Instytut Informatyki, Automatyki i Robotyki

Seria: PREPRINTY nr 34/2006

Hybrydowe algorytmy ewolucyjno-gradientowe dla problemów optymalnego sterowania okresowego z ograniczeniami zasobowo-technologicznymi

(rozprawa doktorska)

Marek Skowron

Promotor: Dr hab. inż. Krystyn Styczeń, prof. PWr

Słowa kluczowe:
- optymalne sterowanie okresowe
- algorytm ewolucyjny
- metoda rzutowanego gradientu
- funkcja kary
- ograniczenie zasobowe
- ograniczenie technologiczne
- ograniczenie stabilnościowe

Wrocław,13.06.2006 r.

INSTYTUT INFORMATYKI, AUTOMATYKI I ROBOTYKI
POLITECHNIKI WROCŁAWSKIEJ

Hybrydowe algorytmy ewolucyjno-gradientowe dla problemów optymalnego sterowania okresowego z ograniczeniami zasobowo-technologicznymi

(rozprawa doktorska)

Marek Skowron

Promotor: dr hab. inż. Krystyn Styczeń, prof. nadzw. PWr.

WROCŁAW

czerwiec 2006

Spis treści

1. Wstęp ... 2
1.1. Dlaczego należy zajmować się optymalnym sterowaniem okresowym? ... 2
1.2. Problem optymalnego sterowania okresowego ... 4
1.3. Cel pracy doktorskiej ... 11
1.4. Zestawienie najważniejszych rezultatów pracy ... 15
1.5. Zawartość pracy ... 16
2. Algorytmy ewolucyjne i sterowanie ... 20
2.1. Wstęp do algorytmów ewolucyjnych ... 20
2.2. Podstawowe pojęcia stosowane w algorytmach ewolucyjnych ... 22
2.3. Kodowanie populacji ... 23
2.3.1. Kodowanie binarne ... 24
2.3.2. Kodowanie rzeczywiste ... 26
2.4. Zarządzanie populacją ... 27
2.4.1. Ruletka ... 31
2.4.2. Reprodukcja turniejowa ... 32
2.4.3. Reprodukcja rangowa ... 33
2.5. Operatory krzyżowania ... 35
2.5.1. Krzyżowanie wymieniające ... 36
2.5.2. Krzyżowanie równomierne ... 38
2.5.3. Krzyżowanie arytmetyczne ... 39
2.6. Operatory mutacji ... 39
2.6.1. Mutacja równomierna ... 41
2.6.2. Mutacja nierównomierna ... 43
2.6.3. Mutacja brzegowa ... 44
2.7. Kryterium stopu algorytmu ... 45
3. Algorytmy ewolucyjne dla problemów GOSO ... 49
3.1. Kodowanie problemu okresowego ... 49
3.2. Algorytm optymalizacji dla problemu GOSO z chwilowymi ograniczeniami zasobowymi ... 52
3.2.1. Problem GOSO z chwilowymi ograniczeniami zasobowymi ... 52
3.2.2. Ewolucyjny algorytm optymalizacji (Algorytm 1) ... 52
3.2.3. Przykłady obliczeniowe ... 57
3.3. Algorytm optymalizacji dla problemu GOSO z ograniczeniami zasobowymi ... 66
3.3.1. Problem GOSO z chwilowymi i uśrednionymi ograniczeniami zasobowymi ... 66
3.3.2. Ewolucyjny algorytm optymalizacji (Algorytm 2) ... 67
3.3.3. Przykłady obliczeniowe ... 72
3.4. Algorytm optymalizacji dla problemów GOSO z ograniczeniami zasobowymi, technologicznymi oraz stabilnościowymi ... 83
3.4.1. Problem GOSO z ograniczeniami zasobowymi, technologicznymi oraz stabilnościowymi ... 83
3.4.2. Ewolucyjny algorytm optymalizacji (Algorytm 3) ... 84
3.4.3. Przykłady obliczeniowe ... 92
4. Algorytmy wspomagające ... 101
4.1. Metoda λ-rzutowanego gradientu ... 103
4.2. Metoda przesuwanej funkcji kary i λ-rzutowanego gradientu ... 107
4.3. Przykłady obliczeniowe ... 113
5. Podsumowanie ... 120
6. Literatura ... 124

1. WSTĘP

1.1. Dlaczego należy zajmować się optymalnym sterowaniem okresowym?

Tradycyjne podejście do problemów optymalnego sterowania autonomicznymi, ciągłymi procesami technologicznymi polega na wyborze ich optymalnego statycznego przebiegu. W praktyce wygląda to tak, że najpierw numerycznie wyznaczane jest optymalne, statyczne rozwiązanie danego problemu, a następnie projektowane są proste, stabilizujące układy automatyki, których zadaniem jest taka regulacja procesu, aby – niezależnie od fluktuacji parametrów obiektu i czynników zewnętrznych – przebiegał on w warunkach zbliżonych do wyznaczonego wcześniej optymalnego rozwiązania statycznego. Inaczej mówiąc, dąży się do tego, by wszystkie zmienne procesowe danego procesu – zarówno sterujące (natężenie dopływu lub stężenie substratów potrzebnych do prowadzenia procesu, itp.) jak i opisujące stan procesu (temperatura i ciśnienie w aparaturze procesowej, itp.) – były stałe w czasie i równe wyznaczonym wartościom optymalnym.

Fakt, że opisana wyżej procedura sterowania ciągłymi procesami technologicznymi jest bardzo często stosowana, nie oznacza, że jest ona optymalna w klasie wszystkich procesów prowadzonych w długim horyzoncie czasowym – horyzoncie znacznie przewyższającym stałe czasowe procesu. Od roku 1935, kiedy to odkryto ekstrakcję pulsacyjną, wiadomo bowiem, iż wymuszenie cyklicznej pracy procesu może znacznie zwiększyć jego wydajność w porównaniu z wydajnością tego procesu w stanie ustalonym [40]. Sterowanie okresowe, w odróżnieniu od sterowania statycznego, wykorzystuje dynamikę procesu, dzięki czemu można kształtować przebieg stanu procesu i jego średnie charakterystyki w szerokim zakresie, a co za tym idzie, można poprawiać wartości wskaźników jakości szeregu procesów produkcyjnych przy zachowaniu średnich wartości ograniczeń charakterystycznych dla sterowania okresowego [40], [60], [65], [72]. Wspomniana poprawa odnosi się m.in. do wskaźników jakości opisujących średnią wydajność procesu, średnie zużycie surowców potrzebnych do prowadzenia procesu, średni koszt poniesiony na prowadzenie procesu, czy średnią selektywność procesu.

Pierwsza ekstrakcja pulsacyjna zapewniała co prawda większą wydajność procesu, jednak sterowanie pulsacyjne nie zaczęło być powszechnie stosowane – w ówczesnych czasach bowiem, koszt realizacji takiego sterowania wielokrotnie przewyższał zyski uzyskane ze zwiększenia wydajności procesu. Dzięki pionierskiemu zastosowaniu eks-

trakcji pulsacyjnej zrozumiano jednak, że optymalne sterowanie statyczne nie zawsze jest najlepsze, dlatego też należy poszukiwać innych sposobów sterowania – m.in. należy starać się, o ile jest to możliwe, stosować sterowanie okresowe. Okresowy sposób prowadzenia procesu wykorzystuje bowiem możliwości tkwiące w dynamice procesu, a jednocześnie pozwala on ograniczyć analizę problemu sterowania do jednego tylko cyklu procesu – dużo krótszego od całego horyzontu czasowego procesu ciągłego.

Dzisiejszy stan techniki pozwala przezwyciężyć trudności naukowców z lat trzydziestych. Nowe technologie, super-szybkie komputery oraz wiedza zdobyta przez szereg lat pozwalają istotnie poszerzyć zakres stosowania sterowania okresowego.

Dziś nikogo nie dziwi już fakt, że w systemach ekologicznych czy agrotechnicznych można jednocześnie zmniejszać negatywny wpływ gospodarki na środowisko i zwiększać wydajność procesów produkcyjnych (np. dzięki cyklicznemu sterowaniu rozwojem populacji biologicznych) [34], [42], [72], [74], [82], [86], [87]; w analogiczny sposób można również zwiększać wydajność systemów biotechnologicznych [1], [21], [63], [69], [76], [92], [94], [102], [122], [128]. Stosując sterowanie okresowe można także osiągać znaczną poprawę wydajności procesu w przypadku wielu reakcji chemicznych [7], [19], [31], [40], [44], [45], [73], [78], [79], [80], [89], [93], [104], [106], [138], [139].

Zastosowanie sterowania okresowego nie ogranicza się oczywiście jedynie do wspomnianych wyżej systemów ekologicznych, biotechnologicznych, czy chemicznych. Również w takich obszarach przemysłu jak elektroenergetyka, termodynamika, mechanika, czy hutnictwo – poprzez cykliczne prowadzenie procesu – można zwiększyć jego wydajność i poprawić jego efektywność [15], [39], [60], [62], [77] [99], [101], [117], [115]. Sterowanie cykliczne stosowane jest także w systemach ekonomiczno-gospodarczych [22], [64], [81], [133], [145], gdzie jest ono wykorzystywane do dynamicznego tworzenia cen w celu zwiększenia średniego zysku, do lepszego zarządzania produkcją i powierzchniami magazynowymi, czy też do szybszego dostosowywania produkcji do zmieniającego się w czasie zapotrzebowania rynku.

Obecnie, gdy z każdym dniem ceny na stacjach benzynowych są coraz wyższe, interesujące mogą być też badania nad zastosowaniem cyklicznego sterowania do zwiększania ekonomiczności pojazdów, a tym samym do zmniejszania zużycia paliwa (zagadnienie to – w odniesieniu głównie do przemysłu lotniczego – zostało szeroko opisane m.in. w [52], [54], [23], [24], [38], [95], [125], [124], [127], [143], [144]).

Z powyższych rozważań wynika, że sterowanie okresowe – poprzez zastosowanie jakie znajduje w różnych gałęziach przemysłu i gospodarki – jest obecne w wielu obszarach ludzkiego życia, często poprawiając jego jakość. Dzięki temu, że ten właśnie rodzaj sterowania zapewnia (w porównaniu ze statycznym prowadzeniem procesu) większą wydajność szeregu procesów produkcyjnych, zainteresowanie sterowaniem okresowym oraz możliwością jego wykorzystania jest coraz większe, co jest chyba wystarczającym argumentem na to, że zajmowanie się tą tematyką nie jest „stratą czasu" tym bardziej, że w dobie globalizacji gospodarki i coraz większej konkurencji między firmami nawet niewielkie zwiększenie wydajności procesu (przenoszące się tym samym na zmniejszenie kosztów produkcji) jest często „na wagę złota". Dlatego też zasadne wydają się próby wdrażania sterowania okresowego do procesów, w których nie zostało ono jeszcze zastosowane; uzasadnione są także badania nad doskonaleniem istniejących już metod optymalizacji okresowej, czy też tworzeniem nowych – jeszcze lepszych – metod zwiększających wydajność procesów produkcyjnych.

1.2. Problem optymalnego sterowania okresowego

Z problemami optymalnego sterowania okresowego można zetknąć się wszędzie tam, gdzie tego rodzaju sterowanie jest stosowane, czyli – jak już wcześniej wspomniano – w chemii, biologii, ekologii, różnych systemach technicznych, czy ekonomicznych. Rozległość dziedzin oraz specyficzne własności poszczególnych problemów powodują, że procesy okresowe mają różnorodny charakter. Na przykład, z uwagi na postać trajektorii stanu (względem czasu) można podzielić je na ciągłe, nieciągłe lub półciągłe. Ze względu na sposób wymuszenia cyklicznej pracy na: autonomiczne (w których cykliczność uzyskuje się dzięki odpowiednio dobranemu sterowaniu) oraz nieautonomiczne (w których cykliczność wymuszona jest przez zakłócenia odziaływujące na prowadzony proces). Podziału można również dokonać biorąc pod uwagę okres procesu; w takim przypadku wyróżnione zostaną: problemy wysokoczęstotliwościowe (w których okres dąży do zera), średnioczęstotliwościowe (w których wartość okresu mniej więcej odpowiada stałym czasowym procesu), czy niskoczęstotliwościowe (w których okres zmierza do nieskończoności).

Istotnym jest, że każdy wspomniany powyżej rodzaj problemu należy traktować indywidualnie, inne są bowiem modele opisujące poszczególne typy problemów; w każdym z typów występują inne, czasami charakterystyczne wyłącznie dla niego ograniczenia; dla każdego rodzaju problemu okresowego inny jest także sposób podejścia do

jego optymalizacji. Z uwagi na powyższe oraz biorąc pod uwagę ograniczone ramy tego opracowania w dalszej części pracy rozważany będzie wyłącznie ciągły, autonomiczny, średnioczęstotliwościowy problem optymalnego sterowania, z którym najczęściej można spotkać się w praktyce. Nie oznacza to jednak, że metody optymalizacyjne przedstawione w tej pracy nie mogą być zastosowane do innych typów problemów okresowych. Wręcz przeciwnie, ewolucyjne podejście optymalizacyjne, będące głównym przedmiotem niniejszego opracowania, może być z powodzeniem zastosowane (oczywiście po nieznacznych modyfikacjach) do prawie każdego problemu okresowego.

Ciągły, autonomiczny, średnioczęstotliwościowy problem globalnie optymalnego sterowania okresowego (problem GOSO) polega na minimalizacji funkcji celu:

$$Q(x,u,\tau) = \nu(q), \tag{1}$$

będącej skalarną funkcją wektora wskaźników charakterystycznych

$$q = \frac{1}{\tau}\int_0^{\tau} h(x(t),u(t))dt \tag{2}$$

procesu opisywanego okresowym równaniem stanu

$$\dot{x}(t) = f(x(t),u(t)), \quad t \in [0,\tau], \quad x(0) = x(\tau), \tag{3}$$

na który to proces narzucone są ograniczenia chwilowe

$$\tau \in \mathcal{T}, \quad x(t) \in X, \quad u(t) \in U, \quad t \in [0,\tau], \tag{4}$$

uśrednione ograniczenia zasobowe

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \tag{5}$$

uśrednione ograniczenia technologiczne

$$\phi(q) = 0, \quad \tilde{\phi}(q) \le 0, \tag{6}$$

oraz ograniczenia stabilnościowe

$$\| s(\Phi(x,u,\tau)) \|_{\infty} \le \alpha. \tag{7}$$

W powyższych równaniach $\tau \in R_+$ oznacza okres procesu sterowania, $x \in \mathcal{H}^1([0,\tau];R^{n_x})$ trajektorię stanu, $u \in \mathcal{H}^0\left([0,\tau];R^{n_u}\right)$ sterowanie, a $q \in R^{n_q}$ jest wektorem wskaźników charakterystycznych procesu. Z kolei funkcje ν, h, f, φ, $\tilde{\varphi}$, ϕ $\tilde{\phi}$ określone są w następujący sposób:

$$v : R^{n_q} \to R, \quad h : R^{n_x} \times R^{n_u} \to R^{n_q}, \quad f : R^{n_x} \times R^{n_u} \to R^{n_x},$$

$$\varphi : R^{n_q} \to R^{n_\varphi}, \quad \tilde{\varphi} : R^{n_q} \to R^{n_{\tilde{\varphi}}}, \quad \phi : R^{n_q} \to R^{n_\phi}, \quad \tilde{\phi} : R^{n_q} \to R^{n_{\tilde{\phi}}}.$$

Wypukłe zbiory $\mathcal{T} \in [\tau^-, \tau^+]$, $X \in [x^-, x^+]$, $U \in [u^-, u^+]$ (gdzie $\tau^\pm \in R$, $x^\pm \in R^{n_x}$, $u^\pm \in R^{n_u}$) określają dopuszczalny zakres zmian zmiennych procesowych. Wartości tworzące te zbiory wynikają z ograniczeń technologicznych aparatury używanej do prowadzenia procesu. Mogą one np. określać dopuszczalny zakres wahań temperatury reaktora (który nie spowoduje jego zniszczenia), czy też maksymalne ciśnienie (którego przekroczenie może przyczynić się do wybuchu reaktora). Ograniczenia te mogą opisywać również maksymalne lub minimalne osiągalne natężenie dopływu surowca potrzebnego do prowadzenia procesu lub jego graniczne osiągalne stężenie.

Uśrednione ograniczenia zasobowe (5) określają średnią dostępność surowca potrzebnego do prowadzenia procesu. Mogą one opisywać np. dopuszczalne średnie zużycie energii elektrycznej lub gazu (którego przekroczenie może być obarczone np. dużymi karami finansowymi). W problemach optymalnego sterowania okresowego uśrednione ograniczenia zasobowe pojawiają się również w przypadku, gdy do prowadzenia danego procesu nie można dostarczać w sposób ciągły niezbędnych surowców – ograniczenia te opisują wtedy maksymalne średnie zużycie surowców w czasie, które nie przyczyni się do opróżnienia zapasów magazynowych, a tym samym do zatrzymania procesu. Dla dużej grupy problemów okresowych funkcje φ, $\tilde{\varphi}$ – opisujące uśrednione ograniczenia zasobowe – mogą przybierać postać:

$$\frac{1}{\tau}\int_0^\tau (u(t) - u_s)dt, \quad \frac{1}{\tau}\int_0^\tau (u^2(t) - u_s)dt, \tag{8}$$

gdzie $u_s \in R^{n_u}$ jest zadanym wektorem opisującym średnią dostępność surowca potrzebnego do prowadzenia procesu.

Uśrednione ograniczenia technologiczne (6) opisywane są przez nieliniowe funkcje zależne od sterowania i stanu. Przy ich pomocy najczęściej określana jest np. dopuszczalna, średnia koncentracja czynnika niepożądanego (bądź nawet szkodliwego) w końcowym produkcie danego procesu. Omawiane ograniczenia mogą również opisywać dopuszczalną, średnią ilość odpadów produkowanych w trakcie trwania procesu, lub też jego selektywność.

Ograniczenia stabilnościowe w opisanym problemie okresowym zdefiniowane są za pomocą największego (co do wartości) multiplikatora Floquet'a oraz parametru $\alpha \in R_+$ określającego dopuszczalny, lokalny poziom stabilności procesu (tzw. poziom Floquet'a). Największa wartość multiplikatora Floquet'a wyznaczana jest w następujący sposób: $\| s \|_\infty = \max_{1 \le i \le n_x} | s_i |$, gdzie $s = (s_i)_{i=1}^{n_x}$ oznacza wektor wartości własnych macierzy monodromii $\Phi(x,u,\tau)$. Z kolei macierz monodromii jest to macierz fundamentalna $\Phi(t)$ w chwili $t = \tau$ zlinearyzowanego równania stanu

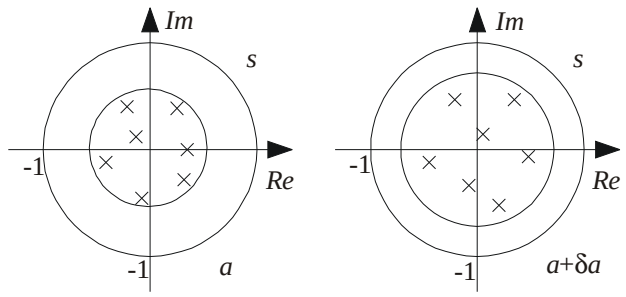
$$\delta\dot{x}(t) = A(t)\delta x(t), \quad t \in [0,\tau], \tag{9}$$

w którym $A(t) = f_x(x(t),u(t))$ jest macierzą Jacobiego okresowego procesu (x,u).

Ograniczenie stabilnościowe (7) wynika wprost z twierdzenia Floquet'a dotyczącego stabilności okresowego układu opisywanego równaniem różniczkowym zwyczajnym [35], [46], [78]. Zgodnie z tym twierdzeniem rozwiązanie okresowe jest lokalnie stabilne, gdy wartości modułów wszystkich charakterystycznych multiplikatorów (wartości własnych macierzy monodromii $\Phi(x,u,\tau)$) systemu (9) są mniejsze od jedności, natomiast jest niestabilne, gdy wartość przynajmniej jednego modułu charakterystycznego multiplikatora systemu (9) jest większa od jedności.

Z przytoczonego twierdzenia wynika, że przyjęcie $\alpha \ge 1$ oznacza poszukiwanie optymalnego rozwiązania, które leży na granicy stabilności lub jest niestabilne; natomiast przyjęcie $\alpha < 1$ jest równoznaczne z poszukiwaniem optymalnego, lokalnie stabilnego rozwiązania problemu (1)-(7) – z zapasem stabilności równym $1-\alpha$.

Przyjęcie F-poziomu $\alpha < 1$ powoduje również, że proces okresowy będzie – przy małych zaburzeniach parametrów równań stanu – lokalnie, asymptotycznie stabilny ze względu na ciągłą zależność – od tych parametrów – macierzy fundamentalnej procesu zlinearyzowanego oraz wartości własnych macierzy monodromii [46]. Oznacza to więc parametryczną stabilność zbioru multiplikatorów procesu okresowego (rysunek 1).



Rysunek 1 Mała zmiana δa parametru a procesu stabilnego nie zaburza jego stabilności

Co więcej, ograniczeniu stabilnościowemu z F-poziomem $\alpha < 1$ można nadać szczególną interpretację (patrz Twierdzenie l poniżej), mającą ważne konsekwencje dla projektowania układów regulacji procesu okresowego.

Niech $x(t, u(t), x(t_0))$ oznacza rozwiązanie równania stanu z warunkiem początkowym $x(t_0)$ i niech warunek $x(t_0)$ zapewnia okresowy przebieg trajektorii stanu przy okresowym sterowaniu $u(t)$. Ponadto, niech prawa strona równania stanu spełnia oszacowanie

$$\| r(\delta x(t), t) \| \leq \kappa \| \delta x(t) \|^{\mu},$$

gdzie $r(\delta x(t), t) = f(x(t) + \delta x(t), u(t)) - f(x(t), u(t)) - f_x(x(t), u(t))\delta x(t)$, zaś κ i μ są stałymi dodatnimi (oszacowanie takie jest charakterystyczne dla badanych w pracy procesów z modelami wielomianowymi względem zmiennych stanu).

Twierdzenie l *Jeśli F-poziom α zlinearyzowanego okresowego procesu sterowania $\delta \dot{x}(t) = f_x(x(t), u(t))\delta x(t)$ jest mniejszy od jedności, to istnieje takie $\eta > 0$, że dla zaburzeń stanu początkowego $\| \delta x(t_0) \| < \eta$, zachodzi:*

$$\| x(t, u(t), x(t_0) + \delta x(t_0)) - x(t, u(t), x(t_0)) \| \leq C\alpha^{t-t_0} \| \delta x(t_0) \|, \quad t \in [t_0, +\infty),$$

gdzie C jest stałą dodatnią. Oznacza to, że okresowa trajektoria stanu jest stabilna wykładniczo z podstawą równą F-poziomowi α.

Dowód. Niech $P(t) \in R^{n_x \times n_x}$ będzie macierzą okresową reprezentacji Floquet'a macierzy fundamentalnej $\Phi(t)$ zlinearyzowanego procesu okresowego (tj. $\Phi(t) = P(t)e^{\Lambda_s t}$), gdzie Λ_s jest macierzą stałą. Niech ponadto $\delta x(t) = x(t, u(t), x(t_0) + \delta x(t_0)) - x(t, u(t), x(t_0))$ będzie odchyleniem trajektorii zaburzonej od badanej trajektorii okresowej. Podstawienie $\delta x(t) = P(t)\delta y(t)$ pozwala uzyskać – na podstawie przekształcenia Lapunowa [35] – zależność

$$\delta y(t) = e^{\Lambda_s (t-t_0)}(\delta y(t_0) + \int_{t_0}^{t} P^{-1}(\theta) r(P(\theta)\delta y(\theta), \theta) d\theta),$$

która implikuje oszacowanie

$$\| \delta\, y(t) \| \le C_1 e^{-\lambda(t-t_0)} (\| \delta\, y(t_0) \| + C_2 \int_{t_0}^{t} e^{-\tilde{\lambda}\tilde{\mu}(t-t_o)} (e^{\lambda(\tilde{t}-t_0)} \| \delta\, y(\tilde{t}) \|)^{\mu}\, dt),$$

gdzie λ jest dowolną liczbą rzeczywistą spełniającą warunek

$$\max_{1 \le i \le n_x} \operatorname{Re} \lambda_i < -\lambda < 0,$$

zaś λ_i są wartościami własnymi macierzy Λ_s, stała C_1 wynika z oszacowania macierzy $e^{\Lambda_s (t-t_0)}$, stała C_2 jest określona jako $C_2 = \kappa \max_{t \in [t_0, t_0+\tau]} \| P^{-1}(t) \|$, a stałe $\tilde{\lambda}$ i $\tilde{\mu}$ spełniają warunki $\tilde{\lambda} < \lambda$ i $\tilde{\mu} < \mu$.

Z lematu Bihariego wynika więc, że

$$e^{\lambda(t-t_0)} \| \delta\, y(t) \| \le \frac{C_1 \| \delta\, y(t_0) \|}{(1-(\mu-1) C_1^{\mu-1} \frac{C_2}{\tilde{\lambda}\tilde{\mu}})^{1/(\mu-1)}}.$$

Biorąc pod uwagę nierówności $\| \delta\, x(t) \| \le \| P(t) \| \| \delta\, y(t) \| \le \max_{t \in [t_0, t_0+\tau]} \| P(t) \| \| \delta\, y(t) \|$ oraz $\| \delta\, y(t_0) \| \le \| P^{-1}(t_0) \| \| \delta\, x(t_0) \|$ i podstawiając $\alpha = e^{-\lambda}$ uzyskuje się ostatecznie tezę twierdzenia.

■

Powtórzenie powyższego rozumowania dla równania stanu zależnego od parametru a (tj. $\dot{x}(t) = f(x(t), u(t), a)$) z parametrem podlegającym fluktuacjom $B(a_0)$ (w otoczeniu jego wartości nominalnej a_0) pozwala stwierdzić jednostajną parametryczną stabilność wykładniczą procesu okresowego.

Własności stabilnościowe okresowego procesu sterowania mają istotny wpływ na efektywność jego realizacji. Wybór w trakcie optymalizacji $\alpha < 1$ oznacza bowiem poszukiwanie rozwiązań okresowych, które łatwo można zaimplementować w praktyce stosując proste i zarazem tanie układy automatycznej regulacji ze sprzężeniem zwrotnym. Natomiast przyjęcie wartości α powyżej jedności (lub całkowite pominięcie ograniczeń (7)) prowadzić może do tego, że – w trakcie optymalizacji – mogą być uzyskiwane lepsze rozwiązania okresowe, jednak będą one niestabilne – bądź na granicy stabilności, w związku z czym ich praktyczna realizacja będzie niemożliwa, lub też związana będzie z koniecznością poniesienia dużych nakładów na zaprojektowanie i realizację skomplikowanych, stabilizujących układów regulacji automatycznej.

Głównym celem optymalizacji procesów okresowych (1)-(7) jest uzyskanie globalnie optymalnego rozwiązania okresowego $(\hat{x}, \hat{u}, \hat{\tau})$, spełniającego ograniczenia (4)-(7), które jednocześnie zapewniałoby lepszą wartość wskaźnika jakości w porównaniu z globalnie optymalnym rozwiązaniem statycznym tego samego problemu (1)-(7).

W przypadku problemu okresowego (1)-(7) związany z nim problem globalnie optymalnego sterowania statycznego polega na minimalizacji wskaźnika jakości:

$$\overline{Q}(\overline{x}, \overline{u}) = \nu(q), \tag{10}$$

przy uwzględnieniu ograniczeń

$$q = h(\overline{x}, \overline{u}), \tag{11}$$

$$0 = f(\overline{x}, \overline{u}), \tag{12}$$

$$\overline{x} \in X, \quad \overline{u} \in U, \tag{13}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \quad \phi(q) = 0, \quad \tilde{\phi}(q) \le 0, \tag{14}$$

$$\| s(e^{f_x(\overline{x}, \overline{u})}) \|_\infty \le \overline{\alpha}, \tag{15}$$

gdzie $\overline{x} \in R^{n_x}$ jest statyczną trajektorią stanu, a $\overline{u} \in R^{n_u}$ jest statycznym sterowaniem.

W powyższym problemie globalnego sterowania statycznego wyjaśnienia wymaga ograniczenie (15). O tym, czy dane rozwiązanie jest stabilne, można dowiedzieć się analizując rzeczywistą część wartości własnych macierzy Jacobiego (gdy wszystkie wartości własne leżą w lewej półpłaszczyźnie zmiennych zespolonych rozwiązanie problemu (10)-(15) jest stabilne). Powyższe nie oznacza, że nierówność (15) jest błędna. Zapis $\| s(e^{f_x(\overline{x}, \overline{u})}) \|_\infty$ jest po prostu przekształceniem, które przeprowadza wszystkie wartości własne macierzy Jacobiego o niedodatnich wartościach części rzeczywistej do koła jednostkowego na płaszczyźnie zmiennej zespolonej. Tak więc, rozwiązanie problemu (10)-(15) jest stabilne, gdy prawdziwa jest nierówność $\| s(e^{f_x(\overline{x}, \overline{u})}) \|_\infty < 1$, natomiast gdy $\| s(e^{f_x(\overline{x}, \overline{u})}) \|_\infty \ge 1$ rozwiązanie jest na granicy stabilności, bądź jest ono niestabilne. Podsumowując, sposób badania stabilności problemu statycznego z wykorzystaniem nierówności (15) został wprowadzony w celu zunifikowania zapisu, dzięki czemu łatwiej będzie porównywać optymalne rozwiązanie zadania statycznego (10)-(15) z rozwiązaniem problemu okresowego (1)-(7).

1.3. Cel pracy doktorskiej

Problem optymalnego sterowania okresowego nie jest nowy i w ciągu ostatnich kilkudziesięciu lat był on przedmiotem rozważań wielu badaczy i tematem wielu prac: opracowano dla niego warunki konieczne i wystarczające optymalności [6], [17], [29], [31], [53], [68], [75], dokonano syntezy optymalnego sterowania w układzie ze sprzężeniem zwrotnym [17], [119], opracowano metody badania dominacji sterowania okresowego nad sterowaniem statycznym [9], [13], [14], [15], [18], [31], [106], [136], [137], [138], opracowano metody badania stabilności procesów okresowych [28], [46], [78], analizowano procesy okresowe z opóźnieniami [1], [29], [30], [31], [32], [33], [88], [93], [121], czy w końcu analizowano problemy sterowania okresowego procesów o parametrach rozłożonych [11], [12].

Jednakże, choć ukazało się wiele publikacji na temat problemu optymalnego sterowania okresowego, to wyznaczenie jego optymalnych rozwiązań stanowi nadal trudne zadanie z uwagi na nieliniowy charakter równań stanu opisujących dynamikę układu, dużą wrażliwość optymalizowanego procesu na zmiany wartości jego zmiennych procesowych, nieznany – optymalny okres procesu, czy nieznany – optymalny stan początkowy procesu [31]. Dodatkowe trudności z wyznaczeniem optymalnego (a zwłaszcza globalnie optymalnego) sterowania okresowego powodowane są często wieloekstremalnym charakterem problemu oraz koniecznością uwzględniania szeregu ograniczeń.

Nie oznacza to jednak, że nie ma metod, przy pomocy których nie można by było znaleźć optymalnego rozwiązania problemu okresowego. Wśród najbardziej znanych metod tego typu należy wymienić: gradientowe metody poprawy [68], metodę drugiej wariacji [13], [17], [31], [126], metody bazujące na zasadzie maksimum [44], [81], metody funkcji bazowych [20], [91], [129], [130] oraz metody wygładzania różniczkowego [134].

Wspólną cechą wymienionych metod jest jednak fakt, iż przy ich pomocy można jedynie wyznaczyć lokalnie optymalne rozwiązanie problemu okresowego, co – biorąc pod uwagę dzisiejszą tendencję w gospodarce do zmniejszania kosztów produkcyjnych przy jednoczesnym zwiększaniu wydajności przedsiębiorstw – jest niewątpliwie wadą tych metod. Pożądane byłyby bowiem metody, dzięki którym możliwe byłoby wyznaczenie globalnie optymalnego rozwiązania problemu. Niestety, z uwagi na wcześniej wymienione utrudnienia związane z problemami sterowania okresowego, wyznaczenie globalnie optymalnego rozwiązania okresowego jest wyjątkowo trudne. Z tego też pew-

nie powodu problematyka globalnej optymalizacji procesów okresowych jest słabo zbadana.

Kolejną wadą wspomnianych wcześniej metod jest fakt, że ich wykorzystanie jest utrudnione (lub wręcz niemożliwe) jeśli zachodzi potrzeba uwzględnienia różnorodnych ograniczeń charakterystycznych dla klasy rozważanych problemów. Prawdziwość tego twierdzenia można dowieść biorąc chociażby pod uwagę uśrednione ograniczenia zasobowe (5). Jednym ze sposobów ich uwzględniania w optymalizowanym problemie jest zastosowanie dodatkowych zmiennych stanu, przy pomocy których ograniczenia te mogą zostać dodane do równań stanu opisujących dany proces. Niestety, dla szerokiej klasy problemów taki sposób uwzględnienia ograniczeń prowadzi do osobliwej macierzy Jacobiego, przez co niemożliwym staje się zastosowanie metody Newtona-Raphsona – głównego narzędzia stosowanego do rozwiązywania okresowych równań stanu. Z kolei ograniczenia chwilowe stanu często powodują, iż redukcja problemu okresowego do przestrzeni sterowania z jednoczesną aproksymacją tych ograniczeń za pomocą dyskretnej siatki czasowej [97] prowadzi do dużej liczby uwikłanych, nieliniowych ograniczeń znacznie zmniejszających skuteczność metod programowania nieliniowego. Wykorzystanie warunków koniecznych optymalności okresowego procesu sterowania wynikających z zasady maksimum komplikuje fakt strukturalnej niestabilności okresowego układu hamiltonowskiego. Układ ten, zgodnie z twierdzeniem Lapunowa-Poincarego [35], nie może być asymptotycznie stabilny. Dlatego jego rozwiązanie za pomocą metody strzałów napotyka na trudności związane przede wszystkim z niestabilnością zmiennych sprzężonych [81].

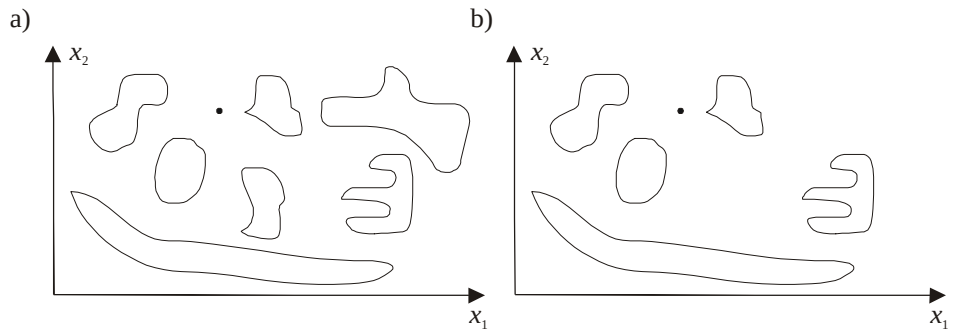
Oczywiście, analizując przedstawione problemy z uwzględnianiem ograniczeń można stwierdzić, że – co prawda nie ma idealnej metody optymalizacyjnej, którą można by było zastosować do każdego typu zadania – jednak ze znanych metod optymalizacyjnych zawsze można wybrać tę, która pozwoli na uzyskanie zadawalającego rozwiązania danego problemu. Stwierdzenie to jest jednak słuszne jedynie w przypadku problemów okresowych, w których występują niezbyt skomplikowane ograniczenia zasobowe lub technologiczne zarówno chwilowe jak i uśrednione. Natomiast, w przypadku, gdy w problemie występują ograniczenia stabilnościowe, dotychczas znane metody optymalizacyjne dla problemów okresowych stają się bezużyteczne, ponieważ „nie potrafią” one – w trakcie poszukiwania rozwiązania optymalnego – uwzględnić ograniczeń o nieanalitycznym i nieróżniczkowalnym charakterze.

Tak więc ułatwienie praktycznej realizacji i rozszerzenie zakresu zastosowań sterowania okresowego związane jest ze zbadaniem nowych aspektów problematyki optymalnego sterowania cyklicznego, którymi są:

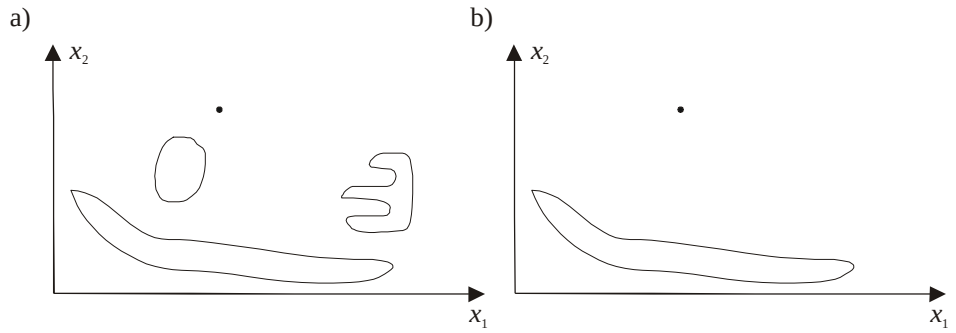
- *wyznaczanie globalnie optymalnego okresowego procesu sterowania* zapewniającego znaczny efekt optymalizacyjny w postaci dużej poprawy wskaźnika jakości procesu w porównaniu z jego wartością dla optymalnego statycznego rozwiązania,
- *dokładne uwzględnianie ograniczeń zasobowo-technologicznych* obejmujących chwilowe i uśrednione ograniczenia stanu i sterowania oraz ograniczenia stabilnościowe warunkujące praktyczną implementowalność procesu okresowego.

Prezentowane w literaturze wyniki badań szeregu problemów optymalnego sterowania okresowego [40], [68], [92], [93] wskazują na istnienie, dla danego problemu, wielu cyklicznych rozwiązań dominujących – co do wartości wskaźnika jakości – nad optymalnym procesem statycznym. Najlepsze spośród tych rozwiązań stanowią lokalne ekstrema problemu (rysunek 2a). Narzucenie jednak klasycznych ograniczeń zasobowo-technologicznych ogranicza liczbę cykli dominujących (rysunek 2b). Jeszcze większe zawężenie klasy cykli dominujących nad optymalnym sterowaniem statycznym powoduje wprowadzenie dodatkowego ograniczenia stabilnościowego (rysunek 3a).

