wybranej liczbie obrazów uczących i klasyfikowanych, poprawa taka ma miejsce, to znaczy modyfikacja korzystnie



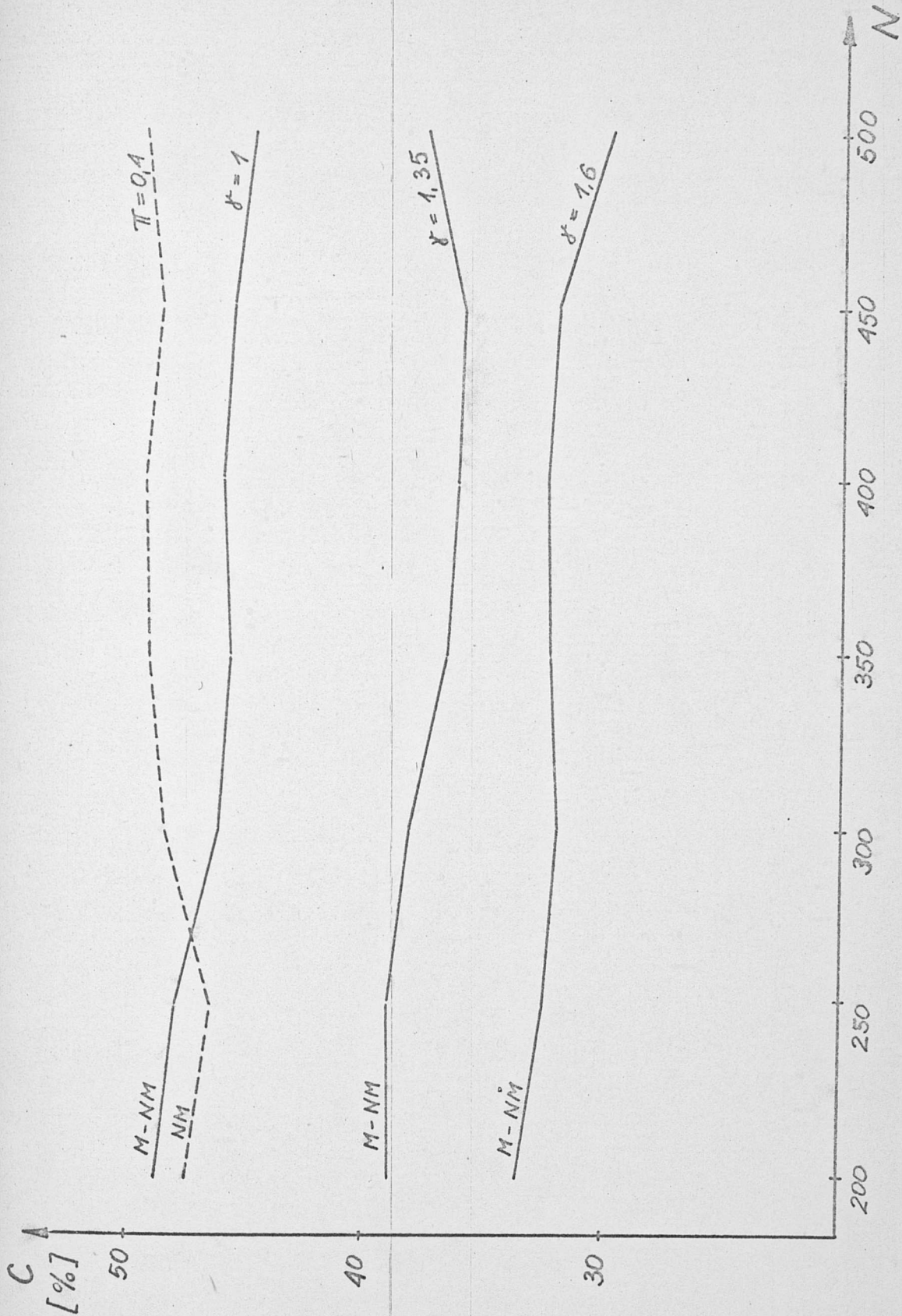
Rys. 5.19. Częstości błędnych klasyfikacji według algorytmów MURTH i M-MURTH przy różnych wartościach parametru ξ i przy $\pi = 0.4$



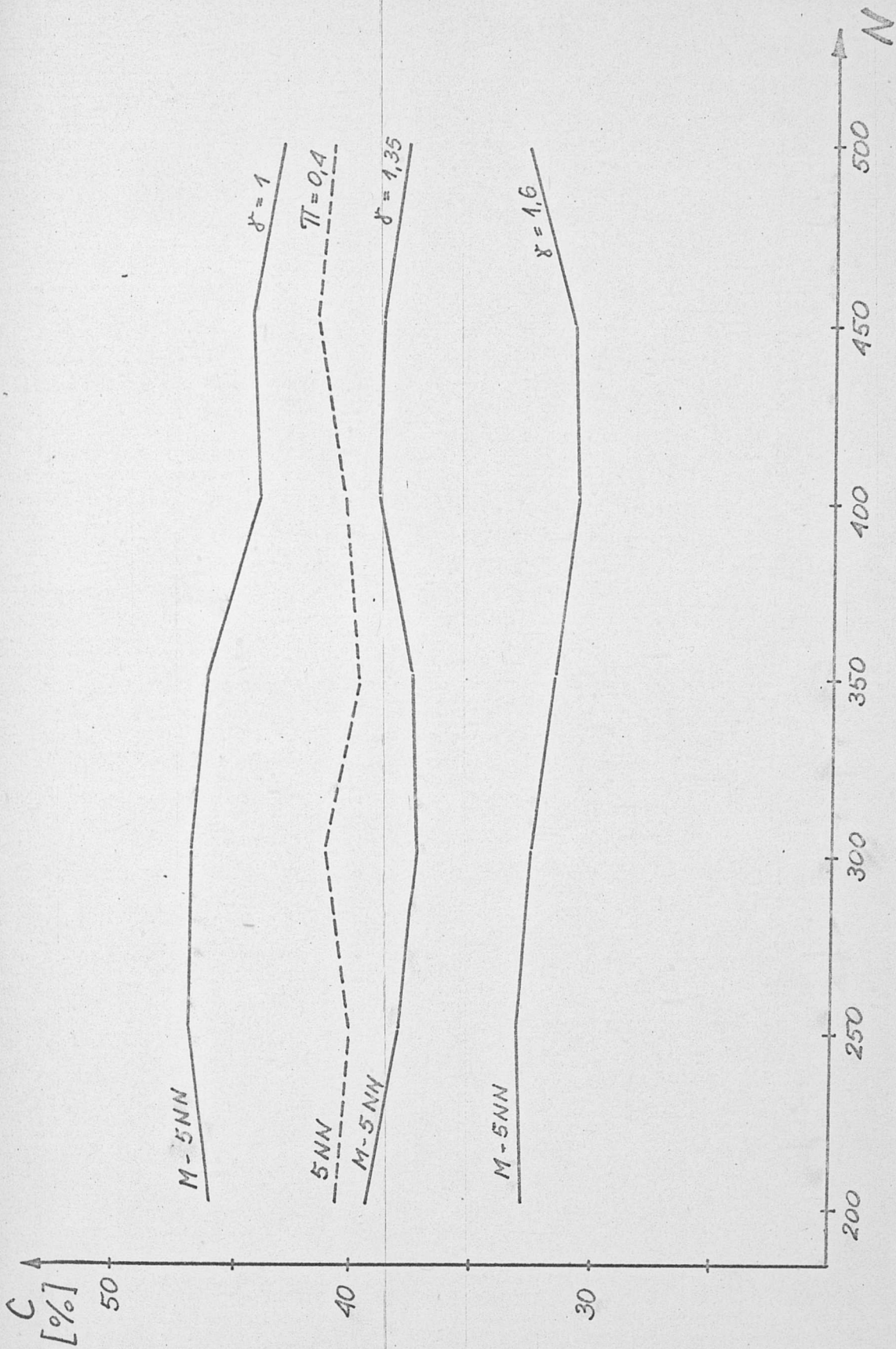
Rys. 5.20. Częstości błędnych klasyfikacji według algorytmów LI i M-LI przy różnych wartościach parametru γ i przy $\pi = 0,4$



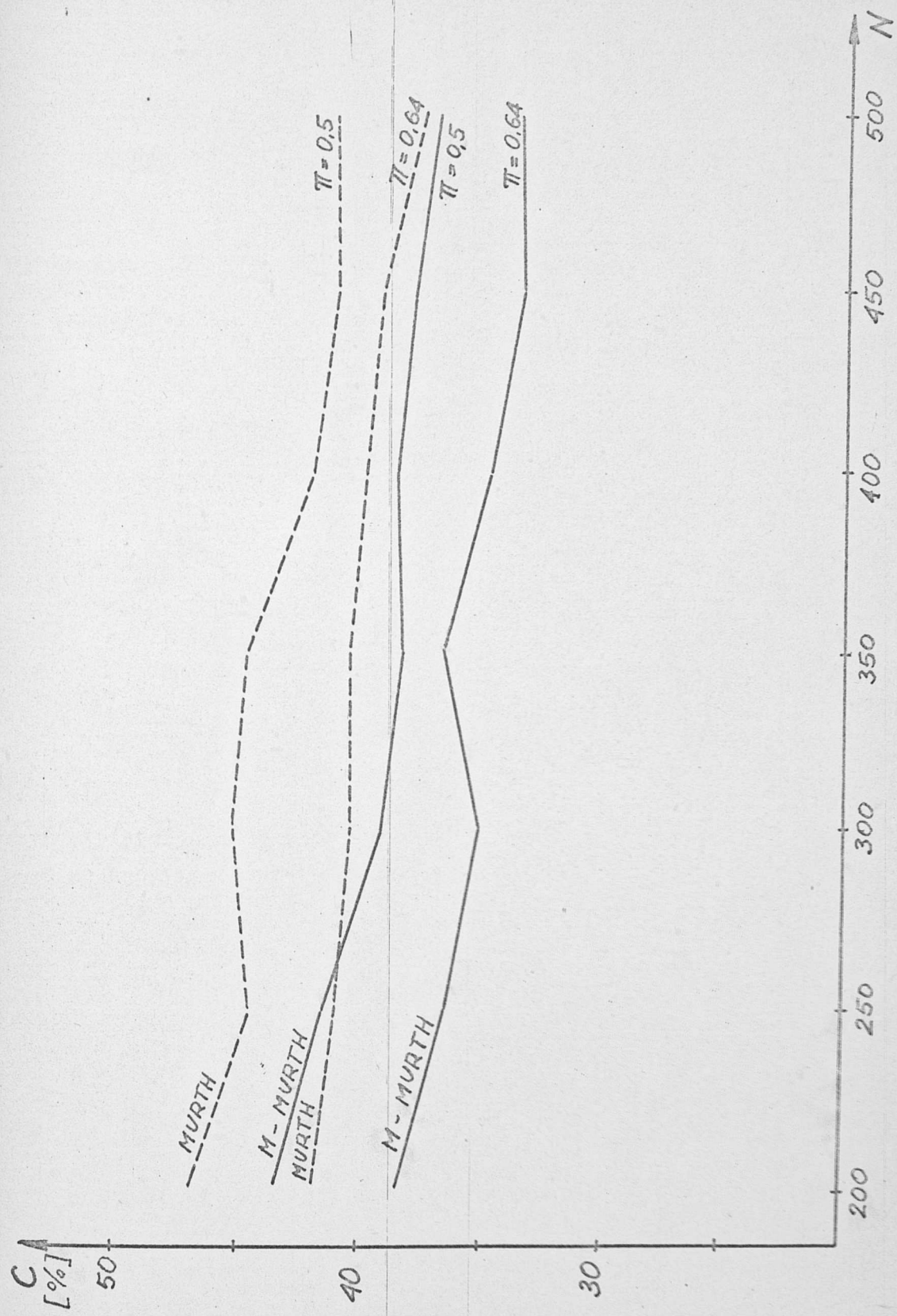
Rys. 5.21. Częstości błędnych klasyfikacji według algorytmów NN i M-NN przy różnych wartościach parametru δ i przy $\pi = 0.4$