Zastosowanie przedstawionego w tej rozprawie algorytmu hybrydowego (algorytmu ewolucyjnego połączonego z algorytmem kierunków poprawy) pozwala wydzielić cykl globalnie optymalny spośród cykli uwzględniających dodatkowe ograniczenia zasobowo-technologiczne oraz ograniczenie stabilnościowe (rysunek 3b).



Rysunek 2 Cykle lokalnie optymalne i optymalne rozwiązanie statyczne (rys. a); cykle lokalnie optymalne i optymalne rozwiązanie statyczne po uwzględnieniu ograniczeń zasobowo-technologicznych (rys. b)



Rysunek 3 Cykle lokalnie optymalne i optymalne rozwiązanie statyczne po uwzględnieniu ograniczenia stabilnościowego (rys. a); cykl globalnie optymalny spełniający różnorodne ograniczenia zasobowo-technologiczne oraz ograniczenie stabilnościowe (rys. b)

Z uwagi na powyższe, celem niniejszej pracy jest przedstawienie wyników prowadzonych przez autora tego opracowania badań nad możliwością wykorzystania algorytmów hybrydowych (łączących ewolucyjne podejście do optymalizacji [107], [108], [111], [113], [114] z szybkimi algorytmami kierunków poprawy [109], [112], [115], [116], [117], [118], [120]) do wyznaczania globalnie optymalnych rozwiązań problemów okresowych. Istotnym przy tym jest, że problemy optymalnego sterowania okresowego ujmowane są tu w nowy, uogólniony sposób (1)-(7), uwzględniający ograniczenia zasobowe (chwilowe i całkowe ograniczenia narzucone na sterowanie), ograniczenia technologiczne (chwilowe i całkowe ograniczenia narzucone na stan) oraz ograniczenia stabilnościowe (w postaci tzw. poziomu Floquet'a).

Wybór algorytmów ewolucyjnych, dla celów optymalizacji procesów cyklicznych, jest szczególnie uzasadniony takimi ich właściwościami, które z jednej strony nawiązują do cyklicznych rozwiązań stosowanych przez naturalną ewolucję organizmów żywych, a z drugiej strony – pozwalają włączać do procesu optymalizacji różnorodne ograniczenia – w tym ograniczenia nieanalityczne, nie dające się wkomponować do znanych metod optymalizacji lokalnej – przy jednoczesnym poszukiwaniu rozwiązania globalnie optymalnego. Z kolei zastosowanie algorytmów kierunków poprawy podyktowane jest chęcią zwiększenia szybkości wyznaczania rozwiązania globalnie optymalnego. W przypadku bowiem problemów okresowych w celu wyznaczenia jednego rozwiązania okresowego optymalizowanego procesu (rozwiązania spełniającego ograniczenie $x(0) = x(\tau)$) należy wielokrotnie rozwiązać równania stanu (3) – które są z reguły nieliniowe – oraz dla każdego rozwiązania równań stanu należy znaleźć odwrotność macierzy $I - \Phi(x,u,\tau)$ (szczegóły w punkcie 3.2.2). Choć wyznaczenie rozwiązania okresowego równań stanu (dzięki wszechobecnym komputerom) nie należy już do zadań niemożliwych do wykonania, jest ono jednak bardzo czasochłonne. Tak więc, algo-

rytmy ewolucyjne, które muszą w jednym kroku swojego działania wyznaczyć wiele rozwiązań okresowych muszą poświęcić bardzo dużo czasu na wyznaczenie globalnie optymalnego sterowania okresowego. Dlatego też uzasadnionym jest stosowanie algorytmu ewolucyjnego tylko do wyznaczania przybliżonego rozwiązania globalnie optymalnego (lub obszaru, w którym to rozwiązanie się znajduje) oraz szybkiego algorytmu kierunków poprawy (algorytmu lokalnej optymalizacji), który startując z rozwiązania dostarczonego przez algorytm ewolucyjny, w krótkim czasie znajdzie rozwiązanie globalnie optymalne.

1.4. Zestawienie najważniejszych rezultatów pracy

Wśród najważniejszych rezultatów pracy doktorskiej można wymienić:

1) zdefiniowanie nowego, uogólnionego sposobu uwzględniania ograniczeń zasobowych, technologicznych oraz stabilnościowych w problemach optymalnego sterowania okresowego,
2) sformułowanie i udowodnienie twierdzenia na temat jednostajnej parametrycznej stabilności wykładniczej procesu okresowego,
3) opracowanie algorytmu ewolucyjnego, przy pomocy którego można znaleźć globalnie optymalne rozwiązanie problemu okresowego z ograniczeniami chwilowymi narzuconymi na sterowanie i okres,
4) opracowanie sposobu uwzględniania ograniczeń całkowych w algorytmach ewolucyjnych – opracowanie algorytmu rekonstrukcji ograniczeń,
5) opracowanie algorytmu ewolucyjnego, przy pomocy którego można wyznaczyć globalnie optymalne rozwiązanie problemu okresowego z ograniczeniami chwilowymi i uśrednionymi narzuconymi na sterowanie oraz z ograniczeniami chwilowymi narzuconymi na okres,
6) opracowanie metody uwzględniania różnorodnych ograniczeń technologicznych i ograniczeń stabilnościowych w algorytmach ewolucyjnych,
7) opracowanie algorytmu ewolucyjnego, przy pomocy którego można wyznaczyć globalnie optymalne sterowanie okresowe procesu z ograniczeniami zasobowymi i technologicznymi (zarówno chwilowymi jak i uśrednionymi) oraz – z bardzo trudnymi do uwzględnienia – ograniczeniami stabilnościowymi,
8) opracowanie nowej metody uwzględniania ograniczeń zasobowych w metodzie rzutowanego gradientu,

9) opracowanie metody λ-rzutowanego gradientu, przy pomocy której można znaleźć lokalnie optymalne rozwiązanie problemu okresowego z ograniczeniami chwilowymi i uśrednionymi narzuconymi na sterowanie,
10) opracowanie algorytmu łączącego metodę przesuwanej funkcji kary i metodę λ-rzutowanego gradientu do rozwiązywania (poszukiwania lokalnego optimum) problemów okresowych z różnymi ograniczeniami zasobowymi i technologicznymi,
11) sformułowanie i udowodnienie twierdzenia dotyczącego zbieżności algorytmu łączącego metodę przesuwanej funkcji kary i metodę λ-rzutowanego gradientu,
12) zaproponowanie hybrydowego algorytmu globalnej optymalizacji łączącego ewolucyjne podejście do optymalizacji z algorytmami kierunków poprawy,
13) zestawienie wyników badań potwierdzających skuteczność wyżej wspomnianych algorytmów optymalizacji.

1.5. Zawartość pracy

W rozdziale 1, tzn. we wstępie niniejszej pracy, przedstawione zostały korzyści, jakie niesie ze sobą stosowanie sterowania okresowego – podano tu dziedziny gospodarki, w których zastosowanie sterowania okresowego wydatnie zwiększyło wydajność różnorodnych procesów produkcyjnych w porównaniu z wydajnością tych procesów osiąganą przy optymalnym sterowaniu statycznym (punkt 1.1). We wstępie zamieszczono również nowe, uogólnione sformułowanie globalnie optymalnego problemu okresowego z ograniczeniami zasobowymi, technologicznymi i stabilnościowymi (punkt 1.2) oraz szczegółowo opisano te ograniczenia. W rozdziale 1 można również znaleźć twierdzenie dotyczące parametrycznej stabilności procesów okresowych, a także sformułowanie globalnie optymalnego problemu statycznego „zanurzonego” w problemie okresowym. Ponadto, wstęp zawiera przegląd literatury z dziedziny optymalnego sterowania okresowego prezentujący pokrótce dotychczasowe osiągnięcia w tym obszarze nauki (punkt 1.3). W końcu w punktach 1.3 i 1.4 przedstawiono cel pracy doktorskiej oraz uwypuklono najważniejsze rezultaty badań prowadzonych nad globalną optymalizacją procesów okresowych.

Rozdział 2 zawiera podstawowe informacje niezbędne do konstruowania algorytmów ewolucyjnych dla problemów optymalnego sterowania okresowego. Rozdział ten rozpoczyna się od wstępu, w którym ogólnie zarysowana została budowa algorytmów ewolucyjnych oraz ich związek z Darwinowską teorią ewolucji. Następnie, wyjaśniane

są podstawowe pojęcia stosowane w teorii algorytmów ewolucyjnych (punkt 2.2). Kolejne punkty rozdziału 2 opisują poszczególne „części” algorytmów ewolucyjnych. W punkcie 2.3 – zatem – opisane zostały sposoby kodowania populacji, które mogą być użyte w przypadku optymalizacji procesów okresowych. Kolejny punkt „Zarządzanie populacją” zawiera szczegółowe informacje o działaniu algorytmów ewolucyjnych (przedstawia on dokładny schemat działania algorytmu wraz z opisem jego poszczególnych etapów pracy; z punktu tego można dowiedzieć się: jak powstaje populacja początkowa; w jaki sposób decyduje się o tym, które osobniki przeżywają, a które umierają; kiedy i na jakich osobnikach należy stosować operatory krzyżowania i mutacji oraz jak tworzone są nowe pokolenia; w punkcie tym opisane są również najczęściej stosowane metody reprodukcji). Punkt 2.5 koncentruje się na opisie podstawowych operatorów krzyżowania, które mogą być zastosowane w algorytmach ewolucyjnych dla problemów okresowych; można w nim znaleźć opis różnych wersji operatorów wymieniających, operatora równomiernego czy operatora arytmetycznego. Z kolei punkt 2.6 poświęcony jest różnorodnym operatorom mutacji – są w nim przedstawione operatory mutacji równomiernej, mutacji nierównomiernej oraz mutacji brzegowej.

Każdy algorytm optymalizacyjny musi się kiedyś zatrzymać. O tym, kiedy należy zatrzymać algorytm ewolucyjny można się dowiedzieć czytając punkt 2.7 rozdziału 2; punk ten zawiera opis sześciu podstawowych kryteriów stopu, które mogą być użyte w trakcie optymalizacji procesów okresowych.

W rozdziale 3 przedstawione zostały algorytmy ewolucyjne, przy pomocy których można rozwiązać różnorodne problemy okresowe. Rozdział ten rozpoczyna punkt zatytułowany Kodowanie problemu okresowego, w którym opisany jest sposób zapisu rozwiązania problemu okresowego w postaci osobnika, na którym można stosować mechanizmy ewolucyjne zamieszczone w rozdziale 2. Punkt 3.2 rozdziału 3 poświęcony jest problemowi globalnie optymalnego sterowania procesem okresowym z ograniczeniami chwilowymi narzuconymi na sterowanie. W omawianej części pracy najpierw sformułowany został sam problem okresowy (punkt 3.2.1), a następnie – po opisaniu sposobu uwzględniania ograniczeń chwilowych narzuconych na sterowanie oraz sposobu uwzględniania okresowości procesu –dla problemu tego skonstruowany został algorytm ewolucyjny (punkt 3.2.2). Punkt 3.2 kończy szereg przykładów z dziedziny chemicznych procesów produkcyjnych, na których przetestowany został algorytm ewolucyjny (punkt 3.2.3). Kolejnym podpunkt rozdziału 3 poświęcony został problemowi globalnie optymalnego sterowania procesami okresowymi z

ograniczeniami chwilowymi i uśrednionymi narzuconymi na sterowanie. Analogicznie jak w punkcie 3.2, punkt 3.3 rozpoczyna się od sformułowania problemu okresowego, dla którego będzie konstruowany algorytm ewolucyjny (punkt 3.3.1). Następnie przedstawiany jest algorytm rekonstrukcji, którego zadaniem jest uwzględnianie ograniczeń uśrednionych narzuconych na sterowanie w trakcie działania algorytmu ewolucyjnego (punkt 3.3.2). Podpunkt 3.3.2 kończy opis algorytmu ewolucyjnego, przy pomocy którego można wyznaczyć globalnie optymalne sterowanie problemu okresowego z ograniczeniami zasobowymi zarówno chwilowymi jak i uśrednionymi. Skuteczność opisywanego w tej części pracy algorytmu ewolucyjnego potwierdzają liczne przykłady, które zostały zamieszczone w podpunkcie 3.3.3. W ostatniej części rozdziału 3 (tzn. w punkcie 3.4) zaprezentowany został algorytm ewolucyjny, przy pomocy którego można wyznaczyć rozwiązanie globalnie optymalne problemu okresowego zawierającego ograniczenia zasobowe i technologiczne zarówno (chwilowe i uśrednione) oraz (bardzo trudne do uwzględnienia) ograniczenia stabilnościowe. Punkt 3.4 rozpoczyna podpunkt 3.4.1, w którym formułowany jest problem globalnie optymalnego sterowania okresowego zawierający wspomniane ograniczenia. Kolejny podpunkt poświęcony jest sposobowi uwzględniania ograniczeń technologicznych i stabilnościowych przy użyciu funkcji kary (punkt 3.4.3). W tym miejscu przedstawiony został również algorytm ewolucyjny dla problemu okresowego sformułowanego w 3.4.1. Rozdział 3 kończą przykłady, na których został przetestowany algorytm ewolucyjny, przy pomocy którego można wyznaczyć globalnie optymalne sterowanie okresowe problemów z ograniczeniami zasobowymi, technologicznymi i stabilnościowymi.

Algorytmy ewolucyjne opisane w rozdziale 3.4.1 są skutecznym narzędziem optymalizacji. Są one jednak bardzo czasochłonne, dlatego też w trakcie globalnej optymalizacji procesów okresowych dobrze jest stosować algorytmy hybrydowe, będące połączeniem (opisanych w rozdziale 3) algorytmów ewolucyjnych z szybkimi algorytmami optymalizacji lokalnej. Rozdział 4 poświęcony jest więc nowym algorytmom kierunków poprawy, przy pomocy których można przyspieszyć proces poszukiwania globalnie optymalnego rozwiązania okresowego. Pierwszy punkt rozdziału 4 zawiera opis metody λ-rzutowanego gradientu, która to metoda może być zastosowana dla problemów okresowych zawierających ograniczenia zasobowe zarówno chwilowe jak i uśrednione. Drugi punkt omawianego rozdziału opisuje natomiast hybrydowy algorytm będący połączeniem metody λ-rzutowanego gradientu oraz metody przesuwanej funkcji kary (algorytm ten umożliwia znalezienie lokalnie optymalnego rozwiązania problemu

okresowego zawierającego ograniczenia chwilowe i uśrednione zarówno narzucone na zmienne sterujące jak i na stan). W tej części pracy można również znaleźć twierdzenie o zbieżności metody przesuwanej kary do optymalnego rozwiązania okresowego. Rozdział 4 kończą przykłady z dziedziny chemicznych procesów produkcyjnych, na których wypróbowane zostały wspomniane algorytmy kierunków poprawy (algorytmy optymalizacji lokalnej).

Niniejszą pracę kończy rozdział 5, w którym przedstawione zostały wnioski z przeprowadzonych badań. W rozdziale tym wskazane są również dalsze kierunki badań nad usprawnianiem algorytmów przedstawionych w niniejszej pracy.

2. ALGORYTMY EWOLUCYJNE I STEROWANIE

Adaptacja algorytmów ewolucyjnych do rozwiązywania problemów optymalnego sterowania okresowego wiąże się z koniecznością wyjaśnienia podstawowych terminów stosowanych w teorii ewolucji i algorytmów ewolucyjnych, a obcych teorii problemów okresowych.

2.1. Wstęp do algorytmów ewolucyjnych

Być może trudno w to uwierzyć, jednak swoje istnienie algorytmy ewolucyjne zawdzięczają Karolowi Darwinowi. Co prawda Darwin był biologiem i nigdy nie zajmował się optymalizacją, a co za tym idzie także algorytmami ewolucyjnymi, jednak formułując w 1895 roku teorię ewolucji oraz obwieszczając ją światu w swoim dziele „*O powstaniu gatunków na drodze doboru naturalnego*” zapoczątkował on rozwój także i tej dziedziny nauki.

Darwin w swojej pracy stwierdził, że wszystkie organizmy żywe mają wspólnego przodka, zaś różnorodność, złożoność i organizacja świata żywego, jaką obserwujemy obecnie, jest wynikiem ewolucji, której głównym mechanizmem jest dobór naturalny.

Choć początkowo wielu naukowców odnosiło się sceptycznie do Darwinowskiej teorii ewolucji, z biegiem lat zyskiwała ona coraz więcej zwolenników, a dziś (po niewielkich zmianach) większość uczonych przyjmuje ją za pewnik.

Z czasem teoria ewolucji zaczęła interesować nie tylko biologów lecz także naukowców z innych dziedzin nauki. Zauważyli oni bowiem, że naturalne procesy zachodzące w przyrodzie przez miliony lat można, dzięki symulacjom komputerowym, wykorzystać do rozwiązywania różnorodnych problemów. Powszechnie uznawanymi pionierami, którzy przyczynili się do rozwoju algorytmów ewolucyjnych, są Lawrence Fogel [50] ze Stanów Zjednoczonych oraz Ingo Rechenberg i Hans-Paul Schwefel z Niemiec [98]. Fogel starał się wykorzystać populację automatów skończonych do rozpoznawania języków. Poddawał on tę populację ciągłym, losowym perturbacjom i procesowi selekcji, dzięki czemu z czasem uzyskał automaty, które nieznane im wcześniej wyrażenia języka rozpoznawały praktycznie bezbłędnie. Natomiast niemieccy uczeni w swoich badaniach nad optymalizacją urządzeń mechanicznych wykorzystywali permutacje rozwiązań początkowych, dzięki czemu przyczynili się oni do sformułowania strategii ewolucyjnej. Jednakże pierwszą osobą, która stworzyła „prawdziwy” algorytm genetyczny,

był John Holland [66], [67], który zajmował się na Uniwersytecie Michigan badaniem procesów adaptacyjnych w odbiornikach sygnałów binarnych. Holland jako pierwszy zastosował przekształcenia na wzór dzisiejszych operatorów krzyżowania, mutacji oraz selekcji, symulujące rzeczywiste procesy ewolucji.

Algorytmy ewolucyjne są to algorytmy, które przeszukują wielokierunkowo, w sposób losowy, przestrzeń dopuszczalnych rozwiązań danego problemu w celu znalezienia jego najlepszego rozwiązania, wykorzystując przy tym mechanizmy znane z Darwinowskiej teorii ewolucji.

Działanie algorytmu ewolucyjnego rozpoczyna się od losowego wygenerowania zbioru rozwiązań dopuszczalnych danego problemu, tzn. od wygenerowania zbioru osobników tworzących populację. Następnie populacja ta, na wzór procesów zachodzących w przyrodzie, poddawana jest nieustannemu działaniu selekcji oraz operatorów genetycznych: krzyżowania i mutacji, dzięki czemu uzyskuje się (w kolejnych krokach) coraz to lepsze rozwiązania danego problemu.

Zadaniem selekcji i operatorów genetycznych, które modelują proces rozmnażania się żywych organizmów, jest realizacja podstawowych założeń teorii ewolucji, tj.:

1) ograniczoności zasobów – osobniki w populacji muszą między sobą konkurować o te same zasoby środowiska, w którym żyją;
2) dostosowania – osobniki tej samej populacji różnią się między sobą; niektóre z nich posiadają cechy, które w danych warunkach środowiska ułatwiają im „walkę” z innymi osobnikami tej samej populacji o zasoby;
3) zróżnicowania przeżywalności – przeżywalność osobników zależy od ich dostosowania do warunków środowiska, w którym żyją – osobniki lepiej dostosowane mają większe szanse przeżycia i wydania potomstwa niż osobniki dostosowane gorzej;
4) dziedziczności – nowe osobniki dziedziczą cechy swoich rodziców;
5) losowej zmienności – dziedziczenie cech rodziców nie jest idealne; u potomstwa mogą w sposób losowy pojawiać się cechy, które nie występowały u rodziców.

Selekcja w algorytmach ewolucyjnych realizuje pierwsze trzy spośród wymienionych wyżej założeń teorii ewolucji. Odpowiada ona procesowi doboru naturalnego, który sprawia, że osobniki najlepiej dostosowane do istniejących warunków środowiska mają największe szanse na przetrwanie. Zadaniem selekcji jest ocena osobników w danej populacji i wskazanie tych osobników, które przeżywają (przechodzą do nowego

pokolenia); przy czym ocena osobników dokonywana jest za pomocą specjalnej funkcji określającej ich stopień dopasowania do środowiska. W przypadku zadań optymalizacyjnych, funkcją określającą stopień dopasowania jest najczęściej funkcja wskaźnika jakości, której wartość podlega optymalizacji. Od stopnia dopasowania danego osobnika (od wartości wskaźnika jakości) zależy prawdopodobieństwo jego przeżycia. Im osobnik (na tle innych osobników tej samej populacji) jest lepiej dopasowany, tzn. jego wartość funkcji celu jest większa w przypadku maksymalizacji (lub mniejsza w przypadku minimalizacji), tym większe ma on szanse na przeżycie i znalezienie się w nowym pokoleniu. Należy tutaj zaznaczyć, że osobniki najgorzej przystosowane nie są pozbawione szans na przetrwanie całkowicie. Tak – jak to bywa w życiu, o tym – czy dany osobnik przeżyje, decyduje los, który jest jednak łaskawszy dla tych osobników, które mają lepszą wartość funkcji dopasowania. Najczęściej stosowane metody selekcji zostały opisane w rozdziale 2.4 niniejszej pracy.

Założenie dziedziczności Darwinowskiej teorii ewolucji realizowane jest przez tzw. operator krzyżowania, który stosowany jest po selekcji – na osobnikach wytypowanych do nowej populacji. Swoim działaniem operator krzyżowania przypomina rozmnażanie żywych organizmów. Przy jego pomocy z losowo wybranych osobników (rodziców), generowane są nowe osobniki (rodzą się dzieci), które charakteryzują się tym, że posiadają cechy, które posiadali ich rodzice. Szczegóły dotyczące działania operatora krzyżowania można znaleźć w rozdziale 2.5.

Ostatnie założenie teorii ewolucji – losowa zmienność – realizowane jest za pomocą operatora mutacji. Pomimo tego, że w znikomym stopniu oddziaływuje on na osobniki w populacji (szczegóły w rozdziale 2.6), można stwierdzić, że jest on bodajże najważniejszym operatorem. Nadaje on bowiem – losowo wybranym z populacji – osobnikom całkowicie nowe cechy, cechy – które nie występowały u ich przodków. Dzięki operatorowi mutacji można zatem uzyskać osobniki, które będą charakteryzowały się o wiele lepszym stopniem przystosowania niż pozostałe osobniki.

2.2. Podstawowe pojęcia stosowane w algorytmach ewolucyjnych

Wśród podstawowych określeń stosowanych w teorii algorytmów ewolucyjnych (niektóre z tych pojęć zostały już wcześniej użyte) należy wymienić [4], [83]:

- osobnik – jest to propozycja postaci rozwiązania danego problemu (punkt przestrzeni poszukiwań), która podlega procesowi ewolucji; osobnik może występować w postaci fenotypu lub genotypu;

- populacja – zbiór osobników (o określonej liczebności) konkurujących między sobą o zasoby środowiska, w którym występują (środowisko dla problemów optymalizacyjnych definiowane jest przez wskaźnik jakości oraz narzucone ograniczenia);
- genotyp – jest to zbiór chromosomów danego osobnika; w przypadku algorytmów ewolucyjnych najczęściej genotyp stanowi jeden chromosom;
- chromosom – łańcuch genów;
- gen – pojedynczy element chromosomu;
- allel – wartość genu (zależy od sposobu kodowania problemu);
- locus – pozycja genu w chromosomie;
- fenotyp – zestaw cech podlegających ocenie środowiska zależny od genotypu; jest to osobnik w formie niezakodowanej;
- funkcja przystosowania – jest to funkcja, która na podstawie fenotypu danego osobnika określa jego przystosowanie do środowiska, w którym on działa; najczęściej jest to funkcja celu rozwiązywanego problemu (ewentualnie nieznacznie zmodyfikowana z uwagi na sposób kodowania osobnika);
- nacisk selektywny – tendencja algorytmu ewolucyjnego do poprawiania średniej wartości przystosowania osobników w populacji; przy pomocy nacisku selektywnego steruje się przeżywalnością osobników – większy nacisk selektywny oznacza, że osobniki lepiej przystosowane mają większe szanse przeżycia niż osobniki gorsze.

Jak widać z zamieszczonej powyżej listy, terminy używane w teorii algorytmów ewolucyjnych są analogiczne do określeń stosowanych w biologii.

2.3. Kodowanie populacji

Algorytmy ewolucyjne są skutecznym narzędziem rozwiązywania różnorodnych problemów pod warunkiem, że rozwiązywany problem zostanie prawidłowo zakodowany, czego elementem m.in. jest właściwa postać chromosomów, na których można stosować operatory genetyczne. Zastosowany sposób kodowania jest jednym z głównych czynników wpływających na jakość działania danego algorytmu ewolucyjnego. Od niego między innymi zależy czas poszukiwania rozwiązania danego problemu oraz to, czy znalezione rozwiązanie jest rozwiązaniem globalnie optymalnym i czy spełnia ono narzucone na nie ograniczenia. Jak już wcześniej stwierdzono, istotą działania algo-

rytmów ewolucyjnych jest losowe przeszukiwanie przestrzeni dopuszczalnych rozwiązań. W przypadku, gdy problem zostanie źle zakodowany, może zdarzyć się sytuacja, w której jakiś fragment przestrzeni rozwiązań nie zostanie przeszukany (przez co nie zostanie znalezione najlepsze rozwiązanie) lub też przeszukiwany zacznie być obszar, w którym znajdują się rozwiązania niedopuszczalne.

Wybór sposobu kodowania jest zadaniem trudnym i zależy on przede wszystkim od rozwiązywanego problemu. Niestety, nie ma gotowej metody wskazującej jednoznacznie sposób kodowania, który należy użyć w przypadku rozwiązywania konkretnego problemu. Wybierając sposób kodowania można się jednak kierować następującymi zasadami [4], [58], [83]:

- musi istnieć funkcja jednoznacznie dekodująca, która zdekoduje dowolny genotyp do postaci fenotypu;
- każde dopuszczalne rozwiązanie zadania można przedstawić jako genotyp;
- kodowanie nie powinno wprowadzać nowych maksimów (lub minimów w przypadku minimalizacji) poza tymi, które wynikają z natury problemu (z jego funkcji celu);
- należy używać minimalnego alfabetu (tzn. należy używać możliwie najprostszego kodowania, które zapewnia możliwie największe podobieństwo zakodowanych osobników do rzeczywistych rozwiązań).

W przypadku rozwiązywania zadań numerycznych najczęściej stosowane jest kodowanie binarne lub rzeczywiste.

2.3.1. Kodowanie binarne

Kodowanie binarne, polegające na zapisie liczb dziesiętnych w systemie dwójkowym, jest kodowaniem, które jako pierwsze było stosowane w algorytmach ewolucyjnych. Główną cechą tego rodzaju kodowania jest to, że dosyć łatwo można je zaimplementować przy użyciu komputera oraz fakt, że istnieją już – dobrze zdefiniowane – operatory krzyżowania i mutacji dla tego kodowania. Ponadto kodowanie binarne jest jedynym kodowaniem, dla którego udało się osiągnąć wyniki teoretyczne dotyczące zbieżności algorytmów ewolucyjnych (teoria schematów [4], [58], [83]).

Kodowanie binarne przebiega w sposób opisany poniżej [103].

Załóżmy, że mamy zadanie optymalizacyjne, którego rozwiązaniem jest wektor liczb rzeczywistych

$$x = [x_1, x_2, x_3, ..., x_n], \tag{16}$$

podlegający ograniczeniu

$$x_i \in [x_i^-, x_i^+] \;\; (i = 1, ..., n), \tag{17}$$

gdzie $x_i^{\pm} \in R$, a n jest rozmiarem zadania.

Osobnik w formie zakodowanej (genotyp) dla tego przykładu reprezentowany jest przez grupę n chromosomów – n ciągów bitowych (zerojedynkowych) o długościach m_i (i=1,...n):

$$\begin{aligned} z_1 &= [z_{1,1}, z_{1,2}, ..., z_{1,m_1}], \\ z_2 &= [z_{2,1}, z_{2,2}, ..., z_{2,m_2}], \\ &\vdots \\ z_n &= [z_{n,1}, z_{n,2}, ..., z_{n,m_n}]. \end{aligned} \tag{18}$$

Długości m_i (i=1,...,n) poszczególnych chromosomów z_i (i=1,...,n) zależą od przyjętej dokładności zapisu zmiennych x_i oraz od wartości ograniczeń (17) i wyznacza się je z zależności:

$$10^{d_i}(x_i^+ - x_i^-) \le 2^{m_i} - 1 \;\; (i = 1, ..., n), \tag{19}$$

gdzie: d_i jest założoną dokładnością reprezentacji zmiennej x_i (jest to dokładność do d_i znaczącego miejsca po przecinku), natomiast m_i jest szukaną długością – najmniejszą liczbą naturalną, dla której spełniona jest nierówność (19).

Przy kodowaniu binarnym fenotyp (wartość zmiennej x_i) wyznaczany jest z genotypu według zależności:

$$x_i = x_i^- + \frac{x_i^+ - x_i^-}{2^{m_i} - 1} \sum_{j=0}^{m_i - 1} z_{ij} 2^j, \tag{20}$$

gdzie z_{ij} jest to j-ty bit i-go ciągu bitowego (i-go chromosomu).

W praktyce, przy kodowaniu binarnym – zamiast n chromosomów – stosowany jest często jeden chromosom:

$$z = [z_{1,1}, ..., z_{1,m_1}, z_{2,1}, ..., z_{2,m_2}, ..., z_{n,m_n}, ..., z_{n,m_n}], \tag{21}$$

będący połączeniem n ciągów bitowych kodujących wektor x.

2.3.2. Kodowanie rzeczywiste

Opisane w poprzednim punkcie kodowanie binarne ma dwie podstawowe wady. Po pierwsze – zbyt długie chromosomy (ciągi bitowe są tym dłuższe im dokładniejsze ma być rozwiązanie). Niestety wydłużenie ciągów bitowych zdecydowanie wydłuża czas obliczeń, co jest szczególnie widoczne i uciążliwe w przypadku problemów wielowymiarowych.

Drugą wadą kodowania binarnego, wynikającą poniekąd z długości ciągów bitowych użytych do kodowania, jest to, że w przypadku tego rodzaju kodowania może występować problem tzw. krawędzi Hamminga [4]. Problem ten związany jest z inną strukturą sąsiedztw punktów w przestrzeni genotypów i fenotypów, która może prowadzić do powstawania dodatkowych minimów/maksimów lokalnych, utrudniających wyznaczenie globalnego optimum danego problemu.

Z uwagi na powyższe, w przypadku rozwiązywania różnorodnych problemów numerycznych, często – zamiast kodowania binarnego – stosowane jest kodowanie rzeczywiste [3], [56], [83], w którym genotyp osobnika reprezentowany jest najczęściej przez jeden chromosom będący wektorem liczb rzeczywistych o tej samej długość co wektor, który jest rozwiązaniem danego zadania. Jeżeli na rozwiązywany problem nie ma nałożonych specyficznych ograniczeń, wówczas – w przypadku kodowania rzeczywistego – fenotyp i genotyp są tymi samymi wektorami liczb rzeczywistych (nie ma więc konieczności stosowania funkcji dekodującej w trakcie działania algorytmu ewolucyjnego).

Kodowanie rzeczywiste wydaje się być lepsze od kodowania binarnego przynajmniej pod kilkoma względami [83]:

- kodowanie rzeczywiste jest bardziej naturalne w przypadku większości rozwiązywanych problemów;
- przy kodowaniu rzeczywistym można osiągnąć bardziej dokładne wyniki niż przy kodowaniu binarnym;
- zastosowanie kodowania rzeczywistego pozwala objąć duże i/lub nieznane dziedziny poszukiwań;
- przy kodowaniu rzeczywistym łatwiej jest uwzględnić nietrywialne ograniczenia niż przy kodowaniu binarnym.

2.4. Zarządzanie populacją

Skuteczność algorytmu ewolucyjnego zależy w głównej mierze od sposobu zarządzania populacją – tzn. od sposobu podejmowania m.in. następujących decyzji:

- które osobniki w bieżącej populacji przeżyją;
- które osobniki umrą;
- ile pokoleń mogą żyć poszczególne osobniki;
- czy dany osobnik może wydawać potomstwo.

W rozdziale 2.1 wspomniano już, iż za podejmowanie tego rodzaju decyzji w algorytmach ewolucyjnych odpowiada proces nazywany selekcją. Proces ten można podzielić na reprodukcję i sukcesję [3].

Aby lepiej zrozumieć istotę zarówno całego procesu selekcji jak i jego części należy nieco dokładniej zapoznać się ze schematem działania algorytmów ewolucyjnych. Schemat przedstawiony w rozdziale 2.1 – jako że jego zadaniem było jedynie wstępne przedstawienie koncepcji działania omawianych algorytmów – jest bowiem dla dalszych rozważań zbyt ogólny. W rzeczywistości działanie algorytmu ewolucyjnego można podzielić na sześć faz: inicjacja populacji bazowej $P^{g=0}$, ocena osobników w populacji, reprodukcja, działanie operatorów genetycznych (krzyżowania i mutacji), ocena populacji uzyskanej w wyniku działania operatorów genetycznych oraz sukcesja. Ostatnie cztery fazy powtarzane są cyklicznie, do momentu – kiedy spełnione zostanie przyjęte kryterium stopu algorytmu (patrz 2.7). Powyższy schemat – bardziej obrazowo – został przedstawiony w postaci „pseudo kodu” na rysunku 4. Algorytm ewolucyjny rozpoczyna się inicjacją populacji bazowej $P^{g=0}$. Osobniki tej populacji generowane są zazwyczaj w sposób losowy, ale mogą być one również wielokrotną kopią rozwiązania dostarczonego przez inny algorytm optymalizacyjny (w przypadku problemów okresowych może to być np. rozwiązanie statyczne lub rozwiązanie statyczne nieznacznie zakłócone). Po wygenerowaniu populacji bazowej dokonuje się oceny populacji P^g (tzn. wyznacza się wartości funkcji przystosowania dla każdego jej osobnika). To właśnie od tej oceny zależy, czy algorytm będzie zatrzymany – czy nie. Jeśli nie jest spełnione kryterium stopu, wówczas – z wykorzystaniem jednej z metod reprodukcji opisanych w rozdziałach 2.4.1-2.4.3 – tworzona jest populacja tymczasowa P_{tmp}^g. Populacja ta poddawana jest następnie działaniu operatorów genetycznych, w wyniku czego uzyskuje się populację P_o^g. W kolejnym kroku populacja P_o^g podlega ocenie, po dokonaniu której tworzona jest (wykorzystując przy tym przyjętą metodę sukcesji)

populacja osobników będąca nowym pokoleniem P^{g+1} (nowa populacja bazowa). Opisany powyżej cykl (z wyłączeniem kroku inicjacji i oceny populacji P^0) powtarzany jest, dopóki kryterium stopu nie zostanie spełnione.

```
AlgorytmEwolucyjny{
  g = 0;
  P^0 = Inicjacja();
  O^g =  Ocena(P^g);
  do{

        if( WarunekStopu(O^0...O^g) )
              break;
        P^g_tmp  = Reprodukcja (O^g ,P^g);
        P^g_o  = OperatoryGenetyczne( P^g_tmp );
        O^g_o  = Ocena( P^g_o );
        P^{g+1} = Sukcesja(O^g , O^g_o ,P^g, P^g_o );
        O^{g+1} ⊂ O^g ∪ O^g_o;
        g = g+1;
  }while (true);
  return P^g;
}
```

Rysunek 4 Poglądowy schemat działania algorytmu ewolucyjnego

Wykorzystywane w algorytmach ewolucyjnych procesy reprodukcji i sukcesji symulują opisany przez Darwina proces doboru naturalnego. Poprzez reprodukcję, z populacji P^g tworzona jest populacja tymczasowa P^g_{tmp}, poddawana następnie działaniu operatorów genetycznych. Sukcesja natomiast tworzy z populacji bieżącej P^g i z populacji P^g_o nową populację bazową P^{g+1}, dla której powtarzany jest cały cykl działania algorytmu ewolucyjnego.

Wybór osobników do tymczasowej populacji w kroku reprodukcji dokonywany jest w sposób losowy, przy czym prawdopodobieństwo, znalezienia się w populacji tymczasowej, jest większe w przypadku osobników o lepszej wartości funkcji przystosowania. Owe preferowanie przez algorytmy ewolucyjne osobników lepiej przystosowanych sprawia, że poszukiwania rozwiązania optymalnego przemieszczają się w przestrzeni rozwiązań dopuszczalnych w kierunku „miejsca”, w którym mogą znajdować się bardziej wartościowe rozwiązania. Jednocześnie losowość wyboru populacji tymczasowej powoduje, że – mimo wszystko – mogą się w niej znaleźć osobniki słabe, co przyczynia się do tego, że algorytm ewolucyjny nie jest zbieżny do rozwiązania optymalnego zbyt szybko, dzięki czemu zmniejsza się prawdopodobieństwo znalezienia rozwiązań jedynie

lokalnie optymalnych. „Pozostawienie przy życiu” słabszych osobników może ponadto zaowocować tym, że w wyniku działania operatorów genetycznych powstanie osobnik o bardzo dobrej wartości funkcji przystosowania, który – w kolejnym kroku algorytmu ewolucyjnego – ukierunkuje jego działanie na przeszukiwanie niebadanego dotychczas obszaru rozwiązań dopuszczalnych (osobnik słabszy może sprawić, że algorytm opuści obszar „przyciągania” optimum lokalnego).

Metod tworzenia nowej populacji bazowej w kroku sukcesji jest kilka. Najczęściej stosowaną i jednocześnie najbardziej trywialną metodą jest tzw. metoda klasyczna, w której nową populację bazową tworzą wyłącznie osobniki wchodzące w skład populacji P_o^g uzyskanej w wyniku działania operatorów krzyżowania. Inną metodą sukcesji jest tzw. metoda z częściową wymianą. W metodzie tej nową populację bazową P^{g+1} tworzą osobniki, z których jedna część pochodzi z populacji P^g, a druga – z populacji P_o^g. Wybór osobników z populacji $P^g \cup P_o^g$ do nowej populacji bazowej może być dokonywany:

1) w sposób losowy;
2) tak, by do nowej populacji trafiły osobniki najmniej do siebie podobne (aby populacja ta zawierała jak najbardziej zróżnicowane osobniki);
3) tak, by do nowej populacji trafiały osobniki o najlepszej wartości przystosowania.

Dosyć często stosowaną metodą sukcesji jest również tzw. sukcesja elitarna, w której do nowej populacji bazowej P^{g+1} wybierane są najlepsze osobniki ze zbioru $P_e^g \cup P_o^g$, gdzie $P_e^g \subset P^g$ jest populacją składającą się z N_e najlepszych osobników wchodzących w skład populacji P^g (zwykle populacja P_e^g zawiera jednego osobnika).

Algorytm ewolucyjny, w którym jako metodę sukcesji zastosowano metodę sukcesji elitarnej lub metodę z częściową wymianą, jest szybciej zbieżny do rozwiązania optymalnego niż algorytm, w którym wykorzystano metodę klasyczną. Koszt uzyskania rozwiązania optymalnego jest więc w takim przypadku mniejszy; istnieje jednak ryzyko (którego nie ma przy zastosowaniu metody klasycznej), że algorytm ewolucyjny „utknie” w optimum lokalnym. Wadą sukcesji klasycznej, natomiast, jest to, że w przypadku jej wykorzystania nowa populacja bazowa może zawierać osobniki gorsze niż populacja bieżąca (której najlepsze osobniki mogą zostać „zniszczone” w wyniku działania operatorów genetycznych).

Operatory genetyczne obecne na schemacie z rysunku 4 mają za zadanie utworzenie populacji P_o^g składającej się z osobników charakteryzujących się cechami innymi niż osobniki należące do populacji P^g. Zmienione osobniki populacji P_o^g (wszystkie bądź tylko część) są następnie – w kroku sukcesji – wprowadzane do nowej populacji bazowej, dzięki czemu algorytm ewolucyjny może posuwać się w kierunku optimum. Wprowadzane przez operatory genetyczne zróżnicowanie populacji P^{g+1} może również przyczynić się do tego, że algorytm opuści obszar przyciągania optimum lokalnego i zacznie przeszukiwać nowe, niezbadane dotąd obszary.

Jak zatem tworzona jest populacja P_o^g? Najczęściej z populacji tymczasowej losowo wybiera się kilka (kilkanaście) osobników, a następnie łączy się je w pary. Każda taka para zostaje poddana działaniu określonego operatora krzyżowania (patrz rozdział 2.5), wynikiem czego jest para nowych osobników nazywanych osobnikami potomnymi. Osobniki potomne zastępują swoich rodziców w populacji tymczasowej. Następuje kolejne losowanie. Każdy, wylosowany z populacji tymczasowej, osobnik jest modyfikowany poprzez działający na niego operator mutacji. W wyniku tej operacji z każdego osobnika – rodzica powstaje nowy osobnik – potomek. Potomek zastępuje w populacji tymczasowej swojego rodzica i w ten sposób powstaje populacja P_o^g.

Pozostaje jeszcze do rozstrzygnięcia kwestia liczebności populacji bazowych w kolejnych cyklach działania algorytmu ewolucyjnego. Prawdopodobieństwo tego, że algorytm ewolucyjny znajdzie rozwiązanie globalnie optymalne zmierza do jedności, gdy liczba osobników w populacji oraz liczba generowanych pokoleń dążą do nieskończoności (czyli, gdy zwiększa się obszar, który algorytm może przeszukać) [83]. Oczywiście – ze względu na ograniczone możliwości obliczeniowe – niemożliwym jest stworzenie sytuacji, w której prawdopodobieństwo znalezienia globalnie optymalnego rozwiązania byłoby równe jedności, dlatego też osoba projektująca algorytm dla danego problemu, korzystając ze swojego doświadczenia, ustala taką wielkość populacji, dla której algorytm ewolucyjny dostarczy zadawalające rozwiązania w racjonalnym czasie. W najczęściej stosowanych, klasycznych algorytmach liczebność populacji jest więc stała i wynika z konieczności zrezygnowania z części walorów jakościowych rozwiązania na rzecz zmniejszenia kosztu jego uzyskania. Oczywiście w literaturze można również znaleźć przykłady, w których populacja bazowa zmienia się w czasie w zależności od czynników takich jak np.: zróżnicowanie populacji, nacisk selektywny, czy wiek poszczególnych osobników [3], [55], [83]; zmiana liczebności populacji w kolejnych

pokoleniach wydaje się bowiem korzystnie wpływać na działanie algorytmu ewolucyjnego. Przykładowo, gdy algorytm znajduje się daleko od optimum, lepiej jest, by liczba osobników w populacji była większa, co wiąże się z większym obszarem poszukiwań; natomiast, w przypadku znalezienia możliwego kierunku poprawy, lepiej jest, żeby populacja zmniejszyła swoją liczebność, co pozwoli algorytmowi na szybszą zbieżność do rozwiązania optymalnego. Z uwagi jednak na fakt, że wyznaczanie rozwiązania problemu okresowego jest bardzo czasochłonne, zaś zwiększenie wielkości populacji w trakcie działania algorytmu mogłoby znacznie wydłużyć czas jego działania, w dalszej części opracowania rozpatrywane będą jedynie przypadki ze stałą liczbą osobników w populacji.

Przedstawiony powyżej sposób zarządzania populacją jest sposobem najczęściej stosowanym, jednakże w literaturze można znaleźć także inne schematy działania algorytmów ewolucyjnych. Dość często opisywane są np. metody selekcji zwane wygaszającymi i zachowującymi. Różnią się one tym, że selekcja zachowująca daje szanse przeżycia każdemu osobnikowi w populacji (prawdopodobieństwo wyboru każdego osobnika jest zawsze różne od zera); w selekcji wygaszającej, natomiast, nie dopuszcza się do reprodukcji najlepszych osobników (tzw. selekcja lewa), bądź też ogranicza się życie takiego osobnika tylko do jednego pokolenia (selekcja wyłączna). Sposobów zarządzania populacją w algorytmach ewolucyjnych jest zresztą wiele, jednak z uwagi na fakt, że temat ten jest dość szeroki i często wychodzi poza ramy optymalizacji okresowej nie zostanie on niestety w pełni zgłębiony w niniejszym opracowaniu.

2.4.1. Ruletka

W poprzednim punkcie wspomniano, że populacja tymczasowa tworzona jest w algorytmie ewolucyjnym w sposób losowy oraz że osobniki lepiej przystosowane mają większe szanse przeżycia niż osobniki „słabe”. Nie powiedziano jednak dotychczas nic na temat sposobu wyznaczania prawdopodobieństwa przetrwania danego osobnika oraz sposobu losowania osobników do populacji tymczasowej; właśnie te zagadnienia będą tematem dalszej części niniejszego podrozdziału oraz podrozdziałów 2.4.2 i 2.4.3.

Jedną z metod tworzenia populacji tymczasowej jest tzw. metoda proporcjonalna [4], [83], [103], która bardzo często określana jest mianem ruletkowej metody reprodukcji. Nazwa oddaje w pełni istotę tej metody, która polega na tym, że każdemu osobnikowi przydzielany jest na kole (ruletce) obszar, którego wielkość zależy od wartości funkcji przystosowania danego osobnika. Następnie owo „koło fortuny” wprowadza się w ruch

i czeka się, aż się ono zatrzyma. Do populacji tymczasowej przechodzi osobnik (a w zasadzie jego kopia) wskazany przez ruletkę po zatrzymaniu się koła. Czynność kręcenia kołem powtarza się tyle razy, ile osobników liczy poddawana reprodukcji populacja bazowa P^g.

Prawdopodobieństwo wylosowania danego osobnika jest w tej metodzie wprost proporcjonalne do jego wartości funkcji przystosowania i wyznacza się je z następującej zależności:

$$p_w^i(z^i) = \frac{Q_\Phi(z^i)}{\sum_{j=1}^{N} Q_\Phi(z^j)}, \quad (i = 1,2,...,N) \tag{22}$$

gdzie: z^i jest i-tym osobnikiem należącym do bieżącej populacji P^g, dla którego wyznaczane jest prawdopodobieństwo wylosowania p_w^i; $Q_\Phi(z^i)$ jest wartością funkcji przystosowania dla osobnika z^i; N jest liczbą osobników w populacji P^g. Natomiast obszar przydzielony danemu osobnikowi na kole wyznaczany jest zgodnie ze wzorem:

$$d(z^i) = p_w^i 100\%. \quad (i = 1,2,...,N) \tag{23}$$

Przedstawiona metoda stosowana jest głównie w algorytmach klasycznych. Jedną z jej wad jest to, że – wprost – można ją stosować wyłącznie do problemów maksymalizacji. Oczywiście w przypadku problemów okresowych, które przeważnie mają na celu minimalizację, nie stanowi to zbyt dużego problemu, ponieważ dość łatwo zadanie minimalizacji można zmienić na odpowiadający mu problem maksymalizacji. Metoda ta ma jednak drugą, o wiele poważniejszą wadę, która w pewnym stopniu ogranicza możliwości jej wykorzystania. Mowa tu o wrażliwości metody ruletki na dodanie wartości stałej do funkcji przystosowania (stała ta wpływa na nacisk selektywny). W przypadku problemów okresowych z ograniczeniami stabilnościowymi (7), które (z uwagi na swój charakter) można uwzględniać jedynie przy użyciu kary w funkcji celu (a tym samym funkcji przystosowania), metoda ruletki może powodować zbyt szybką zbieżność algorytmu do rozwiązania niekoniecznie globalnie optymalnego [3], [83].

2.4.2. Reprodukcja turniejowa

Metoda turniejowa [4], [57], [83], [103], w odróżnieniu od opisanej wcześniej metody proporcjonalnej, nie jest wrażliwa na dodanie stałej do funkcji przystosowania. Metoda ta może ponadto być zastosowana wprost zarówno do zadań maksymalizacji jak i minimalizacji funkcji celu.

W metodzie tej wybór osobników (z populacji bieżącej P^g do populacji tymczasowej P^g_{tmp}) odbywa się na zasadzie „turnieju” pomiędzy losowo wybranymi z populacji bieżącej osobnikami. W praktyce wygląda to tak, że najpierw ustalana jest liczba osobników biorących udział w turnieju ($N_t << N$). Następnie, z populacji P^g losuje się (z jednakowym prawdopodobieństwem) N_t osobników, które będą brać udział w turnieju. Turniej wygrywa osobnik o najlepszej wartości funkcji przystosowania i „w nagrodę” jego kopia zostaje dodana do populacji tymczasowej P^g_{tmp}. Po zakończonym turnieju wszystkie (N_t) osobniki wracają do populacji P^g, a cały cykl powtarzany jest do momentu, kiedy populacja tymczasowa będzie miała tyle samo osobników – co populacja poddawana reprodukcji.

W opisanej metodzie naporem selekcyjnym steruje się dobierając odpowiednią liczbę osobników biorących udział w turnieju (N_t). Im liczba ta jest większa, tym większy jest napór selekcyjny metody. Z reguły przyjmuje się, że osobniki walczą parami, czyli $N_t = 2$.

Przedstawiony powyżej wariant metody turniejowej nazywany turniejem ze zwracaniem jest stosowany najczęściej. Jednakże istnieje również metoda turniejowa bez zwracania. W metodzie tej, osobnik wygrywający turniej przechodzi do populacji tymczasowej, zaś osobniki, które przegrały rywalizację, nie wracają do populacji P^g, lecz są bezpowrotnie kasowane. Cykl turniejów kończy się dopiero wtedy, gdy w populacji P^g nie ma już żadnego osobnika. Zastosowanie metody turniejowej bez zwracania sprawia, że populacja tymczasowa P^g_{tmp} jest mniejsza od populacji poddawanej reprodukcji P^g. Z tego też powodu sposób działania operatorów genetycznych w przypadku tej metody jest inny niż sposób opisany w rozdziale 2.4 – osobniki utworzone przez operatory genetyczne nie zastępują swoich rodziców, lecz odtwarzają liczebność populacji.

2.4.3. Reprodukcja rangowa

Metoda rangowa (rankingowa) [4], [83], [96], [103] jest w dużym stopniu podobna do metody ruletkowej. Również w tej metodzie wybór osobników do populacji tymczasowej następuje w wyniku zakręcenia kołem, na którym każdemu osobnikowi przyporządkowany jest pewien obszar. Obszar ten nie jest jednak bezpośrednio związany z wartością funkcji przystosowania danego osobnika, lecz z nadaną mu liczbą całkowitą

nazywaną rangą. Dzięki zastosowaniu rang (i tym samym rezygnacji z używania wartości funkcji przystosowania) metoda rangowa odporna jest na zwiększanie funkcji przystosowania o stałą. Może ona także być stosowana wprost zarówno do zadań maksymalizacji jak i minimalizacji.

Ranga jest liczbą określającą jakość osobnika na tle populacji. Nadaje się ją po wcześniejszym posortowaniu populacji zgodnie z niemalejącą lub nierosnącą wartością funkcji przystosowania poszczególnych osobników (sposób sortowania zależy od zastosowanej metody rankingowej oraz od charakteru zadania – minimalizacja/maksymalizacja). W posortowanej populacji każdemu osobnikowi przypisywana jest liczba całkowita (ranga) odpowiadająca jego pozycji w uszeregowanym ciągu. Po nadaniu rang można wyznaczyć prawdopodobieństwo wylosowania danego osobnika do populacji tymczasowej.

Przykładowo – system MATLAB, określając prawdopodobieństwo znalezienia się w populacji tymczasowej dla poszczególnych osobników, korzysta z następującej zależności [96]:

$$p_w^i(z^i) = \frac{W_s(r(z^i), s_p, N)}{\sum_{j=1}^{N} W_s(r(z^j), s_p, N)}, \quad (i = 1,2,..., N) \tag{24}$$

gdzie p_w^i jest prawdopodobieństwem wylosowania osobnika z^i, $r(z^i)$ jest rangą osobnika z^i, s_p jest ustalonym naciskiem selekcyjnym, N jest liczbą osobników w populacji, a W_s jest funkcją wyznaczającą współczynnik selekcji osobnika. Użytkownik systemu MATLAB, chcący zastosować metodę rankingową (24), ma do wyboru metodę liniową i nieliniową. W metodzie liniowej funkcja W_s zdefiniowana jest następująco:

$$W_s(r(z^i), s_p, N) = 2 - s_p + \frac{2(s_p - 1)(r(z^i) - 1)}{N - 1}, \tag{25}$$

natomiast w metodzie nieliniowej funkcja W_s ma postać:

$$R(r(z^i), s_p, N) = \frac{Nw(s_p)^{(r(z^i)-1)}}{\sum_{j=1}^{N} w(s_p)^{(j-1)}}, \tag{26}$$

gdzie $w(s_p)$ jest rozwiązaniem równania:

$$(s_p - N)w^{N-1} + s_p w^{N-2} + ... + s_p w + s_p = 0. \tag{27}$$

Różnica pomiędzy metodami liniową i nieliniową polega na tym, że w metodzie nieliniowej w szerszym zakresie można wpływać na nacisk selektywny (dla metody liniowej $s_p \in [1,2]$, natomiast dla metody nieliniowej $s_p \in [1, N-2]$); ponadto uważana jest ona za bardziej efektywną.

W literaturze można oczywiście spotkać jeszcze inne funkcje pozwalające określić prawdopodobieństwo wylosowania osobnika. Przykładowo, dla liniowej metody rankingowej prawdopodobieństwo przeżycia danego osobnika może zostać wyznaczone w następujący sposób [3], [83]:

$$p_w^i(z^i) = b_1 + b_2(1 - \frac{r(z^i)}{N}), \quad (i = 1,2,..., N) \tag{28}$$

$$p_w^i(z^i) = b_1 - b_2(r(z^i) - 1), \quad (i = 1,2,..., N) \tag{29}$$

natomiast w przypadku metody nieliniowej można zastosować funkcje:

$$p_w^i(z^i) = b_1 + b_2(N - r(z^i))^{b_3}, \quad (i = 1,2,..., N) \tag{30}$$

$$p_w^i(z^i) = b_1(1 - b_1)^{r(z^i)-1}, \quad (i = 1,2,..., N) \tag{31}$$

gdzie parametry b_i $(i = 1,2,3)$ określają napór selekcyjny, a dobiera się je tak, by spełnione były warunki: $\sum_{i=1}^{N} p_w^i(z^i) = 1$, $0 \le p_w^i \le 1$.

Należy tutaj zaznaczyć, że wzory (26)-(27) odpowiadają przypadkowi, w którym najlepszemu osobnikowi przyporządkowana jest ranga $r = N$, a najgorszemu $r = 1$. Natomiast wzory (28)-(31) opisują sytuację, w której najlepszemu osobnikowi przyporządkowana jest ranga $r = 1$, a najgorszemu $r = N$.

2.5. Operatory krzyżowania

Zdolność rozmnażania się osobników jest jednym z warunków koniecznych ewolucji. W procesie zapłodnienia, czyli w wyniku połączenia się gamet (komórek rozrodczych) – pochodzących najczęściej od dwóch różnych osobników (rodziców), powstaje jedna komórka (zygota), dająca początek nowemu życiu. Nie zagłębiając się w szczegóły można powiedzieć, że w trakcie zapłodnienia dochodzi do losowej wymiany materiału genetycznego rodziców tzn. do wymiany funkcjonalnie odpowiadających sobie, ale

nie identycznych części chromosomów. W wyniku takiej wymiany powstaje nowy osobnik, który wprowadza zróżnicowanie do populacji przez to, że – mimo iż posiada cechy obojga rodziców – nie jest dokładną repliką żadnego z nich. W algorytmach ewolucyjnych opisany powyżej proces rozmnażania symulowany jest za pomocą operatorów krzyżowania.

Z uzyskanej po reprodukcji populacji tymczasowej losowo wybierana jest grupa osobników, która stanowi zwykle od 25% do 100% populacji. Następnie, grupę tę losowo dzieli się na mniejsze podgrupy, które poddawane są krzyżowaniu. Wynikiem tej operacji jest przynajmniej jeden nowy osobnik potomny. Nowe osobniki dodaje się do populacji, lub zastępuje się nimi osobniki poddawane krzyżowaniu. Liczba osobników podlegających krzyżowaniu oraz liczba osobników będących wynikiem krzyżowania zależą od przyjętej strategii krzyżowania. Najczęściej jednak operatory krzyżowania stosuje się na dwóch osobnikach i dwa osobniki potomne uzyskuje się w wyniku krzyżowania.

Najpopularniejszymi operatorami krzyżowania, znajdującymi zastosowanie także przy rozwiązywaniu zadań okresowych, są [4], [71], [83], [103], [123]:

- operator krzyżowania wymieniającego (w wersji jednopunktowej, dwupunktowej lub wielopunktowej);
- operator krzyżowania równomiernego;
- operator krzyżowania arytmetycznego.

Dwa pierwsze typy operatorów odpowiadają wprost procesom zachodzącym w przyrodzie w trakcie rozmnażania – dostarczają one osobniki potomne, których materiał genetyczny powstaje w wyniku wymieszania się między sobą funkcjonalnie analogicznych genów krzyżowanych rodziców (mieszać mogą się ze sobą pojedyncze geny – operator równomierny, albo całe łańcuchy genów – operator wymieniający). Trzeci rodzaj operatora, jak wskazuje jego nazwa, tworzy genotyp nowych osobników poprzez liniową kombinację wartości odpowiadających sobie genów rodziców.

2.5.1. Krzyżowanie wymieniające

Operator krzyżowania wymieniającego może być stosowany w przypadkach zarówno kodowania binarnego jak i kodowania rzeczywistego. Istota działania tego operatora polega na tym, że osobniki podlegające krzyżowaniu wymieniają się między sobą funkcjonalnie odpowiadającymi sobie genami lub grupami genów.

Podstawową wersją operatora krzyżowania wymieniającego jest tzw. operator jednopunktowy, który z dwóch – dowolnie wybranych do krzyżowania – rodziców: $z^1 = (z_1^1, z_2^1, ..., z_m^1)$, $z^2 = (z_1^2, z_2^2, ..., z_m^2)$ tworzy parę nowych chromosomów ($z^{'}$, $z^{''}$) takich, że:

$$\begin{aligned} z^{'} &= (z_1^1, z_2^1, ..., z_c^1, z_{c+1}^2, z_{c+2}^2, ..., z_m^2), \\ z^{''} &= (z_1^2, z_2^2, ..., z_c^2, z_{c+1}^1, z_{c+2}^1, ..., z_m^1), \end{aligned} \tag{32}$$

przy czym $c \in \{1,2,...,m-1\}$ jest, wyznaczanym w sposób losowy, punktem podziału.

Jednopunktowy operator krzyżowania wymieniającego ma dwie wady. Po pierwsze jest on obciążony (tzn. w jednym krzyżowaniu nie można osiągnąć wszystkich kombinacji genów osiągalnych poprzez wielokrotne krzyżowanie jednopunktowe), przez co zwiększa się ryzyko porażki w poszukiwaniu rozwiązania globalnie optymalnego. Sytuacja taka może mieć miejsce w przypadku, gdy wszystkie osobniki osiągalne w wyniku jednego krzyżowania dla danej populacji mają gorsze wartości przystosowania niż osobniki aktualnie tworzące tę populację. W wyniku naporu selekcyjnego nowe osobniki będą zawsze odrzucane, przez co nie będzie można otrzymać osobników o wartości funkcji przystosowania lepszej od wartości funkcji dla osobników tworzących bieżącą populację; fakt, iż osobnik potomny (będący rezultatem jednokrotnego krzyżowania) jest gorzej przystosowany nie oznacza bowiem, że wartość funkcji przystosowania jego potomków również będzie zła - u potomków tych mogą bowiem ujawnić się cechy recesywne, które sprawią, że osobniki te będą osobnikami "wybitnymi".

Drugą wadą jednopunktowego operatora krzyżowania jest natomiast to, że – w przypadku problemów okresowych (1)-(7) – operator ten narusza ograniczenia uśrednione narzucone na sterowanie. O ile pierwsza wada może być łatwo zniwelowana poprzez odpowiedni dobór naporu selekcyjnego (np. pozostawianie przy życiu – w pewnych przypadkach – gorszych osobników) lub też zastosowanie operatora mutacji, o tyle wyeliminowanie drugiej wady wymaga zaprojektowania nowego operatora.

Jeżeli ograniczenia (5) są wypukłe (a z takimi najczęściej można się spotkać w przypadku problemów okresowych) i stosowane jest kodowanie rzeczywiste, wówczas drugą wadę jednopunktowego operatora wymieniającego można wyeliminować poprzez zastosowanie jednopunktowej wersji operatora wymieniającego, która wykorzystuje właściwości przestrzeni wypukłych [83]. W takim wypadku nowe chromosomy ($z^{'}$, $z^{''}$) – spełniające ograniczenia (5) – tworzone są w następujący sposób:

$$
\begin{aligned}
z^{'} &= (z_1^1, z_2^1, ..., z_c^1, (1-a)z_{c+1}^1 + az_{c+1}^2, (1-a)z_{c+2}^1 + az_{c+2}^2, ..., (1-a)z_m^1 + az_m^2), \\
z^{''} &= (z_1^2, z_2^2, ..., z_c^2, (1-a)z_{c+1}^2 + az_{c+1}^1, (1-a)z_{c+2}^2 + az_{c+2}^1, ..., (1-a)z_m^2 + az_m^1),
\end{aligned} \tag{33}
$$

gdzie $c \in \{1,2,...,m-1\}$ jest, wyznaczanym w sposób losowy, punktem podziału, natomiast $a \in [0,1]$ jest największą liczbą dla której $z^{'}$ i $z^{''}$ spełniają ograniczenia (5).

W literaturze można również spotkać się z dwupunktową wersją operatora wymieniającego [3]. Operator taki losowo dzieli dwa krzyżowane chromosomy na trzy części, a następnie tworzy nowe chromosomy wymieniając tylko środkowe fragmenty krzyżowanych chromosomów. Drugi punkt podziału wprowadzony został w celu wyeliminowania pierwszej wady jednopunktowego operatora krzyżowania, tzn. w celu stworzenia operatora, który nie byłby obciążony. Niestety, cel ten nie został osiągnięty, a operator dwupunktowy nadal jest – jak łatwo to udowodnić – operatorem obciążonym. Dążąc do usunięcia tej cechy operatora dodawano kolejne punkty cięć, tworząc tym samym tzw. wielopunktowy operator wymieniający [71].

2.5.2. Krzyżowanie równomierne

Dodawanie kolejnych punktów cięć do operatora wymieniającego przyczyniło się do powstania operatora krzyżowania równomiernego [123]. Operator ten tworzy nowe osobniki wymieniając między osobnikami krzyżowanymi nie całe ciągi genów, lecz jedynie pojedyncze geny. O tym, którymi genami mają się wymienić pomiędzy sobą osobniki rodzicielskie, decyduje los.

W przypadku dwóch chromosomów $z^1 = (z_1^1, z_2^1, ..., z_m^1)$, $z^2 = (z_1^2, z_2^2, ..., z_m^2)$ (zarówno bitowych jak i rzeczywistych) operator krzyżowania równomiernego tworzy geny nowych dwóch chromosomów według następującego schematu:

$$
\begin{aligned}
z_i^{'} &= \begin{cases} z_i^2 & \text{dla} \quad a_i < p_w, \\ z_i^1 & \text{dla} \quad a_i \geq p_w, \end{cases} \\
z_i^{''} &= \begin{cases} z_i^2 & \text{dla} \quad z_i^{'} = z_i^1, \\ z_i^1 & \text{dla} \quad z_i^{'} \neq z_i^1, \end{cases}
\end{aligned} \quad (i = 1,2,...,m), \tag{34}
$$

gdzie $a_i \in [0,1]$ jest zmienną losową o rozkładzie jednostajnym, natomiast $p_w \in (0,1)$ jest ustalonym prawdopodobieństwem wymiany genów między krzyżowanymi osobnikami.

Jak łatwo zauważyć, krzyżowanie równomierne nie jest – w przeciwieństwie do opisanych wcześniej operatorów wymieniających – operatorem obciążonym. Niestety wa-

dą omawianego operatora jest to, że w przypadku zastosowania go do rozwiązywania problemów okresowych z ograniczeniami uśrednionymi (5) narzuconymi na sterowanie, zwróci on osobniki, które naruszać będą wspomniane ograniczenia.

2.5.3. Krzyżowanie arytmetyczne

Operator krzyżowania arytmetycznego [4], [83] może być stosowany wyłącznie w przypadku kodowania rzeczywistego, ponieważ tworzone przez niego nowe osobniki są wynikiem liniowej kombinacji cech ich rodziców. Przy pomocy tego operatora nowe chromosomy $z^{'}$ i $z^{''}$ wyznaczane są z rodziców $z^1 = (z_1^1, z_2^1, ..., z_m^1)$ i $z^2 = (z_1^2, z_2^2, ..., z_m^2)$ według następującej zależności

$$\begin{aligned} z^{'} &= pz^1 + (1-p)z^2, \\ z^{''} &= pz^2 + (1-p)z^1, \end{aligned} \tag{35}$$

gdzie $p \in [0,1]$ jest zmienną losową wygenerowaną z rozkładem jednostajnym.

Jak widać, operator krzyżowania arytmetycznego (w odróżnieniu od jednopunktowego operatora wymieniającego) oddziałuje na wartość genów krzyżowanych chromosomów. Dzięki temu jest on operatorem nieobciążonym. Dodatkową zaletą operatora arytmetycznego jest to, że w przypadku gdy chromosomy podlegające krzyżowaniu należą do zbioru wypukłego, również ich potomkowie będą należeć do tego samego zbioru wypukłego. Zaleta ta ma duże znaczenie przy problemach okresowych. Oznacza ona bowiem, że osobniki potomne – oprócz tego, że spełniać będą ograniczenia chwilowe (4) – spełniać również będą ograniczenia uśrednione (5), specyficzne dla tego rodzaju problemów.

2.6. Operatory mutacji

Mutacja w przyrodzie jest jednym z ważniejszych mechanizmów ewolucji. Polega ona na losowej (choć mogącej podlegać pewnym wpływom środowiska), nagłej zmianie materiału genetycznego, która to zmiana jest następnie dziedziczona. Pomimo tego, że mutacja zachodzi w przyrodzie dosyć rzadko i najczęściej niesie ze sobą negatywne lub obojętne dla zmutowanego osobnika skutki, jest ona niezwykle istotna, ponieważ dzięki niej osobnik może nabyć nową cechę, która nie występowała u jego rodziców, a która zdecydowanie zwiększy jego wartość przystosowawczą.

Przykładem niebagatelnej roli, jaką mutacja odgrywa w przyrodzie może być przypadek komarów z rodziny Anopheles [140]. Drobna zmiana w genie kodującym enzym

u tego gatunku komarów (wymiana guaniny na adeninę w genie i glicyny na serynę w białku) spowodowała, że owady te uodporniły się na niektóre pestycydy stosowane do ich zwalczania. Dzięki tej mutacji komary z rodziny Anopheles (na nieszczęście człowieka) przeżywały opryski, których celem było ich całkowite wyniszczenie.

Operator mutacji w algorytmach ewolucyjnych [4], [83], [103], [123] (tak samo jak mutacja w naturze) wprowadza dodatkowe, losowe zmiany do populacji – zmiany, dzięki którym algorytm ewolucyjny nabiera zdolności „omijania" minimów lokalnych. W wyniku działania operatora mutacji niektóre osobniki mogą zyskać nowe cechy, które spowodują, że algorytm ewolucyjny zacznie przeszukiwać obszar (nieosiągalny np. w wyniku działania operatora krzyżowania), w którym znajdują się lepsze rozwiązania.

Operator mutacji dokonuje „drastycznych" zmian w populacji, dlatego nie należy stosować go zbyt często. Nadmierne stosowanie mutacji może bowiem doprowadzić do sytuacji, w której algorytm ewolucyjny nie będzie zbieżny do rozwiązania optymalnego – gdyż mutacja będzie niszczyć dobre rozwiązania. Zbyt rzadkie stosowanie mutacji może z kolei spowodować, że algorytm ewolucyjny będzie dostarczał jedynie lokalnie optymalne rozwiązania.

Z uwagi na powyższe, mutacji poddawanych jest zwykle nie więcej niż 10-15% osobników w populacji. Najpierw ustala się prawdopodobieństwo mutacji p_m, a następnie – w zależności od przyjętego sposobu kodowania – postępuje się zgodnie z procedurą opisana poniżej

- w przypadku kodowania binarnego dla każdego bitu (genu) – w każdym chromosomie z populacji tymczasowej – losowana jest z rozkładem jednostajnym liczba $p \in [0,1]$; jeśli (dla danego bitu) wylosowana liczba p jest mniejsza lub równa przyjętemu prawdopodobieństwu mutacji p_m, wówczas bit ten jest negowany (w przypadku kodowania binarnego prawdopodobieństwo mutacji p_m jest zwykle na poziomie 0.01);
- w przypadku kodowania rzeczywistego dla każdego chromosomu w populacji tymczasowej losuje się liczbę $p \in [0,1]$; jeżeli dla danego chromosomu zachodzi nierówność $p \leq p_m$, wówczas osobnik poddawany jest działaniu jednego z operatorów mutacji opisanych poniżej.

2.6.1. Mutacja równomierna

W literaturze można znaleźć dwa (podstawowe) rodzaje operatorów mutacji równomiernej: operator, który modyfikuje wszystkie geny mutowanego chromosomu [4] oraz operator, który modyfikuje tylko jeden losowo wybrany gen chromosomu [83].

Zanim jednak zostaną przedstawione równomierne operatory mutacji, należy zdefiniować funkcję nasycenia:

$$Sat(y, y^-, y^+) = \begin{cases} y^- & \text{dla} \quad y < y^-, \\ y & \text{dla} \quad y^- \le y \le y^+, \\ y^+ & \text{dla} \quad y > y^+, \end{cases} \tag{36}$$

która używana będzie przez te operatory w celu uwzględnienia ograniczeń chwilowych (4) występujących w problemach okresowych.

Równomierny operator mutacji[1], który modyfikuje każdy gen danego chromosomu poprzez dodanie do niego zmiennej losowej, definiowany jest w sposób następujący: jeżeli chromosom w formie zakodowanej reprezentowany jest przez wektor $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$), to – w wyniku działania na nim operatora mutacji równomiernej – otrzymuje się chromosom $z' \in R^m$, którego geny wyznaczane są z zależności:

$$z_i' = Sat(z_i + \xi_i, z_i^-, z_i^+), \ (i = 1,2,...,m), \tag{37}$$

przy czym ξ_i jest zmienną losową o rozkładzie normalnym lub o rozkładzie Cauchy'go (z zerową średnią i ustalonym odchyleniem standardowym).

Drugi (częściej stosowany) operator mutacji, który poprzez dodanie zmiennej losowej – modyfikuje tylko jeden, wybrany gen określonego chromosomu $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$) wyznacza geny nowego chromosomu $z' \in R^m$ według wzoru:

$$z_i' = \begin{cases} Sat(z_i + \xi, z_i^-, z_i^+) & \text{dla} \quad i = k, \\ z_i & \text{dla} \quad i \ne k, \end{cases} \ (i = 1,2,...,m), \tag{38}$$

[1] Użycie rozkładu normalnego lub Cauchy'ego w operatorze mutacji równomiernej powoduje, że nazwa tego operatora nie jest adekwatna do sposobu jego działania. Ponieważ jednak omawiany operator może zmieniać wartość genu w całym jego dopuszczalnym zakresie, nazwa "operator mutacji równomiernej" będzie używana, aby odróżnić przedmiotowy operator od operatora nierównomiernego (opisanego w kolejnym punkcie).

gdzie $k \in [1, m]$ jest losowo wybieranym genem, który będzie modyfikowany; natomiast ξ jest zmienną losową o rozkładzie normalnym lub o rozkładzie Cauchy'go.

Opisane wyżej operatory mutacji modyfikują cały mutowany chromosom lub tylko wybrany jego gen. Z przeprowadzonych eksperymentów (przedstawionych w następnym rozdziale) wynika jednak, że w przypadku problemów okresowych – zadawalające wyniki – uzyskuje się stosując operator mutacji będący połączeniem operatorów mutacji równomiernej i operatora mutacji stosowanego dla kodowania binarnego. Operator ten zdefiniowany jest w sposób następujący: jeżeli mutacji podlega chromosom $z = (z_1, z_2, ..., z_m)$ (gdzie: $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$), to geny zmutowanego chromosomu $z' \in R^m$ wyznaczane są z zależności:

$$z_i' = \begin{cases} Sat(z_i + \xi_i, z_i^-, z_i^+) & \text{dla} \quad p_i \le p_m, \\ z_i & \text{dla} \quad p_i > p_m, \end{cases} \quad (i = 1,2,...,m), \tag{39}$$

gdzie ξ_i jest zmienną losową o rozkładzie normalnym lub o rozkładzie Cauchy'go, $p_i \in [0,1]$ jest zmienną losową o rozkładzie jednostajnym, zaś $p_m \in [0,1]$ jest przyjętym prawdopodobieństwem mutacji genu. Jak można się domyślić, (w odróżnieniu od operatorów (37)–(38)) operator (39) stosowany jest dla każdego chromosomu w populacji.

Operator mutacji równomiernej wprowadza do populacji dosyć duże zróżnicowanie, dlatego jego wykorzystanie uzasadnione jest zwłaszcza na początku działania algorytmu ewolucyjnego – w momencie, kiedy konieczne jest przeszukanie całego obszaru rozwiązań dopuszczalnych w celu wyznaczenia obszaru, w którym może znajdować się rozwiązanie globalnie optymalne. Stosowanie operatora mutacji równomiernej na początku działania algorytmu ewolucyjnego jest szczególnie uzasadnione w przypadku, gdy populacja jest mało zróżnicowana (np. gdy wszystkie osobniki w populacji początkowej są kopią rozwiązania dostarczonego przez któryś z „algorytmów wspomagających" – opisanych w rozdziale 4).

Wprowadzając do populacji duże zróżnicowanie, operator mutacji równomiernej może zarówno negatywnie jak i pozytywnie wpływać na działanie algorytmu ewolucyjnego w końcowej fazie jego działania. Pozytywnie – gdyż dzięki działaniu tego operatora algorytm może opuścić minimum lokalne, w którym utknął; negatywnie – dlatego, że pod koniec działania algorytmu zbyt duże zmiany w populacji nie są pożądane z uwagi na fakt, iż mogą one niszczyć dobre rozwiązania oraz powodować, że algorytm nie dostroi się do rozwiązania optymalnego (ma to szczególne znaczenie wtedy, gdy algorytm

przeszukuje obszar, w którym znajduje się globalnie optymalne rozwiązanie danego problemu).

2.6.2. Mutacja nierównomierna

Operator mutacji nierównomiernej jest operatorem, który został stworzony w celu wyeliminowania wady operatora mutacji równomiernej objawiającej się pod koniec działania algorytmu ewolucyjnego. W początkowej fazie działania algorytmu operator ten działa tak samo – jak operator mutacji równomiernej, tzn. wprowadza dużą zmienność do populacji. Jednak w miarę trwania algorytmu wpływ operatora mutacji nierównomiernej na populację jest zmniejszany – wraz z pojawianiem się nowych pokoleń operator ten zawęża swój obszar działania do lokalnego otoczenia osobnika, który jest poddawany jego działaniu.

Operator mutacji nierównomiernej, będący odpowiednikiem operatora (37), jest definiowany następująco: jeżeli chromosom w formie zakodowanej reprezentowany jest przez wektor $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$), to wynikiem działania operatora mutacji nierównomiernej na tym wektorze jest wektor $z' \in R^m$, którego elementy wyznaczane są według zależności [83]:

$$z_i' = \begin{cases} z_i + F(g, z_i^+ - z_i) & \text{dla} \quad p_{1,i} \le 0.5, \\ z_i - F(g, z_i - z_i^-) & \text{dla} \quad p_{1,i} > 0.5, \end{cases} \quad (i = 1,2,...,m), \tag{40}$$

w której funkcja F opisywana jest wzorem:

$$F(g, a) = a p_2 (1 - \tfrac{g}{G_{\max}})^b, \tag{41}$$

natomiast $p_{ji} \in [0,1]$ $(j = 1,2;\ i = 1,2,...,n)$ jest zmienną losową o rozkładzie jednostajnym, G_{max} jest maksymalną liczbą pokoleń, g jest numerem bieżącego pokolenia poddawanego działaniu operatora mutacji, a b jest parametrem określającym niejednorodność.

W analogiczny sposób definiowany jest operator mutacji nierównomiernej, będący modyfikacją operatora (38). W przypadku tego operatora geny nowego chromosomu $z' \in R^m$ ($z_i' \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$) wyznaczane są według zależności

$$z_i^{'} = \begin{cases} z_i + F(g, z_i^+ - z_i) & \text{dla} \quad i = k, p \le 0.5, \\ z_i - F(g, z_i - z_i^-) & \text{dla} \quad i = k, p > 0.5, \ (i = 1,2,...,m), \\ z_i & \text{dla} \quad i \ne k, \end{cases} \tag{42}$$

gdzie $k \in [1, m]$ jest losowo wybieranym genem, który będzie modyfikowany; $p \in [0,1]$ jest zmienną losową o rozkładzie jednostajnym; g jest numerem bieżącego pokolenia; zaś funkcja F ma postać wzoru (41).