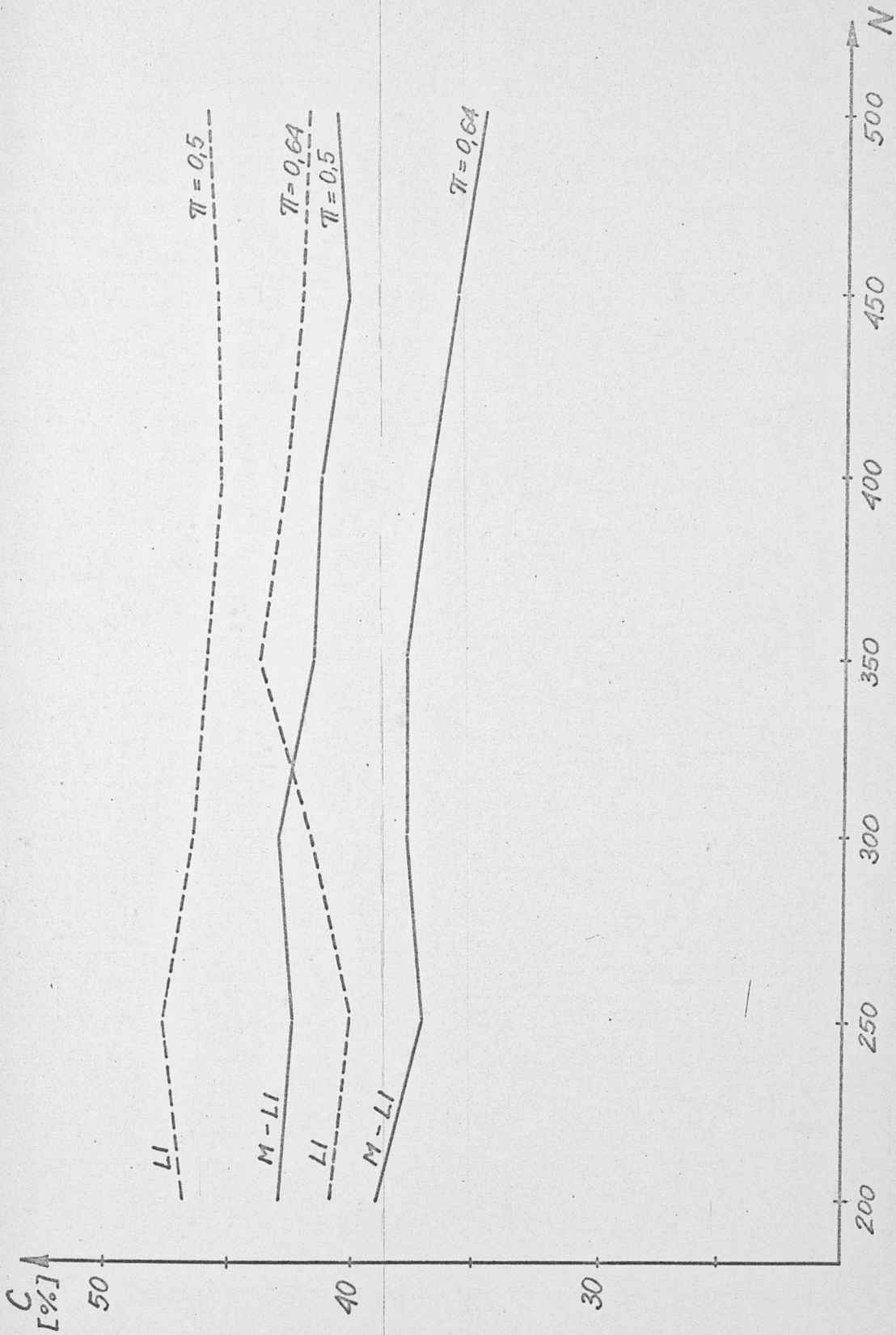
Rys. 5.22. Częstości błędnych klasyfikacji według algorytmów NM i M-NM przy różnych wartościach parametru χ i przy $\pi = 0.4$



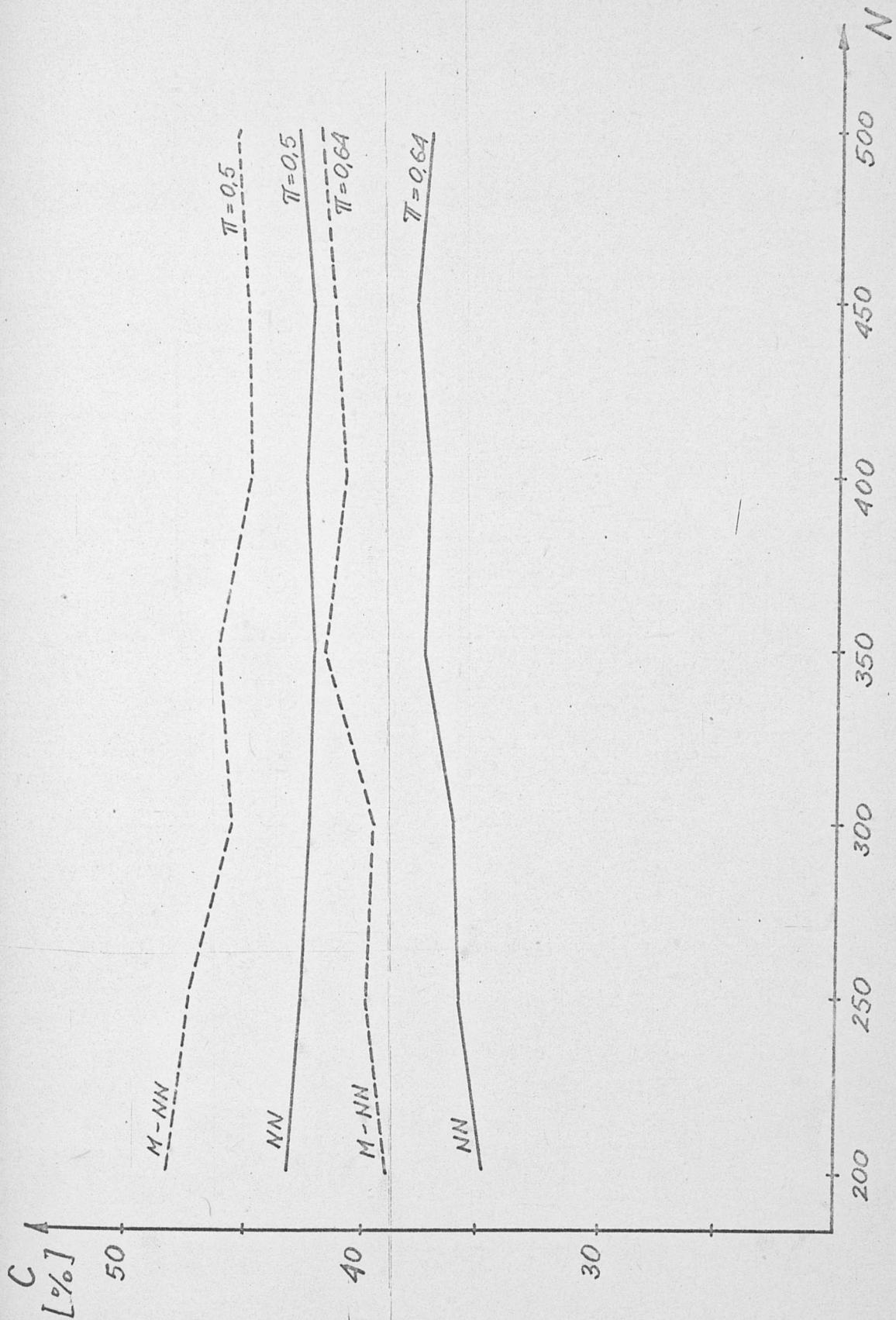
Rys. 5.23. Częstości błędnych klasyfikacji według algorytmów 5NN i M-5NN przy różnych wartościach parametru δ i przy $\pi = 0.4$.



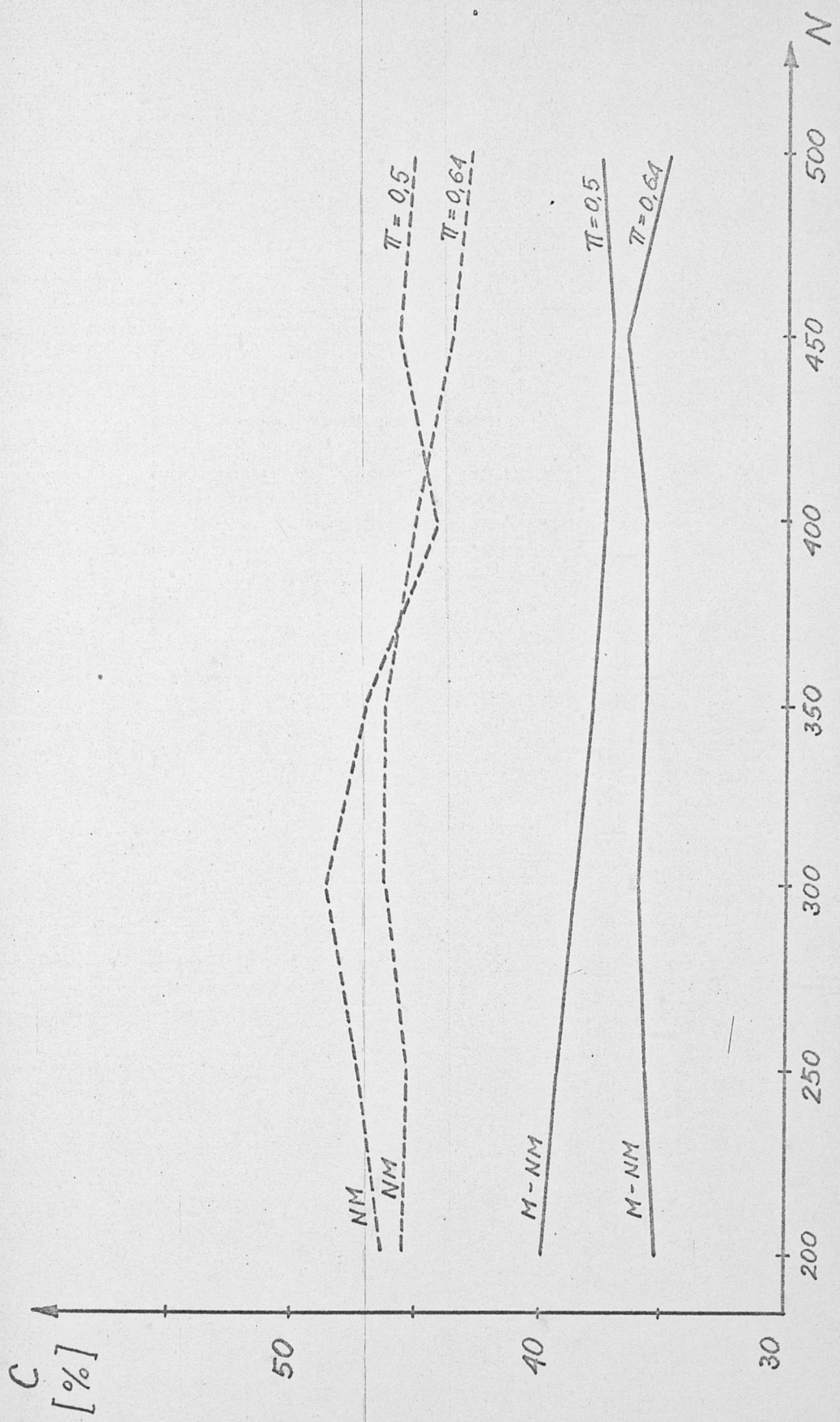
Rys. 5.24. Częstości błędnych klasyfikacji według algorytmów MURTH i M-MURTH przy różnych wartościach π i przy $\delta = 1.4$