Nierównomierny operator mutacji, będący odpowiednikiem operatora (39), definiowany jest natomiast w sposób następujący: jeżeli mutowany jest chromosom $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$), to wynikiem działania operatora mutacji nierównomiernej jest chromosom $z^{'} \in R^m$, którego geny spełniają zależność:

$$z_i^{'} = \begin{cases} z_i + F(g, z_i^+ - z_i) & \text{dla} \quad p_{1i} \le p_m, p_{2i} \le 0.5, \\ z_i - F(g, z_i - z_i^-) & \text{dla} \quad p_{1i} \le p_m, p_{2i} > 0.5, \ (i = 1,2,...,m), \\ z_i & \text{dla} \quad p_{1i} > p_m, \end{cases} \tag{43}$$

gdzie $p_{ji} \in [0,1]$ ($j = 1,2$) jest zmienną losową o rozkładzie jednostajnym, $p_m \in [0,1]$ jest przyjętym prawdopodobieństwem mutacji genu , g jest numerem bieżącego pokolenia, zaś funkcja F opisywana jest wzorem (41).

Z przeprowadzonych przez Michalewicza badań [83] wynika, że algorytm ewolucyjny, w którym zastosowany był operator mutacji nierównomiernej jest szybciej zbieżny do rozwiązania optymalnego niż algorytm, w którym zastosowano operator mutacji równomiernej. Ponadto, rozwiązania uzyskiwane, przez algorytm „wspomagane” mutacją nierównomierną były zdecydowanie lepsze.

2.6.3. Mutacja brzegowa

Ponieważ często zdarza się, że sterowanie – będące optymalnym rozwiązaniem problemów okresowych – jest typu bang-bang należy wspomnieć o jeszcze jednym operatorze mutacji – tj. o operatorze mutacji brzegowej. Sposób działania tego operatora polega na tym, że nadaje on mutowanemu genowi tylko minimalną lub maksymalną, dopuszczalną wartość. Dzięki takiemu postępowaniu operator mutacji brzegowej jest operatorem skutecznym, który znacznie skraca poszukiwania rozwiązań optymalnych typu bang-bang.

Pierwsza wersja operatora mutacji brzegowej, w której modyfikacji podlegają wszystkie geny mutowanego chromosomu $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$) definiowana jest w sposób następujący:

$$z_i^{'} = \begin{cases} z_i^+ & \text{dla} \quad p_i \le 0.5, \\ z_i^- & \text{dla} \quad p_i > 0.5, \end{cases} \quad (i = 1,2,...,m), \tag{44}$$

gdzie $p_i \in [0,1]$ jest zmienną losową o rozkładzie jednostajnym, natomiast $z^{'} \in R^m$ jest osobnikiem uzyskanym w wyniku działania omawianego operatora mutacji.

Istotą drugiej (częściej stosowanej) wersji operatora mutacji brzegowej jest modyfikacja tylko jednego, wybieranego losowo genu. Dla genotypu w postaci wektora $z = (z_1, z_2, ..., z_m)$ (gdzie $z_i \in [z_i^-, z_i^+]$ dla $i = 1,2,...,m$) geny nowego osobnika $z^{'} \in R^m$ wyznaczane są w tym przypadku z zależności:

$$z_i^{'} = \begin{cases} z_i^+ & \text{dla} \quad i = k, p \le 0.5, \\ z_i^- & \text{dla} \quad i = k, p > 0.5, \\ z_i & \text{dla} \quad i \ne k, \end{cases} \quad (i = 1,2,...,m), \tag{45}$$

w której k – oznaczające modyfikowany gen – jest losową wartością z przedziału $[0, m]$, zaś $p \in [0,1]$ jest zmienną losową o rozkładzie jednostajnym.

Operator mutacji brzegowej będący połączeniem operatorów (44)–(45) oraz operatora mutacji dla kodowania binarnego jest zdefiniowany następująco:

$$z_i^{'} = \begin{cases} z_i^+ & \text{dla} \quad p_{1i} \le p_m, p_{2i} \le 0.5, \\ z_i^- & \text{dla} \quad p_{1i} \le p_m, p_{2i} > 0.5, \\ z_i & \text{dla} \quad p_{1i} > p_m, \end{cases} \quad (i = 1,2,...,m) \tag{46}$$

gdzie $p_{ji} \in [0,1]$ ($j = 1,2$) jest zmienną losową o rozkładzie jednostajnym, a $p_m \in [0,1]$ jest przyjętym prawdopodobieństwem mutacji genu.

2.7. Kryterium stopu algorytmu

Jak już wspomniano wcześniej, prawdopodobieństwo znalezienia optymalnego rozwiązania – w przypadku stosowania algorytmów ewolucyjnych – wzrasta wraz ze wzrostem liczby osobników w populacji oraz liczby pokoleń. Niestety, im większa jest liczba pokoleń i/lub liczebność populacji, tym dłuższy jest czas niezbędny algorytmowi na znalezienie optymalnego rozwiązania. Oczywistym jest, że w praktyce algorytm ewolu-

cyjny nie może działać w nieskończoność, dlatego też konieczne jest ustalenie kryterium stopu algorytmu będące kompromisem pomiędzy czasem działania algorytmu, a jakością dostarczonego przez niego rozwiązania.

Najczęściej stosowanymi kryteriami stopu algorytmów ewolucyjnych, wykorzystywanych do rozwiązywania zadań numerycznych, są:

1) ustalony czas pracy,
2) ustalona liczba pokoleń,
3) osiągnięcie ustalonego poziomu funkcji celu (funkcji przystosowania),
4) osiągnięcie minimalnej szybkości poprawy,
5) zanik różnorodności populacji,
6) kombinacja wyżej wymienionych kryteriów.

Dwa pierwsze wymienione kryteria są stosunkowo proste w realizacji i przez to – często stosowane. Wykorzystując pierwsze kryterium, ustala się maksymalny czas pracy algorytmu T_{max}, po przekroczeniu którego algorytm jest zatrzymywany. Analogicznie – w przypadku drugiego kryterium algorytm ewolucyjny zatrzymuje się, gdy liczba wygenerowanych pokoleń osiągnie, ustaloną przez użytkownika, maksymalną wartość G_{max}.

Wadą kryteriów 1 i 2 jest to, że nie uwzględniają one wartości funkcji celu rozwiązywanego problemu (funkcji przystosowania populacji); dlatego też – stosując te kryteria należy zdawać sobie sprawę z tego, że gdy działanie algorytmu ewolucyjnego zakończy się, uzyskane rozwiązanie może być wysoce niezadowalające. Mimo wszystko, kryteria 1 i 2 stosowane są zawsze w przypadku optymalizacji „online” – tzn. gdy optymalizacja musi odbywać się w czasie rzeczywistym (na przykład optymalizując układ, w którym stosowane jest sterowanie okresowe, algorytm musi zakończyć swoje działanie przed rozpoczęciem się nowego okresu sterowania).

Trzecie spośród wymienionych wyżej kryteriów stopu algorytmu znajduje zastosowanie w sytuacji, gdy – dla danego problemu – znane jest już jakieś rozwiązanie (uzyskane np. algorytmem lokalnej optymalizacji), które nie jest jednak satysfakcjonujące. W takim przypadku algorytm ewolucyjny wykorzystywany jest do poprawy rozwiązania optymalnego, a jego kryterium stopu sprawia, że algorytm zatrzymuje się, gdy znalezione zostanie rozwiązanie lepsze o określony przez użytkownika poziom. Kryterium osiągnięcia ustalonego poziomu funkcji celu idealnie nadaje się do optymalizacji problemów okresowych. Jak wspomniano w rozdziale 1, sterowanie okresowe często stosowane jest zamiast sterowania statycznego w celu poprawy wydajności procesu. Tak

więc, dla problemów optymalizacji okresowej omawiane kryterium stopu może być sformułowane w następując sposób: zatrzymaj działanie algorytmu ewolucyjnego, gdy prawdziwa jest nierówność

$$Q(z^{\min}) \le \hat{\bar{Q}} - \varepsilon, \tag{47}$$

gdzie $\hat{\bar{Q}} \in R$ jest wartością wskaźnika jakości dla optymalnego statycznego procesu sterowania, $\varepsilon \in R_+$ jest zakładanym stopniem poprawy sterowania statycznego, $z^{\min}$ jest najlepszym znalezionym osobnikiem, dla którego wskaźnik jakości $Q \in R$ przyjmuje minimalną wartość. Kryterium (47) zostało sformułowane dla problemów, w których minimalizuje się wartość wskaźnika jakości. W przypadku zadania polegającego na maksymalizacji wydajności procesu okresowego, kryterium to przybiera postać:

$$Q(z^{\max}) \ge \hat{\bar{Q}} + \varepsilon, \tag{48}$$

gdzie $z^{\max}$ jest osobnikiem, dla którego $Q \in R$ przyjmuje maksymalną wartość.

Wadą kryterium (47)–(48) jest to, że w określonym przypadku algorytm ewolucyjny może działać bez końca. Stanie się tak wówczas, gdy nie istnieje sterowanie okresowe zapewniające wydajność procesu lepszą od optymalnego sterowania statycznego o zadaną wartość $\varepsilon \in R_+$. Dlatego też kryterium osiągnięcia ustalonego poziomu funkcji celu należy zawsze stosować w połączeniu z kryterium 1 lub 2.

Kryterium nr 4 – kryterium osiągnięcia minimalnej szybkości poprawy jest ulepszoną wersją kryterium drugiego. W swoim działaniu kryterium to uwzględnia nie tylko liczbę pokoleń, ale także wartość funkcji celu. Zgodnie z tym kryterium, algorytm ewolucyjny powinien zostać zatrzymany w momencie, gdy wartość funkcji celu rozwiązywanego problemu – wyznaczona dla najlepszego osobnika w bieżącej populacji – w ciągu ostatnich g_p pokoleń uległa poprawie co najwyżej o zadaną przez użytkownika wartość $\varepsilon \in R_+$. Dla problemu okresowego, w którym minimalizuje się wartość funkcji celu, kryterium osiągnięcia minimalnej szybkości poprawy można zapisać następująco: zatrzymaj algorytm ewolucyjny, gdy spełniona jest nierówność

$$Q(z^{\min}(g - g_p)) - Q(z^{\min}(g)) \le \varepsilon, \tag{49}$$

gdzie g oznacza numer bieżącego pokolenia, zaś g_p jest zadaną przez użytkownika liczbą, określającą – dla ilu pokoleń wstecz ocena poprawy funkcji celu ma być wykony-

wana. Analogicznie, omawiane kryterium – dla zadania maksymalizacji – można zdefiniować jako: zatrzymaj algorytm ewolucyjny, gdy prawdziwa jest nierówność

$$Q(z^{\max}(g)) - Q(z^{\max}(g - g_p)) \leq \varepsilon. \tag{50}$$

Zazwyczaj w zależnościach (49)–(50) przyjmuje się $\varepsilon = 0$, co oznacza, że optymalizacja z wykorzystaniem algorytmu ewolucyjnego zatrzymywana jest, gdy po g_p pokoleniach funkcja celu w ogóle nie uległa poprawie.

W praktyce dobrze jest – ustalając kryterium stopu dla algorytmu ewolucyjnego – połączyć kryterium osiągnięcia minimalnej szybkości poprawy z kryterium ustalonego czasu pracy algorytmu ewolucyjnego. Dzięki takiemu połączeniu można bowiem zabezpieczyć się przed zbyt długą pracą algorytmu, która może mieć miejsce w przypadku dobrania zbyt małej wartości ε.

Oprócz wymienionych wyżej kryteriów stopu stosowane jest również kryterium 5 – badające zróżnicowanie populacji. Kryterium to ocenia zdolność algorytmu ewolucyjnego do opuszczania optimów lokalnych. Im bardziej bowiem osobniki w populacji są zróżnicowane, tym większe jest prawdopodobieństwo, że algorytm opuści obszar przeszukiwania lokalnego – większe jest prawdopodobieństwo, że – w wyniku działania operatorów genetycznych – powstaną osobniki „bardziej oddalone” w przeszukiwanej przestrzeni od swoich rodziców. Inaczej mówiąc, małe zróżnicowanie populacji oznacza, że algorytm zawęził swoje pole działania do bliskiego otoczenia optimum globalnego lub optimum lokalnego (którego raczej nie opuści); a więc – gdy zróżnicowanie populacji spadnie poniżej zadanej wartości – bardziej opłaca się zatrzymać algorytm ewolucyjny i zastąpić go jednym z algorytmów optymalizacji lokalnej opisanych w rozdziale 4 (zwykle koszt obliczeniowy algorytmu ewolucyjnego jest bowiem większy od kosztu obliczeniowego tradycyjnych metod optymalizacji).

Wśród najczęściej stosowanych ocen zróżnicowania populacji można wymienić [4]: badanie standardowych odchyleń poszczególnych genów w genotypie, badanie wartości dyskrepancji zbioru chromosomów oraz rozkład macierzy kowariancji na wartości i wektory własne.

3. ALGORYTMY EWOLUCYJNE DLA PROBLEMÓW GOSO

3.1. Kodowanie problemu okresowego

Przedstawiony we wstępie niniejszej pracy ogólny zapis ciągłego, średnioczęstotliwościowego, autonomicznego problemu globalnie optymalnego sterowania okresowego (1)-(7) jest z punktu widzenia konstruowania algorytmów optymalizacji cyklicznej i ich późniejszego testowania niewygodny. Niewygoda ta spowodowana jest choćby tym, że zmieniająca się w trakcie poszukiwania rozwiązania optymalnego wartość okresu utrudnia przedstawianie, a co za tym idzie, analizowanie wyników bezpośrednio z przebiegów stanu i sterowania. Ponadto zapis (1)-(7) utrudnia wyznaczanie pochodnych wskaźnika jakości względem okresu, sterowania i stanu. Co prawda, w rozdziale dotyczącym algorytmów ewolucyjnych wyznaczanie pochodnych nie będzie konieczne (algorytmy te nie korzystają bowiem z pochodnych), jednakże – z uwagi na fakt, że algorytmy pomocnicze, o których mowa w rozdziale 4, wymagają wyznaczenia pochodnych wskaźnika jakości – nie należy tej niedogodności bagatelizować.

Konstruowanie algorytmów optymalizacji dla problemów okresowych ułatwia ich transformacja do przedziału czasowego [0,1]. Dokonuje się jej stosując następujące podstawienia $t = \tau\,\tilde{t}$,, $u(t) = u(\tau\,\tilde{t}) = \tilde{u}(\tilde{t})$, $x(t) = x(\tau\,\tilde{t}) = \tilde{x}(\tilde{t})$, gdzie $\tilde{t} \in [0,1]$.

Wprowadzenie nowych zmiennych z falą do równań (1)-(7), a następnie opuszczenie fal nad zmiennymi powoduje, że dowolny problem globalnie optymalnego sterowania okresowego, który opisywany jest modelem (1)-(7) (sformułowanym we wstępie), może być zapisany w następującej, równoważnej, znormalizowanej postaci: zminimalizuj wskaźnik jakości

$$Q(x,u,\tau) = \nu(q), \tag{51}$$

przy ograniczeniach

$$q = \int_0^1 h(x(t),u(t))dt \tag{52}$$

$$\dot{x}(t) = \tau\ f(x(t),u(t)), \quad t \in [0,1], \quad x(0) = x(1), \tag{53}$$

$$\tau \in \mathcal{T}, \quad x(t) \in X, \quad u(t) \in U, \quad t \in [0,1], \tag{54}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \tag{55}$$

$$\phi(q) = 0, \quad \tilde{\phi}(q) \le 0, \tag{56}$$

$$\| s(\Phi(x,u,1)) \|_{\infty} \le \alpha. \tag{57}$$

Jak widać z zależności (51)-(57), problem (1)-(7) został zredukowany do przedziału czasowego [0,1] z okresem występującym jedynie w okresowym równaniu stanu (53).

Z uwagi na fakt, że znormalizowane zadanie optymalizacji okresowej (51)-(57) jest problemem ciągłym, natomiast przedstawione w poprzednim rozdziale operatory genetyczne działają na liczbach w systemie dwójkowym lub na wektorach liczb rzeczywistych, konieczne jest odpowiednie przekształcenie problemu (51)-(57). W tym celu sterowanie ciągłe $u(t)$ będzie aproksymowane za pomocą sterowania dyskretnego $u_K = (u_0, u_1, ..., u_{K-1})$, zdefiniowanego w K chwilach czasowych w następujący sposób:

$$\tilde{u}(t) = u_k \quad \text{dla} \quad t \in [k/K, (k+1)/K), \tag{58}$$

gdzie $u_k \in R^{n_u}$ $(k = 0,1,...,K-1)$.

Ponadto zakłada się, że znormalizowane nieliniowe równanie stanu

$$\dot{x}(t) = \tau \; f(x(t), \tilde{u}(t)), \quad t \in [0,1], \quad x(0) = x(1) \tag{59}$$

posiada lokalnie jednoznaczne rozwiązanie $x(t, \tau, x(0), u_K) \in X$ dla każdego dopuszczalnego wektora $(\tau, x(0), u_K)$ zmiennych procesowych.

Uwzględniając powyższe, problem globalnie optymalnego sterowania okresowego może zostać zapisany w następującej – znormalizowanej, dyskretnej postaci: zminimalizować wskaźnik jakości

$$Q(\tilde{z}) = \nu(q), \tag{60}$$

będący skalarną funkcją wektora wskaźników charakterystycznych procesu

$$q = \int_0^1 h(x(t, \tilde{z}), \tilde{u}(t)) dt \tag{61}$$

przy uwzględnieniu: ograniczenia na okresowość procesu

$$x(0) - x(1, \tilde{z}) = 0, \tag{62}$$

ograniczeń chwilowych

$$\tau \in \mathcal{T}, \quad x(0) \in X, \quad u_k \in U, \quad (k = 0,1,...,K-1),$$
$$x(t) \in X, \quad t \in [0,1], \tag{63}$$

uśrednionych ograniczeń zasobowych

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \tag{64}$$

uśrednionych ograniczeń technologicznych

$$\phi(q) = 0, \quad \tilde{\phi}(q) \le 0 \tag{65}$$

oraz ograniczenia stabilnościowego

$$\| s(\Phi(1, \tilde{z})) \|_\infty \le \alpha. \tag{66}$$

W powyższych zależnościach $\tilde{z} = (\tau, x(0), u_K)$ jest dyskretnym wektorem zmiennych procesowych, natomiast $x(t, \tilde{z})$ jest rozwiązaniem równania (59) dla danego $\tilde{z}$.

Analizując nowy sposób zapisu problemu globalnie optymalnego sterowania okresowego (60)-(66) oczywistym jest, że w algorytmach ewolucyjnych (stosowanych do rozwiązywania problemów sterowania okresowego) osobnik w formie zakodowanej powinien być reprezentowany przez wektor liczb rzeczywistych

$$z = (z_i)_{i=1}^m \ (m = 1 + n_x + n_u K), \tag{67}$$

którego każdy element odpowiada zmiennym procesowym: okresowi $z_1 = \tau$, stanowi początkowemu procesu $z_{1+i} = x_i(0)$ $(i = 1,2,..,n_x)$ oraz dyskretnemu sterowaniu $z_{2+n_x+k+(i-1)K} = u_{i,k}$ $(k = 0,1,...,K-1;\ i = 1,2,...,n_u)$; przy czym poszczególne elementy wektora z powinny podlegać ograniczeniom brzegowym $z_i \in [z_i^-, z_i^+]$ $(i = 1,2,...,m)$, wynikającym ze zbioru ograniczeń $Z = \mathcal{T} \times X \times U$.

Biorąc pod uwagę fakt, iż zwykle procesy okresowe opisywane są modelami zawierającymi od trzech do pięciu zmiennych stanu oraz od jednej do trzech zmiennych sterujących, stosowanie klasycznego kodowania binarnego w przypadku tego typu problemów jest niewskazane z uwagi na zbyt duży rozmiar osobnika. Dla zadania optymalizacyjnego, które zawiera dwie zmienne stanu oraz dwie zmienne sterujące, zdefiniowany dla dwudziestu chwil czasowych osobnik musiałby bowiem być zakodowany za pomocą 563 bitów (oczywiście przy założeniu, że wszystkie zmienne procesowe należą do

przedziału [0,2], a osobnik zakodowany jest z dokładności do czterech cyfr po przecinku).

3.2. Algorytm optymalizacji dla problemu GOSO z chwilowymi ograniczeniami zasobowymi

3.2.1. Problem GOSO z chwilowymi ograniczeniami zasobowymi

W celu przetestowania ewolucyjnego podejścia do rozwiązywania problemów globalnie optymalnego sterowania okresowego, jako pierwszy zostanie rozpatrzony przypadek, w którym występują ograniczenia na okresowość procesu oraz ograniczenia chwilowe narzucone na zmienne sterujące (na stan i sterowanie).

Znormalizowany i zdyskretyzowany problem GOSO z chwilowymi ograniczeniami zasobowymi polega na minimalizacji wskaźnik jakości

$$Q(z) = \nu(q), \tag{68}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t,z), \tilde{u}(t))dt\,, \tag{69}$$

$$\dot{x}(t,z) = \tau\ f(x(t,z), \tilde{u}(t)), \quad t \in [0,1], \tag{70}$$

$$x(0) - x(1,z) = 0, \tag{71}$$

$$\tau \in \mathcal{T}, \quad x(0) \in X\ , \quad u_k \in U, \quad (k = 0,1,...,K-1). \tag{72}$$

W powyższym problemie, zamiast wektora $\tilde{z}$, został użyty osobnik w formie zakodowanej z, który od tego momentu będzie występował przy każdym opisie problemu okresowego.

3.2.2. Ewolucyjny algorytm optymalizacji (Algorytm 1)

W przypadku zadania (68)-(72) żaden z operatorów genetycznych – opisanych w rozdziale 2 – nie narusza ograniczeń chwilowych (72), którym podlega okres oraz dyskretne sterowanie. Pewien problem stanowią tu jednak ograniczenia (71), które wymuszają okresową pracę danego procesu przez co komplikują proces projektowania algorytmu ewolucyjnego. Jednym z rozwiązań tego problemu jest uwzględnienie ograniczenia $x(0) - x(1,z) = 0$ za pomocą metody Newtona-Raphsona [68]. W metodzie tej, dla

ustalonego okresu τ, sterowania u_K oraz stanu początkowego procesu $x(0)$, nową wartość stanu początkowego $x^{i+1}(0)$ wyznacza się według zależności:

$$x^{i+1}(0) = x^i(0) - (I - \Phi^i(1,z))^{-1}(x^i(0) - x^i(1,z)) . \tag{73}$$

Formuła (73) powtarzana jest do czasu, aż spełnione będzie równanie $x(0) - x(1,z) = 0$. Oczywiście, ze względów praktycznych, wyznaczanie stanu początkowego za pomocą formuły (73) może być przerwane, gdy spełniona będzie nierówność $(x(0) - x(1,z))^T (x(0) - x(1,z)) \le \varepsilon_N$ (gdzie ε_N jest zadaną dokładnością, z którą ma być spełnione ograniczenie na okresowość procesu (71)).

Przedstawiona powyżej metoda Newtona-Raphsona w większości przypadków działa bez zastrzeżeń. Można jednak spotkać się z zadaniami, w których wystąpi problem ze znalezieniem macierzy odwrotnej macierzy $(I - \Phi(1,z))$. Dla takich zadań (w których wspomniana macierz jest osobliwa) można zastosować metodę Gaussa-Newtona [37], w której nową wartość stanu wyznacza się z zależności:

$$x^{i+1}(0) = x^i(0) + \kappa_{GN}\vartheta \tag{74}$$

gdzie ϑ jest rozwiązaniem równania $(I - \Phi^i(1,z))\vartheta = -(x^i(0) - x^i(1,z))$, zaś $\kappa_{GN} \in [0,1]$ dobiera się tak, by – dla nowego stanu początkowego – wyrażenie $(x^{i+1}(0) - x^{i+1}(1,z))^T (x^{i+1}(0) - x^{i+1}(1,z))$ przyjmowało najmniejszą wartość.

Mając powyższe, algorytm ewolucyjny dla problemu globalnie optymalnego sterowania okresowego (68)-(72) można zdefiniować następująco:

Algorytm 1a (algorytm z turniejową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; $N_t < N$ – ilość osobników biorących udział w turnieju; K – poziom dyskretyzacji sterowania; ε, g_p – parametry kryterium stopu, badającego osiągnięcie przez algorytm minimalnej szybkości poprawy (49); $G_{\max}$ – maksymalna liczba pokoleń; $p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (72) (tzn. by $z \in Z$) Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Stosując metodę Newtona (73) lub Gaussa-Newtona (74) doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (71) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu (przystosowania) $Q(z^i)$ $(i = 1,2,..., N)$. Wśród osobników spełniających ograniczenia (71)- (72) znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q oraz osobnika $z^{\max}$ z największą wartością funkcji celu Q.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$ to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Powtarzaj poniższe czynności do momentu, aż bieżąca populacja P^g będzie pusta:

- wylosuj z bieżącej populacji N_t osobników;
- weź z N_t osobników do populacji tymczasowej P^g_{tmp} osobnika z najmniejszą wartością funkcji celu Q;
- usuń z bieżącej populacji N_t osobników.

Krok 7 (krzyżowanie) Skopiuj do nowej populacji P^{g+1} osobniki z populacji tymczasowej P^g_{tmp}. Następnie, powtarzaj poniższe czynności tak długo, aż nowa populacja P^{g+1} będzie składać się z N osobników:

- wylosuj dwa osobniki z populacji tymczasowej P^g_{tmp} i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania, natomiast krzyżowane osobniki zwróć do populacji P^g_{tmp}.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w populacji działaniu tego operatora – przy czym otrzymanymi, nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano ope-

rator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,..., N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \le p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

Algorytm ewolucyjny dla problemu (68)-(72), wykorzystujący jako metodę reprodukcji metodę ruletkową 2.4.1 lub metodę rangową 2.4.3, może być zdefiniowany następująco:

Algorytm 1b (algorytm z ruletkową lub rangową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; K – poziom dyskretyzacji sterowania; ε, g_p - parametry kryterium stopu, badającego osiągnięcie przez algorytm minimalnej szybkości poprawy (49); $G_{\max}$ – maksymalna liczba pokoleń; $p_c \in (0,1]$ – prawdopodobieństwo krzyżowania; $p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Spośród metod reprodukcji (metody ruletkowej i metod rangowych) wybierz metodę, według której będziesz wyznaczać prawdopodobieństwo przeżycia osobnika. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (72) (tzn. by $z \in Z$) Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Stosując metodę Newtona (73) lub Gaussa-Newtona (74) doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (71) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu (przystosowania) $Q(z^i)$ $(i = 1,2,..., N)$. Wśród osobników spełniających ograniczenia (71)- (72) znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q oraz osobnika $z^{\max}$ z największą wartością funkcji celu Q.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$ to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Stosując wybraną metodę reprodukcji wyznacz prawdopodobieństwo przeżycia (wylosowania) $p_w^i(z^i)$ $(i = 1,2,...,N)$ dla każdego osobnika w populacji, a następnie – dla każdego osobnika z^i $(i = 1,2,...,N)$ – wyznacz dystrybuantę $d_i = \sum_{j=1}^{i} p_w^j$ $(i = 1,2,...,N)$. Powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^g będzie zawierać N osobników:

- wylosuj z równomiernym rozkładem prawdopodobieństwa liczbę $p \in [0,1]$;
- wstaw do populacji tymczasowej P_{tmp}^g kopię osobnika z^i $(i = 1,2,...,N)$, dla którego prawdziwa jest nierówność $d_{i-1} < p \leq d_i$.

Krok 7 (krzyżowanie) Dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji tymczasowej wylosuj $p_i \in [0,1]$. Przenieś do nowej populacji P^{g+1} te osobniki z populacji tymczasowej P_{tmp}^g, dla których $p_i > p_c$. Jeżeli populacja P_{tmp}^g zawiera nieparzystą liczbę osobników, to losowo wybierz z niej jednego osobnika i przenieś go do populacji P^{g+1}. Następnie, powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^g będzie pusta:

- wylosuj dwa osobniki z populacji tymczasowej P_{tmp}^g i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania;
- usuń krzyżowane osobniki z populacji P_{tmp}^g.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w populacji działaniu tego operatora – przy czym otrzymanymi nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano ope-

rator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,..., N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \le p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

3.2.3. Przykłady obliczeniowe

Przykład 1.

Działanie algorytmów ewolucyjnych opisanych w poprzednim punkcie zostanie zilustrowane na przykładzie problemu GOSO dla ciągłego procesu produkcyjnego, którego istotą jest równoległa reakcja chemiczna $A \to B$, $A \to C$ zachodząca w reaktorze zbiornikowym z wymieszaniem.

Zadaniem poszukiwanego sterowania jest minimalizacja wskaźnika jakości:

$$Q(x,u,\tau) = -\frac{1}{\tau}\int_0^{\tau} x_2(t)dt \tag{75}$$

procesu opisanego równaniami stanu

$$\dot{x}_1(t) = 1 - x_1(t) - u(t)x_1^2(t) - u^{0.75}(t)x_1(t), \tag{76}$$

$$\dot{x}_2(t) = -x_2(t) + u(t)x_1^2(t) \tag{77}$$

przy uwzględnieniu ograniczeń

$$x(0) = x(\tau), \tag{78}$$

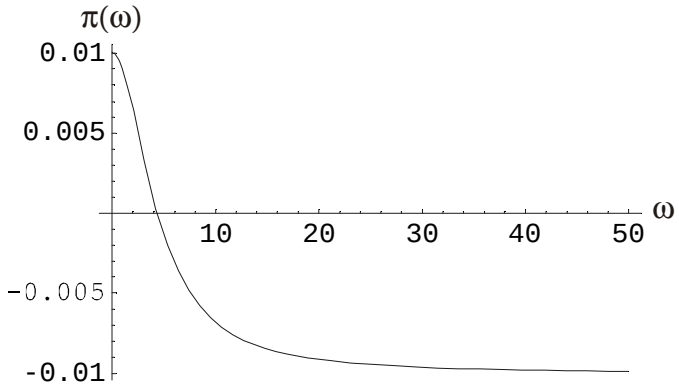
$$\tau \in [0.1,5], \quad 1 \le u(t) \le 5, \quad t \in [0,\tau]. \tag{79}$$

W omawianym przykładzie zmienna sterująca $u(t)$ oznacza znormalizowaną temperaturę reaktora, $x_1(t)$ – stężenie składnika A w reaktorze, natomiast $x_2(t)$ – stężenie składnika B w reaktorze. Wskaźnik jakości omawianego problemu odpowiada maksymalizacji średniego stężenia produktu B.

W przypadku poszukiwania optymalnego rozwiązania okresowego danego procesu produkcyjnego, pierwszym krokiem jaki należy wykonać, jest sprawdzenie, czy zastosowanie sterowania okresowego dla danego procesu jest uzasadnione (tzn. czy istnieje

sterowanie okresowe, które poprawi wydajność procesu w porównaniu z jego wydajnością dla optymalnego sterowania statycznego). Algorytmy optymalizacji okresowej, a w szczególności algorytmy ewolucyjne przedstawione w punkcie 3.2.2, są bowiem bardzo czasochłonne, dlatego też przed rozpoczęciem poszukiwania optymalnego sterowania okresowego należy najpierw – przy pomocy „szybkiego” testu π [13], [15] – sprawdzić, czy istnieje sterowanie okresowe poprawiające sterowanie statyczne.

Dla zadania (75)-(79) istnieje sterowanie okresowe, które zapewni mniejszą średnią wartość wskaźnika jakości Q w porównaniu z wartością optymalnego wskaźnika jakości $\hat{\bar{Q}}$ problemu statycznego. Potwierdza to test π, którego wynik przedstawiony jest na rysunku 5. O tym, że istnieje sterowanie okresowe „dominujące” nad sterowaniem statycznym, świadczą ujemne wartości przyjmowane przez krzywą $\pi(\omega)$. Z testu π można także dowiedzieć się, ile w przybliżeniu może wynosić optymalna wartość okresu sterowania okresowego. Wartość tę wyznacza się z zależności $\tau = 2\pi / \hat{\omega}$, gdzie $\hat{\omega}$ jest częstością, dla której krzywa $\pi(\omega)$ przyjmuje minimalną wartość.



Rysunek 5 Wynik testu π dla przykładu 1

Ponieważ test π dla rozpatrywanego zadania reakcji równoległej wykazał, że istnieje sterowanie okresowe, które może poprawić wydajność procesu w porównaniu ze sterowaniem statycznym, można rozpocząć poszukiwanie optymalnego sterowania okresowego, korzystając przy tym z omówionych w punkcie 3.2.2 algorytmów ewolucyjnych. Wspomniane algorytmy zostały zaimplementowane w systemie Mathematica [142].

Wyniki otrzymane dla zadania (75)-(79) – za pomocą Algorytmu 1a i Algorytmu 1b – zostały zebrane w tabeli 1; natomiast najlepsze, znalezione rozwiązanie okresowe, dla którego wskaźnik jakości $\hat{Q} = -0.196397$, zostało przedstawione na rysunkach 7-9. W tabeli 1 K-(34) oraz K-(35) oznacza, że wyniki zostały uzyskane odpowiednio przy

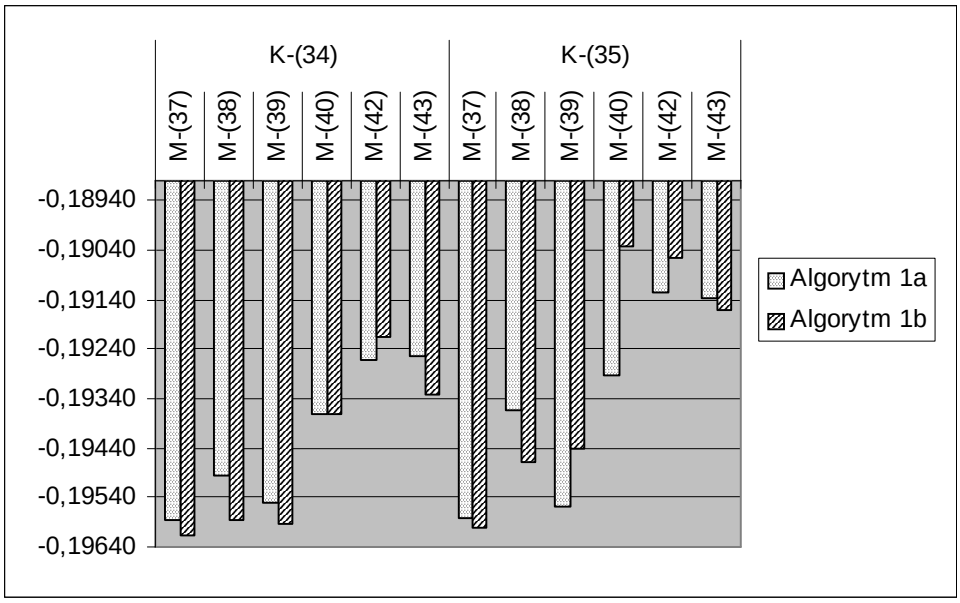
użyciu równomiernego (34) i arytmetycznego (35) operatora krzyżowania; M-(38), M-(39) oznaczają równomierne operatory mutacji (37), (38), (39), zaś M-(40), M-(42), M-(43) – operatory nierównomierne (40), (42), (43). Wyniki przedstawione w tabeli 1 uzyskane zostały dla następujących parametrów: $G_{\max} = 3000$, $\varepsilon = 10^{-6}$, $g_p = 500$, $N = 20$, $N_t = 2$, $K = 20$, $p_c = 0.5$. Prawdopodobieństwo mutacji, gdy używano operatorów mutacji (39) i (43), było równe $p_m = 0.015$, natomiast dla pozostałych operatorów wynosiło ono $p_m = 0.15$.

Tabela 1 Wyniki algorytmu ewolucyjnego dla zadania (75)-(79)

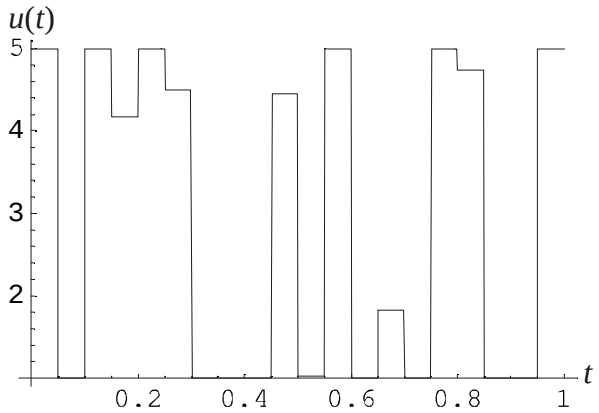
Algorytm 1a							
Próba		**12**	**2**	**3**	**4**	**5**	**Wynik średni**
K-(34)	**M-(37)**	**-0,196397**	-0,195902	-0,195790	-0,196268	-0,194865	-0,195844
	M-(38)	-0,194846	-0,195069	-0,194983	-0,195049	-0,194917	-0,194973
	M-(39)	-0,194996	-0,195714	-0,195729	-0,196266	-0,194886	-0,195518
	M-(40)	-0,193368	-0,194066	-0,194167	-0,193565	-0,193458	-0,193725
	M-(42)	-0,191380	-0,193705	-0,192458	-0,192287	-0,193238	-0,192614
	M-(43)	-0,192649	-0,192862	-0,192374	-0,192372	-0,192438	-0,192539
K-(35)	**M-(37)**	-0,195872	-0,195923	-0,195779	-0,195825	-0,195775	-0,195835
	M-(38)	-0,193522	-0,193697	-0,193561	-0,193762	-0,193692	-0,193647
	M-(39)	-0,195691	-0,195587	-0,195449	-0,195568	-0,195618	-0,195583
	M-(40)	-0,193019	-0,192730	-0,193260	-0,192813	-0,192844	-0,192933
	M-(42)	-0,191198	-0,191368	-0,191281	-0,190820	-0,191672	-0,191268
	M-(43)	-0,191001	-0,191423	-0,191191	-0,191361	-0,191894	-0,191374
Algorytm 1b (rangowa metoda reprodukcji (29))							
Próba		**1**	**2**	**3**	**4**	**5**	**Wynik średni**
K-(34)	**M-(37)**	-0,196291	-0,196285	-0,195944	-0,196246	-0,196107	-0,196174
	M-(38)	-0,196089	-0,195955	-0,195806	-0,195531	-0,195934	-0,195863
	M-(39)	-0,195750	-0,196022	-0,196022	-0,195947	-0,195908	-0,195930
	M-(40)	-0,193618	-0,194056	-0,194042	-0,193216	-0,193728	-0,193732
	M-(42)	-0,192399	-0,192114	-0,191811	-0,191976	-0,192544	-0,192169
	M-(43)	-0,194223	-0,193135	-0,192697	-0,193697	-0,192852	-0,193321
K-(35)	**M-(37)**	-0,196149	-0,195846	-0,196113	-0,195952	-0,196066	-0,196025
	M-(38)	-0,194532	-0,194743	-0,194716	-0,194560	-0,194858	-0,194682
	M-(39)	-0,194909	-0,195672	-0,192538	-0,193973	-0,194998	-0,194418
	M-(40)	-0,190327	-0,190099	-0,190401	-0,190340	-0,190477	-0,190329
	M-(42)	-0,190715	-0,190815	-0,190295	-0,190329	-0,190694	-0,190570
	M-(43)	-0,191765	-0,191725	-0,191553	-0,191475	-0,191529	-0,191609

Aby ułatwić porównanie jakości rozwiązań dostarczanych przez Algorytm 1a i Algorytm 1b, na rysunku 6 przedstawione zostały średnie wyniki z pięciu niezależnych prób. Jak widać, algorytmy ewolucyjne dostarczyły najlepszych rozwiązań dla równomiernych operatorów mutacji. Rodzaj użytego krzyżowania nie miał natomiast dużego wpływu na jakość wyznaczonego rozwiązania. Porównując oba zastosowane algorytmy,

można stwierdzić, że Algorytm 1b jest nieco skuteczniejszy od Algorytmu 1a, bowiem dostarczył on 8 najlepszych wyników, podczas gdy Algorytm 1a wygenerował ich tylko 4.