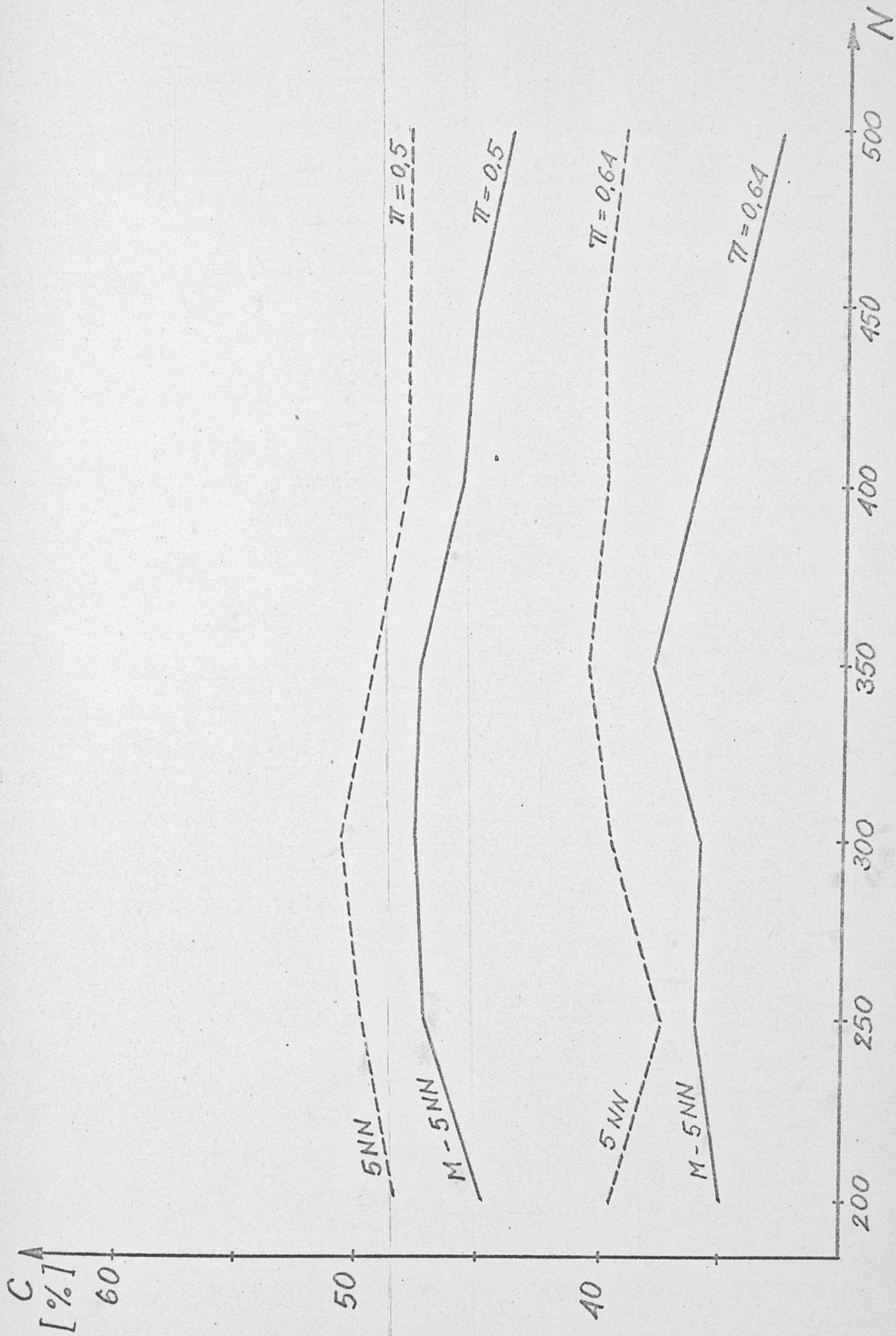
Rys. 5.25. Częstości błędnych klasyfikacji według algorytmów LI i M-LI przy różnych wartościach π i przy $\delta = 1.4$



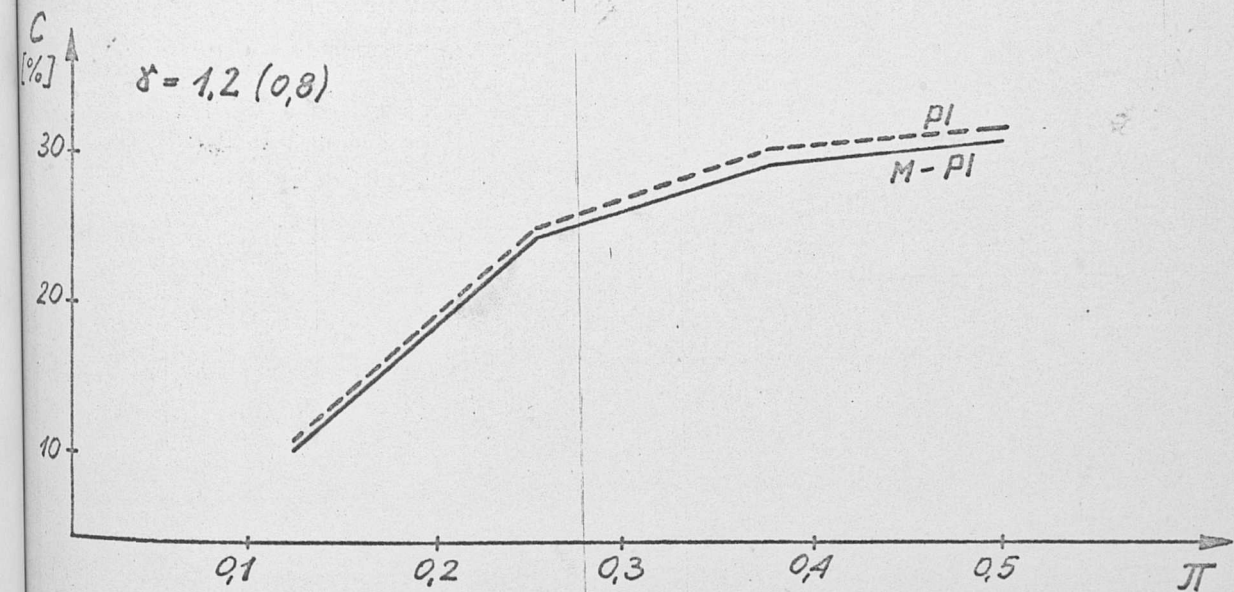
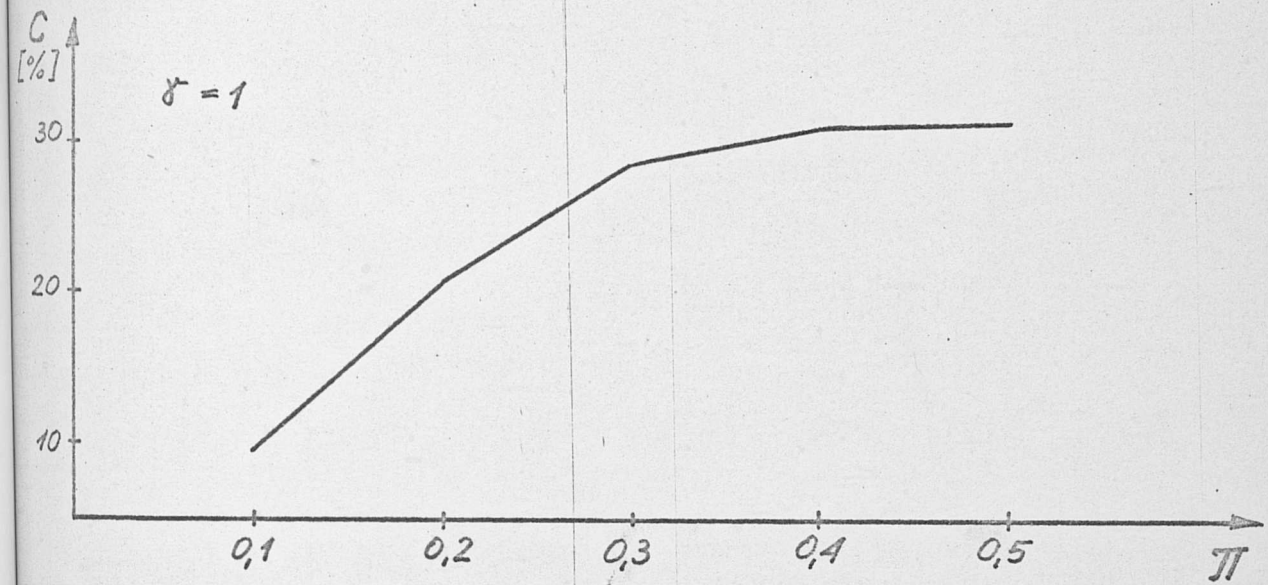
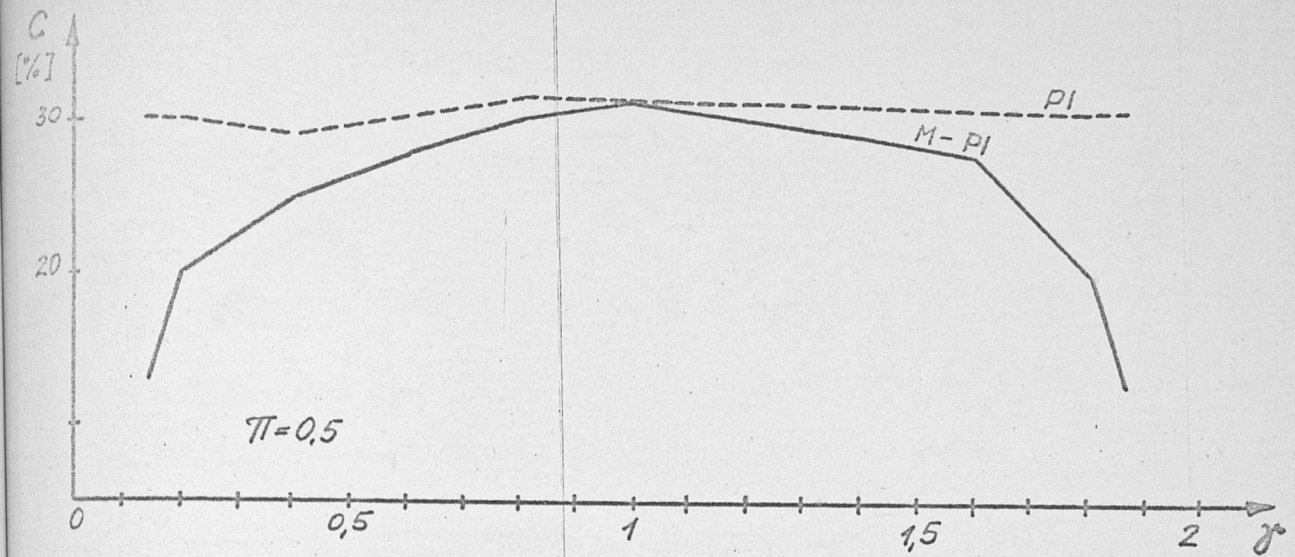
Rys. 5.26. Częstości błędnych klasyfikacji według algorytmów NN i M-NN przy różnych wartościach π i przy $\delta = 1.4$.



Rys. 5.27. Częstości błędnych klasyfikacji według algorytmów NM i M-NM przy różnych wartościach π i przy $\gamma = 1.4$



Rys. 5.28. Częstości błędnych klasyfikacji według algorytmów 5NN i M-5NN przy różnych wartościach π i przy $\gamma = 1.4$



Rys. 5.29. Częstości błędnych klasyfikacji według algorytmów przy pełnej informacji probabilistycznej

wpływa na jakość klasyfikacji zmniejszając w widoczny sposób częstość błędnych rozpoznań, jedynie przy pewnym określonym stopniu zależności - ci pomiędzy poszczególnymi klasami obiektów i zachodzi tym wyraźniej, im suma prawdopodobieństw przejścia bardziej różni się od jedności (podobnego rezultatu spodziewać się należy dla γ z przedziału $0 \div 1$). Tak więc dla badanych długości ciągów uczących uwzględnienie w algorytmach z uczeniem zależności Markowa zmniejsza częstość błędnych rozpoznań jedynie wtedy, gdy zależność ta występuje w sposób wyraźny.

W przeciwnej sytuacji modyfikacja doprowadzić może do pogorszenia jakości klasyfikacji w porównaniu z algorytmami niezmodyfikowanymi. Tłumaczyłoby to większą liczbę błędnych decyzji według algorytmu M-II w porównaniu z jego wersją niezmodyfikowaną w poprzednich eksperymentach.

W wyniku jakościowego porównania badanych algorytmów stwierdzono przykładowo, że dla $\gamma = 1.6$ modyfikacja jest najkorzystniejsza w przypadku algorytmu M - NM zmniejsza bowiem częstość błędnych decyzji w granicach 15 % oraz algorytmów M-MURTH i M-NN (poprawa w granicach 10 %).

Oczywiście parametr γ nie ma wpływu na częstość c według algorytmów niezmodyfikowanych, stąd tylko 1 wykres dla różnych jego wartości.

Kształtowanie się częstości c błędnych rozpoznań dla badanych algorytmów dla różnych prawdopodobieństw początkowych przy zachowaniu stałej sumy prawdopodobieństw przejścia przedstawiają rysunki 5.24 ÷ 5.28. Uzyskane rezultaty wskazują, że częstości błędnych rozpoznań według wszystkich badanych algorytmów maleją ze wzrostem różnicy pomiędzy prawdopodobieństwami początkowymi. Na podstawie przeprowadzonych eksperymentów stwierdzono ponadto, że prawdopodobieństwa początkowe nie wpływają w istotny sposób na wynik jakościowego porównania badanych algorytmów z ich wersjami niezmodyfikowanymi.

Wszystkie powyższe wnioski znalazły swoje wyraźne potwierdzenie na rysunku 5.29. przedstawiającym zachowanie się częstości c według algorytmów z pełną informacją probabilistyczną dla różnych wartości π i γ .

Wszystkie powyższe wnioski znalazły swoje wyraźne potwierdzenie na rysunku 5.29. przedstawiającym zachowanie się częstości c według algorytmów z pełną informacją probabilistyczną dla różnych wartości π i γ .

6. PRZYKŁAD ZASTOSOWANIA PRAKTYCZNEGO

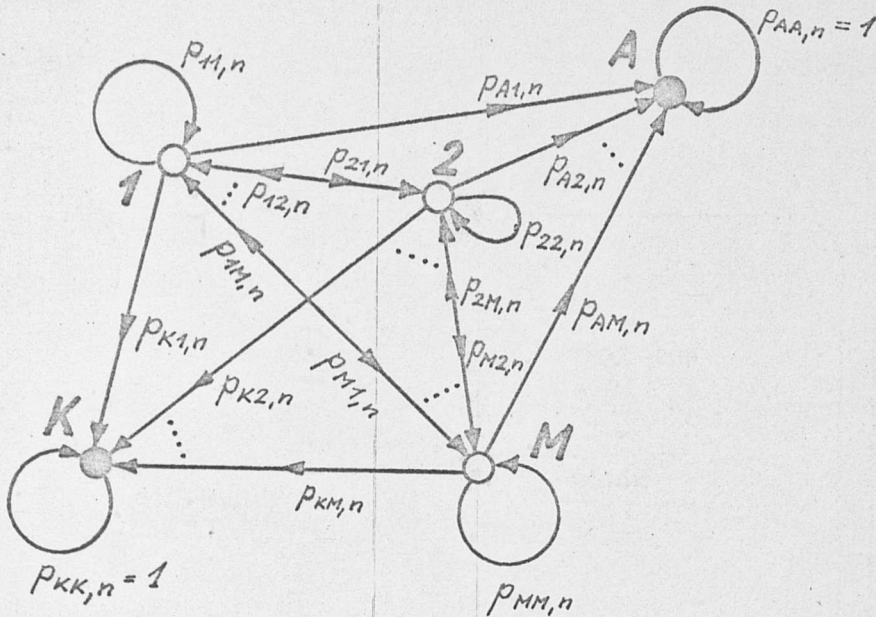
Często podczas kontroli przebiegów procesów przemysłowych oraz w kompleksowej ich automatyzacji zachodzi konieczność rozpoznania sytuacji technologicznej (oceny stanu procesu w jakim się on znalazł), polegającej na przetworzeniu dużej ilości informacji otrzymanych w wyniku pomiaru charakterystycznych parametrów tego procesu. Może się ona pojawić jako zadanie samodzielne (diagnostyka) albo też może wystąpić w pewnych zadaniach sterowania procesem [2,34].

Z reguły stan charakterystyczny w pewnej fazie procesu technologicznego zależy od stanu, jaki wystąpił w poprzedniej fazie tego procesu. W tym przypadku badany proces można traktować jako proces wielostapowy a odpowiednie rozpoznawanie stanów charakterystycznych oraz ewentualne wykorzystanie wyników rozpoznawania do sterowania tym procesem, wymaga uwzględnienia zależności pomiędzy poszczególnymi etapami badanego procesu.

Przedstawimy obecnie pewien prosty przykład sterowania programowego z rozpoznawaniem sytuacji technologicznej, w którym wynikiem pomiarów parametrów procesu (sytuacjom technologicznym) przyporządkowuje się jedną ze skończonej ilości możliwych nastaw sterujących tak, aby jego stan charakterystyczny osiągnął pewien stan pożądaný lub aby zminimalizować pewne kryterium określające straty spowodowane nie uzyskaniem takiego stanu. Ograniczymy się tutaj jedynie do wyznaczenia algorytmu rozpoznawania pomijając zagadnienie określenia algorytmu sterowania (wyboru nastawy sterującej w zależności od rozpoznanego stanu) oraz bardzo istotną w praktycznych zastosowaniach problematykę dotyczącą sposobu przygotowania pomiarów, ich wstępnej obróbki, wprowadzenia na maszynę cyfrową rozpoznającą sytuacje itp.

Rozważmy pewien reaktor chemiczny, w którym zachodzący proces może przyjmować jeden z M możliwych stanów charakterystycznych powiązanych w prosty łańcuch Markowa. Oznacza to, że stan jaki wystąpił w pewnej fazie reakcji chemicznej zależy od tego, jaki wystąpił w fazie poprzedniej, uzależniając jednocześnie od siebie

stan charakterystyczny w momencie następnym. Probabilistyczne powiązania pomiędzy stanami tego procesu można przedstawić schematycznie przy pomocy następującego grafu:



W którym węzły oznaczają wyróżnione stany charakterystyczne, a gałęzie odpowiednie prawdopodobieństwa przejścia (nie koniecznie stacjonarne).

W procesie (grafie) wyróżnia się dwa stany:

1. stan K (pochłaniający dla każdej możliwej macierzy P) mający sens stanu końcowego procesu,
2. stan A (również pochłaniający dla każdej możliwej macierzy P) - stan alarmowy - jego osiągnięcie oznacza, że reakcja zaczyna podążać w niekontrolowanym i niebezpiecznym (groźnym wybuchem, zniszczeniem reaktora itp.) kierunku.

Łatwo zauważyć, że wystąpienie stanów pochłaniających (łańcuch Markowa nie jest więc łańcuchem nieprzywiedlnym) powoduje, że proces nie jest w pełni sterowalny.

Reakcja chemiczna zachodzi w ten sposób, że w chwili jej uruchomienia proces przyjmuje jeden z wyróżnionych stanów z odpowiednim prawdopodobieństwem początkowym i dalej przebiega na zasadzie błędzenia przypadkowego z dwoma stanami pochłaniającymi /stan końcowy i alarmowy/ i ulega zakończeniu z chwilą osiągnięcia jednego z nich.

Założmy, że możliwość ingerencji zewnętrznej /sterowania/ w zachodzącą reakcję jest niewielka i ogranicza się do zmiany w pewnych granicach prawdopodobieństw przejścia

$$P = P(u) ,$$

gdzie u oznacza wektor sterujący

oraz możliwe ponadto jest zatrzymanie reakcji.

Założmy dalej, że znajomość zbioru parametrów (x, a) , gdzie x i a oznaczają odpowiednio k - oraz p - wymiarowe wektory liczbowe mierzonych parametrów fizycznych / temperatura, ciśnienie, natężenie prądu itp./ oraz nie mierzonych lub niemierzalnych parametrów, określa jednoznacznie stan charakterystyczny

$$j = \Phi(x, a) .$$

Zakładając również będziemy, że istnieją rozkłady prawdopodobieństwa $f(x/i)$ / $i=1,2,\dots,M$ / .

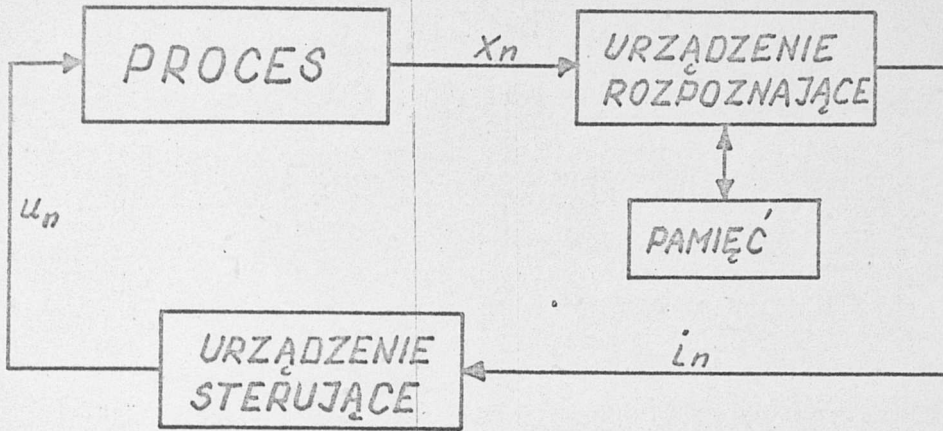
Na podstawie zgromadzonych danych statystycznych o procesie określono dla potrzeb sterowania nim model matematyczny reakcji składający się z:

1. wektora prawdopodobieństw początkowych,
2. zestawu, zależnych od zewnętrznych nastaw, macierzy przejścia,
3. wektorów pomiarów średnich \bar{x}_i wyróżnionych stanów charakterystycznych / $i = 1,2,\dots,M$ / .

Dla tak określonej sytuacji zaprojektować należy układ sterowania, to jest wyznaczyć algorytm rozpoznawania sytuacji technologicznej oraz algorytm sterowania / wybór odpowiedniej macierzy przejścia, a dla stanów końcowego i alarmowego - przerwanie procesu / w zależności

od rozpoznanego stanu.

W rachubę wchodzi tylko sterowanie w układzie zamkniętym



Stan informacji apriorycznej określa jednoznacznie algorytm rozpoznawania, jaki należy zastosować w urządzeniu rozpoznającym. Będzie to zmodyfikowany algorytm "najbliższa średnia" (3.46).

Natomiast odpowiedniego wyboru/w sensie określonego kryterium/ macierzy przejścia w zależności od rozpoznanego stanu dokonać można w oparciu o teorię sterowanych łańcuchów Markowa [8, 25].

7. WNIOSKI I UWAGI KOŃCOWE

W pracy sformułowane zostały w oparciu o nieparametryczną estymację gęstości prawdopodobieństwa pewne analityczne algorytmy rozpoznawania z uczeniem dla przypadku, gdy ciąg klas powiązany jest w prosty, stacjonarny, nieprzywiedlny i jednorodny łańcuch Markowa. Wszystkie wykorzystane estymatory są zgodne (zbieżne według prawdopodobieństwa), a zatem, na mocy udowodnionych twierdzeń, uzyskane w oparciu o nie algorytmy rozpoznawania są asymptotycznie optymalne, to znaczy ryzyko dla tych algorytmów dąży do ryzyka Bayesa.

Przedstawione algorytmy rozpoznawania z uczeniem dla prostych łańcuchów Markowa stanowią pewną ilustrację ogólnej metody konstruowania takich algorytmów. Algorytmy tego typu mają ogólnie dość skomplikowaną postać i ponadto niestacjonarny charakter. Pomimo tego stosowanie ich przy rozpoznawaniu prostych łańcuchów Markowa wydaje się być celowe z uwagi na asymptotyczne własności tych algorytmów (ich zbieżność do reguły Bayesa) oraz z uwagi na fakt, że algorytm przy pełnej informacji probabilistycznej uwzględniającej zależność typu Markowa gwarantuje lepszą jakość rozpoznawania w porównaniu z algorytmem nie uwzględniającym takiej zależności. (rys. 5.18, 5.29)

W pracy zostały przedstawione także wyniki eksperymentalnego badania algorytmów rozpoznawania prostych łańcuchów Markowa, a najważniejsze wnioski i spostrzeżenia jakie z nich wynikają są następujące:

1. uwzględnienie markowskiej zależności w badanych algorytmach w minimalnym stopniu wydłuża czas klasyfikacji w porównaniu z wersjami niezmodyfikowanymi tych algorytmów;
2. parametry warunkowych rozkładów oraz prawdopodobieństwa początkowe łańcucha Markowa mają istotny wpływ na kształtowanie się częstości błędnych klasyfikacji algorytmów, nie wpływają natomiast w widoczny sposób na wynik jakościowego porównania badanych algorytmów z ich wersjami niezmodyfikowanymi;
3. modyfikacja algorytmów MURTH, LL, NN, NM i NN do postaci

M-MURTH, M-LI, M-NN, M-NM i M-~~Q~~NN w przypadku klasyfikacji prostego łańcucha Markowa okazuje się tym bardziej celowa, im zależność między poszczególnymi klasami obiektów jest silniejsza; w przypadku natomiast klasyfikacji niezależnego ciągu obrazów modyfikacja taka prowadzi do pogorszenia jakości klasyfikacji.