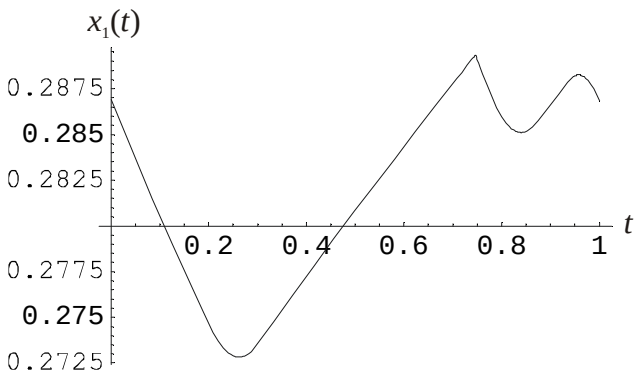
Rysunek 6 Średnie wyniki z pięciu prób uzyskane przy wykorzystaniu Algorytmu 1 dla problemu (75)-(79)



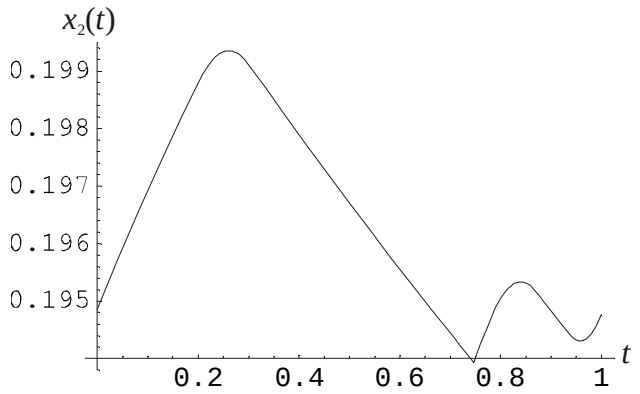
Rysunek 7 Sterowanie optymalne $\hat{u}(t)$ wyznaczone algorytmem ewolucyjnym, zapewniające najlepszą wartość wskaźnika jakości dla problemu (75)-(79)

Zadanie (75)-(79) analizowane było także przez Horna i Lina [68] oraz przez Fjelda [48]. Horn i Lin – wykorzystując spadkową metodę optymalizacyjną – uzyskali wskaźnik jakości $Q = -0.1931$. Natomiast Fjeld – stosując metodę płaszczyzn fazowych – wyznaczył sterowanie, dla którego wskaźnik jakości był równy $Q = -0.195$. Jak widać z wyników zamieszczonych w tabeli 1, algorytmy ewolucyjne (z równomiernym operatorem mutacji) w analizowanym przykładzie dostarczyły rozwiązania lepsze bądź rów-

ne rozwiązaniom znalezionym przez algorytmy klasyczne (oczywiście przy zaokrągleniu ich do trzeciego miejsca po przecinku). Najlepsze rozwiązanie uzyskane algorytmem ewolucyjnym ($\hat{Q} = -0.196397$) poprawia rozwiązanie Fjelda o niecały 1%, zaś rozwiązanie uzyskane przez Horna i Lina – o 1.7%; natomiast w porównaniu z rozwiązaniem statycznym, które wynosi $\hat{\overline{Q}} = -0.185$, najlepsze rozwiązanie okresowe (wyznaczone z wykorzystaniem algorytmu ewolucyjnego) poprawia wydajność procesu o ponad 6%.



Rysunek 8 Optymalny przebieg stanu $\hat{x}_1(t)$, odpowiadający sterowaniu z rysunku 7, który z okresem $\hat{\tau} = 0.1$ zapewnia wskaźnik jakości $\hat{Q} = -0.196397$



Rysunek 9 Optymalny przebieg stanu $\hat{x}_2(t)$, odpowiadający sterowaniu z rysunku 7, który z okresem $\hat{\tau} = 0.1$ zapewnia wskaźnik jakości $\hat{Q} = -0.196397$

Przykład 2.

Kolejnym przykładem, na którym przetestowane zostanie ewolucyjne podejście do optymalizacji procesów okresowych, jest ciągły proces produkcyjny, którego istotą jest równoległa reakcja chemiczna $A \to B,\ A \to C$ zachodząca w przepływowym reaktorze zbiornikowym z mieszalnikiem.

Zminimalizuj wskaźnik jakości:

$$Q(x,u,\tau) = -\frac{1}{\tau}\int_0^{\tau} x_2(t)dt \tag{80}$$

procesu opisanego równaniami stanu

$$\dot{x}_1(t) = 1 - 10^4 x_1^2(t)e^{-1/x_3(t)} - 400x_1(t)e^{-0.55/x_3(t)} - x_1(t), \tag{81}$$

$$\dot{x}_2(t) = 10^4 x_1^2(t)e^{-1/x_3(t)} - x_2(t), \tag{82}$$

$$\dot{x}_3(t) = u(t) - x_3(t) \tag{83}$$

przy uwzględnieniu ograniczeń

$$x(0) = x(\tau), \tag{84}$$

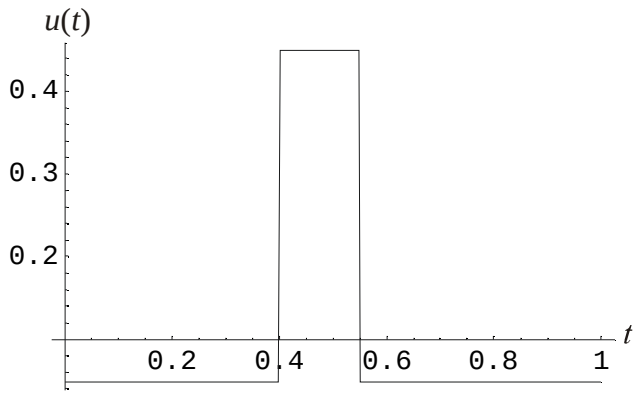
$$\tau \in [1,3], \quad 0.049 \le u(t) \le 0.449, \quad t \in [0,\tau]. \tag{85}$$

W powyższym przykładzie zmienna sterująca $u(t)$ odpowiada przepływowi ciepła przez ścianę reaktora, $x_1(t)$ – stężeniu składnika A w reaktorze, $x_2(t)$ – stężeniu produktu użytecznego B w reaktorze (C jest produktem ubocznym reakcji). Wskaźnik jakości omawianego problemu odpowiada maksymalizacji średniego stężenia produktu użytecznego B.

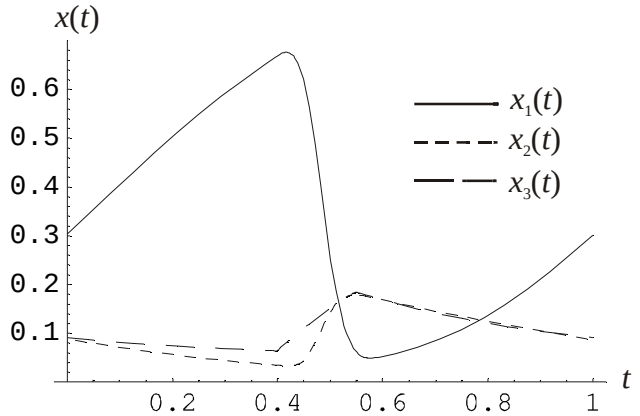
Przedstawiony przykład był już rozpatrywany przez Baileya, Horna i Lina [10]. Naukowcy ci wykazali, że optymalne sterowanie statyczne $\hat{\bar{u}} = 0.149$, dla którego wskaźnik jakości jest równy $\hat{\bar{Q}} = -0.0846$, zapewnia gorszą wydajność procesu w porównaniu z wydajnością uzyskaną przy sterowaniu okresowym. Bailey, Horn i Lin wyznaczyli (z wykorzystaniem metod numerycznych) optymalne sterowanie okresowe, dla którego wskaźnik jakości $Q = -0.0923$ poprawiał rozwiązanie statyczne o 9%.

W niniejszej pracy do rozwiązania zadania (80)-(85) wykorzystany został Algorytm 1a; przy czym wykonanych było pięć prób z wykorzystaniem jednopunktowego operatora krzyżowania wymieniającego (32) i mutacji brzegowej (45) oraz pięć – z wykorzystaniem arytmetycznego operatora krzyżowania (35) i równomiernego operatora mutacji (38). We wszystkich próbach przyjęto następujące wartości parametrów: $N = 20$, $N_t = 2$, $K = 20$, $p_m = 0.15$.

Najlepsze wyniki uzyskano w próbach, w których wykorzystywano operator krzyżowania wymieniającego i operator mutacji brzegowej. Średnia wartość wskaźnika jakości z tych prób po 2000 pokoleń wynosiła $\hat{Q}_s = -0.09450$. Natomiast najlepszy znaleziony wskaźnik jakości, odpowiadający sterowaniu i stanowi przedstawionemu na rysunkach 10-12, jest równy $\hat{Q} = -0.0968$. Wyznaczone z wykorzystaniem algorytmu ewolucyjnego najlepsze rozwiązanie okresowe poprawia zatem rozwiązanie uzyskane przez Baileya, Horna i Lina o blisko 5%.



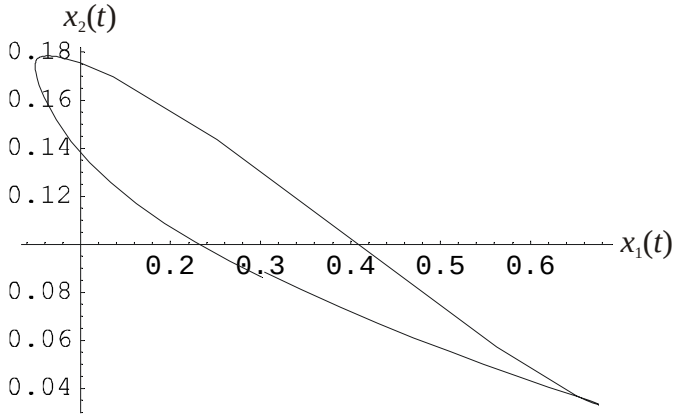
Rysunek 10 Optymalne sterowanie okresowe $\hat{u}(t)$ wyznaczone dla zadania (80)-(85) z wykorzystaniem algorytmu ewolucyjnego



Rysunek 11 Przebieg optymalnego stanu procesu $\hat{x}(t)$, który – z optymalnym okresem $\hat{\tau} = 2.55$ – zapewnia wskaźnik jakości $\hat{Q} = -0.0968$

W przypadku Algorytmu 1a w wersji z krzyżowaniem arytmetycznym i mutacją równomierną, uzyskane wyniki były nieznacznie gorsze od tych przedstawionych powyżej, jednak nawet one poprawiały rozwiązanie Baileya, Horna i Lina. Średni wskaźnik jakości uzyskany po pięciu próbach z wykorzystaniem tej wersji algorytmu wynosił $\hat{Q}_s = -0.0936$, natomiast najlepsze wyznaczone rozwiązanie okresowe dawało wskaźnik jakości $\hat{Q} = -0.0939$. Należy tutaj zaznaczyć, że wyniki uzyskane przez Algorytm

1a (w którym zastosowane było krzyżowanie arytmetyczne oraz mutacja równomierna) otrzymano po 10000 pokoleń.



Rysunek 12 Cykl, dla którego wskaźnik jakości $\hat{Q} = -0.0968$

Przykład 3.

Ostatnim przykładem obrazującym rozwiązywanie problemu globalnie optymalnego sterowania okresowego będzie ciągły proces produkcyjny (analizowany wcześniej przez Watanabe [138]), w którym zachodzi pojedyncza, nieizotermiczna reakcja chemiczna $A \to B$.

Zminimalizuj wskaźnik jakości:

$$Q(x,u,\tau) = \frac{0.1}{\tau}\int_0^\tau \frac{u_2(t)}{25-u_2(t)}dt - \frac{1}{\tau}\int_0^\tau u_1(t)x_2(t)dt \tag{86}$$

procesu opisanego równaniami stanu

$$\dot{x}_1(t) = u_1(t) - u_1(t)x_1(t) - 10^{23}e^{-128/x_3(t)}x_1(t), \tag{87}$$

$$\dot{x}_2(t) = -u_1(t)x_2(t) + 10^{23}e^{-128/x_3(t)}x_1(t), \tag{88}$$

$$\dot{x}_3(t) = u_1(t)(1.1 - x_3(t)) + u_2(t)(2.5 - x_3(t)) + 10^{23}e^{-128/x_3(t)}x_1(t) \tag{89}$$

przy uwzględnieniu ograniczeń

$$x(0) = x(\tau), \tag{90}$$

$$\tau \in [0.1,4], \quad u_1(t) = 1, \quad 0 \le u_2(t) \le 20, \quad t \in [0,\tau]. \tag{91}$$

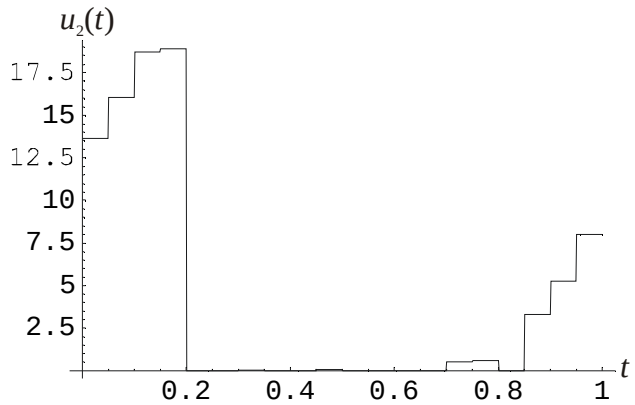
W problemie (86)-(91) $x_1(t)$ jest koncentracją surowca A w reaktorze w chwili t, $x_2(t)$ reprezentuje produkt użyteczny B reakcji w chwili t, $x_3(t)$ jest temperaturą re-

agującej mieszaniny w reaktorze w chwili t, $u_1(t)$ jest natężeniem dopływu surowca A do reaktora w chwili t, zaś $u_2(t)$ opisuje wymianę ciepła pomiędzy zawartością reaktora i zanurzonego w nim urządzenia grzewczego. Wskaźnik jakości powyższego problemu odpowiada maksymalizacji średniej zawartości produktu użytecznego B przy jednoczesnej minimalizacji kosztów poniesionych na chłodzenie reaktora.

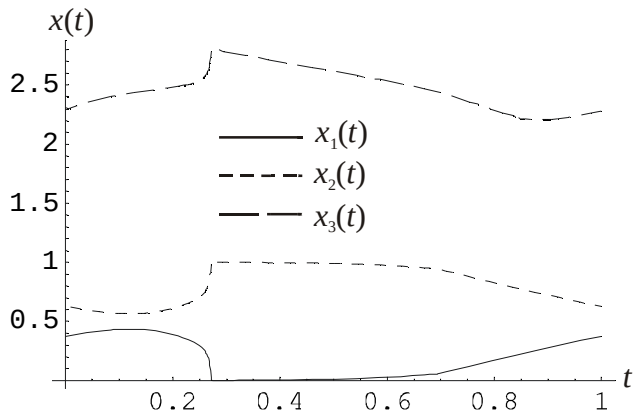
Analizowany przykład najpierw rozwiązany został z wykorzystaniem Algorytmu 1a, w którym został zastosowany arytmetyczny operator krzyżowania (35) oraz równomierny operator mutacji (38); przy czym przyjęto, że $N = 20$, $N_t = 2$, $K = 20$, $p_m = 0.15$. Po pięciu próbach wskaźnik jakości dla najlepszego rozwiązania wynosił $\hat{Q} = -0.6640$, natomiast wartość średnia wskaźnika jakości z pięciu prób była równa $\hat{Q}_s = -0.6510$. Niestety, w tym przypadku algorytm ewolucyjny okazał się gorszy od algorytmu gradientowego zastosowanego przez Watanabe. Watanabe uzyskał bowiem wskaźnik jakości $Q = -0.7433$.

Po zmianie operatora mutacji na operator równomierny (39) oraz po zmniejszeniu prawdopodobieństwa mutacji ($p_m = 0.015$), wyniki uzyskane przez algorytm ewolucyjny poprawiły się. Średni wskaźnik jakości z pięciu prób wyniósł $\hat{Q}_s = -0.7590$, natomiast rozwiązanie najlepsze – przedstawione na rysunkach 13-15 – dostarczyło wskaźnik jakości równy $\hat{Q} = -0.771062$. Rozwiązanie optymalne uzyskane przez Watanabe algorytmem gradientowym zostało zatem poprawione o prawie 4%.

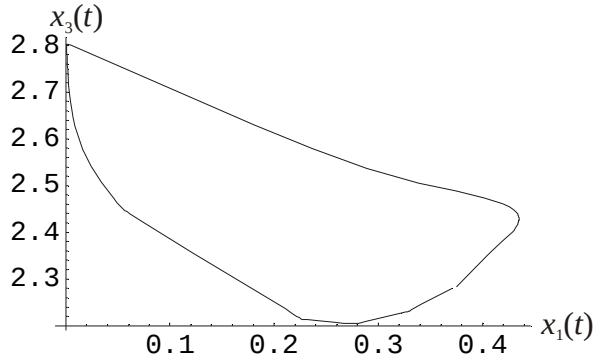
Wszystkie wyniki dla zadania (86)-(91) zostały uzyskane po 10000 pokoleń.



Rysunek 13 Optymalne sterowanie okresowe $\hat{u}(t)$ wyznaczone dla zadania (86)-(91) z wykorzystaniem algorytmu ewolucyjnego



Rysunek 14 Przebieg optymalnego stanu procesu $\hat{x}(t)$, który – z optymalnym okresem $\hat{\tau} = 1.32$ – zapewnia wskaźnik jakości $\hat{Q} = -0.771062$



Rysunek 15 Cykl, dla którego wskaźnik jakości $\hat{Q} = -0.771062$

3.3. Algorytm optymalizacji dla problemu GOSO z ograniczeniami zasobowymi

3.3.1. Problem GOSO z chwilowymi i uśrednionymi ograniczeniami zasobowymi

Rozpatrywany w rozdziale 3.2 problem okresowy z ograniczeniami zasobowymi w rzeczywistości występuje stosunkowo rzadko. Znacznie częściej można się spotkać z problemami okresowymi z chwilowymi i uśrednionymi ograniczeniami zasobowymi; właśnie tego rodzaju problemy i algorytmy ewolucyjne służące do ich rozwiązywania będą przedmiotem rozważań przedstawionych w dalszej części tego rozdziału.

Znormalizowany i zdyskretyzowany problem GOSO z chwilowymi i uśrednionymi ograniczeniami zasobowymi polega na minimalizacji wskaźnika jakości

$$Q(\tilde{z}) = \nu(q), \tag{92}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t,z), \tilde{u}(t))dt\,, \tag{93}$$

$$\dot{x}(t,z) = \tau\ f(x(t,z),\tilde{u}(t)), \quad t \in [0,1], \tag{94}$$

$$x(0) - x(1,z) = 0, \tag{95}$$

$$\tau \in \mathcal{T}, \quad x(0) \in X\ , \quad u_k \in U, \quad (k = 0,1,...,K-1), \tag{96}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0. \tag{97}$$

3.3.2. Ewolucyjny algorytm optymalizacji (Algorytm 2)

Problem globalnie optymalnego sterowania okresowego z ograniczeniami chwilowymi i uśrednionymi narzuconymi na sterowanie można rozwiązać (a raczej spróbować rozwiązać) przy pomocy Algorytmu 1 przedstawionego w rozdziale 3.2.2; aby wspomniany algorytm – dla omawianego problemu – zadziałał, muszą być jednak spełnione następujące warunki:

1) ograniczenia uśrednione (97) muszą być opisane funkcjami wypukłymi,
2) w algorytmie 1 nie można używać operatora mutacji,
3) jako operator krzyżowania musi być zastosowany operator arytmetyczny (35) lub jednopunktowy operator wymieniający (33) (który nie narusza ograniczeń wypukłych).

Rozwiązania dostarczane przez tak „zmieniony” Algorytm 1 będą wysoce niezadowalające, co wynika przed wszystkim z faktu, iż osobniki nie są w tym przypadku poddawane działaniu mutacji. Konsekwencją braku mutacji jest również to, że wszystkie rozwiązania będące wynikiem działania Algorytmu 1 zależą od populacji wygenerowanej w kroku inicjalizacji; stosując bowiem wyłącznie operator krzyżowania, algorytm ewolucyjny może przeszukiwać jedynie przestrzeń utworzoną przez populację początkową (na przykład, gdy wygenerowana zostanie populacja N osobników podlegająca ograniczeniu chwilowemu $z_i^{\pm} \in [0,2]$ $(i = 1,2,...,m)$, w której gen z_i^j $(1 \le j \le N)$ przyjmuje wartości z przedziału [0.5,1.5], wówczas algorytm ewolucyjny – w omawianej postaci – nigdy nie wygeneruje osobnika, którego gen z_j^i przyjmie wartości spoza przedziału [0.5,1.5]).

Wszystko to powoduje, iż koniecznym jest opracowanie metody, która nie eliminowałaby z algorytmu ewolucyjnego operatora mutacji i dokładnie uwzględniałaby ograniczenia uśrednione (97) narzucone na sterowanie. Aby tego dokonać, zakłada się, że ograniczenia uśrednione (97) mają postać:

$$\int_0^1 u_i(t)dt = u_{s_i}, \quad (i = 1,2,...,n_u), \tag{98}$$

która dla sterowania dyskretnego u_K (58) przyjmuje równoważną formę

$$\frac{1}{K}\sum_{k=0}^{K-1} u_{i,k} = u_{s_i}, \tag{99}$$

gdzie $u_s \in R^{n_u}$ jest zadanym wektorem opisującym średnią dostępność surowca potrzebnego do prowadzenia procesu.

Zasadność przyjętej postaci ograniczeń uśrednionych (98) potwierdza fakt, iż takie właśnie ograniczenia są typowe dla większości problemów okresowych. Ponadto zapis (98) ułatwia omówienie metody uwzględniania ograniczeń uśrednionych w algorytmach ewolucyjnych - metody, która może być stosunkowo łatwo zaadaptowana do innego typu ograniczeń uśrednionych.

Ograniczenia uśrednione problemu (92)-(97) w postaci dyskretnej (99) najlepiej jest uwzględnić, przy poszukiwaniu globalnie optymalnego rozwiązania (z wykorzystaniem algorytmu ewolucyjnego), za pomocą następującego algorytmu rekonstrukcji:

Algorytm rekonstrukcji ograniczeń

Krok 1 *(inicjalizacja)* Wybierz sterowanie u_i $(i = 1,2,...,n_u)$, które narusza ograniczenia (99) oraz utwórz pusty zbiór wykluczeń $W \subset D = \{0,1,...,K-1\}$.

Krok 2 *(ustalenie elementu do naprawy)* W sposób losowy (z równomiernym rozkładem prawdopodobieństwa) wybierz sterowanie $u_{i,j}$ $(j \in D \setminus W)$, które zostanie zmodyfikowane.

Krok 3 *(naprawa)* Wyznacz nową wartość sterowania $u_{i,j}$ według zależności:

$$u_{i,j} = Sat(Ku_{s_i} - \sum_{k \in D \setminus \{j\}} u_{i,k}, u_i^-, u_i^+), \tag{100}$$

gdzie funkcja *Sat* jest zdefiniowana jak w (36).

Krok 4 *(modyfikacja zbioru wykluczeń)* Podstaw $W = W \cup \{j\}$.

Krok 5 *(kryterium stopu)* Jeżeli spełnione jest ograniczenie (99), zatrzymaj algorytm naprawy, w przeciwnym przypadku idź do kroku 2.

Mając powyższy algorytm rekonstrukcji ograniczeń, ewolucyjny algorytm optymalizacji dla problemu (92)-(97) z uśrednionymi ograniczeniami zasobowymi (98) może być sformułowany następująco:

Algorytm 2a (algorytm z turniejową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; $N_t < N$ – ilość osobników biorących udział w turnieju; K – poziom dyskretyzacji sterowania; ε, g_p – parametry kryterium stopu, badającego osiągnięcie przez algorytm minimalnej szybkości poprawy (49); $G_{\max}$ – maksymalna liczba pokoleń; $p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (96) (tzn. by $z \in Z$). Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Na osobnikach naruszających ograniczenia uśrednione (97) zastosuj Algorytm rekonstrukcji ograniczeń. Następnie, stosując metodę Newtona (73) lub Gaussa-Newtona (74), doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (95) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu (przystosowania) $Q(z^i)$ $(i = 1,2,..., N)$. Wśród osobników spełniających ograniczenia (95)- (97) znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q oraz osobnika $z^{\max}$ z największą wartością funkcji celu Q.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$ to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Powtarzaj poniższe czynności do momentu, aż bieżąca populacja P^g będzie pusta:

- wylosuj z bieżącej populacji N_t osobników;
- weź z N_t osobników do populacji tymczasowej P_{tmp}^g osobnika z najmniejszą wartością funkcji celu Q;
- usuń z bieżącej populacji N_t osobników.

Krok 7 (krzyżowanie) Skopiuj do nowej populacji P^{g+1} osobniki z populacji tymczasowej P_{tmp}^g. Następnie, powtarzaj poniższe czynności tak długo, aż nowa populacja P^{g+1} będzie składać się z N osobników:

- wylosuj dwa osobniki z populacji tymczasowej P_{tmp}^g i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania, natomiast krzyżowane osobniki zwróć do populacji P_{tmp}^g.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w populacji działaniu tego operatora – przy czym otrzymanymi nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano operator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \le p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

Algorytm ewolucyjny dla problemu (92)-(97) z ograniczeniami uśrednionymi w postaci (98), wykorzystujący jako metodę reprodukcji metodę ruletkową 2.4.1 lub metodę rangową 2.4.3, może być zdefiniowany następująco:

Algorytm 2b (algorytm z ruletkową lub rangową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; K – poziom dyskretyzacji sterowania; ε, g_p - parametry kryterium stopu, badającego osiągnięcie przez algorytm minimalnej szybkości poprawy (49); $G_{\max}$ – maksymalna liczba pokoleń; $p_c \in (0,1]$ – prawdopodobieństwo krzyżowania oraz

$p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Spośród metod reprodukcji (metody ruletkowej i metod rangowych) wybierz metodę, według której będziesz wyznaczać prawdopodobieństwo przeżycia osobnika. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (96) (tzn. by $z \in Z$). Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Na osobnikach naruszających ograniczenia uśrednione (97) zastosuj Algorytm rekonstrukcji ograniczeń. Następnie, stosując metodę Newtona (73) lub Gaussa-Newtona (74), doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (95) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu (przystosowania) $Q(z^i)$ $(i = 1,2,...,N)$. Wśród osobników spełniających ograniczenia (95)-(97), znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q oraz osobnika $z^{\max}$ z największą wartością funkcji celu Q.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$, to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Stosując wybraną metodę reprodukcji wyznacz prawdopodobieństwo przeżycia (wylosowania) $p_w^i(z^i)$ $(i = 1,2,...,N)$ dla każdego osobnika w populacji, a następnie – dla każdego osobnika z^i $(i = 1,2,...,N)$ – wyznacz dystrybuantę $d_i = \sum_{j=1}^{i} p_w^j$ $(i = 1,2,...,N)$. Powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^g będzie zawierać N osobników:

- wylosuj z równomiernym rozkładem prawdopodobieństwa liczbę $p \in [0,1]$;

- wstaw do populacji tymczasowej P_{tmp}^{g} kopię osobnika z^i $(i = 1,2,...,N)$, dla którego prawdziwa jest nierówność $d_{i-1} < p \le d_i$.

Krok 7 (krzyżowanie) Dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji tymczasowej wylosuj $p_i \in [0,1]$. Przenieś do nowej populacji P^{g+1} te osobniki z populacji tymczasowej P_{tmp}^{g}, dla których $p_i > p_c$. Jeżeli populacja P_{tmp}^{g} zawiera nieparzystą liczbę osobników, to losowo wybierz z niej jednego osobnika i przenieś go do populacji P^{g+1}. Następnie, powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^{g} będzie pusta:

- wylosuj dwa osobniki z populacji tymczasowej P_{tmp}^{g} i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania;
- usuń krzyżowane osobniki z populacji P_{tmp}^{g}.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w populacji działaniu tego operatora – przy czym otrzymanymi nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano operator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \le p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

3.3.3. Przykłady obliczeniowe

Przykład 4.

Dla sprawdzenia skuteczności działania – omówionego w punkcie 3.3.2 – algorytmu ewolucyjnego, rozwiązane zostało następujące zadanie: zminimalizuj wskaźnik jakości

$$Q(x,u,\tau) = \frac{1}{\tau}\int_0^{\tau}(x_1(t) + x_2(t))dt \tag{101}$$

procesu opisanego równaniami stanu

$$\dot{x}_i(t) = u_i(t) - x_i(t) - 2x_1^2(t)x_2(t) \quad (i = 1,2) \tag{102}$$

przy uwzględnieniu ograniczeń

$$x(0) = x(\tau), \tag{103}$$

$$\tau = 6, \quad 0 \le u_i(t) \le 2, \quad t \in [0,\tau], \quad (i = 1,2) \tag{104}$$

$$\frac{1}{\tau}\int_0^\tau u_i(t)dt = 1 \quad (i = 1,2)\,. \tag{105}$$

Powyższe zadanie opisuje ciągły proces produkcyjny polegający na izotermicznej reakcji syntezy $A_1 + A_2 \to B$, w której $x_i(t)$ $(i = 1,2)$ jest koncentracją składnika A_i w reaktorze w chwili t, natomiast $u_i(t)$ $(i = 1,2)$ oznacza koncentrację składnika A_i w strumieniu wejściowym. Wskaźnik jakości problemu (102)-(105) odpowiada maksymalizacji produktu użytecznego B.

Rozpatrywane zadanie zostało wcześniej rozwiązane z wykorzystaniem metody λ-rzutowanego gradientu przez Skowrona i Stycznia [118]. Autorzy ci wyznaczyli optymalne sterowanie statyczne badanego problemu $\hat{\bar{u}} = (1,1)$ oraz odpowiadający mu optymalny stan statyczny $\hat{\bar{x}} = (0.5897, 0.5897)$, które zapewniają wskaźnik jakości $\hat{\bar{Q}}(\hat{\bar{x}}, \hat{\bar{u}}) = 1.1794$. Wykazali oni ponadto, że optymalne rozwiązanie statyczne może być poprawione za pomocą okresowego prowadzenia procesu – wykorzystanie metody rzutowanego gradientu pozwala bowiem na znalezienie optymalnego sterowania okresowego, dla którego wskaźnik jakości jest równy $Q = 0.9522$.

W niniejszym opracowaniu zadanie (102)-(105) zostało rozwiązane zarówno z wykorzystaniem Algorytmu 2a jak i Algorytmu 2b (w którym zastosowano rangową metodę reprodukcji (29)). Obydwa algorytmy były testowane dla następujących parametrów: $G_{\max} = 2000$, $\varepsilon = 0$, $g_p = 2000$, $N = 20$, $N_t = 2$, $K = 20$, $p_c = 0.5$. Wyniki uzyskane po 2000 pokoleń z pięciu niezależnych prób zostały przedstawione w tabeli 2; natomiast ich średnia wartość została pokazana na rysunku 16. K-(34) oraz K-(35) w tabeli 2 i na rysunku 16 oznaczają, że wyniki zostały uzyskane odpowiednio dla: równomiernego (34) i arytmetycznego (35) operatora krzyżowania. W analogiczny sposób zostały oznaczone użyte operatory mutacji; M-(37), M-(38), M-(39) oznaczają równomierne operatory mutacji (37), (38), (39), natomiast M-(40), M-(42), M-(43) – operatory nie-

równomierne (40), (42), (43). Istotnym jest, że prawdopodobieństwo mutacji w przypadku operatorów (39) i (43) było równe $p_m = 0.015$, natomiast dla pozostałych operatorów wynosiło ono $p_m = 0.15$.

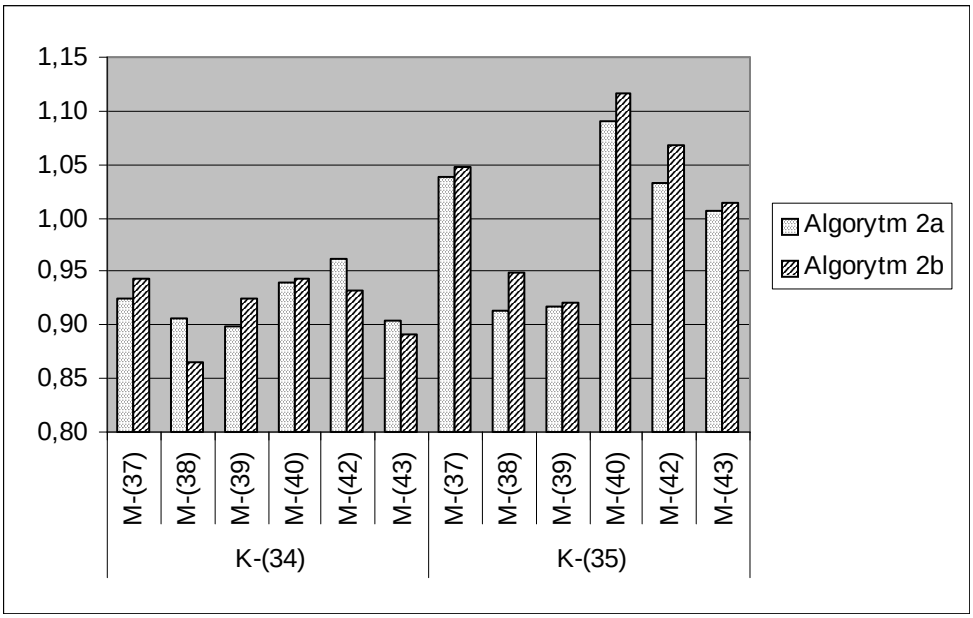
Tabela 2 Wyniki uzyskane po 2000 pokoleń dla problemu (102)-(105).

Algorytm 2a							
Próba		**1**	**2**	**3**	**4**	**5**	**Wynik średni**
K-(34)	**M-(37)**	0,915194	0,935040	0,947281	0,904435	0,924425	0,925275
	M-(38)	0,879506	0,892232	0,909211	0,899921	0,950377	0,906249
	M-(39)	0,889840	0,906150	0,903260	0,889973	0,900977	0,898040
	M-(40)	0,967567	0,970835	0,920236	0,924683	0,916358	0,939936
	M-(42)	1,046252	0,962640	0,893240	0,975046	0,932850	0,962006
	M-(43)	0,904147	0,890923	0,932908	0,880055	0,912906	0,904188
K-(35)	**M-(37)**	1,051270	1,022210	1,048789	1,033716	1,031229	1,037443
	M-(38)	1,033470	0,876604	0,859917	0,918357	0,879759	0,913621
	M-(39)	0,954732	0,880411	0,951453	0,900526	0,895683	0,916561
	M-(40)	1,106894	1,079968	1,076409	1,100913	1,084611	1,089759
	M-(42)	1,043577	0,990470	1,068756	1,014508	1,042967	1,032055
	M-(43)	1,041608	1,005432	0,970868	0,982511	1,033701	1,006824
Algorytm 2b (rangowa metoda reprodukcji (29))							
Próba		**1**	**2**	**3**	**4**	**5**	**Wynik średni**
K-(34)	**M-(37)**	0,934585	0,971992	0,928448	0,900355	0,982083	0,943493
	M-(38)	0,882989	0,876813	**0,802253**	0,880519	0,878976	0,864310
	M-(39)	0,923533	0,933743	0,925459	0,914766	0,929699	0,925440
	M-(40)	0,943353	0,919026	0,929517	0,969508	0,953041	0,942773
	M-(42)	0,951193	0,897730	1,029865	0,885283	0,893681	0,931550
	M-(43)	0,832605	0,914850	0,913029	0,876968	0,920086	0,891508
K-(35)	**M-(37)**	1,070090	1,066240	1,034460	1,038135	1,024675	1,046720
	M-(38)	0,953336	0,937804	0,956829	0,940757	0,954384	0,948622
	M-(39)	0,904097	0,932641	0,909894	0,913872	0,943994	0,920899
	M-(40)	1,126904	1,111774	1,107768	1,117446	1,116944	1,116167
	M-(42)	1,052039	1,055860	1,100595	1,047770	1,067942	1,068042
	M-(43)	1,006210	1,006633	1,011195	1,014775	1,033001	1,014363

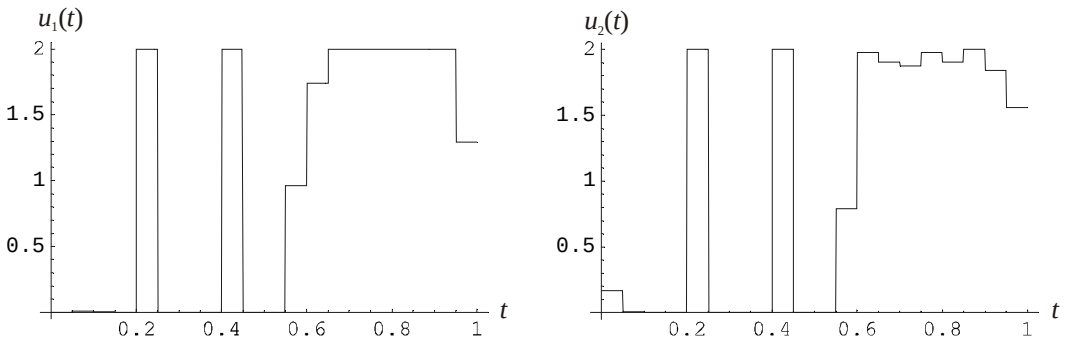
W przeciwieństwie do wyniku uzyskanego w przykładzie 1, tym razem skuteczniejszy okazał się algorytm z turniejową metodą reprodukcji. W 9 (na 12) przypadkach średnie rozwiązania dostarczone przez Algorytm 2a okazały się bowiem lepsze od rozwiązań wyznaczonych przez Algorytm 2b. W przypadku problemu (102)-(105) okazało się również, że krzyżowanie arytmetyczne dostarcza o wiele gorsze rozwiązania od krzyżowania równomiernego (oczywiście nie biorąc pod uwagę wyników uzyskanych dla operatorów mutacji (38), (39)). Wyniki uzyskane dla tego operatora były gorsze nawet od wyniku uzyskanego metodą λ-rzutowanego gradientu. Porównując wyniki uzyskane dla różnych operatorów mutacji można zauważyć, że – w przypadku krzyżo-

wania równomiernego – otrzymane rezultaty były do siebie zbliżone. Natomiast analizując wyniki dla krzyżowania arytmetycznego widać, że (podobnie jak miało to miejsce przy zadaniu z przykładu 1) zdecydowanie najgorsze wyniki zostały uzyskane dla nierównomiernego operatora mutacji.