W konkluzji do sformułowanych wniosków należy zaznaczyć, że nie posiadają one charakteru ostatecznego jako działającego prawa, lecz stanowią pewne stwierdzenia natury hipotetycznej.

W wielu wypadkach na podstawie otrzymanych wyników nie można jeszcze ocenić asymptotycznych własności badanych algorytmów. Świadczy to o zbyt małej długości wykorzystywanych ciągów uczących.

W przeprowadzonych eksperymentach częstość błędnej klasyfikacji przy ustalonym N wyznaczana była tylko dla jednego ciągu uczącego. Nie można jej zatem nadawać interpretacji średniego ryzyka, co wymagałoby uśrednienia wartości c dla wszystkich możliwych ciągów uczących.

Formułując stosunek do przeprowadzonych badań eksperymentalnych należy podkreślić, że z uwagi na trudności natury numerycznej ograniczono się tylko do najprostszyc przypadków, stąd też uzyskane rezultaty posiadają charakter fragmentaryczny. Zachodzi więc dalsza potrzeba przeprowadzenia dodatkowych badań, by ich rezultaty mogły mieć znaczenie w pełni praktyczne.

Poczuwam się do miłego obowiązku serdecznego podziękowania Promotorowi Panu prof. dr hab. inż. Zdzisławowi Bubnickiemu za zaproponowanie atrakcyjnego tematu oraz za liczne cenne rady i wskazówki, które ułatwiły mi rozwiązanie niejednego problemu w niniejszej pracy.

Również wszystkim moim życzliwym Kolegom z Zakładu Systemów Sterowania, a w szczególności Koledze mgr inż. Zygmuntowi Hasiewiczowi, którzy nie szczędzili mi w toku pisania pracy swojej pomocy, wyrażam szczerą wdzięczność.

LITERATURA

1. Billingsley P., Statistical Methods in Markov Chains, Ann. Math. Statist., vol.32, pp.12 - 40.
2. Bubnicki Z., Rozpoznawanie sytuacji technologicznych w systemach kompleksowej automatyzacji, w : Systemy automatyki kompleksowej, Ossolineum, Wrocław 1969.
3. Bubnicki Z., Algorytmy rozpoznawania dla prostych łańcuchów Markowa, Prace Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, Materiały i Studia, zeszyt 1, 1972.
4. Bubnicki Z., Zbieżność procesów automatycznej aproksymacji w układach dyskretnych, Zeszyty Naukowe Politechniki Wrocławskiej, Automatyka V, 1966.
5. Bubnicki Z., Least Interval Pattern Recognition and its Application to Control Systems, Materiały IV Kongresu IFAC, zeszyt 21, Warszawa 1969.
6. Chu J.T., Error Bounds for a Contextual Recognition Procedure, IEEE Trans. on Computers, October 1971.
7. Cover T.M. Hart P.E., Nearest Neighbour Pattern Classification, IEEE Trans. on Inf. Theory, vol.13, No 1.
8. Derman C., Finite State Markovian Decision Processes, N. York 1970, Academic Press.
9. Edwards J.T., Chambers R.L., Can a priori Probabilities help in Character Recognition?, J. Assoc. Comput. Mach., vol.14, 1964.
10. Fisz M., Rachunek prawdopodobieństwa i statystyka matematyczna, PWN, Warszawa 1969.
11. Goldstein M., K_N - nearest Neighbour Classification, IEEE Trans. on Inf. Theory, vol.18, Sept. 1972.
12. Greblicki W., Nieparametryczna estymacja w uczeniu rozpoznawania, Podstawy Sterowania, t.2, z.3, 1972.
13. Hasiewicz Z., Eksperymentalne badanie algorytmów rozpoznawania z uczeniem dla prostych łańcuchów Markowa, Podstawy Sterowania, t.2, z.1, 1972.
14. Hasiewicz Z., Kurzyński M., Wyniki eksperymentalnego badania pewnych algorytmów rozpoznawania z uczeniem dla prostych łańcuchów

Markowa, w : Probabilistyczne problemy rozpoznawania, Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, Wrocław 1974.

15. Hasiewicz Z., Kurzyński M., Algorytmy rozpoznawania z uczeniem dla prostych łańcuchów Markowa, Materiały VI Krajowej Konferencji Automatyki /w druku/.
16. Hilborn Ch., Unsupervised Learning Minimum Risk Pattern Classification for dependent Hypotheses and dependent Measurements, IEEE Trans. System Sci. and Cybernetics, vol.5, no.2, 1969.
17. Kaczkowski A., Eksperymenty z rozpoznawaniem liter przez maszynę cyfrową z zastosowaniem różnych metod uczenia, Prace Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej, z.1, 1970.
18. Konakov, A Nonparametric Estimate of the Density of the Distribution of Probabilities, Teoria werojatkostej i jejo primenenija, tom 17, wyp.3, 1972.
19. Krejn S. O., Analiza funkcjonalna, PWN, Warszawa 1967.
20. Loftsgaarden D.O., Quessenberry C.P., A Nonparametric Estimate of Multivariate Density Function, Ann. Math. Statist. vol.36, 1965.
21. Łojasiewicz, Wstęp do teorii funkcji rzeczywistych, PWN, Warszawa 1973.
22. Murthy V.K., Estimation of Probability Density, Ann. Math. Stat. vol.36, pp.1027 - 1031.
23. Neveu J., Matematyckeskoje osnovy teori werojatkostej, Mir, Moskwa 1969.
24. Nilsson N.J., Maszyny uczące się, PWN, Warszawa 1968.
25. Pallu de la Barrière R., Matematyczne podstawy teorii sterowania automatycznego, PWN, Warszawa 1972.
26. Papoulis A., Prawdopodobieństwo, zmienne losowe i procesy stochastyczne, WNT, Warszawa 1972.
27. Parzen E., On estimation of Probability Density Function and Mode, Ann. Math. Statist., vol.33, 1962.
28. Preston P.F., An Empirical Bayes Problem with a Markovian Parameter, Biometrika, vol58, no.3, 1971.
29. Raviv J., Decision Making in Markov Chains applied to the Prob-

lem of Pattern Recognition, IEEE Trans. on Inform. Theory, October 1967.

30. Robbins H., The Empirical Bayes Approach to Statistical Decision Problems, Ann. Math. Statist., vol.35, 1964.
31. Specht F., Series Estimation of a Probability Density Function, Technometrics, vol.13, no.2, 1971.
32. Tou J., Advances in Information System Science, vol.1, Plenum Press, N.York 1969.
33. Warmus M., Wykłady z probabilistyki, PWN, Warszawa 1973.
34. Węgrzyn S., Sterowanie procesami kompleksowymi w oparciu o rozpoznawanie ich stanów, Prace IV K.K.A., Kraków 1967.
35. Systemy programowania m. c. ODRA 1204, Procedury cz. 2, 1204 VIII - 11, Wrocław 1970.
36. Paszkowski S., Algol 60, PWN, Warszawa 1968.
37. Jerzykiewicz K., Szczepkowiec J., Algol 1204, PWN, Warszawa 1973.

D O D A T E K

BIBLIOTEKA PODPROGRAMÓW

OBLICZANIE PRAWDOPODOBIENSTW POCZĄTKOWYCH I PRZEJSCIA ŁAŃCUCHA MARKOWA

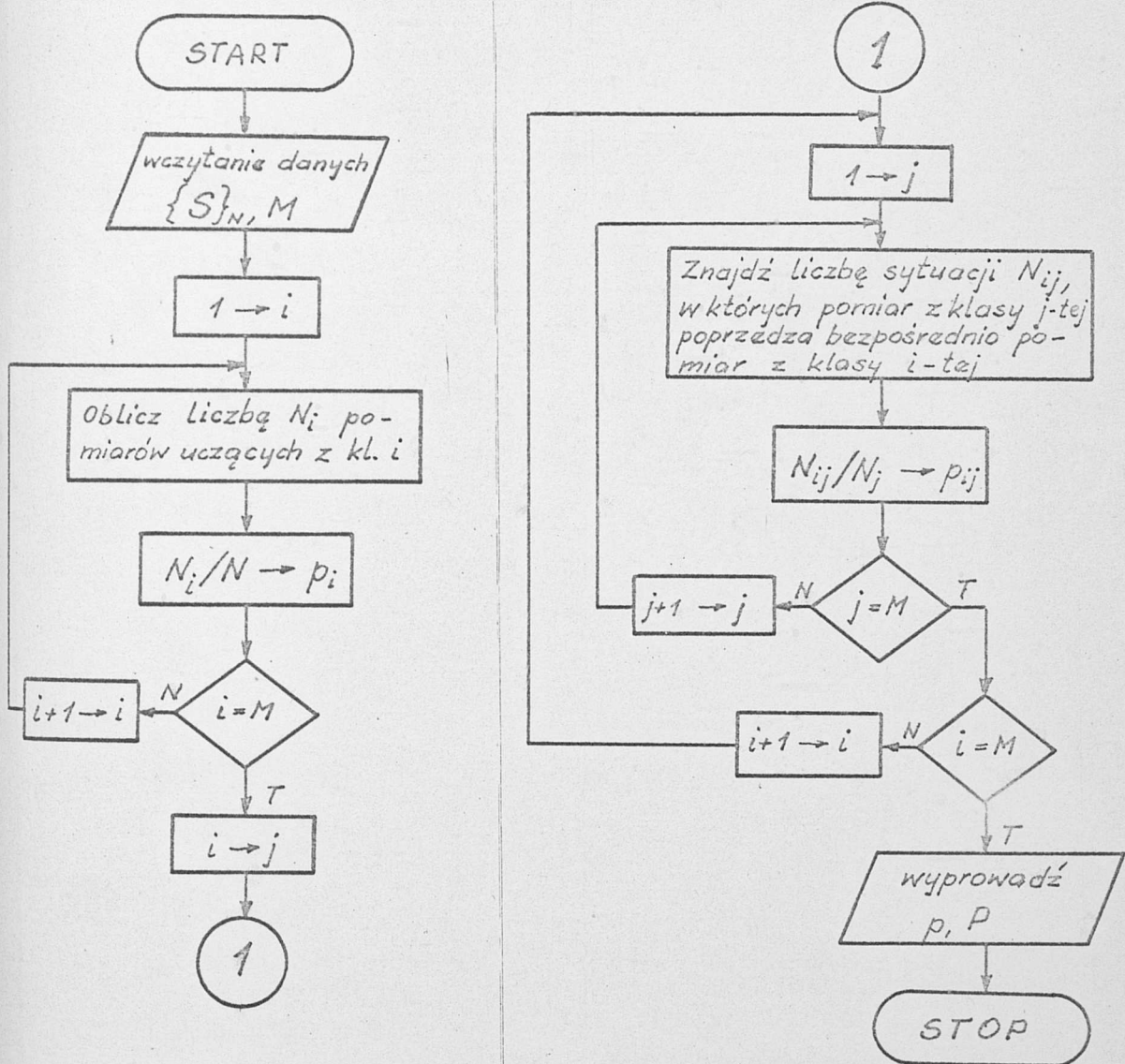
1. Nazwa procedury.

procedure PRAWDOPODOBIENSTWO (I, N, M, P, Q)

2. Funkcja procedury i opis metody.

Procedura PRAWDOPODOBIENSTWO oblicza M wymiarowy wektor prawdopodobieństw początkowych P i macierz M×M prawdopodobieństw przejścia Q według wzoru (3.3).

3. Schemat blokowy.



4. Dane i parametry.

$I[1:N]$ - jednowskaźnikowa tablica typu integer prawidłowych klasyfikacji obrazów uczących,

N - długość ciągu uczącego - zmienna typu integer,

M - ilość klas - zmienna typu integer.

5. Wyniki.

Po wyjściu z procedury jednowskaźnikowa tablica typu real - $P [1:M]$ będzie zawierać prawdopodobieństwa początkowe, a dwuwskaźnikowa tablica typu real $Q [1:M,1:M]$ - prawdopodobieństwa przejścia $P[i]=p_i$, $Q[i,j]=p_{ij}$ /.

6. Tabulogram.

```

procedure PRAWDOPODOBIENSTWO (I) dlugosc ciagu uczacego:(N)
      ilosc klas:(M) prawdopodobienstwo poczatkowe:(P)
      prawdopodobienstwo przejscia:(Q);
value N,M;
integer N,M;
integer array I;
array P,Q;
begin comment procedura oblicza na podstawie ciagu uczacego
      M prawdopodobienstw poczatkowych oraz M*M praw-
      dopodobienstw przejscia lancucha Markowa;
  integer i,j,k;
  array S[1:M],L[1:M,1:M];
  for i:=1 step 1 until M do
  begin
    S[i]:=0.0;
    for j:=1 step 1 until N do
      if I[j]=i then S[i]:=S[i]+1.0;
    P[i]:=S[i]/N;
  end;
  for i:=1 step 1 until M do
  for j:=1 step 1 until M do
  begin
    L[i,j]:=0.0;
    for k:=2 step 1 until N do
      if I[k-1]=j and I[k]=i then L[i,j]:=L[i,j]+1.0;
    Q[i,j]:=L[i,j]/S[j];
  end;
end PRAWDOPODOBIENSTWO;

```

ZMODYFIKOWANY ALGORYTM WYKORZYSTUJĄCY ESTYMATOR GĘSTOŚCI PARZENA

1. Nazwa i typ procedury.

integer procedure MPAR(I,U,W,p,M,P,Q,X,N)

2. Funkcja procedury i opis metody.

Procedura MPAR zalicza dany p wymiarowy obraz X do jednej z M klas przy użyciu algorytmu (3.17) wykorzystującego nieparametryczny estymator gęstości Parzena z funkcją jądra (3.23).

3. Schemat blokowy.

Rysunek 3.2

4. Dane i parametry.

I [1:N] - jednowskaźnikowa tablica typu integer prawidłowych klasyfikacji obrazów uczących,

U [1:N,1:p] - dwuwskaźnikowa tablica typu real pomiarów uczących,

W [1:M] - jednowskaźnikowa tablica typu real współczynników wagowych,

p - wymiar przestrzeni obrazów - zmienna typu integer,

M - ilość klas - zmienna typu integer,

P [1:M] - jednowskaźnikowa tablica typu real prawdopodobieństw początkowych,

Q [1:M,1:M] - dwuwskaźnikowa tablica typu real prawdopodobieństw przejścia,

X [1:p] - jednowskaźnikowa tablica typu real pomiarów aktualnie klasyfikowanego obrazu,

N - długość ciągu uczącego - zmienna typu integer.

5. Wyniki.

Po wyjściu z procedury pod jej nazwę podstawiony zostanie numer klasy, a tablica W [1:M] zawierać będzie nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MPAR należy obliczyć prawdopodobieństwa początkowe i przejścia przy użyciu procedury PRAWDOPODOBIEŃSTWO.

7. Uwagi.

Przed pierwszym wywołaniem procedury MPAR należy dokonać podstawienia

$$W[i] := (P[i] \times N)^{1/(p/2)} / N, \quad (i=1, 2, \dots, M).$$

8. Tabulogram.

integer procedure MPAR(I) pomiary uczace:(U) współczynniki wagi we:(W) wymiar obrazow:(p) ilosc klas:(M) prawdopodobienstwa poczatkowe:(P) prawdopodobienstwa przejścia:(Q) klasyfikowany obiekt:(X) dlugosc ciagu uczacego:(N);

value p,M,N;

integer p,M,N;

integer array I;

array U,W,P,Q,X;

begin comment procedura MPAR zalicza dany p wymiarowy pomiar X do jednej z M klas przy uzyciu algorytmu wykorzystujacego nieparametryczny estymator gestosci Parzena;

integer i,j,k;

real A,B,C;

array D[1:M];

for i:=1 step 1 until M do

begin

D[i]:=0.0;

B:=P[i]*N;

for j:=1 step 1 until N do

if I[j]=i then

begin

A:=1.0;

for k:=1 step 1 until p do

A:=exp(-(X[k]-U[j,k])²*B/2)*A;

D[i]:=A+D[i];

end;

D[i]:=D[i]*W[i];

end;

C:=0.0;

for i:=1 step 1 until M do

C:=C+D[i];

for i:=1 step 1 until M do

D[i]:=D[i]/C;

A:=D[1];

MPAR:=1;

for i:=2 step 1 until M do

if D[i]>A then

```
begin  
  A:=D[i];  
  MPAR:=i;  
end;  
for i:=1 step 1 until M do  
begin  
  A:=0.0;  
  for j:=1 step 1 until M do  
    A:=A+Q[i,j]*D[j];  
    W[i]:=A/(sqrt(B)*p*B);  
  end;  
end MPAR;  
?
```

ZMODYFIKOWANY ALGORYTM „h(N) - TY NAJBLIŻSZY SĄSIAD”

1. Nazwa i typ procedury.

integer procedure MHNNN(I,U,W,p,M,P,Q,X,N)

2. Funkcja procedury i opis metody.

Procedura MHNNN zalicza dany p wymiarowy obraz X do jednej z M klas przy użyciu algorytmu (3.11) wykorzystującego nieparametryczny estymator gęstości Loftsgaardena i Quesenbery'ego. Przyjęto w nim $h(N_i) = \sqrt{N_i}$.

3. Schemat blokowy.

Rysunek 3.1.

4. Dane i parametry.

Patrz procedura MPAR.

5. Wyniki.

Po wyjściu z procedury pod jej nazwą podstawiony zostanie numer klasy, a tablica W[1:M] będzie zawierać nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MHNNN należy obliczyć prawdopodobieństwa początkowe i prawdopodobieństwa przejścia łańcucha Markowa przy użyciu procedury PRAWDOPODOBIENSTWO.

7. Uwagi.

Przed pierwszym wywołaniem procedury MHNNN należy dokonać podstawienia

$$W[i] := N / (\text{entier}(\text{sqrt}(P[i] \times N)) - 1) \quad , \quad i=1,2,\dots,M$$

i w trakcie klasyfikacji kolejnych pomiarów, elementów tablicy W nie zmieniać.

8. Tabulogram.

integer procedure MHNNN(I) pomiary uczące:(U) współczynniki wagowe:(W) wymiar obrazów:(p) ilość klas:(M) prawdopodobieństwa początkowe:(P) prawdopodobieństwa przejścia:(Q) klasyfikowany pomiar:(X) długość ciągu uczącego(N);

value p,M,N;

integer p,M,N;


```

integer array I;
array U,W,P,Q,X;
begin comment procedura MHNNN zalicza dany p wymiarowy pomiar X
do jednej z M klas przy uzyciu zmodyfikowanego al
gorytmu h(N)-ty najblizszy sasiad;
integer i,j,k,m;
real A;
array D[1:N],AD[1:M];
integer array H[1:M],J[1:N];
for j:=1 step 1 until N do
begin
D[j]:=0.0;
J[j]:=I[j];
for k:=1 step 1 until p do
D[j]:=D[j]+(U[j,k]-X[k])2;
D[j]:=sqrt(D[j]);
end;
begin comment metoda Shella porzadkowania tablicy;
m:=-N;
for m:=m÷2 while m<0 do
for j:=1 step 1 until N+m do
begin
for k:=j step m until 1 do
if D[k]<D[k-m] then go to END
else begin
exch(1,D[k],D[k-m]);
exch(1,J[k],J[k-m]);
end;
END: end;
end SORT;
for i:=1 step 1 until M do
H[i]:=entier(sqrt(P[i]*N));
for i:=1 step 1 until M do
begin
k:=0;
for j:=1 step 1 until N do
if J[j]=i then
begin
k:=k+1;
if k=H[i] then
begin

```

```

        AD[i]:=D[k]p*W[i];
        go to LOOP;
    end;
end;
LOOP: end;
A:=0.0;
for i:=1 step 1 until M do
A:=A+AD[i];
for i:=1 step 1 until M do
AD[i]:=AD[i]/A;
A:=AD[1];
MHNNN:=1;
for i:=2 step 1 until M do
if AD[i]<A then
begin
A:=AD[i];
MHNNN:=i;
end;
for i:=1 step 1 until M do
begin
A:=0.0;
for j:=1 step 1 until M do
A:=A+Q[i,j]/AD[j];
W[i]:=(P[i]*N)/((H[i]-1)*A);
end;
end MHNNN;
?
```

ZMODYFIKOWANY ALGORYTM WYKORZYSTUJĄCY ESTYMATOR GĘSTOSCI MURTHY'EGO

1. Nazwa i typ procedury.

integer procedure MMURTH(I,U,W,p,M,P,Q,X,N)

2. Funkcja procedury i opis metody.

Procedura MMURTH zalicza dany p wymiarowy obraz X do jednej z M klas przy użyciu algorytmu (3.34) wykorzystującego nieparametryczny estymator gęstości Murthy'ego.

3. Schemat blokowy.

Rysunek 3.3.

4. Dane i parametry.

Patrz procedura MPAR.

5. Wyniki.

Po wyjściu z procedury pod jej nazwą zostanie podstawiony numer klasy, a tablica $W[1:M]$ będzie zawierać nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MMURTH należy obliczyć prawdopodobieństwa początkowe i przejścia łańcucha Markowa przy użyciu procedury PRAWDOPODOBIENSTWO.

7. Uwagi.

Przed pierwszym wywołaniem procedury MMURTH należy dokonać podstawienia

$$W[i] := 1/N, \quad i=1,2,\dots,M,$$

i w trakcie klasyfikacji kolejnych pomiarów, elementów tablicy W nie zmieniać.

8. Tabulogram.

integer procedure MMURTH(I) pomiary uczace:(U) współczynniki wagowe:(W) wymiar obrazow:(p) ilosc klas:(M) prawdopodobieństwa początkowe:(P) prawdopodobieństwa przejścia:(Q) klasyfikowany obiekt:(X) długość ciągu uczacego:(N);

value p,M,N;

integer p,M,N;

integer array I;

array U,W,P,Q,X;

begin comment procedura MMURTH zalicza dany p wymiarowy pomiar X do jednej z M klas przy uzyciu algorytmu wyko rzystujacego nieparametryczny estymator gestosci Murthy 'ego;

```

integer i,j,k;
real A,B,C;
array D[1:M];
for i:=1 step 1 until M do
begin
  D[i]:=0.0;
  B:=(P[i]*N)†(1/(p*2));
  for j:=1 step 1 until N do
  if I[j]=i then
  begin
    A:=1.0;
    for k:=1 step 1 until p do
    A:=A*sin(B*(U[j,k]-X[k]))/(U[j,k]-X[k]);
    D[i]:=A+D[i];
  end;
  D[i]:=D[i]*W[i];
end;
C:=0.0;
for i:=1 step 1 until M do
C:=C+D[i];
for i:=1 step 1 until M do
D[i]:=D[i]/C;
A:=D[1];
MMURTH:=1;
for i:=2 step 1 until M do
if D[i]>A then
begin
  A:=D[i];
  MMURTH:=i;
end;
for i:=1 step 1 until M do
begin
  A:=0.0;
  for j:=1 step 1 until M do
  A:=A+Q[i,j]*D[j];
  W[i]:=A/(P[i]*N);

```

end ;

end MMURTH;

?

ZMODYFIKOWANY ALGORYTM „NAJMNIEJSZY PRZEDZIAŁ”

1. Nazwa i typ procedury.

integer procedure MLI(U,H,M,p,W,X,Q)

2. Funkcja procedury i opis metody.

Procedura MLI zalicza dany p wymiarowy obraz X do jednej z M klas przy użyciu zmodyfikowanego algorytmu "najmniejszy przedział" (3.41)

3. Schemat blokowy.

Rysunek 3.4.

4. Dane i parametry.

$U[1:N,1:p]$ - dwuwskaźnikowa tablica typu real uporządkowanych pomiarów uczących,

$H[0:M]$ - jednowskaźnikowa tablica typu integer wykazu par granicznych,

pozostałe parametry jak w algorytmie MPAR.

5. Wyniki.

Po wyjściu z procedury MLI pod jej nazwą podstawiony zostanie numer klasy, a tablica $W[1:M]$ będzie zawierać nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MLI należy obliczyć prawdopodobieństwa przejścia przy użyciu procedury PRAWDOPODOBIENSTWO oraz uporządkować tablicę pomiarów uczących przy użyciu procedury SORT.

7. Uwagi.

Przed pierwszym wywołaniem procedury MLI należy dokonać podstawienia

$$W[i]=1.0, \quad i=1,2,\dots,M.$$

Jeżeli klasyfikowany pomiar X leży poza zestawem obrazów uczących to pod nazwą MLI podstawione zostanie 0, a tablica nowych współczynników wagowych będzie nieokreślona.