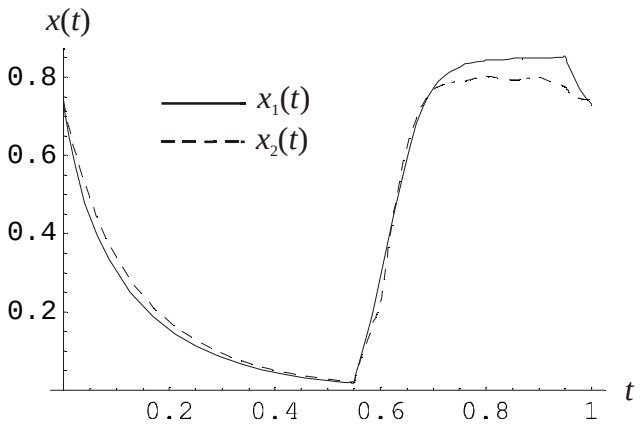
Jak widać z wyników zamieszczonych w z tabeli 1, najlepsze rozwiązanie, które zostało przedstawione na rysunkach 17-19, zapewniło wskaźnik jakości $\hat{Q} = 0.802253$. Rozwiązanie to poprawia rozwiązanie okresowe wyznaczone algorytmem gradientowym [118] o blisko 16%, a optymalne rozwiązanie statyczne – o 32%.



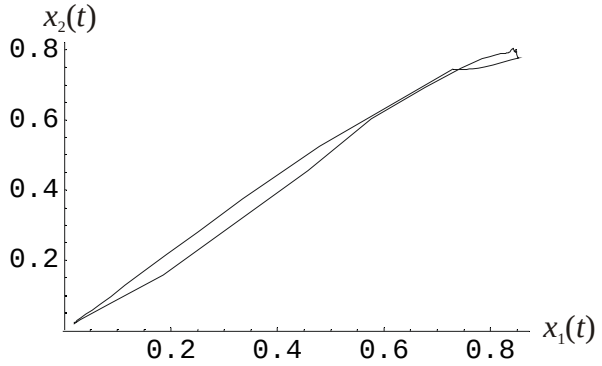
Rysunek 16 Średnie wyniki z pięciu prób uzyskane przy wykorzystaniu Algorytmu 2 po 2000 pokoleń dla problemu (102)-(105)



Rysunek 17 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające najmniejszą wartość wskaźnika jakości dla zadania (102)-(105)



Rysunek 18 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 17, który zapewnia wskaźnik jakości $\hat{Q} = 0.802253$



Rysunek 19 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = 0.802253$

Przykład 5.

Ewolucyjny Algorytm 2b przetestowany został także na problemie optymalizacji ciągłego procesu okresowego – przemiany $A \to B$ – prowadzonego w kaskadzie nie-izotermicznych reaktorów, w którym występują energetyczne ograniczenia uśrednione. Problem ten polegał na minimalizacji wskaźnika jakości

$$Q(x,u,\tau) = \tfrac{1}{\tau}\int_0^{\tau} x_2(t)dt \tag{106}$$

procesu opisanego równaniami stanu

$$\dot{x}_1(t) = 1 - x_1(t) - 100e^{-5/u_1(t)}x_1^2(t), \tag{107}$$

$$\dot{x}_2(t) = x_1(t) - x_2(t) - 100e^{-5/u_2(t)}x_2^2(t), \qquad (108$$

przy uwzględnieniu ograniczeń

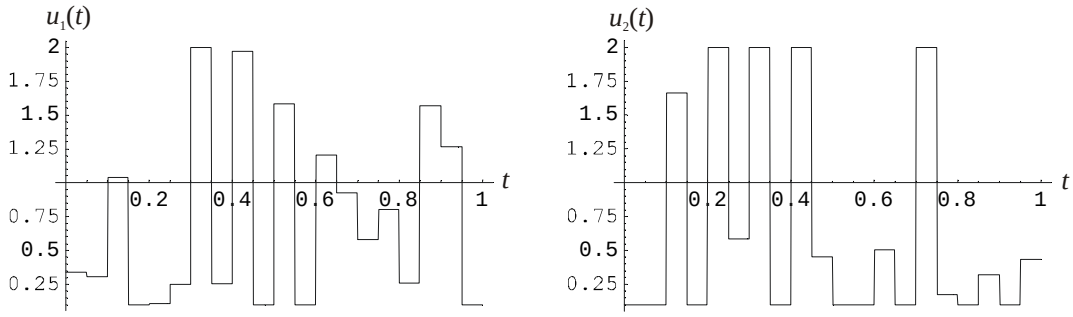
$$x(0) = x(\tau), \tag{109}$$

$$\tau = 0.314, \quad 0.1 \le u_i(t) \le 2, \quad t \in [0, \tau], \quad (i = 1,2) \tag{110}$$

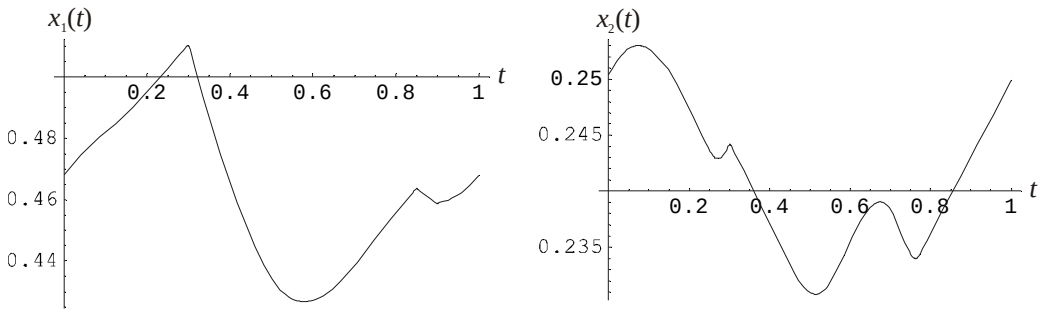
$$\frac{1}{\tau}\int_0^{\tau} u_i^2(t)dt \le 2 \quad (i = 1,2)\,. \tag{111}$$

W powyższym problemie $x_i(t)$ $(i = 1,2)$ jest stężeniem substancji surowcowej w *i*–tym reaktorze, zaś $u_i(t)$ $(i = 1,2)$ jest napięciem obwodu grzejnego *i*–tego reaktora. Wskaźnik jakości w tym przypadku odpowiada natomiast minimalizacji średniego stężenia nieprzereagowanego surowca na wyjściu drugiego reaktora.

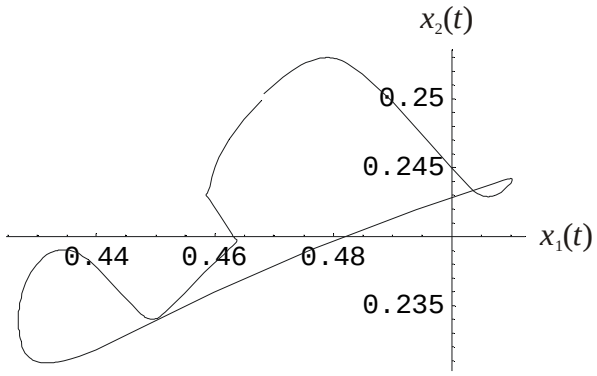
Powyższy problem został rozwiązany z wykorzystaniem Algorytmu 2b (przy zastosowaniu reprodukcji rangowej (29), krzyżowania równomiernego (34) oraz mutacji równomiernej (39)) dla następujących parametrów: $G_{\max} = 1000$, $\varepsilon = 0$, $g_p = 1000$, $N = 20$, $p_c = 0.5$, $p_m = 0.015$. Najlepsze rozwiązanie (z pięciu prób), które znalazł algorytm ewolucyjny w 1000 pokoleniu przedstawione zostało na rysunkach 20-22. Wartość wskaźnika jakości dla tego rozwiązania wynosi $\hat{Q} = 0.241378$ i poprawia rozwiązanie statyczne $\overline{\hat{Q}} = 0.5094$ o blisko 53%. Znalezione za pomocą algorytmu ewolucyjnego rozwiązanie jest również o 25% lepsze od rozwiązania tego samego problemu wyznaczonego przy wykorzystaniu gradientowego algorytmu lokalnej optymalizacji, które wynosi $Q = 0.3234$ [115]. Z kolei wartość średnia wskaźnika jakości z pięciu niezależnych prób jest równa $\hat{Q} = 0.2517$ i również poprawia rozwiązanie dostarczone algorytmem lokalnej optymalizacji.



Rysunek 20 Sterowanie okresowe *u*(*t*) wyznaczone algorytmem ewolucyjnym zapewniające najmniejszą wartość wskaźnika jakości dla zadania (106)-(111)



Rysunek 21 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 20, który zapewnia wskaźnik jakości $\hat{Q} = 0.241378$



Rysunek 22 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = 0.241378$

Przykład 6.

Skuteczność Algorytmu 2a potwierdza przykład problemu GOSO dla ciągłego procesu produkcyjnego, którego istotą jest równoległa reakcja chemiczna $A \to B,\ A \to C$ zachodząca w reaktorze z wymieszaniem. Równania stanu opisujące ten proces są następujące:

$$\dot{x}_1(t) = -u_2(t)x_1^2(t) - 400 * 10^{-2.2} u_2^{0.55}(t)x_1(t) + u_1(t)(1 - x_1(t)), \tag{112}$$

$$\dot{x}_2(t) = u_2(t)x_1^2(t) - u_1(t)x_2(t), \tag{113}$$

gdzie: $x_1(t)$ – stężenie składnika A w reaktorze, $x_2(t)$ – stężenie składnika B w reaktorze, $u_1(t)$ – natężenie przepływu mieszaniny przez reaktor, $u_2(t)$ – temperatura reaktora. W opisanym wyżej problemie należy znaleźć takie sterowanie okresowe minimalizujące wskaźnik jakości

$$Q = -\frac{1}{\tau}\int_0^{\tau} u_1(t)x(t)dt, \tag{114}$$

dla którego spełnione będą następujące ograniczenia:

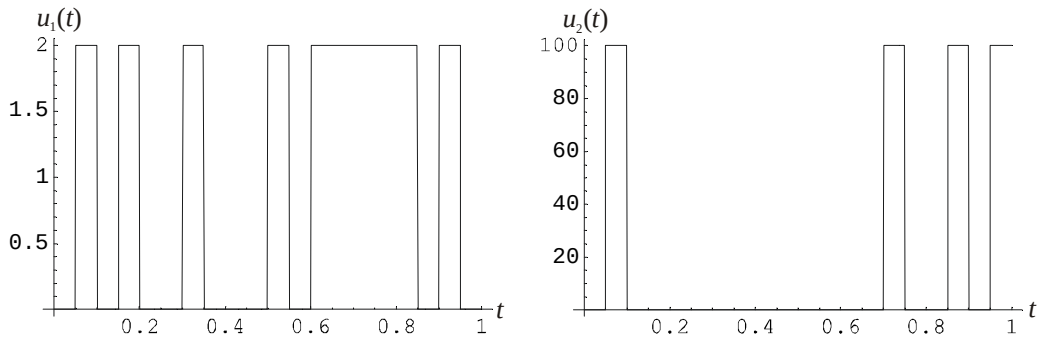
$$x(0) = x(\tau), \tag{115}$$

$$\tau \in [0.4,1.2], \quad 0 \le u_1(t) \le 2, \quad 0.1 \le u_2(t) \le 100, \quad t \in [0,\tau]), \tag{116}$$

$$\frac{1}{\tau}\int_0^{\tau} u_1(t)dt = 1. \tag{117}$$

Powyższy problem został rozwiązany z wykorzystaniem dwóch wersji Algorytmu 2a. W pierwszej – użyty był wymieniający operator krzyżowania (32) i mutacja brzegowa (46), natomiast w drugiej wersji zastosowano równomierny operator krzyżowania (34) i równomierny operator mutacji (39). Obydwie wersje algorytmu zostały uruchomione dla następujących parametrów: $N = 20$, $N_t = 2$, $K = 20$; przy czym algorytm z mutacją brzegową został zatrzymany po 500 pokoleniach, zaś drugi algorytm – po 5000 pokoleniach.

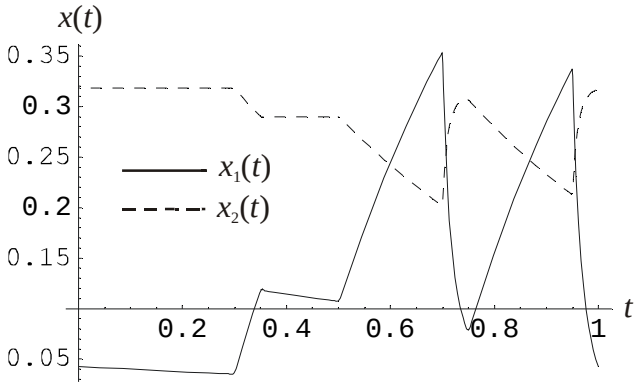
Najlepsze rozwiązanie, które przedstawione jest na rysunkach 23-25, zostało wyznaczone przy pomocy algorytmu korzystającego z krzyżowania wymieniającego i z mutacji brzegowej. Rozwiązanie to zapewnia wskaźnik jakości $\hat{Q} = -0.272907$, który jest o ponad 21% lepszy od rozwiązania wyznaczonego przez Fjelda metodą gradientową (zapewniającego wskaźnik jakości $Q = -0.225$ [49]).



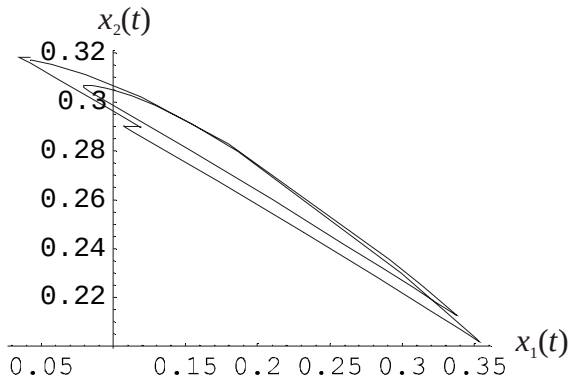
Rysunek 23 Sterowanie okresowe $u(t)$, wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 0.93$ najmniejszą wartość wskaźnika jakości dla problemu (112)-(117)

Druga wersja algorytmu dostarczyła nieznacznie gorsze rozwiązanie, które zapewniło wskaźnik jakości równy $Q = -0.268888$. Rozwiązanie to uzyskane zostało jednak dopiero w 5000 pokoleniu. W pokoleniu 500, najlepszy osobnik w algorytmie z krzyżowaniem i mutacja równomierną zapewniał wskaźnik jakości równy $Q = -0.241843$.

Jak widać, zastosowanie operatora mutacji brzegowej znacznie przyspiesza proces wyznaczania rozwiązania optymalnego.



Rysunek 24 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 23, który zapewnia wskaźnik jakości $\hat{Q} = -0.272907$



Rysunek 25 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = -0.272907$

Przykład 7.

Algorytm 2b został przetestowany również na ciągłym procesie produkcyjnym (rozpatrywanym przez Watanabe, Kurimoto, Matsubare i Onogi [138]), w którym zachodzi równoległa reakcja $A \to B$, $A \to C$.

Zminimalizuj wskaźnik jakości:

$$Q = \frac{0.4}{\tau}\int_0^{\tau} u_1(t)x_3(t)dt - \frac{1}{\tau}\int_0^{\tau} u_1(t)x_2(t)dt \tag{118}$$

przy uwzględnieniu ograniczeń

$$\dot{x}_1(t) = u_1(t) - u_1(t)x_1(t) - 10^{12} e^{-1000/x_4(t)} x_1^2(t) - 10^9 e^{-750/x_4(t)} x_1(t), \tag{119}$$

$$\dot{x}_2(t) = -u_1(t)x_2(t) + 10^{12} e^{-1000/x_4(t)} x_1^2(t), \tag{120}$$

$$\dot{x}_3(t) = -u_1(t)x_3(t) + 10^9 e^{-750/x_4(t)} x_1(t), \tag{121}$$

$$\begin{aligned}\dot{x}_4(t) &= u_1(t)(30 - x_4(t)) + u_2(t)(45 - x_4(t)) \\ &+ 10^{12} e^{-1000/x_4(t)} x_1^2(t) + 0.5 * 10^9 e^{-750/x_4(t)} x_1(t),\end{aligned} \tag{122}$$

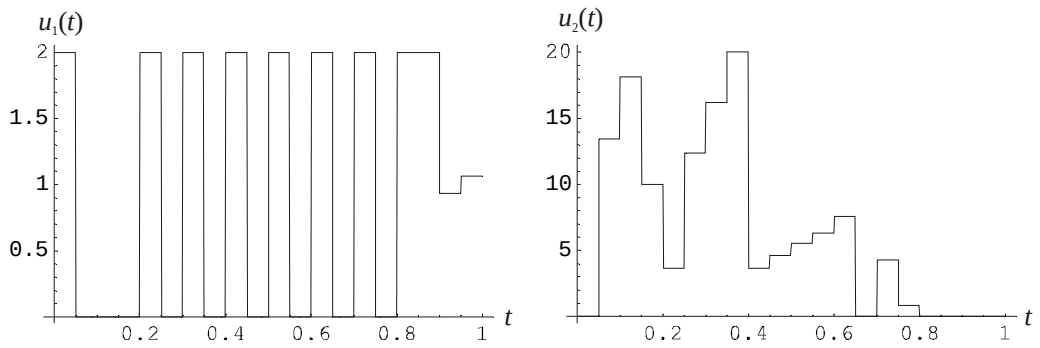
$$x_i(0) = x_i(\tau), \quad (i = 1,2,3,4), \tag{123}$$

$$\tau \in [0.1,10], \quad 0 \le u_1(t) \le 2, \quad 0 \le u_2(t) \le 20, \quad t \in [0,\tau], \tag{124}$$

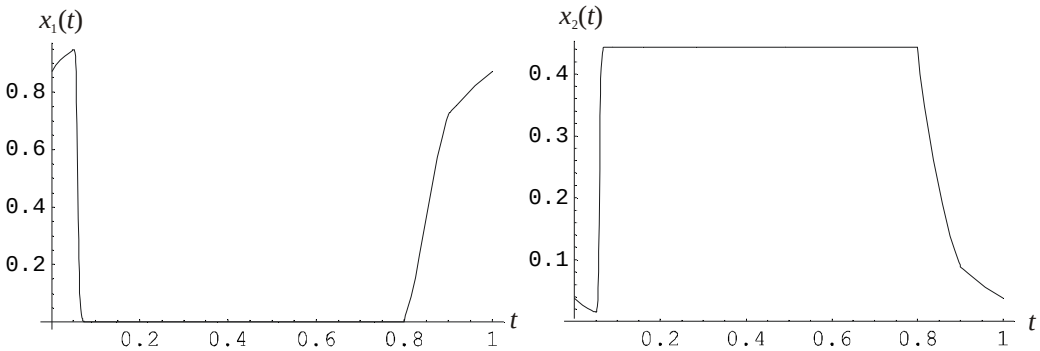
$$\frac{1}{\tau}\int_0^\tau u_1 dt = 1. \tag{125}$$

W problemie (118)-(125) $x_1(t)$ jest koncentracją surowca A w reaktorze w chwili t, $x_2(t)$ reprezentuje produkt użyteczny B reakcji w chwili t, $x_3(t)$ jest produktem ubocznym C reakcji w chwili t, $x_4(t)$ jest temperaturą reagującej mieszaniny w reaktorze w chwili t, $u_1(t)$ jest natężeniem dopływu składnika A do reaktora w chwili t, zaś $u_2(t)$ opisuje wymianę ciepła pomiędzy zawartością reaktora i zanurzonego w nim urządzenia grzewczego. Wskaźnik jakości powyższego problemu odpowiada maksymalizacji średniej zawartości produktu użytecznego B przy jednoczesnej minimalizacji produktu ubocznego C.

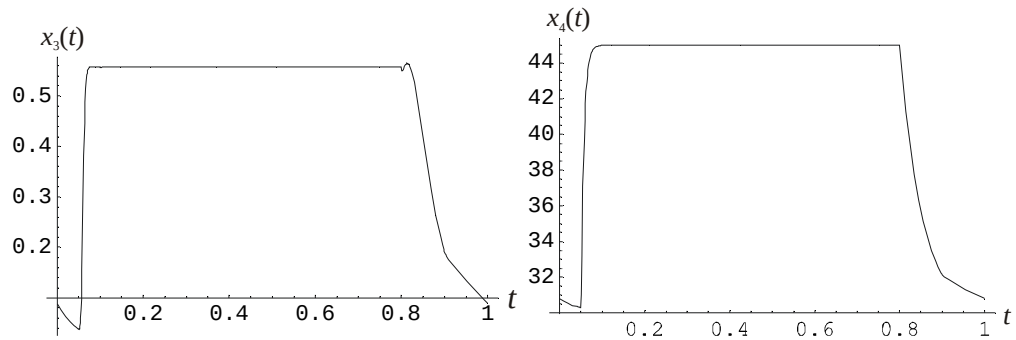
Problem (118)-(125) został rozwiązany Algorytmem 2b (z reprodukcją rangową (29), krzyżowaniem równomiernym (34) oraz mutacją równomierną (39)) dla następujących parametrów: $G_{\max} = 3000$, $\varepsilon = 0$, $g_p = 3000$, $N = 20$, $p_c = 0.5$, $p_m = 0.015$. Najlepsze otrzymane rozwiązanie, które zapewnia wskaźnik jakości $\hat{Q} = -0.145961$, przedstawione zostało na rysunkach 26-29. Rozwiązanie to poprawia wydajność procesu o 111% w porównaniu z rozwiązaniem dostarczonym przez algorytmem lokalnej optymalizacji [138] – zapewniającym wskaźnik jakości $Q = -0.0692$. Łatwo zauważyć, że – podobnie jak w innych przykładach – i tym razem algorytm ewolucyjny okazał się zdecydowanie lepszy od tradycyjnych metod optymalizacyjnych.



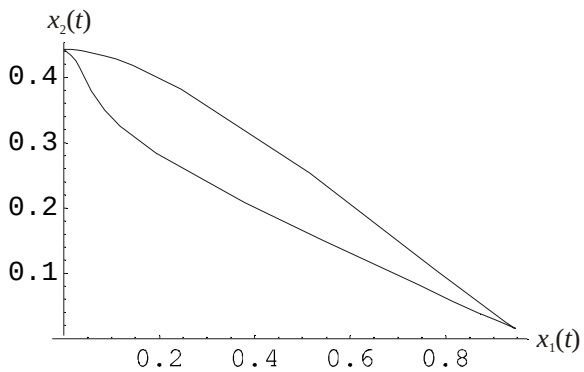
Rysunek 26 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 10$ najmniejszą wartość wskaźnika jakości dla problemu (118)-(125)



Rysunek 27 Przebieg stanu odpowiadający sterowaniu z rysunku 26, który zapewnia wskaźnik jakości $\hat{Q} = -0.145961$



Rysunek 28 Przebieg stanu odpowiadający sterowaniu z rysunku 26, który zapewnia wskaźnik jakości $\hat{Q} = -0.145961$



Rysunek 29 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = -0.145961$

3.4. Algorytm optymalizacji dla problemów GOSO z ograniczeniami zasobowymi, technologicznymi oraz stabilnościowymi

3.4.1. Problem GOSO z ograniczeniami zasobowymi, technologicznymi oraz stabilnościowymi

Problem globalnie optymalnego sterowania okresowego, w którym występują wszystkie ograniczenia (tzn. chwilowe i uśrednione ograniczenia zasobowe, chwilowe i uśrednione ograniczenia technologiczne oraz ograniczenia stabilnościowe) został już sformułowany we wstępie tej pracy, jednak gwoli przypomnienia, zostanie on ponownie przedstawiony w znormalizowanej, dyskretnej postaci.

Zminimalizuj wskaźnik jakości

$$Q(z) = \nu(q), \tag{126}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t,z), \tilde{u}(t))dt\,, \tag{127}$$

$$\dot{x}(t,z) = \tau\ f(x(t,z), \tilde{u}(t)), \quad t \in [0,1], \tag{128}$$

$$x(0) - x(1,z) = 0, \tag{129}$$

$$\tau \in \mathcal{T}, \quad u_k \in U \quad (k = 0,1,...,K-1), \quad x_l \in X \quad (l = 0,1,...,L-1), \tag{130}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \tag{131}$$

$$\phi(q) = 0, \quad \tilde{\phi}(q) \le 0, \tag{132}$$

$$\| s(\Phi(1,z)) \|_\infty \le \alpha. \tag{133}$$

gdzie $x_l \in R^{n_x}$ jest dyskretnym stanem procesu (zdefiniowanym za pomocą $L \ge K$ chwil czasowych) , którego wartość wyznacza na jest w sposób następujący: $x_l = x(\frac{l}{L})$ $(l = 0,1,...,L-1)$. Zmienna ta została wprowadzona do opisu problemu w celu aproksymacji ciągłych ograniczeń chwilowych $x(t) \in X$; łatwiej jest bowiem sprawdzić, czy ograniczenia nie są naruszone w konkretnych momentach niż w całym horyzoncie czasowym.

Wprowadzenie dyskretnego ograniczenia chwilowego na stan nie powoduje utraty informacji o tym, czy funkcja ciągła stanu przyjęła niedopuszczalne wartości. W przypadku problemu okresowego, w którym sterowanie jest aproksymowane sterowaniem dyskretnym u_K, przebieg stanu ma charakter segmentowy z ograniczoną pochodną (przyjmuje minimalne/maksymalne wartości w chwili czasowej $t = \frac{k}{K}$); tak więc, aby dowiedzieć się, czy ograniczenie $x(t) \in X$ jest spełnione, wystarczy sprawdzić – czy jest ono spełnione na dostatecznie gęstej siatce $t = \frac{l}{L}$ $(l = 0,1,...,L-1)$.

3.4.2. Ewolucyjny algorytm optymalizacji (Algorytm 3)

Ograniczenia w algorytmach ewolucyjnych można uwzględniać za pomocą [4], [26], [83]-[85]:

1) wyspecjalizowanych operatorów lub mechanizmów rekonstruujących osobniki naruszające ograniczenia;
2) metod wielokryterialnych;
3) funkcji kary.

Pierwszy – spośród wymienionych wyżej sposobów – stosowany jest przy rozwiązywaniu tych problemów okresowych, w których występują chwilowe i uśrednione ograniczenia narzucone na sterowanie (Algorytm rekonstrukcji ograniczeń oraz metoda Newtona-Raphsona). Jest to dosyć dobry sposób uwzględniania ograniczeń, gdyż (w odróżnieniu od metod wielokryterialnych, czy funkcji kary) pozwala on, by algorytm ewolucyjny działał zawsze na rozwiązaniach dopuszczalnych. Niestety, w zadaniu optymalizacji okresowej, w której występują chwilowe i uśrednione ograniczenia stanu oraz ograniczenia stabilnościowe, nie można zastosować tej metody, bowiem zaprojektowanie mechanizmu rekonstruującego ograniczenia jest (zgodnie z dzisiejszym stanem wiedzy) niemożliwe (głownie z powodu nieanalitycznego charakteru ograniczeń stabilnościowych).

Z uwagi na fakt, że optymalizacja procesów okresowych z wykorzystaniem algorytmów ewolucyjnych jest zadaniem skomplikowanym, do uwzględniania ograniczeń problemu (126)-(133) nie można również użyć metody wielokryterialnej. W algorytmach ewolucyjnych, wykorzystujących wspomnianą metodę uwzględniania ograniczeń, optymalizacja odbywa się według wielu wskaźników jakości (tworzonych przez funkcję celu problemu i ograniczenia [2], [27]); w porównaniu z optymalizacją problemu z jednym wskaźnikiem jakości taka optymalizacja jest bardzo trudna i czasochłonna, dlatego

nie wskazanym jest stosowanie metody wielokryterialnej do uwzględniania ograniczeń w problemach okresowych.

Dosyć powszechnie stosowana jest natomiast metoda kary. Najprostsza jej wersja polega na tym, że z populacji usuwany jest osobnik naruszający ograniczenia, a na jego miejsce wstawiany jest nowo-wygenerowany osobnik, który spełnia wszystkie narzucone ograniczenia (osobnik dopuszczalny). Mimo swojej prostoty ta wersja metody kary używana jest dość rzadko, bowiem jest ona obarczona poważnymi wadami. Jedną z nich jest np. to, że – w przypadku ograniczeń równościowych – wylosowanie osobnika dopuszczalnego jest niemalże niemożliwe.

W innej wersji metody kary (tzw. statycznej) wartość wskaźnika jakości zwiększana jest o $\sum_{i=1}^{m_o} \rho_i c_i(z)$ (gdzie ρ_i jest karą za przekroczenie ograniczenia, $c_i(z)$ jest miarą przekroczenia ograniczenia i przez osobnika z, natomiast m_o jest liczbą ograniczeń). Wadą tej metody jest problem doboru wartości ρ_i. Zbyt duża wartość kary powoduje, że algorytm może szybko wyznaczyć rozwiązanie, które niekoniecznie musi być rozwiązaniem globalnie optymalnym. Zbyt mała wartość ρ_i, z kolei, może być przyczyną tego, że w zadanym czasie algorytm nie znajdzie rozwiązania dopuszczalnego. Udoskonaloną wersją metody statycznej jest dynamiczna metoda kary zaproponowana przez Joinesa i Houcka [70], w której współczynnik kary zależny jest od liczby pokoleń oraz od trzech parametrów, których prawidłowy dobór może nadal niestety nastręczać pewne problemy.

Wad metody statycznej i dynamicznej nie jest również pozbawiona samoadaptująca się metoda kary zaproponowana przez Farmaniego i Wrighta [47]. Co prawda metoda ta nie wymaga ustalania aż tylu parametrów, bowiem dostraja się sama – korzystając z wartości funkcji przystosowania najlepszego i najgorszego osobnika w populacji oraz osobnika najgorszego, niedopuszczalnego. Niestety także i w tej metodzie występuje parametr (tzw. współczynnik skalowania), który należy prawidłowo ustalić, aby metoda dawała dobre rezultaty.

Dla rozpatrywanego problemu globalnie optymalnego sterowania okresowego zostanie zastosowana zmodyfikowana wersja statycznej metody kary. Modyfikacja będzie polegać na tym, że algorytm ewolucyjny „nie będzie karał” osobników niedopuszczalnych, jeśli ich średnia, znormalizowana wartość przekroczenia ograniczeń nie będzie przekraczać ustalonego, dopuszczalnego poziomu naruszenia ograniczeń χ. Poziom χ

jest liczbą – automatycznie ustalającą się przy starcie algorytmu ewolucyjnego, której wartość w trakcie działania algorytmu zmniejsza się do 0 według ustalonej funkcji.

W celu wyznaczenia średniej, znormalizowanej wartości przekroczenia ograniczeń przez osobnika z tworzony jest m_o elementowy wektor (m_o jest liczbą wszystkich ograniczeń w problemie (126)-(133)), którego każdy element opisuje przekroczenie i-tego $(i = 1,2,...,m_o)$ ograniczenia. Wektor ten jest wyznaczany według zależności:

$$c_i(z) = \begin{cases} \max(0, \| s(\Phi(1,\tilde{z}) \|_\infty - \alpha), & (i = 1) \\ \max(0, \psi_{i-1}(z)), & (i = 2,...,n_\psi + 1) \\ \max(0, | \phi_{i-n_\psi-1}(q) | - \delta_\phi), & (i = n_\psi + 2,...,n_\psi + n_\phi + 1) \\ \max(0, \tilde{\phi}_{i-n_\psi-n_\phi-1}(q)), & (i = n_\psi + n_\varphi + 2,...,m_o) \end{cases} \tag{134}$$

gdzie δ_ϕ jest tolerancją, z jaką ograniczenia równościowe (132) mają być spełnione, $|\cdot|$ oznacza wartość absolutną, $m_o = n_\psi + n_\varphi + n_{\tilde{\varphi}} + 1$, natomiast wektor $\psi \in R_+^{n_x \times L}$ wyznaczany jest następująco:

$$\psi_{1+i+(j-1)L}(z) = \begin{cases} 0 & \text{dla} \quad x_{ji} \in X, \\ | x_j^- - x_{ji} | & \text{dla} \quad x_{ji} < x_j^-, \\ | x_j^+ - x_{ji} | & \text{dla} \quad x_{ji} > x_j^+, \end{cases} \quad (j = 1,...,n_x; i = 0,...,L-1). \tag{135}$$

Za wartość, według której normalizowane będą ograniczenia, przyjmowane jest największe przekroczenie ograniczenia w pierwszym pokoleniu (w populacji, od której zaczyna pracę algorytm ewolucyjny). Wyznacza się ją następująco:

$$c_{\max} = \max_{z \in P^1 \wedge i=1,...,m_o} c_i(z). \tag{136}$$

Mając powyższe, zmodyfikowany wskaźnik jakości dla problemu (126)-(133) (uwzględniający karę za przekroczenie ograniczeń o więcej niż χ) jest wyznaczany jak poniżej:

$$\tilde{Q}(z) = Q(z) + \rho \eta_p(z) c_s(z), \tag{137}$$

gdzie

$$c_s(z) = \frac{1}{m_o} \sum_{i=1}^{m_o} \frac{c_i(z)}{c_{\max}}, \tag{138}$$

$$\eta_p(z) = \begin{cases} 0 & dla \quad c_s(z) \le \chi(g), \\ 1 & dla \quad c_s(z) > \chi(g), \end{cases} \tag{139}$$

natomiast współczynnik kary ρ jest dobierany tak, by – dla tych osobników z w bieżącej populacji, dla których $n_p(z) \neq 0$ – prawdziwa była nierówność $\rho\eta_p(z)c_s(z) > Q(z^{\max})$ ($z^{\max}$ jest to osobnik, dla którego funkcja celu Q przyjmuje największą wartość).

W trakcie działania algorytmu, wartość dopuszczalnego przekroczenia ograniczeń χ wyznaczana jest następująco:

$$\chi(g+1) = \begin{cases} \max\limits_{z \in P^1}(c_s(z)) & dla \quad g = 1, \\ (1 - 2\dfrac{g}{G_{\max}})^{\chi_p} & dla \quad 1 < g \le 0.5G_{\max} \wedge (G_\chi \bmod g) = 0, \\ \chi(g) & dla \quad 1 < g \le 0.5G_{\max} \wedge (G_\chi \bmod g) \neq 0, \\ 0 & dla \quad g > 0.5G_{\max}, \end{cases} \tag{140}$$

gdzie $G_{\max}$ jest maksymalną liczbą pokoleń (którą może wygenerować algorytm), $1 < G_\chi << 0.5G_{\max}$ i $\chi_p \ge 1$ są parametrami ustalanymi przez użytkownika (decydują one o tym, jak szybko – w poszczególnych pokoleniach – wartość χ ma maleć).

Zaletą zaproponowanej dla problemów okresowych metody kary (137) jest to, że ma ona tylko dwa parametry, które musi określić użytkownik: G_χ – od którego zależy, jak często zmieniany ma być dopuszczalny poziom naruszenia ograniczeń oraz χ_p – ustalający funkcję, zgodnie z którą zmieniany jest dopuszczalny poziom naruszenia ograniczeń. Dodatkową zaletą omawianej metody kary jest fakt, iż w początkowej fazie działania algorytmu nie „ocenia ona źle” osobników nieznacznie naruszających ograniczenia, dzięki czemu algorytm może przemieszczać się przez obszary niedopuszczalne w celu poszukiwania rozwiązania, które będzie spełniać ograniczenia narzucone na rozwiązywany problem.

Przedstawiona metoda kary (137), bazuje na metodzie Takahama i Sakai [132]. Naukowcy ci nie stosują jednak funkcji kary, lecz specjalnie zdefiniowane operatory: <, ≤, które wykorzystywane są do porównywania osobników (i ich oceniania). Operatory te – na podstawie wartości funkcji celu oraz wartości określającej przekroczenie ograniczeń – informują, który z dwóch porównywanych osobników jest w danym momencie działania algorytmu lepiej przystosowany.

Mając zdefiniowany sposób uwzględniania ograniczeń technologicznych (chwilowych i uśrednionych) oraz ograniczenia stabilnościowego, można skonstruować algorytm ewolucyjny globalnej optymalizacji dla problemów okresowych ze wspomnianymi ograniczeniami.

Algorytm 3a (algorytm z turniejową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; $N_t < N$ – ilość osobników biorących udział w turnieju; K – poziom dyskretyzacji sterowania; ε, g_p – parametry kryterium stopu (49); $G_{\max}$ – maksymalna liczba pokoleń; G_χ, χ_p – parametry funkcji kary (137); $p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (130) (tzn. by $z \in Z$). Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Na osobnikach naruszających ograniczenia uśrednione (131) zastosuj Algorytm rekonstrukcji ograniczeń. Następnie, stosując metodę Newtona (73) lub Gaussa-Newtona (74), doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (129) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu $Q(z^i)$ $(i = 1,2,...,N)$ oraz – według zależności (137) – jej zmodyfikowaną wersję $\tilde{Q}(z^i)$. W śród osobników spełniających ograniczenia (129)-(133) znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q. Jeżeli wszystkie osobniki naruszają ograniczenia (129)-(133), wówczas za $z^{\min}$ przyjmij osobnika, który ma najmniejszą wartość naruszenia ograniczeń c_s (138). Znajdź osobnika $z^{\max}$ z największą wartością zmodyfikowanej funkcji celu $\tilde{Q}$.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $z^{\min}(g-1)$ oraz $z^{\min}(g)$ spełniają ograniczenia (129)-(133) i zachodzi $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$. W przeciwnym wypadku, zastąp osobnika

$z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$ tylko wówczas, gdy prawdziwa jest nierówność $c_s(z^{\min}(g-1)) < c_s(z^{\min}(g))$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$, to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Powtarzaj poniższe czynności do momentu, aż bieżąca populacja P^g będzie pusta:

- wylosuj z bieżącej populacji N_t osobników;
- z N_t osobników do populacji tymczasowej P^g_{tmp} skopiuj osobnika o najmniejszej wartości zmodyfikowanej funkcji celu $\tilde{Q}$;
- usuń z bieżącej populacji N_t osobników.