8. Tabulogram.

integer procedure MLI(U) pary graniczne:(H) ilosc klas:(M) ilosc wymiarow obrazow:(p) wspolczynniki wagowe:(W) klasyfikowany pomiar:(X) prawdopodobienstwa przejścia:(Q);

```

value M,p;
integer M,p;
integer array H;
array U,W,X,Q;
begin comment procedura MLI zalicza dany p wymiarowy pomiar X
do jednej z M klas przy uzyciu zmodyfikowanego
algorytmu najmniejszy przedzial;
  integer i,j,k,t;
  real A;
  array DEL[1:M];
  integer array L[1:M];
  for t:=1 step 1 until M do
    L[t]:=0;
    t:=1;
    for i:=1 step 1 until M do
      begin
        for k:=1 step 1 until p do
          if X[k]<U[H[i-1]+1,k] or X[k]>2*U[H[i],k]-U[H[i-1],k]
          then begin
            L[t]:=i;
            t:=t+1;
            go to END;
          end;
        END: end;
        if t=M+1 then
          begin
            MLI:=0;
            go to STOP;
          end;
        for i:=1 step 1 until M do
          begin
            DEL[i]:=5&150;
            for j:=1 step 1 until M do
              if L[j]=i then go to LOOP;
              DEL[i]:=1.0;
              for k:=1 step 1 until p do
                for j:=H[i-1]+1 step 1 until H[i] do
                  if X[k]<U[j,k] then DEL[i]:=DEL[i]*abs(U[j,k]-U[j-1,k]);
                  DEL[i]:=DEL[i]*W[i];
            LOOP: end;
            A:=DEL[1];

```

```
MLI:=1;
for i:=2 step 1 until M do
if DEL[i]<A then
begin
    A:=DEL[i];
    MLI:=i;
end;
for i:=1 step 1 until M do
begin
    A:=0.0;
    for j:=1 step 1 until M do
    A:=A+Q[i,j]/DEL[j];
    W[i]:=(H[i]-H[i-1])tp/A;
end;
end MLI;
?
```


PORZĄDKOWANIE CIĄGU UCZĄCEGO DLA POTRZEB ALGORYTMU MLI

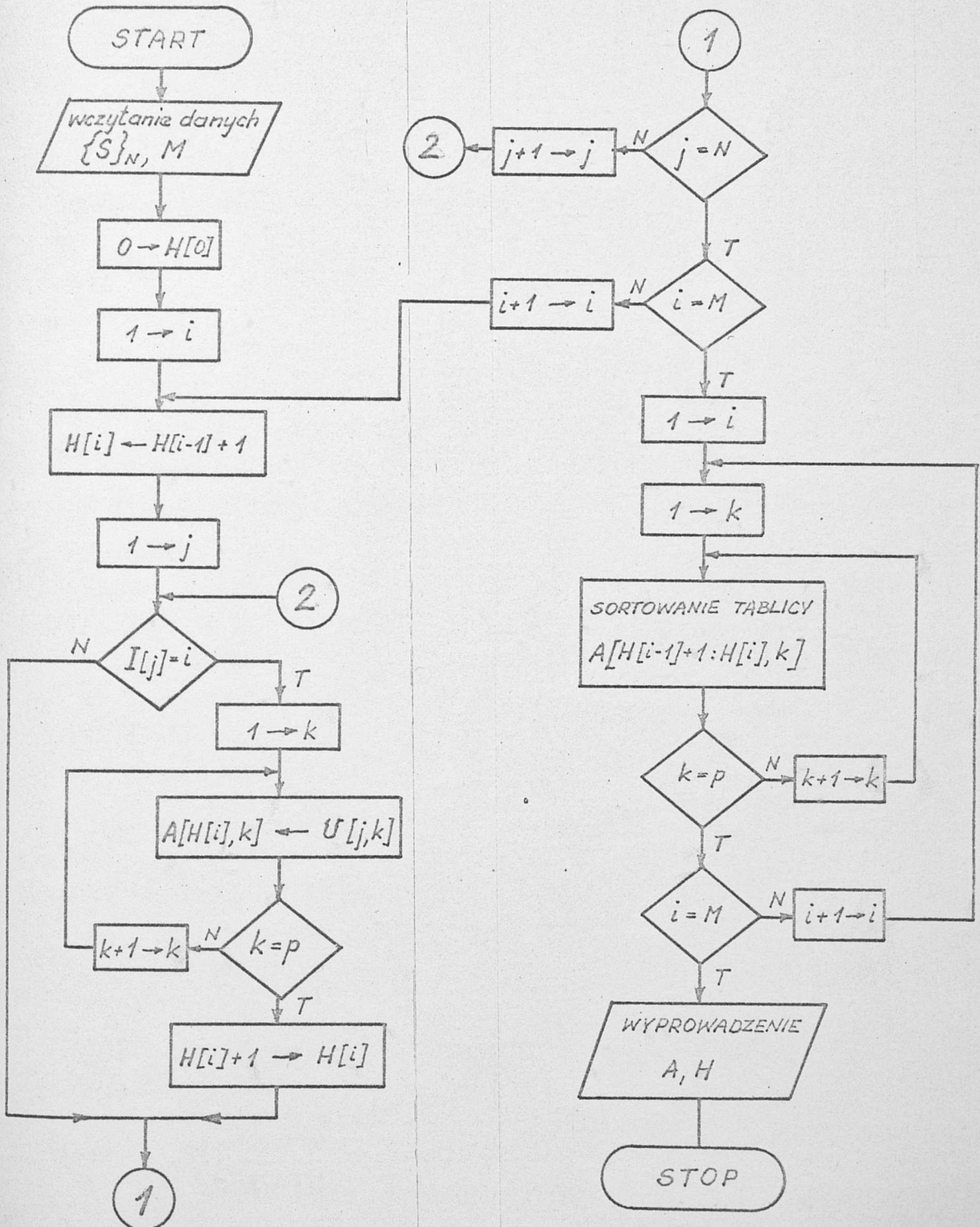
1. Nazwa procedury.

procedure SORT(I,U,p,M,N,A,H)

2. Funkcja procedury i opis metody.

Procedura SORT porządkuje rosnąco współrzędne pomiarów uczących z poszczególnych klas według metody Shella [37].

3. Schemat blokowy.



4. Dane i parametry.

$I[1:N]$ - jednowskaźnikowa tablica typu integer prawidłowych klasyfikacji obrazów uczących,

$U[1:N,1:p]$ - dwuwskaźnikowa tablica typu real pomiarów uczących,
 p - wymiar przestrzeni obrazów - zmienna typu integer,

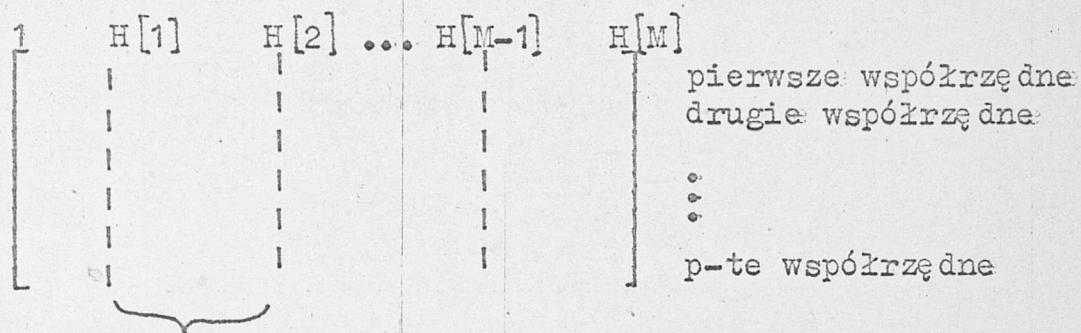
M - ilość klas - zmienna typu integer,

N - długość ciągu uczącego - zmienna typu integer.

5. Wyniki.

Po wyjściu z procedury SORT dwuwskaźnikowa tablica typu real

$A[1:N,1:p]$ będzie zawierać obrazy uczące uporządkowane w następujący sposób:



}
 obrazy uczące z drugiej klasy
 uszeregowane rosnąco

a jednowskaźnikowa tablica typu integer $H[0:M]$ zawiera numery kolumn rozdzielających obrazy uczące na M klas.

6. Tabulogram.

```
procedure SORT(I) pomiary uczace:(U) wymiar obrazow:(p) ilosc
  klas:(M) dlugosc ciagu uczacego:(N) uporządkowane
  obrazy uczace:(A) wykaz par granicznych:(H);
```

```
  value p,M,N;
```

```
  integer p,M,N;
```

```
  integer array I,H;
```

```
  array U,A;
```

```
  begin comment procedura SORT porządkuje rosnaco według metody
    Shella pomiary uczace dla potrzeb algorytmu MLI;
```

```
    integer i,j,k,l,m,t;
```

```
    H[0]:=0;
```

```
    for i:=1 step 1 until M do
```

```
      begin
```

```
        H[i]:=H[i-1]+1;
```

```
        for j:=1 step 1 until N do
```

```

  if I[j]=i then
    begin
      for k:=1 step 1 until p do
        A[H[i],k]:=U[j,k];
        H[i]:=H[i]+1;
      end;
    end;
  for i:=1 step 1 until M do
  for k:=1 step 1 until p do
  begin
    t:=H[i]-H[i-1];
    m:=-t;
    for m:=m÷2 while m<0 do
    for j:=1 step 1 until t+m do
    begin
      for l:=j step m until 1 do
      if A[l+H[i-1],k]<A[l-m+H[i-1],k] then go to END
      else exch(1,A[l+H[i-1],k],A[l-m+H[i-1],k]);
    end;
  END:end;
  end;
end SORT;
?
```

ZMODYFIKOWANY ALGORYTM "NAJBLIŻSZA ŚREDNIA"

1. Nazwa i typ procedury.

integer procedure MNM(S,W,p,M,P,Q,X)

2. Funkcja procedury i opis metody.

Procedura MNM zalicza dany p wymiarowy pomiar X do jednej z M klas przy użyciu zmodyfikowanego algorytmu "najbliższa średnia"(3.46).

3. Schemat blokowy.

Rysunek 3.5.

4. Dane i parametry.

S [1:M,1:p] - dwuwskaznikowa tablica typu real obrazów średnich, pozostałe parametry jak w algorytmie MPAR.

5. Wyniki.

Po wyjściu z procedury pod jej nazwą zostanie podstawiony numer klasy, a tablica W [1:M] będzie zawierać nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MNM należy obliczyć prawdopodobieństwa początkowe i prawdopodobieństwa przejścia łańcucha Markowa przy użyciu procedury PRAWDOPODOBIENSTWO oraz na podstawie ciągu uczącego wyznaczyć obrazy średnie wywołując procedurę SREDNIA.

7. Uwagi.

Przed pierwszym wywołaniem procedury MNM należy dokonać podstawienia

$$W[i]:=1.0, \quad i=1,2,\dots,M$$

i następnie elementów tablicy W nie zmieniać.