Krok 7 (krzyżowanie) Skopiuj do nowej populacji P^{g+1} osobniki z populacji tymczasowej P^g_{tmp}. Następnie, powtarzaj poniższe czynności tak długo, aż nowa populacja P^{g+1} będzie składać się z N osobników:

- wylosuj dwa osobniki z populacji tymczasowej P^g_{tmp} i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania, natomiast krzyżowane osobniki zwróć do populacji P^g_{tmp}.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w populacji działaniu tego operatora – przy czym otrzymanymi nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano operator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \leq p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

Algorytm ewolucyjny dla problemu (126)-(133), wykorzystujący jako metodę reprodukcji metodę ruletkową 2.4.1 lub metodę rangową 2.4.3, może być zdefiniowany następująco:

Algorytm 3b (algorytm z ruletkową lub rangową metodą reprodukcji)

Krok 1 (inicjalizacja). Przyjmij: N – ilość osobników z w populacji; K – poziom dyskretyzacji sterowania; ε, g_p – parametry kryterium stopu (49); $G_{\max}$ – maksymalna liczba pokoleń; G_χ, χ_p – parametry funkcji kary (137); $p_c \in (0,1]$ – prawdopodobieństwo krzyżowania; $p_m \in (0,0.15]$ – prawdopodobieństwo mutacji. Wybierz (spośród operatorów genetycznych opisanych w rozdziale 2) po jednym operatorze krzyżowania i mutacji. Spośród metod reprodukcji (metody ruletkowej i metod rangowych) wybierz metodę, według której będziesz wyznaczać prawdopodobieństwo przeżycia osobnika. Z równomiernym rozkładem prawdopodobieństwa wylosuj N osobników z tak, by były spełnione ograniczenia chwilowe (130) (tzn. by $z \in Z$). Podstaw $g = 1$, gdzie g jest numerem bieżącego pokolenia.

Krok 2 (naprawa) Na osobnikach naruszających ograniczenia uśrednione (131) zastosuj Algorytm rekonstrukcji ograniczeń. Następnie, stosując metodę Newtona (73) lub Gaussa-Newtona (74) doprowadź do sytuacji, w której – dla każdego osobnika z w populacji – spełnione będzie ograniczenie (129) na okresowość procesu.

Krok 3 (ocena) Dla każdego osobnika w populacji wyznacz wartość funkcji celu $Q(z^i)$ $(i = 1,2,...,N)$ oraz – według zależności (137) – jej zmodyfikowanej wersji $\tilde{Q}(z^i)$. W śród osobników spełniających ograniczenia (129)-(133) znajdź osobnika $z^{\min}$ z najmniejszą wartością funkcji celu Q. Jeżeli wszystkie osobniki naruszają ograniczenia (129)-(133), wówczas za $z^{\min}$ przyjmij osobnika, który ma najmniejszą wartość naruszenia ograniczeń c_s (138). Znajdź osobnika $z^{\max}$ z największą wartością zmodyfikowanej funkcji celu $\tilde{Q}$.

Krok 4 (sukcesja) Niech $z(g)$ oznacza osobnika z w pokoleniu g. Jeżeli $z^{\min}(g-1)$ oraz $z^{\min}(g)$ spełniają ograniczenia (129)-(133) i zachodzi $Q(z^{\min}(g-1)) < Q(z^{\min}(g))$, wtedy zastąp w bieżącym pokoleniu osobnika

$z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$. W przeciwnym wypadku, zastąp osobnika $z^{\max}(g)$ osobnikiem $z^{\min}(g-1)$ tylko wówczas, gdy prawdziwa jest nierówność $c_s(z^{\min}(g-1)) < c_s(z^{\min}(g))$.

Krok 5 (kryterium stopu) Jeżeli $g \geq G_{\max}$ lub $Q(z^{\min}(g-g_p)) - Q(z^{\min}(g)) \leq \varepsilon$, to zatrzymaj algorytm i przyjmij $z^{\min}(g)$ za znalezione optimum. W przeciwnym wypadku kontynuuj działanie algorytmu.

Krok 6 (reprodukcja) Stosując wybraną metodę reprodukcji i używając wartości zmodyfikowanej funkcji celu $\tilde{Q}$ wyznacz prawdopodobieństwo przeżycia (wylosowania) $p_w^i(z^i)$ $(i = 1,2,...,N)$ dla każdego osobnika w populacji. Następnie – dla każdego osobnika z^i $(i = 1,2,...,N)$ – wyznacz dystrybuantę $d_i = \sum_{j=1}^{i} p_w^j$ $(i = 1,2,...,N)$. Powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^g będzie zawierać N osobników:

- wylosuj z równomiernym rozkładem prawdopodobieństwa liczbę $p \in [0,1]$;
- wstaw do populacji tymczasowej P_{tmp}^g kopię osobnika z^i $(i = 1,2,...,N)$, dla którego prawdziwa jest nierówność $d_{i-1} < p \leq d_i$.

Krok 7 (krzyżowanie) Dla każdego osobnika z^i $(i = 1,2,...,N)$ w populacji tymczasowej wylosuj $p_i \in [0,1]$. Przenieś do nowej populacji P^{g+1} te osobniki z populacji tymczasowej P_{tmp}^g, dla których $p_i > p_c$. Jeżeli populacja P_{tmp}^g zawiera nieparzystą liczbę osobników, to losowo wybierz jednego osobnika i przenieś go do populacji P^{g+1}. Następnie, powtarzaj poniższe czynności do momentu, aż populacja tymczasowa P_{tmp}^g będzie pusta:

- wylosuj dwa osobniki z populacji tymczasowej P_{tmp}^g i poddaj je działaniu wybranego (w kroku inicjalizacji) operatora krzyżowania;
- dodaj do nowej populacji P^{g+1} osobniki będące wynikiem krzyżowania;
- usuń krzyżowane osobniki z populacji P_{tmp}^g.

Krok 8 (mutacja) Podstaw $g = g + 1$. Jeżeli w kroku inicjalizacji – jako operator mutacji – został wybrany operator (39), (43), (46), to poddaj każdego osobnika w po-

pulacji działaniu tego operatora – przy czym otrzymanymi nowymi genami zastępuj geny, które podlegały mutacji. Jeżeli jako operator mutacji wybrano operator inny niż (39), (43), (46), to dla każdego osobnika z^i $(i = 1,2,..., N)$ w populacji P^g wylosuj liczbę $p_i \in [0,1]$. Gdy $p_i \le p_m$, to poddaj osobnika z^i działaniu wybranego operatora mutacji – przy czym osobnikiem będącym wynikiem mutacji zastępuj osobnika mutowanego. Po wykonaniu mutacji idź do kroku 2 niniejszego algorytmu.

3.4.3. Przykłady obliczeniowe

Przykład 8.

Pierwszym problemem okresowym, na którym przetestowany został ewolucyjny Algorytm 3a, było następujące zadanie: zminimalizuj wskaźnik jakości

$$Q(x,u,\tau) = \frac{1}{\tau}\int_0^\tau (x_1(t) + x_2(t))dt \tag{141}$$

procesu opisanego równaniami stanu

$$\dot{x}_i(t) = u_i(t) - x_i(t) - 2x_1^2(t)x_2(t), \quad (i = 1,2), \tag{142}$$

przy uwzględnieniu ograniczeń:

$$x(0) = x(\tau), \tag{143}$$

$$\tau \in [0.1,10], \quad 0 \le u_i(t) \le 2, \quad t \in [0,\tau], \quad (i = 1,2) \tag{144}$$

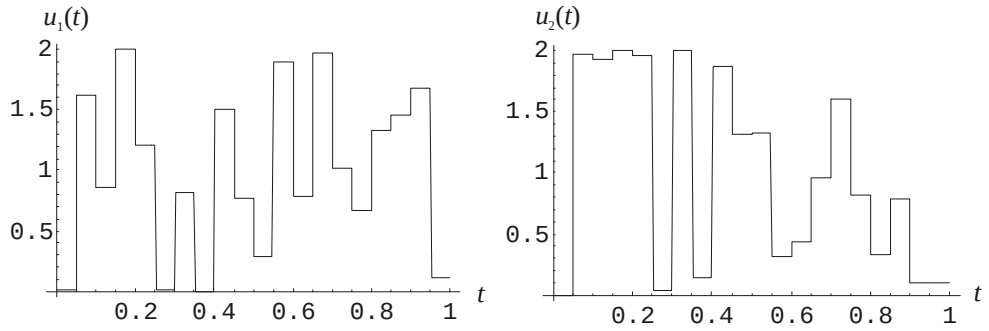
$$\frac{1}{\tau}\int_0^\tau u_i(t)dt = 1 \quad (i = 1,2). \tag{145}$$

$$0 \le x_1(t) \le 0.6, \quad 0.2 \le x_2(t) \le 0.6, \quad t \in [0,\tau]. \tag{146}$$

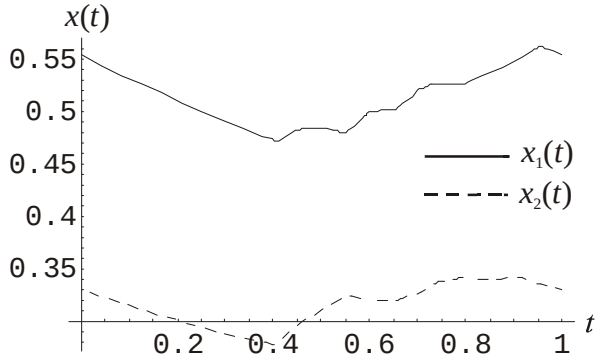
Powyższe zadanie opisuje ciągły proces produkcyjny polegający na izotermicznej reakcji syntezy $A_1 + A_2 \to B$, w której $x_i(t)$ $(i = 1,2)$ jest koncentracją składnika A_i w reaktorze w chwili t, natomiast $u_i(t)$ $(i = 1,2)$ oznacza koncentrację składnika A_i w strumieniu wejściowym. Wskaźnik jakości problemu (141)-(146) odpowiada maksymalizacji produktu użytecznego B.

Problem (141)-(146) rozwiązany został z wykorzystaniem arytmetycznego operatora krzyżowania (35) oraz z wykorzystaniem równomiernego operatora mutacji (38) dla następujących parametrów: $G_{\max} = 1000$, $\varepsilon = 0$, $g_p = 1000,$, $N = 20$, $N_t = 2$, $K = 20$, $p_m = 0.15$. Najlepsze uzyskane sterowanie zostało przedstawione na rysunkach 30-32. Zapewnia ono wskaźnik jakości $\hat{Q} = 0.828933$ i jest ono lepsze od optymalnego rozwiązania statycznego analizowanego problemu (zapewniającego wskaźnik jakości $\hat{\bar{Q}} = 1.1794$) o około 30%.

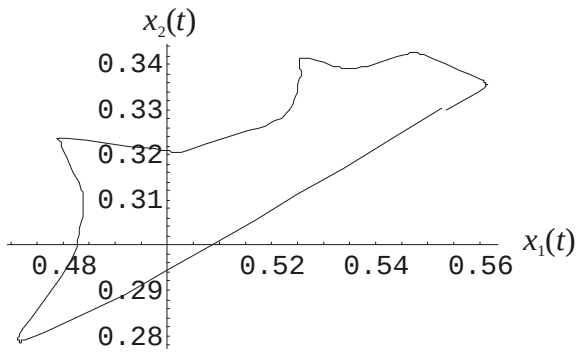
Problem (141)-(146) – z ustalonym okresem oraz bez chwilowych ograniczeń stanu – był rozważany w przykładzie 4. W przykładzie tym, dla najlepszego rozwiązania dostarczonego przez ewolucyjny Algorytm 2a – wykorzystujący arytmetyczny operator krzyżowania (35) oraz równomierny operator mutacji (38) – wskaźnik jakości był równy $Q = 0.859917$. Jak widać, możliwość zmiany okresu pozwoliła Algorytmowi 3a (mimo narzuconych ograniczeń chwilowych na stan) wyznaczyć lepsze rozwiązanie.



Rysunek 30 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 0.3$ najmniejszą wartość wskaźnika jakości dla problemu (141)-(146)



Rysunek 31 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 30, który zapewnia wskaźnik jakości $\hat{Q} = 0.828933$



Rysunek 32 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q}$ = 0.828933

Przykład 9.

Algorytm 3b z rangową reprodukcją (29), krzyżowaniem równomiernym (34) oraz mutacją równomierną (39) został przetestowany na ciągłym procesie katalitycznego utleniania etylenu w przepływowym reaktorze z wymieszaniem. Równania stanu opisujące dynamikę tego procesu dane są następującymi zależnościami:

$$\dot{x}_1(t) = u_1(t)(1 - x_1(t)x_4(t)), \tag{147}$$

$$\begin{aligned}\dot{x}_2(t) = u_1(t)(u_2(t) - x_2(t)x_4(t)) - 92.8e^{-8.13/x_4(t)}(x_2(t)x_4(t))^{0.5} \\ -12.66e^{-7.12/x_4(t)}(x_2(t)x_4(t))^{0.25}\end{aligned} \tag{148}$$

$$\begin{aligned}\dot{x}_3(t) = -u_1(t)x_3(t)x_4(t)) + 92.8e^{-8.13/x_4(t)}(x_2(t)x_4(t))^{0.5} \\ -2412.71e^{-11.07/x_4(t)}(x_3(t)x_4(t))^{0.5}\end{aligned} \tag{149}$$

$$\begin{aligned}\dot{x}_4(t) = x_1^{-1}(t)(u_1(t)(1 - x_4(t)) + 7.32e^{-8.13/x_4(t)}(x_2(t)x_4(t))^{0.5} \\ +10.39e^{-7.12/x_4(t)}(x_2(t)x_4(t))^{0.25} \\ +2170.5e^{-11.07/x_4(t)}(x_3(t)x_4(t))^{0.5} - 7.02(x_4(t) - 1))\end{aligned} \tag{150}$$

gdzie $x_1(t)$ jest zawartością mieszaniny gazów, $x_2(t)$ jest stężeniem etylenu, $x_3(t)$ jest stężeniem tlenku etylenu, zaś $x_4(t)$ oznacza temperaturę; z kolei $u_1(t)$ i $u_2(t)$ – to odpowiednio: natężenie dopływu i stężenie etylenu.

W powyższym procesie należy znaleźć takie sterowanie, które zapewnia najmniejszą wartość wskaźnika jakości

$$Q = \frac{\int_0^\tau u_1(t)x_3(t)x_4(t)dt}{\int_0^\tau u_1(t)u_2(t)dt}, \tag{151}$$

przy ograniczeniach:

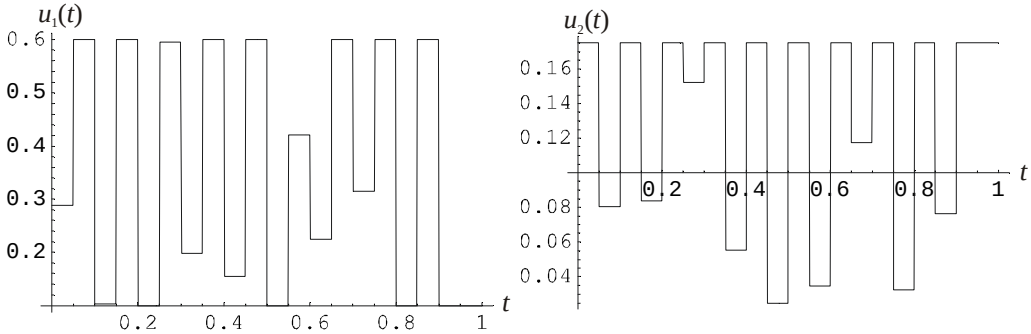
$$x_i(0) = x_i(\tau), \quad (i = 1,2,3,4), \tag{152}$$

$$\tau \in [0.1,150], \quad 0.041 \le u_1(t) \le 0.659, \quad 0 \le u_2(t) \le 0.2, \quad t \in [0,\tau] \tag{153}$$

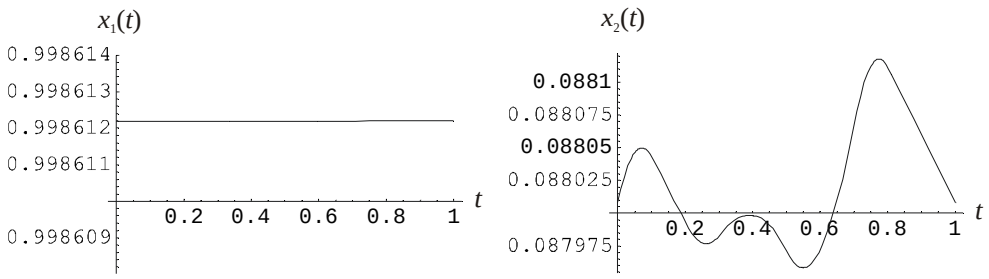
$$\frac{1}{\tau}\int_0^{\tau} u_1(t)dt = 0.35, \quad \frac{1}{\tau}\int_0^{\tau} u_1(t)u_2(t)dt = 0.035, \tag{154}$$

$$x_2(t) \le 0.1. \tag{155}$$

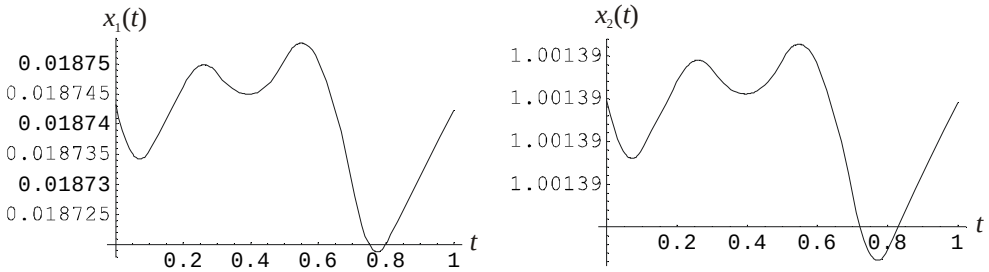
Optymalne rozwiązanie problemu (147)-(155) wyznaczone metodą strzałów [91] zapewnia wskaźnik jakości $Q = -0.108$; rozwiązanie wyznaczone z wykorzystaniem algorytmu gradientowego [20] daje wskaźnik jakości $Q = -0.127$; natomiast wskaźnik jakości dla najlepszego rozwiązania przedstawionego na rysunkach 33-35 – wyznaczonego algorytmem ewolucyjnym – jest równy $\hat{Q} = -0.187652$. Rozwiązanie ewolucyjne poprawia rozwiązanie wyznaczone metodą gradientową o prawie 48%, a optymalne rozwiązanie statyczne $\hat{\overline{Q}} = -0.095$ o 97%. Należy tutaj zaznaczyć, że rozwiązanie optymalne zostało wyznaczone algorytmem ewolucyjnym dla następujących parametrów: $G_{\max} = 500$, $\varepsilon = 0$, $g_p = 500$, $N = 20,$, $K = 20$, $p_c = 0.5$, $p_m = 0.015$.



Rysunek 33 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 0.1$ najmniejszą wartość wskaźnika jakości dla problemu (147)-(155)



Rysunek 34 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 33, który zapewnia wskaźnik jakości $\hat{Q} = -0.187652$



Rysunek 35 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 33, który zapewnia wskaźnik jakości $\hat{Q} = -0.187652$

Przykład 10.

Zadaniem, zawierającym ograniczenia stabilnościowe, na którym testowany był Algorytm 3 był (wymuszany przez katalizator fluidalny K gromadzący się w reaktorze) ciągły proces produkcyjny, w którym zachodzi reakcja chemiczna $A \to B$.

Zadanie to polega na minimalizacji wskaźnika jakości

$$Q(x,u,\tau) = \frac{1}{\tau}\int_0^{\tau}(x_1(t) + x_2(t))dt \tag{156}$$

procesu opisanego równaniami stanu:

$$\dot{x}_1(t) = u_1(t) - x_1(t) - x_1^2(t)x_2^3(t) \ , \tag{157}$$

$$\dot{x}_2(t) = u_2(t) - x_2(t) - x_1^2(t)x_2^{0.5}(t), \tag{158}$$

przy uwzględnieniu ograniczeń:

$$x(0) = x(\tau), \tag{159}$$

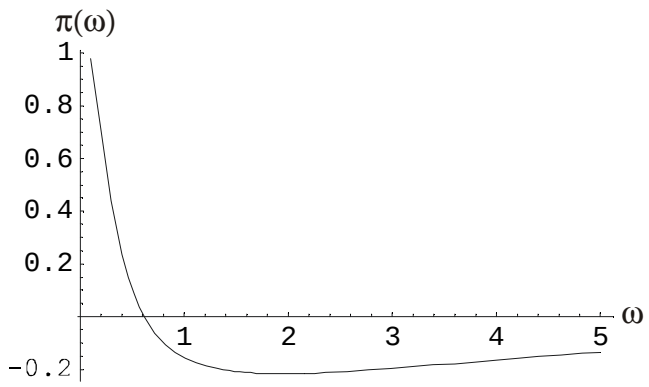
$$\tau \in [0.1,10], \quad 0 \le u_i(t) \le 4, \quad t \in [0,\tau], \quad (i = 1,2) \tag{160}$$

$$\frac{1}{\tau}\int_0^{\tau} u_i(t)dt = 2 \quad (i = 1,2), \tag{161}$$

$$\| s(\Phi(x,u,\tau)) \|_{\infty} \le \alpha. \tag{162}$$

W powyższym problemie $x_1(t)$ oznacza stężenie surowca A w reaktorze, $x_2(t)$ jest stężeniem katalizatora K w reaktorze, zaś $u_1(t)$ i $u_2(t)$ oznaczają odpowiednio: koncentrację A i K w strumieniu wejściowym. Wskaźnik jakość odpowiada tu natomiast maksymalizacji produktu użytecznego B, przy jednoczesnej minimalizacji zawartości katalizatora w strumieniu wyjściowym z reaktora.

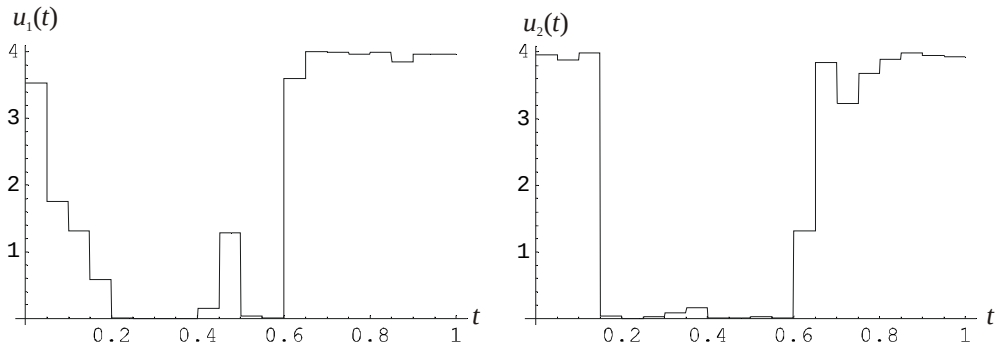
Optymalne rozwiązanie statyczne tego problemu ($\hat{\bar{u}} = (2,2)$, $\hat{\bar{x}} = (1,1)$), zapewniające wskaźnik jakości $\hat{\bar{Q}} = 2$, jest niestabilne. Multiplikatory Floquet'a są bowiem dla niego równe (0.0081, 1.3658). Z testu π wynika (patrz rysunek 36), że istnieje sterowanie okresowe, które może zapewnić lepszą wydajność procesu w porównaniu z optymalnym sterowaniem statycznym. Dlatego też powyższy problem został rozwiązany Algorytmem 3a przy założeniu $\alpha = 1$ (tzn. poszukiwane było optymalne, stabilne sterowanie okresowe).



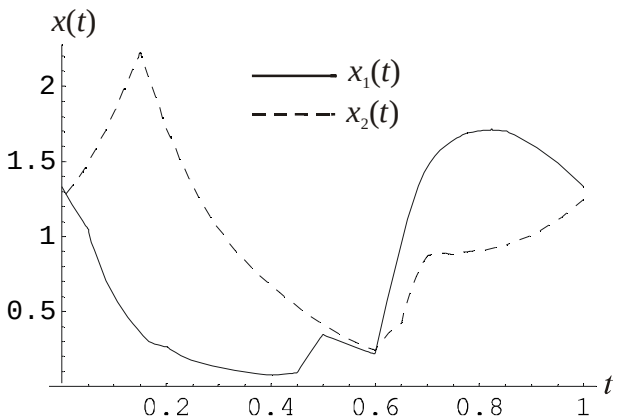
Rysunek 36 Wynik testu π dla przykładu problemu (156)-(162)

Przy zastosowaniu równomiernego operatora krzyżowania (34) oraz równomiernego operatora mutacji (39) Algorytm 3 znalazł w 10000 pokoleniu rozwiązanie okresowe zapewniające wskaźnik jakości $\hat{Q} = 1.7824$ (patrz rysunki 37-39). Rozwiązanie to nie tylko poprawia wydajność procesu statycznego o prawie 11%, lecz jest ono także stabilne – największy multiplikator Floquet'a ma bowiem w tym przypadku wartość 0.994079.

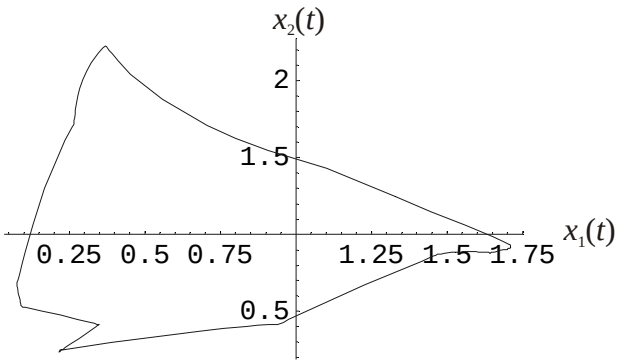
Należy jeszcze wspomnieć, że wynik przedstawiony na rysunkach 37-39 został wyznaczony Algorytmem 3a dla następujących parametrów: $G_{\max} = 10000$, $\varepsilon = 0$, $g_p = 10000,$, $N = 20$, $N_t = 2$, $K = 20$, $p_m = 0.015$.



Rysunek 37 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 4.88$ najmniejszą wartość wskaźnika jakości dla problemu (156)-(162)



Rysunek 38 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 37, który zapewnia wskaźnik jakości $\hat{Q} = 1.7824$



Rysunek 39 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = 1.7824$

Przykład 11.

Ostatnim przedstawionym w tej pracy przykładem, który został rozwiązany algorytmem ewolucyjnym jest problem GOSO dla ciągłego procesu produkcyjnego, którego istotą jest nieodwracalna, egzotermiczna reakcja $A \to B$, $A \to C$ zachodząca w przepływowym reaktorze z mieszalnikiem.

Zminimalizuj wskaźnik jakości

$$-\frac{1}{\tau}\int_0^{\tau} x_2(t)dt \tag{163}$$

procesu opisanego równaniami stanu

$$\dot{x}_1(t) = 1 - x_1 - 10^2 e^{28} e^{-100/x_3(t)} x_1^2(t) - 10^2 e^{23} e^{-85/x_3(t)} x_1(t), \tag{164}$$

$$\dot{x}_2(t) = 10^2 e^{28} e^{-100/x_3(t)} x_1^2(t) - x_2(t), \tag{165}$$

$$\begin{aligned} \dot{x}_3(t) = {} & 10^2 e^{28} e^{-100/x_3(t)} x_1^2(t) - 1.2 * 10^2 e^{23} e^{-85/x_3(t)} x_1(t) \\ & - (x_3(t) - 2.5)u - (x_3(t) - 2.7), \end{aligned} \tag{166}$$

przy uwzględnieniu ograniczeń:

$$x(0) = x(\tau), \tag{167}$$

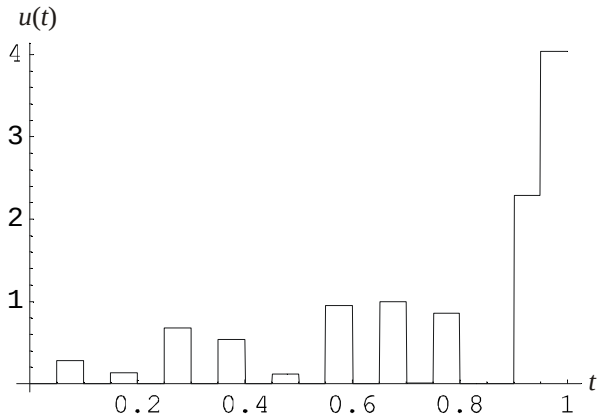
$$\tau \in [0.1, 10], \quad 0 \le u(t) \le 5, \quad t \in [0, \tau], \tag{168}$$

$$\| s(\Phi(x, u, \tau)) \|_\infty \le 1. \tag{169}$$

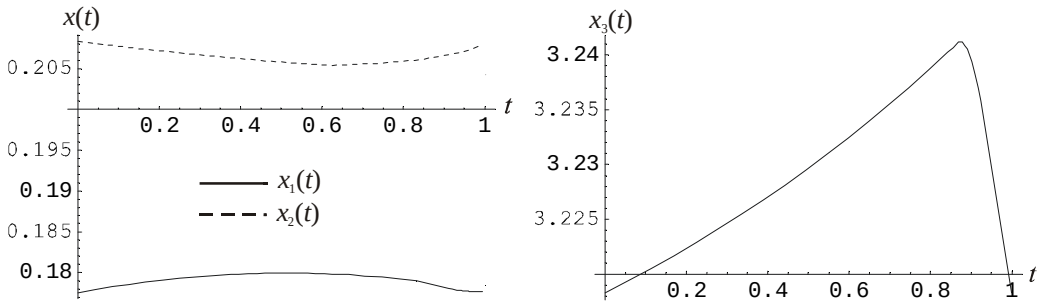
W powyższym problemie $x_1(t)$ i $x_2(t)$ oznaczają odpowiednio stężenie składnika A i B w reaktorze, $x_3(t)$ jest znormalizowaną temperaturą mieszaniny w reaktorze, zaś $u(t)$ jest natężeniem dopływu czynnika chłodzącego.

Problem (163)-(169) został rozwiązany z wykorzystaniem Algorytmu 3b – zawierającego rangową reprodukcją (29), krzyżowanie równomierne (34) oraz mutację równomierną (39) – dla następujących parametrów: $G_{\max} = 10000$, $\varepsilon = 0$, $g_p = 10000$, $N = 20,$, $K = 20$, $p_c = 0.5$, $p_m = 0.015$. Najlepsze, stabilne rozwiązanie problemu przedstawione zostało na rysunkach 40-41. Rozwiązanie to zapewnia wskaźnik jakości $\hat{Q} = -0.205982$, który o prawie 6% poprawia optymalne, niestabilne rozwiązanie ($Q = -0.1946$) wyznaczone przez Matsurbare i Onogi [80]. Znalezione rozwiązanie

„ewolucyjne” poprawia wydajność procesu także w porównaniu z rozwiązaniem statycznym (o 7.5%, $\hat{\overline{Q}} = -0.1917$).



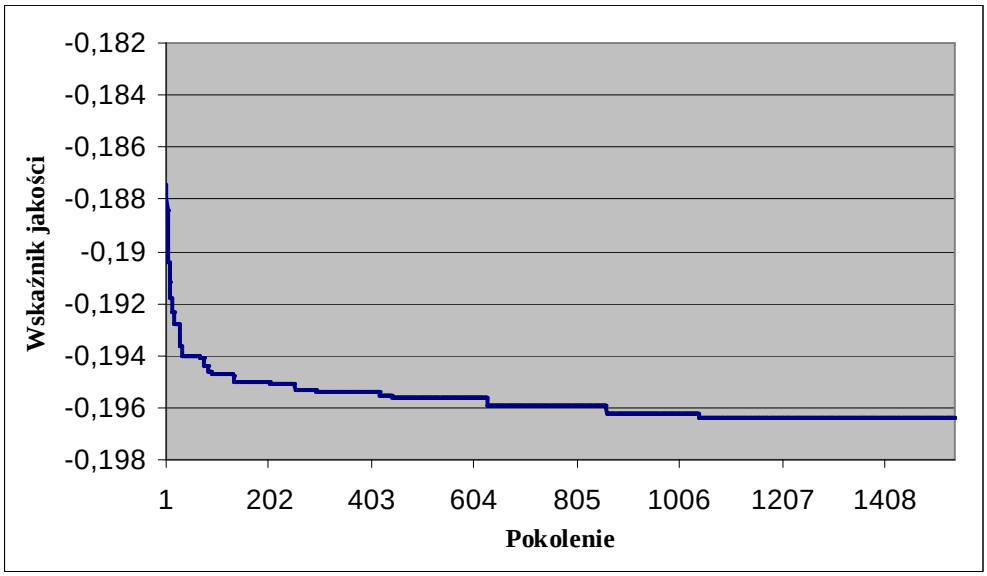
Rysunek 40 Sterowanie okresowe $u(t)$ wyznaczone algorytmem ewolucyjnym zapewniające z okresem $\hat{\tau} = 4.88$ najmniejszą wartość wskaźnika jakości dla problemu (163)-(169)



Rysunek 41 Przebieg stanu $x(t)$, odpowiadający sterowaniu z rysunku 40, który zapewnia wskaźnik jakości $\hat{Q} = -0.205982$

4. ALGORYTMY WSPOMAGAJĄCE

Analizując zachowanie się wartości wskaźnika jakości najlepszego osobnika w populacji w trakcie działania algorytmu ewolucyjnego (rysunek 42), można zauważyć, że stopień jego poprawy jest zdecydowanie większy w początkowej fazie działania algorytmu niż w fazie końcowej. Fakt ten świadczy o tym, że algorytm dosyć szybko znajduje obszar, do którego należy rozwiązanie optymalne; dokładne wyznaczenie tego rozwiązania zajmuje jednak algorytmowi znacznie więcej czasu. Można zatem powiedzieć, że wadą algorytmów ewolucyjnych stosowanych do rozwiązywania problemów okresowych jest ich czasochłonność[1].



Rysunek 42 Wartość wskaźnika jakości (odpowiadająca najlepszemu osobnikowi w populacji) w funkcji generowanych pokoleń, dla rozwiązania z przykładu 1 zapewniającego najmniejszy wskaźnik jakości

Opisane wyżej „zachowanie" algorytmów ewolucyjnych wynika ze sposobu działania operatorów genetycznych. Operatory te są bowiem projektowane tak, by algorytm nie był wrażliwy na przyciąganie optimów lokalnych. Ów brak wrażliwości algorytmów ewolucyjnych uzyskuje się dzięki temu, że geny (w wyniku działania operatorów genetycznych na osobnikach) mogą przyjmować dowolne, dopuszczalne wartości. W początkowej fazie działania algorytmu jest to korzystne, gdyż pozwala algorytmowi na poruszanie się w kierunku globalnego optimum i jednocześnie daje mu możliwość opuszczania optimów lokalnych. Możliwość przyjmowania przez geny dowolnych (do-

[1] Wygenerowanie 2000 pokoleń dla przykładów z poprzedniego rozdziału na komputerze PC z procesorem Sempron 2000 zajmowało od 2-6 godzin.

puszczalnych) wartości jest jednak niekorzystne pod koniec działania algorytmu, gdy jest on blisko optimum globalnego, bowiem duże zmiany wartości genów powodują wówczas, że algorytm nie poprawia dotychczas znalezionego rozwiązania. W końcowej fazie działania algorytmu ewolucyjnego pożądane byłyby więc niewielkie zmiany wartości genów osobników w bliskim ich otoczeniu.

W celu przyspieszenia działania algorytmu ewolucyjnego w końcowej fazie jego działania można zastosować na przykład nierównomierny operator mutacji (patrz rozdział 2.6.2). Istotą działania tego operatora jest zmniejszający się zakres zmian wartości genu w trakcie działania algorytmu; dzięki czemu w początkowej fazie algorytm przeszukuje dosyć duży obszar, natomiast pod koniec działania przeszukuje on tylko najbliższe otoczenie osobników w populacji. Zastosowanie nierównomiernego operatora mutacji w algorytmie ewolucyjnym może więc przyczynić się do tego, że algorytm ten znajdzie lepsze rozwiązanie w krótszym czasie. Wadą tego rozwiązania jest jednak fakt, iż – w przypadku gdy źle dobrane zostaną parametry omawianego operatora mutacji – algorytm utknie w optimum lokalnym, ponieważ małe zmiany wartości nie pozwolą algorytmowi wydostać się z obszaru przyciągania optimum lokalnego.

Z uwagi na powyższe, zasadnym wydaje się poszukiwanie innych sposobów przyspieszania wyznaczania optimum globalnego. Takim sposobem może być zastosowanie metody hybrydowej. W metodzie tej – w pierwszej fazie – działa algorytm ewolucyjny, którego zadaniem jest wyznaczenie wyłącznie obszaru, w którym znajduje się rozwiązanie globalnie optymalne. Po wyznaczeniu tego obszaru „pracę” zaczyna z kolei szybki algorytm optymalizacji lokalnej.

W dalszej części tego rozdziału opisane zostały algorytmy lokalnej optymalizacji, które mogą zostać użyte w hybrydowej metodzie optymalizacji procesów okresowych. Jednak z uwagi na fakt, że opisane algorytmy są metodami gradientowymi, mogą one być wykorzystane wyłącznie do optymalizacji procesów okresowych, w których występują ograniczenia chwilowe oraz uśrednione ograniczenia zasobowe i technologiczne (jedynie takie ograniczenia mogą być bowiem uwzględniane w trakcie optymalizacji przez metody gradientowe). Nie oznacza to jednak, że opisanych niżej algorytmów nie można użyć do rozwiązywania problemów z ograniczeniami stabilnościowymi. W przypadku takich problemów, metody te mogą być jednakże użyte wyłącznie do poprawy dostarczonego przez algorytm ewolucyjny najlepszego rozwiązania w najbliższym jego otoczeniu – w otoczeniu, w którym występują wyłącznie rozwiązania spełniające ograniczenia stabilnościowe. W przypadku – gdy opisane poniżej metody zaczną do-

starczać rozwiązania, dla których ograniczenia stabilnościowe są naruszone, należy zakończyć poszukiwanie optimum z wykorzystaniem proponowanych niżej algorytmów lokalnej optymalizacji.