8. Tabulogram.

integer procedure MNM(S) współczynniki wagowe:(W) wymiar obrazow:
 (p) ilosc klas:(M) prawdopodobienstwa poczatkowe:(P) prawdopodobienstwa przejscia:(Q) klasyfikowany obiekt:(X);

value M,p;

integer M,p;

array S,W,P,Q,X;

begin comment procedura MNM zalicza dany p wymiarowy pomiar X do jednej z M klas przy użyciu zmodyfikowanego algorytmu najbliższa średnia;

```

integer i,j,k;
real A,B;
real array DEL[1:M];
for i:=1 step 1 until M do
begin
  DEL[i]:=0.0;
  for k:=1 step 1 until p do
  DEL[i]:=(S[i,k]-X[k])2+DEL[i];
  DEL[i]:=sqrt(DEL[i])*W[i];
end;
A:=0.0;
for i:=1 step 1 until M do
A:=A+DEL[i];
for i:=1 step 1 until M do
DEL[i]:=DEL[i]/A;
A:=DEL[1];
MNM:=1;
for i:=2 step 1 until M do
if DEL[i]<A then
begin
  A:=DEL[i];
  MNM:=i;
end;
for i:=1 step 1 until M do
begin
  B:=0.0;
  for j:=1 step 1 until M do
  B:=B+Q[i,j]/DEL[j];
  W[i]:=P[i]/B;
end;
end MNM;
?
```

OBLICZANIE OBRAZÓW ŚREDNICH

1. Nazwa procedury.

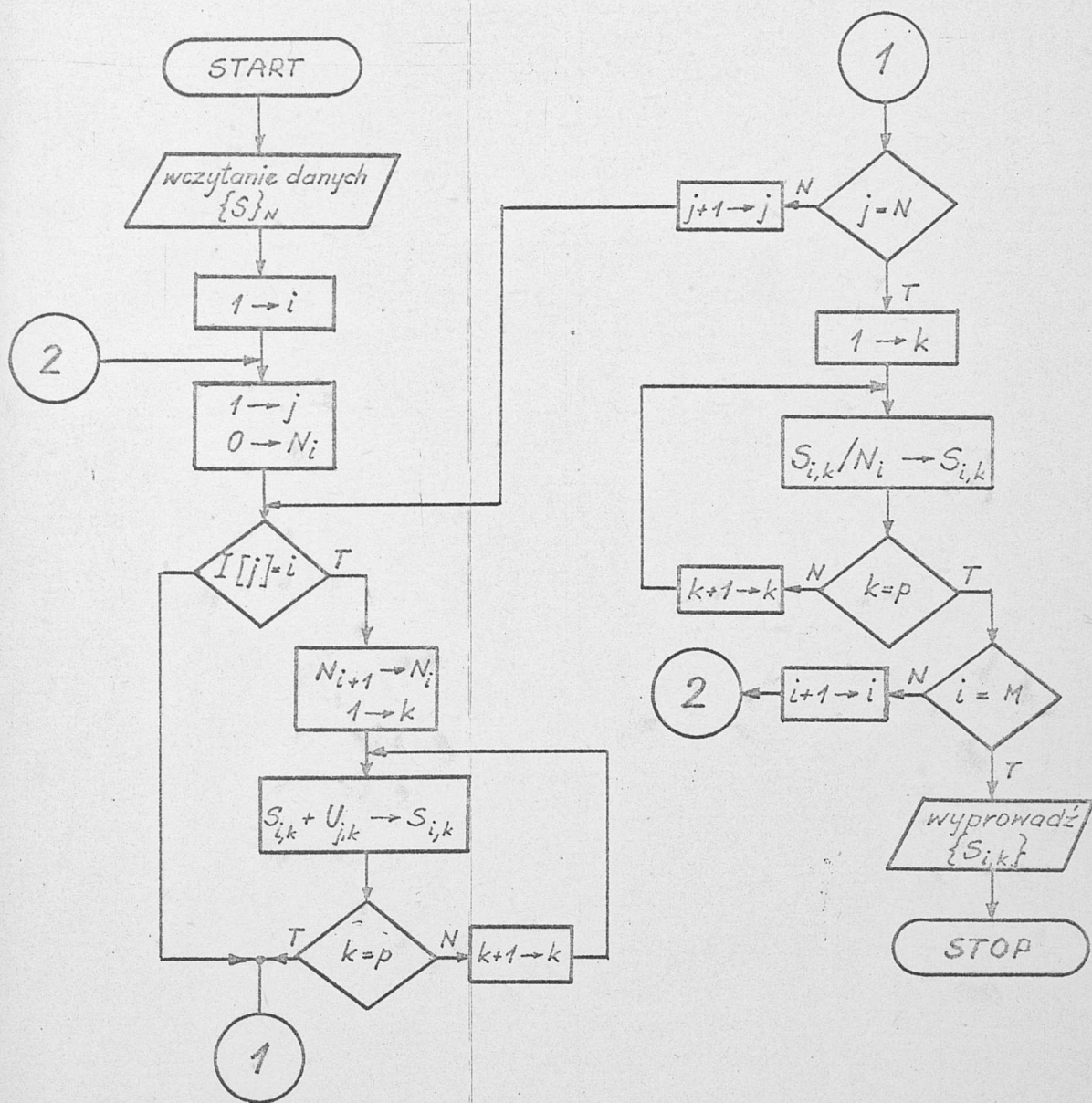
procedure SREDNIA(I,U,N,p,M,S)

2. Funkcja procedury i opis metody.

Procedura SREDNIA oblicza dla potrzeb algorytmu MNM M obrazów średnich p wymiarowych

$$S_{i,k} = \frac{1}{N_i} \sum_{j=1}^{N_i} U_{j,k} \quad , \quad k=1,2,\dots,p.$$

3. Schemat blokowy.



4. Dane i parametry.

$I[1:N]$ - jednowskaźnikowa tablica typu integer prawidłowych klasy-

fikacji obrazów uczących,

$U[1:N, 1:p]$ - dwuwskaznikowa tablica typu real pomiarów uczących,

N - długość ciągu uczącego - zmienna typu integer,

p - wymiar przestrzeni obrazów - zmienna typu integer,

M - ilość klas - zmienna typu integer.

5. Wyniki.

Po wyjściu z procedury dwuwskaznikowa tablica typu real $S[1:M, 1:p]$ będzie zawierać współrzędne obrazów średnich:

$S[i, k]$ - k -ta współrzędna średniego obrazu z i -tej klasy.

6. Tabulogram.

procedure SREDNIA(I) pomiary uczace:(U) dlugosc ciagu uczacego:(N)
wymiar obrazow:(p) ilosc klas:(M) pomiary srednie:(S);

value p, M, N;

integer p, M, N;

integer array I;

array U, S;

begin comment procedura oblicza na podstawie ciagu uczacego M
obrazow srednich p wymiarowych;

integer i, j, k, l;

for i:=1 step 1 until M do

for k:=1 step 1 until p do

$S[i, k] := 0.0;$

for i:=1 step 1 until M do

begin

l:=0;

for j:=1 step 1 until N do

if I[j]=i then

begin

l:=l+1;

for k:=1 step 1 until p do

$S[i, k] := S[i, k] + U[j, k];$

end;

for k:=1 step 1 until p do

$S[i, k] := S[i, k] / l;$

end;

end SREDNIA;

?

ZMODYFIKOWANY ALGORYTM "NAJBLIŻSZY SĄSIAD"

1. Nazwa i typ procedury.

integer procedure MNN(I,U,W,p,M,P,Q,X,N)

2. Funkcja procedury i opis metody.

Procedura MNN zalicza dany p wymiarowy obraz X do jednej z M klas przy użyciu zmodyfikowanego algorytmu "najbliższy sąsiad" (3.48).

3. Schemat blokowy.

Rysunek 3.6.

4. Dane i parametry.

Patrz procedura MPAR.

5. Wyniki.

Po wyjściu z procedury pod jej nazwą zostanie podstawiony numer klasy, a tablica $W[1:M]$ będzie zawierać nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MNN należy obliczyć prawdopodobieństwa początkowe i przejścia łańcucha Markowa wykorzystując procedurę PRAWDOPODOBIENSTWO.

7. Uwagi.

Przed pierwszym wywołaniem procedury MNN należy dokonać podstawienia

$$W[i] := 1.0, \quad i=1,2,\dots,M$$

i w trakcie klasyfikacji kolejnych pomiarów, elementów tablicy W nie zmieniać.

8. Tabulogram.

```
integer procedure MNN(I) pomiary uczace:(U) współczynniki wago
we:(W) wymiar obrazow:(p) ilosc klas:(M) praw
dopodobienstwa poczatkowe:(P) prawdopodobien
stwa przejścia:(Q) klasyfikowany obiekt:(X)
długosc ciagu uczacego:(N);
```

```
value p,M,N;
```

```
integer p,M,N;
```

```
integer array I;
```

```
array U,W,P,Q,X;
```

```
begin comment procedura MNN zalicza dany p wymiarowy obiekt X
do jednej z M klas przy użyciu zmodyfikowanego
```



```

        algorytmu najblizszy sasiad;
integer i,j,k;
real A,B;
real array DEL[1:M];
for i:=1 step 1 until M do
begin
    DEL[i]:=5&150;
    for j:=1 step 1 until N do
    if I[j]=i then
    begin
        B:=0.0;
        for k:=1 step 1 until p do
        B:=(U[j,k]-X[k])2+B;
        B:=sqrt(B);
        if DEL[i]>B then DEL[i]:=B;
    end;
    DEL[i]:=DEL[i]*W[i];
end;
A:=0.0;
for i:=1 step 1 until M do
A:=A+DEL[i];
for i:=1 step 1 until M do
DEL[i]:=DEL[i]/A;
A:=DEL[1];
MNN:=1;
for i:=2 step 1 until M do
if DEL[i]<A then
begin
    A:=DEL[i];
    MNN:=i;
end;
for i:=1 step 1 until M do
begin
    B:=0.0;
    for j:=1 step 1 until M do
    B:=B+Q[i,j]/DEL[j];
    W[i]:=P[i]/B;
end;
end MNN;
?

```

ZMODYFIKOWANY ALGORYTM "ALFA NAJBLIŻSZYCH SĄSIADÓW"

1. Nazwa i typ procedury.

integer procedure MALFANN(I,U,W,p,M,P,Q,X,N,ALFA)

2. Funkcja procedury i opis metody.

Procedura MALFANN zalicza dany p wymiarowy pomiar X do jednej z M klas przy użyciu zmodyfikowanego algorytmu "alfa najbliższych sąsiadów" (3.51).

3. Schemat blokowy.

Rysunek 3.7.

4. Dane i parametry.

ALFA - ilość najbliższych sąsiadów - zmienna typu integer, pozostałe parametry jak w algorytmie MPAR.

5. Wyniki.

Po wyjściu z procedury pod jej nazwą podstawiony zostanie numer klasy, a tablica $W[1:M]$ zawierać będzie nowe współczynniki wagowe.

6. Wykorzystywane podprogramy.

Przed pierwszym wywołaniem procedury MALFANN obliczyć należy prawdopodobieństwa początkowe i przejścia łańcucha Markowa wykorzystując procedurę PRAWDOPODOBIENSTWO.

7. Uwagi.

Przed pierwszym wywołaniem procedury MALFANN należy dokonać podstawienia

$$W[i] := 1.0, \quad i=1,2,\dots,M$$

i w trakcie klasyfikacji kolejnych pomiarów, elementów tablicy W nie zmieniać.

8. Tabulogram.

integer procedure MALFANN(I) pomiary uczace:(U) współczynniki wagowe:(W) wymiar obrazow:(p) ilosc klas:(M) prawdopodobienstwa poczatkowe:(P) prawdopodobienstwa przejścia:(Q) klasyfikowany pomiar:(X) dlugosc ciagu uczacego:(N) ilosc najblizszych sasiadow:(ALFA);

value p,M,N,ALFA;

integer p,M,N,ALFA;

integer array I;

```

array U,W,P,Q,X;
begin comment procedura MALFANN zalicza dany p wymiarowy pomiar
      X do jednej z M klas przy uzyciu zmodyfikowanego
      algorytmu ALFA najblizszych sasiadow;
  integer i,j,k,l;
  real A,B;
  boolean C;
  integer array L[1:ALFA];
  array D[1:ALFA],AD[1:M];
  for l:=1 step 1 until ALFA do
    D[l]:=5&150;
  for j:=1 step 1 until N do
    begin
      A:=0.0;
      for k:=1 step 1 until p do
        A:=(X[k]-U[j,k])2+A;
      A:=sqrt(A);
      C:=false;
      for l:=1 step 1 until ALFA do C:=C or A<D[l];
      if C then
        begin
          B:=D[l];
          k:=1;
          for l:=2 step 1 until ALFA do
            if B<D[l] then
              begin
                B:=D[l];
                k:=l;
              end;
          D[k]:=A;
          L[k]:=I[j];
        end;
    end;
  for i:=1 step 1 until M do
    begin
      k:=0;
      for l:=1 step 1 until ALFA do
        if L[l]=i then k:=k+1;
      AD[i]:=k*W[i];
    end;
end;

```

```
A:=0.0;
for i:=1 step 1 until M do
A:=A+AD[i];
for i:=1 step 1 until M do
AD[i]:=AD[i]/A;
A:=AD[1];
MALFANN:=1;
for i:=2 step 1 until M do
if AD[i]>A then
begin
A:=AD[i];
MALFANN:=i;
end;
for i:=1 step 1 until M do
begin
A:=0.0;
for j:=1 step 1 until M do
A:=Q[i,j]*AD[j]+A;
W[i]:=A/P[i];
end;
end MALFANN;
?
```