4.1. Metoda λ-rzutowanego gradientu

Przedstawiony poniżej algorytm – oparty na metodzie rzutowanego gradientu – może być stosowany do wyznaczania optymalnego rozwiązywania problemów okresowych zawierających chwilowe i uśrednione ograniczenia zasobowe. Dla przypomnienia, problem ten (zapisany w znormalizowanej postaci) polega na minimalizacji wskaźnika jakości:

$$Q(x,u,\tau) = \nu(q), \tag{170}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t),u(t))dt \tag{171}$$

$$\dot{x}(t) = \tau\ f(x(t),u(t)), \quad t \in [0,1], \quad x(0) = x(1), \tag{172}$$

$$\tau \in \mathcal{T}, \quad x(t) \in X, \quad u(t) \in U, \quad t \in [0,1], \tag{173}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \le 0, \tag{174}$$

gdzie $\tau \in R_+$ oznacza okres, $x \in \mathcal{H}^1([0,1];R^{n_x})$ – trajektorię stanu, $u \in \mathcal{H}^0\left([0,1];R^{n_u}\right)$ – sterowanie, zaś $q \in R^{n_q}$ jest wektorem wskaźników charakterystycznych procesu; zbiory $\mathcal{T} \in [\tau^-,\tau^+]$, $X \in [x^-,x^+]$, $U \in [u^-,u^+]$ (gdzie $\tau^\pm \in R$, $x^\pm \in R^{n_x}$, $u^\pm \in R^{n_u}$) są wypukłe, a funkcje ν, h, f, φ, $\tilde{\varphi}$, określone są w następujący sposób:

$$\nu : R^{n_q} \to R, \quad h : R^{n_x} \times R^{n_u} \to R^{n_q}, \quad f : R^{n_x} \times R^{n_u} \to R^{n_x},$$

$$\varphi : R^{n_q} \to R^{n_\varphi}, \quad \tilde{\varphi} : R^{n_q} \to R^{n_{\tilde{\varphi}}}.$$

Założenie 1 *Funkcje h i f są dwukrotnie różniczkowalne w sposób ciągły na zbiorze $X \times U$, natomiast funkcje ν, φ, $\tilde{\varphi}$ są dwukrotnie różniczkowalne w sposób ciągły na zbiorze $h(X \times U)$.*

Jakoż iż zapis problemu optymalnego sterowania okresowego (170)-(174) nie jest wygodny zostanie on zmodyfikowany na potrzeby algorytmu rzutowanego gradientu. Sterowanie ciągłe $u(t)$ będzie więc aproksymowane sterowaniem dyskretnym $u_K = (u_0, u_1, ..., u_{K-1})$ w sposób opisany w rozdziale 3.1 (patrz zależność (58)). Istotnym jest, że przyjmuje się, iż dyskretne sterowanie u_K należy do przestrzeni Hilberta $\mathcal{H}^0_{n_u K}$ i że wyposażone jest ono w iloczyn skalarny $< u_K, \breve{u}_K >= \frac{1}{K}\sum_{k=0}^{K-1} < u_k, \breve{u}_k >$ oraz w normę $\| u_K \| = (\frac{1}{K}\sum_{k=0}^{K-1} | u_k |^2)^{0.5}$, gdzie $| u_k | = (\sum_{i=1}^{n_u} u_{ik}^2)^{0.5}$ jest normą Euklidesową.

Założenie 2 *Dla sterowania dyskretnego u_K, ograniczenia zasobowe (173)-(174) odpowiadają ograniczeniom $u_K \in \mathcal{U}$, gdzie*

$$\mathcal{U} = \left\{ u_K \in H^0_{n_u K} : u_k \in U \ (k = 0,1,..,K-1), \frac{1}{\mathrm{K}}\sum_{k=0}^{K-1} \varphi_k^d(u_k) = 0, \frac{1}{\mathrm{K}}\sum_{k=0}^{K-1} \tilde{\varphi}_k^d(u_k) \le 0 \right\}, \quad (175)$$

$\varphi_k^d : R^{n_u} \to R^{n_\varphi}$ *są to funkcje afiniczne, zaś* $\tilde{\varphi}_k^d : R^{n_u} \to R^{n_{\tilde{\varphi}}}$ *są to funkcje wypukłe.*

Założenie 3 *Rozwiązanie okresowe $x(t) \in X$ równania stanu (172) można jednoznacznie wyznaczyć dla (spełniających ograniczenia) danego sterowania u_K i okresu τ.*

Mając powyższe, problem (170)-(174) może zostać zapisany w następującej znormalizowanej, dyskretnej postaci, zredukowanej do przestrzeni sterowania i okresu: zminimalizuj wskaźnik jakości

$$J(\tau, u_K) = \nu(q), \tag{176}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t), \tilde{u}(t))dt, \tag{177}$$

$$\dot{x}(t) = \tau \ f(x(t), \tilde{u}(t)), \quad t \in [0,1], \quad x(0) = x(1), \tag{178}$$

$$\tau \in \mathcal{T}, \quad u_K \in \mathcal{U}, \tag{179}$$

Z problemem (176)-(179) związać można następujący funkcjonał Lagrange'a

$$
\begin{aligned}
\mathcal{L}(\mu,\eta,q,x,\tau,u_K) &= \nu(q)+\mu^T(\int_0^1 h(x(t),\tilde{u}(t))dt-q) \\
&\quad +\int_0^1 \eta^T(t)(\dot{x}(t)-\tau f(x(t),\tilde{u}(t)))dt+\eta^T(0)(x(0)-x(1)) \\
&= \nu(q)-\mu^T q-\int_0^1 H(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt \\
&\quad -\int_0^1 \dot{\eta}^T(t)x(t)dt-(\eta^T(1)-\eta^T(0))x(1),
\end{aligned}
$$

gdzie $\mu \in R^{n_q}$ oraz $\eta \in \mathcal{H}^1([0,1];R^{n_x})$ są zmiennymi sprzężonymi (mnożnikami Lagrange'a), natomiast

$$
H(\mu,\eta(t),x(t),\tau,\tilde{u}(t)) = -\mu^T h(x(t),\tilde{u}(t))+\eta^T(t)\,\tau\, f(x(t),\tilde{u}(t))
$$

jest hamiltonianem problemu.

Z zależności $\mathcal{L}_x = 0$, $J_\tau = \mathcal{L}_\tau$, i $J_{u_K} = \mathcal{L}_{u_K}$ można wyznaczyć poszczególne składowe gradientu wskaźnika jakości postaci:

$$
J_\tau(\tau,u_K) = -\int_0^1 H_\tau(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt, \tag{180}
$$

$$
J_{u_K}(\tau,u_K) = (J_{u_k}(\tau,u_K))_{k=0}^{K-1}, \quad J_{u_k}(\tau,u_K) = -\int_{k/K}^{(k+1)/K} H_{\tilde{u}}(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt, \tag{181}
$$

gdzie zmienne sprzężone μ, η spełniają równania

$$
\mu = \nu_q^T(q), \quad \dot{\eta}(t) = -H_x^T(\mu,\eta(t),x(t),\tau,\tilde{u}(t)), \quad t\in[0,1], \quad \eta(0)=\eta(1).
$$

Rzut P okresu τ na zbiór $\mathcal{T}$ można znaleźć z poniższej zależności

$$
P_{\mathcal{T}}(\tau) = Sat(\tau,\tau^-,\tau^+). \tag{182}
$$

Z kolei rzut $\breve{u}_K$ dyskretnego sterowania u_K na wypukły zbiór ograniczeń $\mathcal{U}$ może być wyznaczony z następującego, pomocniczego problemu optymalizacyjnego [36]: zminimalizuj

$$
\frac{1}{K}\Big(\sum_{k=0}^{K-1}|\breve{u}_k-u_k|^2+\lambda^T\sum_{k=0}^{K-1}\varphi_k^d(\breve{u}_k)+\tilde{\lambda}^T\sum_{k=0}^{K-1}\tilde{\varphi}_k^d(\breve{u}_k)\Big) \tag{183}
$$

z uwzględnieniem ograniczenia

$$\breve{u}_k \in U \quad (k = 0,1,...,K-1),$$

gdzie: $\lambda \in R^{n_\varphi}$, $\tilde{\lambda} \in R_+^{n_{\tilde{\varphi}}}$.

Ponieważ pomocniczy problem optymalizacyjny (183) można traktować rozłącznie względem $\breve{u}_k$, dlatego też jego optymalne rozwiązanie będzie oznaczane jako $\hat{\breve{u}}_k = \hat{\breve{u}}_k(u_k, \breve{\lambda})$ $(k = 0,1,...,K-1)$, gdzie parametr $\breve{\lambda} = (\lambda^T, \tilde{\lambda}^T)^T$ ($\breve{\lambda} \in R^{n_\varphi} \times R_+^{n_{\tilde{\varphi}}}$) jest rozwiązaniem zbioru poniższych równań i nierówności

$$\varphi^d(\breve{\lambda}, u_K) = 0, \quad \tilde{\varphi}^d(\breve{\lambda}, u_K) \le 0, \tag{184}$$

przy czym $\varphi^d(\breve{\lambda}, u_K) = \frac{1}{K}\sum_{k=0}^{K-1}\varphi_k^d(\hat{\breve{u}}_k(u_k, \breve{\lambda}))$, $\tilde{\varphi}^d(\breve{\lambda}, u_K) = \frac{1}{K}\sum_{k=0}^{K-1}\tilde{\varphi}_k^d(\hat{\breve{u}}_k(u_k, \breve{\lambda}))$ są to tzw. $\breve{\lambda}$-funkcje.

W przypadku ograniczeń (174), które dla zdecydowanej większości problemów optymalnego sterowania okresowego mają postać ($u_s \in R^{n_u}$ jest zadanym wektorem):

$$\frac{1}{\tau}\int_0^\tau (u(t) - u_s)dt = 0, \quad \frac{1}{\tau}\int_0^\tau (u(t) - u_s)dt \le 0,$$

$$\frac{1}{\tau}\int_0^\tau (u^2(t) - u_s)dt = 0, \quad \frac{1}{\tau}\int_0^\tau (u^2(t) - u_s)dt \le 0,$$

$\breve{\lambda}^*$ (będące rozwiązaniem zadania (184)) można wyznaczyć np. z wykorzystaniem metody złotego podziału (patrz 4.3). Jeżeli ograniczenia (174) mają bardziej skomplikowaną formę, wówczas $\breve{\lambda}^*$ – spełniające równania i nierówności (184) – może być wyznaczone z następującego problemu optymalizacyjnego:

$$\min\left\{|\varphi^d(\breve{\lambda}, u_K)|^2 + |\tilde{\varphi}_+^d(\breve{\lambda}, u_K)|^2 : \breve{\lambda} \in R^{n_\varphi} \times R_+^{n_{\tilde{\varphi}}}\right\}, \tag{185}$$

gdzie $\tilde{\varphi}_+^d = (\max(\tilde{\varphi}_i^d, 0))_{i=1}^{n_{\tilde{\varphi}}}$.

Problem (185) może zostać rozwiązany za pomocą np. metody gradientowej [25], [105].

Wyznaczywszy $\breve{\lambda}^*$, rzut dyskretnego sterowania u_K na wypukły zbiór ograniczeń $\mathcal{U}$ można wyrazić jako

$$P_{\mathcal{U}}(u_K) = \hat{\breve{u}}_k(u_k, \breve{\lambda}^*). \tag{186}$$

Biorąc pod uwagę powyższe zależności, możliwe jest skonstruowanie algorytmu (wykorzystującego metodę rzutowanego gradientu), który będzie wyznaczał lokalnie optymalne rozwiązanie problemu okresowego (170)-(174) z ograniczeniami zasobowymi (chwilowymi i uśrednionymi).

Algorytm 4

Krok 1 Przyjmij: parametr kryterium stopu $\varepsilon_g \geq 0$, początkowy okres τ oraz sterowanie u_K (może być to np. najlepsze rozwiązanie dostarczone przez algorytm ewolucyjny lub okres wyznaczony z testu π i optymalne sterowanie statyczne). Podstaw $i = 1$.

Krok 2 Wykorzystując rzuty (182), (186) wyznacz nowe rozwiązanie (okres i sterowanie) z następujących zależności

$$\tau^{i+1} = P_{\mathcal{T}}(\tau^i - \hat{\gamma}_s J_\tau(\tau^i, u_K^i)),$$

$$u_K^{i+1} = P_{\mathcal{U}}(u_K^i - \hat{\gamma}_s J_{u_K}(\tau^i, u_K^i)),$$

gdzie $\hat{\gamma}_s$ jest rozwiązaniem problemu

$$\min_{\gamma_s \geq 0} J(P_{\mathcal{T}}(\tau^i - \gamma_s J_\tau(\tau^i, u_K^i)), P_{\mathcal{U}}(u_K^i - \hat{\gamma}_s J_{u_K}(\tau^i, u_K^i))).$$

Krok 3 Podstaw $i = i + 1$. Jeżeli $J(\tau^{i-1}, u_K^{i-1}) - J(\tau^i, u_K^i) \leq \varepsilon_g$, zatrzymaj algorytm i przyjmij τ^i, u_K^i za znalezione lokalne optimum. W przeciwnym wypadku – idź do kroku 2.

4.2. Metoda przesuwanej funkcji kary i λ-rzutowanego gradientu

Problem optymalnego sterowania z ograniczeniami zasobowymi i technologicznymi polegający na minimalizacji wskaźnika jakości

$$Q(x, u, \tau) = \nu(q), \tag{187}$$

przy uwzględnieniu następujących ograniczeń

$$q = \int_0^0 h(x(t), u(t))dt \tag{188}$$

$$\dot{x}(t) = \tau\ f(x(t), u(t)), \quad t \in [0,1], \quad x(0) = x(1), \tag{189}$$

$$\tau \in \mathcal{T}, \quad u(t) \in U, \quad t \in [0, \tau], \tag{190}$$

$$\varphi(q) = 0, \quad \tilde{\varphi}(q) \leq 0, \tag{191}$$

$$x(t) \in X, \tag{192}$$

$$\phi(q) = 0, \quad \tilde{\phi}(q) \leq 0, \tag{193}$$

może zostać rozwiązany przy pomocy algorytmu hybrydowego, łączącego metodę λ-rzutowanego gradientu z metodą przesuwanej kary. Algorytm ten, w trakcie poszukiwania optymalnego rozwiązania problemu (187)-(193), wykorzystuje metodę λ-rzutowanego gradientu do dokładnego uwzględniania ograniczeń zasobowych (190)-(191) (zarówno chwilowych jak i uśrednionych); natomiast metoda kary wykorzystywana jest w nim do uwzględniania ograniczeń technologicznych (192)-(193).

Biorąc pod uwagę informacje, o których mowa w punkcie 4.1, problem (187)-(193) może zostać zapisany w następującej znormalizowanej, dyskretnej postaci zredukowanej do przestrzeni sterowania i okresu: zminimalizuj wskaźnik jakości

$$J(\tau, u_K, \xi_p) = \tilde{\nu}(q, \xi_p), \tag{194}$$

przy uwzględnieniu ograniczeń

$$q = \int_0^1 h(x(t), \tilde{u}(t))dt, \tag{195}$$

$$\dot{x}(t) = \tau \; f(x(t), \tilde{u}(t)), \quad t \in [0,1], \quad x(0) = x(1), \tag{196}$$

$$\tau \in \mathcal{T}, \quad u_K \in \mathcal{U}., \tag{197}$$

gdzie $\xi_p = (\rho, s, \tilde{s})$ jest parametrem przesuwanej funkcji kary $\tilde{\nu}$, zdefiniowanej w sposób następujący

$$\tilde{\nu}(q, \xi_p) = \nu(q) + 0.5\rho(|\phi(q) - s|^2 + |(\tilde{\phi}(q) - \tilde{s}, 0)_+|^2), \tag{198}$$

przy czym $\rho \in R_+$ jest współczynnikiem kary, natomiast $s \in R^{n_\phi}$, $\tilde{s} \in R^{n_{\tilde{\phi}}}$ oznaczają przesunięcie kary.

Z problemem (194)-(197) związać można następujący funkcjonał Lagrange'a

$$
\begin{aligned}
\mathcal{L}(\mu,\eta,q,x,\tau,u_K) &= \tilde{v}(q,\xi_p)+\mu^T(\int_0^1 h(x(t),\tilde{u}(t))dt-q) \\
&\quad +\int_0^1 \eta^T(t)(\dot{x}(t)-\tau f(x(t),\tilde{u}(t)))dt+\eta^T(0)(x(0)-x(1)) \\
&= \tilde{v}(q,\xi_p)-\mu^T q-\int_0^1 H(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt \\
&\quad -\int_0^1 \dot{\eta}^T(t)x(t)dt-(\eta^T(1)-\eta^T(0))x(1),
\end{aligned}
$$

gdzie $\mu \in R^{n_q}$ oraz $\eta \in \mathcal{H}^1([0,1];R^{n_x})$ są zmiennymi sprzężonymi (mnożnikami Lagrange'a), natomiast

$$
H(\mu,\eta(t),x(t),\tau,\tilde{u}(t)) = -\mu^T h(x(t),\tilde{u}(t))+\eta^T(t)\,\tau\, f(x(t),\tilde{u}(t))
$$

jest hamiltonianem problemu.

Z zależności $\mathcal{L}_x = 0$, $J_\tau = \mathcal{L}_\tau$, i $J_{u_K} = \mathcal{L}_{u_K}$ można wyznaczyć poszczególne składowe gradientu wskaźnika jakości postaci:

$$
J_\tau(\tau,u_K,\xi_p) = -\int_0^1 H_\tau(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt, \tag{199}
$$

$$
\begin{aligned}
&J_{u_K}(\tau,u_K,\xi_p) = (J_{u_k}(\tau,u_K,\xi_p))_{k=0}^{K-1}, \\
&J_{u_k}(\tau,u_K,\xi_p) = -\int_{k/K}^{(k+1)/K} H_{\tilde{u}}(\mu,\eta(t),x(t),\tau,\tilde{u}(t))dt,
\end{aligned} \tag{200}
$$

gdzie zmienne sprzężone μ, η spełniają poniższe równania

$$
\mu = \tilde{v}_q^T(q,\xi_p), \quad \dot{\eta}(t) = -H_x^T(\mu,\eta(t),x(t),\tau,\tilde{u}(t)), \quad t\in[0,1], \quad \eta(0)=\eta(1).
$$

Mając powyższe oraz pamiętając o zależnościach na wyznaczanie rzutu okresu na zbiór $\mathcal{T}$ i sterowania na zbiór $\mathcal{U}$ (patrz punkt 4.1), można zdefiniować algorytm optymalizacji, który umożliwi znalezienie lokalnie optymalnego rozwiązania problemu (187)-(193).

Algorytm 5

Krok 1 Przyjmij: $\varepsilon_g \geq 0$ - parametr kryterium stopu pętli λ-rzutowanego gradientu; $\varepsilon_s \geq 0$ - parametr stopu algorytmu; $\xi_p = (\rho, s, \tilde{s})$ - parametr funkcji kary;

$\alpha_p \in (0,1)$, $\beta_p > 0$ $\varepsilon_p > \varepsilon_s$ parametry pętli kary; τ - początkowy okres; u_K - sterowanie (może być to np. najlepsze rozwiązanie dostarczone przez algorytm ewolucyjny lub okres wyznaczony z testu π i optymalne sterowanie statyczne). Podstaw $i = 1$.

Pętla rzutowanego gradientu

Krok 2 Wykorzystując rzuty (182), (186) wyznacz nowe rozwiązanie (okres i sterowanie) z następujących zależności:

$$\tau^{i+1} = P_{\mathcal{T}}(\tau^i - \hat{\gamma}_s J_\tau(\tau^i, u_K^i, \xi_p)),$$

$$u_K^{i+1} = P_{\mathcal{U}}(u_K^i - \hat{\gamma}_s J_{u_K}(\tau^i, u_K^i, \xi_p)),$$

gdzie $\hat{\gamma}_s$ jest rozwiązaniem problemu

$$\min_{\gamma_s \geq 0} J(P_{\mathcal{T}}(\tau^i - \gamma_s J_\tau(\tau^i, u_K^i, \xi_p)), P_{\mathcal{U}}(u_K^i - \hat{\gamma}_s J_{u_K}(\tau^i, u_K^i, \xi_p)), \xi_p).$$

Krok 3 Podstaw $i = i + 1$. Jeżeli $J(\tau^{i-1}, u_K^{i-1}, \xi_p) - J(\tau^i, u_K^i, \xi_p) > \varepsilon_g$, idź do kroku 2.

Pętla kary

Krok 4 Jeżeli $|\phi(q^i)|^2 + |\tilde{\phi}_+(q^i)|^2 \leq \varepsilon_s$, zatrzymaj algorytm i przyjmij τ^i, u_K^i za znalezione lokalne optimum. W przeciwnym wypadku – idź do kroku 5.

Krok 5 Jeżeli $|\phi(q^i)|^2 + |\tilde{\phi}_+(q^i)|^2 \leq \varepsilon_p$, podstaw:

$$s = -(\phi(q) - s), \quad \tilde{s} = -(\tilde{\phi}(q) - \tilde{s})_+, \quad \varepsilon_p = \alpha\varepsilon_p.$$

W przeciwnym wypadku – podstaw:

$$s = -\tfrac{1}{\beta}(\phi(q) - s), \quad \tilde{s} = -\tfrac{1}{\beta}(\phi(q) - \tilde{s})_+, \quad \rho = \beta\rho.$$

Idź do kroku 2 niniejszego algorytmu.

Zbieżność pętli rzutowanego gradientu do rozwiązania spełniającego konieczny warunek optymalności (metody rzutowanego gradientu) jest zagwarantowana dla skończonej liczby dyskretnych kroków $\hat{\gamma}$ minimalizacji kierunkowej [41].

Zbieżność pętli kary związana jest ze stabilnością Lipschitza (L-stabilności) warunków koniecznych optymalności pierwszego rzędu dla rozwiązania problemu optymalnego sterowania okresowego bez funkcji kary, który to problem można zapisać następująco: zminimalizuj zredukowany wskaźnik jakości

$$J(\tau, u_K) \tag{201}$$

przy uwzględnieniu ograniczeń

$$\Psi(\tau, u_K) = 0\,, \tag{202}$$

$$\tilde{\Psi}(\tau, u_K) \le 0\,, \tag{203}$$

gdzie

$$\Psi(\tau, u_K) = \begin{pmatrix} \varphi(q(\tau, u_K)) \\ \phi(q(\tau, u_K)) \end{pmatrix}, \quad \tilde{\Psi}(\tau, u_K) = \begin{pmatrix} \tilde{\varphi}(q(\tau, u_K)) \\ \tilde{\phi}(q(\tau, u_K)) \\ \Lambda \circ (\tau, u_K) \end{pmatrix},$$

zaś afiniczna funkcja $\Lambda : R^{n_o} \to R^{2n_o}$ $((n_o = 1 + n_u K))$, reprezentująca ograniczenia chwilowe $(\tau, u_K) \in \mathcal{T} \times U$, jest zdefiniowana następując $\Lambda \circ (\tau, u_K) = (-\tau + \tau^-, \tau - \tau^+,$ $-u_{ik} + u_i^-, u_{ik} - u_i^+ \;\; (i = 1,...,n_u;\;\; k = 0,1,...,K-1))$.

Niech $(\hat{\tau}, \hat{u}_K)$ oznacza lokalnie optymalne rozwiązanie problemu (201)-(203), natomiast $\tilde{\Psi}_a(\hat{\tau}, \hat{u}_K)$ będzie zbiorem aktywnych ograniczeń nierównościowych dla rozwiązania optymalnego $(\hat{\tau}, \hat{u}_K)$. Ponadto, niech $\Gamma = \Psi'(\hat{\tau}, \hat{u}_K)$, $\tilde{\Gamma}_a = \tilde{\Psi}'(\hat{\tau}, \hat{u}_K)$, gdzie apostrof oznacza pochodną względem (τ, u_K).

Założenie 4 *Macierz*

$$\begin{pmatrix} \Gamma \\ \tilde{\Gamma}_a \end{pmatrix}$$

jest pełnego rzędu, tj. gradienty ograniczeń aktywnych są liniowo niezależne.

Powyższe założenie implikuje istnienie mnożników Lagrange'a $\hat{\varsigma} \in R^{n_\varphi + n_\phi}$, $\hat{\tilde{\varsigma}} \in R_+^{n_{\tilde{\varphi}} + n_{\tilde{\phi}} + 2n_o}$, które pozwalają zapisać konieczny warunek optymalności pierwszego rzędu problemu (201)-(203) w następującej postaci:

$$\mathcal{L}'(\hat{\tau}, \hat{u}_K, \hat{\varsigma}, \hat{\tilde{\varsigma}}) = 0\,, \quad \psi(\hat{\tau}, \hat{u}_K) = 0\,, \quad \tilde{\psi}(\hat{\tau}, \hat{u}_K) \le 0\,, \quad \hat{\tilde{\varsigma}}^T \tilde{\psi}(\hat{\tau}, \tilde{u}_K) = 0\,, \tag{204}$$

gdzie

$$\mathcal{L}(\tau, u_K, \varsigma, \tilde{\varsigma}) = J(\tau, u_K) + \varsigma^T \psi(\tau, u_K) + \tilde{\varsigma}^T \tilde{\psi}(\tau, u_K)$$

jest funkcjonałem Lagrange'a rozważanego problemu.

Niech $\tilde{\Gamma}_+ = \tilde{\psi}'_+(\hat{\tau},\hat{u}_K)$, gdzie $\tilde{\psi}_+(\hat{\tau},\hat{u}_K)$ jest zbiorem nierównościowych, aktywnych ograniczeń odpowiadających $(\hat{\tau},\hat{u}_K)$, dla których $\hat{\tilde{\varsigma}} > 0$.

Założenie 5 *Nierówność*

$$\left\langle (\tau,u_K), \mathcal{L}''(\hat{\tau},\hat{u}_K,\hat{\varsigma},\hat{\tilde{\varsigma}}) \circ (\tau,u_K) \right\rangle > 0$$

jest spełniona dla każdego niezerowego (τ,u_K), *spełniającego następujące równania*

$$\Gamma \circ (\tau,u_K) = 0, \quad \tilde{\Gamma}_+ \circ (\tau,u_K) = 0,$$

(tzn. wystarczający warunek optymalności drugiego rzędu jest spełniony).

Twierdzenie l *Pętla kary jest zbieżna do lokalnego optimum* $(\hat{\tau},\hat{u}_K)$ *problemu optymalnego sterowania okresowego dla skończonego współczynnika kary* ρ *wyznaczonego w skończonej liczbie kroków.*

Dowód. Założenia 4, 5 gwarantują, że warunki konieczne optymalności (204) są silnie regularne w sensie Robinsona [100]. To oznacza, że parametryczny problem optymalizacji, polegający na minimalizacji $J(\tau,u_K)$ przy ograniczeniach $\Psi(\tau,u_K) = a$, $\tilde{\Psi}(\tau,u_K) \le \tilde{a}$ (gdzie: $a \in R^{n_\varphi + n_\phi}$, $\tilde{a} \in R^{n_{\tilde{\varphi}} + n_\phi + 2n_o}$ są parametrami), ma w sąsiedztwie $(a_0,\tilde{a}_0) = (0,0)$ jednoznaczne rozwiązanie $(\hat{\tau}(a,\tilde{a}), \hat{u}_K(a,\tilde{a}))$ spełniające – razem z jednoznacznie określonymi mnożnikami Lagrange'a $(\hat{\varsigma}(a,\tilde{a}), \hat{\tilde{\varsigma}}(a,\tilde{a}))$ – warunek Lipschitza w postaci:

$$\| \hat{\tau}(a,\tilde{a}) - \hat{\tau}(a_0,\tilde{a}_0) \| + \| \hat{u}_K(a,\tilde{a}) - \hat{u}_K(a_0,\tilde{a}_0) \| + \| \hat{\varsigma}(a,\tilde{a}) - \hat{\varsigma}(a_0,\tilde{a}_0) \|$$

$$+ \| \hat{\tilde{\varsigma}}(a,\tilde{a}) - \hat{\tilde{\varsigma}}(a_0,\tilde{a}_0) \| \le M_L \| (a,\tilde{a}) - (a_0,\tilde{a}_0) \| .$$

Wynika stąd, że problem (201)-(203) jest L-stabilny, co z kolei pociąga za sobą zbieżność algorytmu przesuwanej funkcji kary [141].

■

4.3. Przykłady obliczeniowe

Przykład 12.

Metoda λ-rzutowanego gradientu została przetestowana m.in. na następującym problemie – opisującym ciągły proces produkcyjny, w którym zachodzi izotermiczna reakcja $A \to B$, $A \to C$: zminimalizuj wskaźnik jakości

$$-\tfrac{1}{\tau}\int_0^{\tau} x_2(t)dt \tag{205}$$

procesu opisanego okresowymi równaniami stanu:

$$\dot{x}_1 = 2(u_1(t) - x_1(t)) - 5u_2^2(t)x_1^2(t) - 7u_2(t)x_1(t), \quad x(0) = x(\tau), \tag{206}$$

$$\dot{x}_2 = -2x_2(t) + 5u_2^2(t)x_1^2(t), \tag{207}$$

przy uwzględnieniu ograniczeń chwilowych:

$$\tau \in [0.1,\ 10], \quad 0 \le x_i(t), \quad 0 \le u_i(t) \le 2 \quad (i = 1,2), \quad t \in [0, \tau], \tag{208}$$

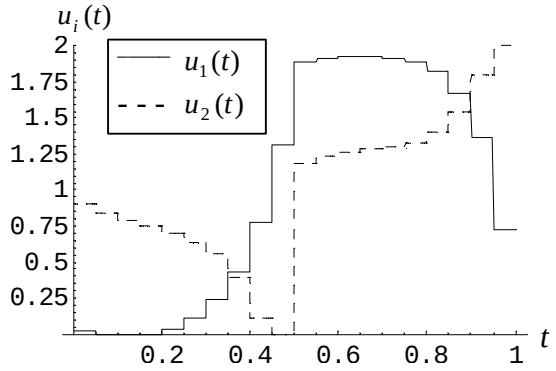
oraz ograniczeń całkowych

$$\tfrac{1}{\tau}\int_0^{\tau} u_i(t)dt = 1, \quad (i = 1,2), \tag{209}$$

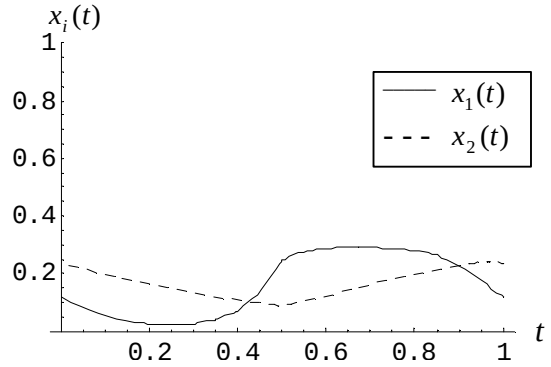
gdzie: $x_1(t)$ – koncentracja składnika A w reaktorze, $x_2(t)$ – koncentracja składnika B w reaktorze, $u_1(t)$ – koncentracja składnika A w strumieniu wejściowym, $u_2(t)$ – znormalizowana temperatura procesu.

Optymalne rozwiązanie statyczne dla problemu przedstawionego powyżej ($\hat{\bar{x}} = (0.1, 0.2)$, $\hat{\bar{u}} = (1,1)$) zapewnia wskaźnik jakości $\hat{\bar{Q}} = -0.2$. Z testu π wynika, że wydajność procesu opisanego równaniami stanu (206)-(207) może zostać poprawiona przez zastosowanie sterowania okresowego. W związku z powyższym, problem (205)-(209) został rozwiązany z wykorzystaniem metody λ-rzutowanego gradientu; przy czym Algorytm 4 został uruchomiony dla okresu początkowego $\tau = 7$ (odczytanego z testu π) oraz dla nieznacznie zakłóconego optymalnego sterowania statycznego. Ponadto, proces był optymalizowany tylko względem sterowania (przez cały czas działania Algorytmu 4 $\tau = 7$).

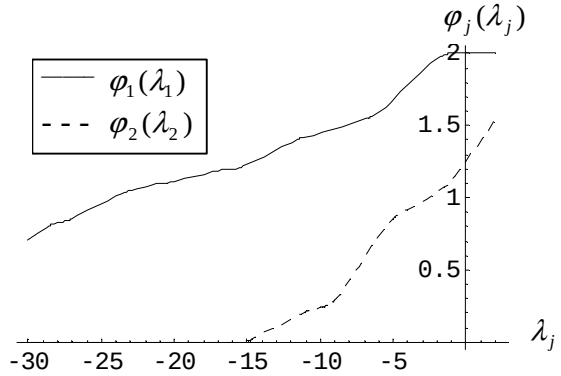
Na rysunkach 43÷45 przedstawione zostały pośrednie wyniki symulacji sterowania okresowego i stanu układu oraz funkcji $\varphi^d(\breve{\lambda}, u_K)$ uzyskane za pomocą metody λ-rzutowanego gradientu. Dla sterowania przedstawionego na rysunku 43 wartość wskaźnika jakości wynosi $Q = -0.162927$.



Rysunek 43 Sterowanie okresowe, dla którego $Q = -0.162927$



Rysunek 44 Trajektoria stanu układu odpowiadająca sterowaniu z rysunku 43

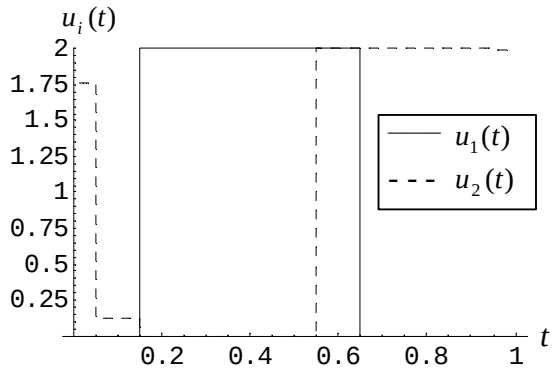


Rysunek 45 Pomocnicza funkcja $\varphi^d(\breve{\lambda}, u_K)$

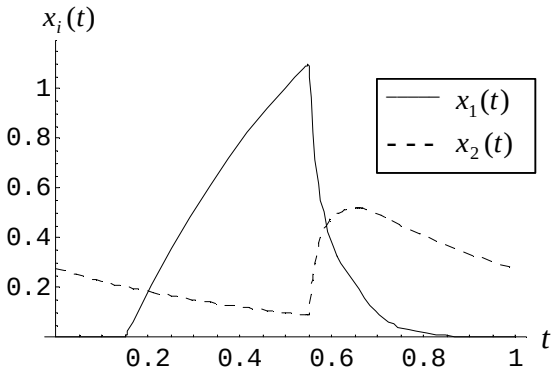
Na rysunkach 46÷47 przedstawiono z kolei optymalne rozwiązanie okresowe problemu (205)-(209) zapewniające wskaźnik jakości $\hat{Q} = -0.274428$. Jak widać, w tym przypadku sterowanie okresowe poprawia rozwiązanie statyczne o 37%.

W przypadku problemu (205)-(209) wyznaczenie rzutu sterowania nie było skomplikowane. Optymalne rozwiązanie pomocniczego problemu (183), które wynosi $\hat{\breve{u}}_K = Sat(u_K + \frac{\breve{\lambda}}{2}, 0, 2)$, można było wyznaczyć analitycznie; natomiast wyznaczenie $\breve{\lambda}^*$ (rozwiązania problemu (184)) sprowadzało się do rozwiązania – metodą złotego podziału – następujących dwóch niezależnych równań:

$$\frac{1}{K}\sum_{k=0}^{K-1} Sat(u_{i,k} + \frac{\breve{\lambda}_i}{2}, 0, 2) = 1 \quad (i = 1,2). \tag{210}$$



Rysunek 46 Optymalne sterowanie okresowe, dla którego $\hat{Q} = -0.274428$



Rysunek 47 Trajektoria stanu – odpowiadająca sterowaniu z rysunku 46

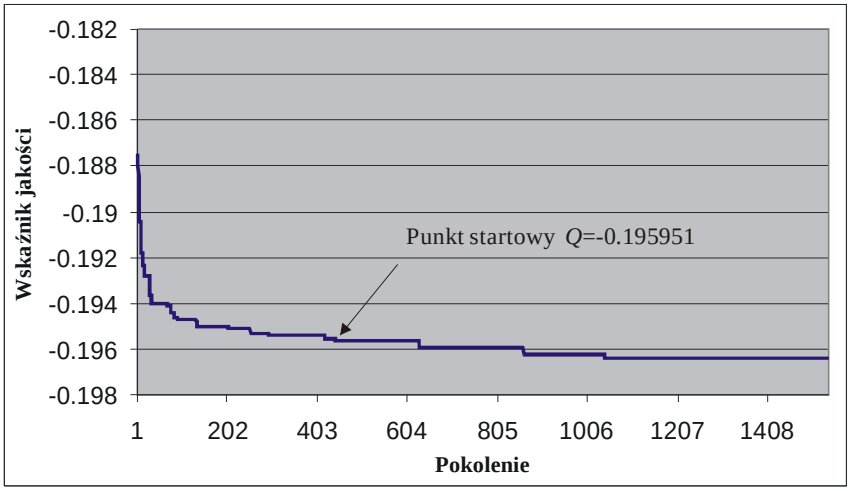
Przykład 13.

W tym miejscu pokazane zostaną wyniki użycia algorytmu hybrydowego (ewolucyjno-gradientowego) dla zadania z przykładu 1.

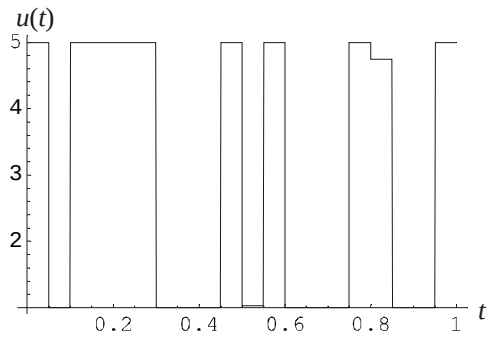
Na rysunku 48 przedstawiona została wartość wskaźnika jakości (najlepszego osobnika w populacji) w funkcji generowanych pokoleń dla przypadku, w którym Algorytm

1 dostarczył najlepszy wskaźnik jakości (tj. $\hat{Q} = -0.196397$). Jak widać z tego rysunku, algorytm ewolucyjny potrzebował, aż 568 pokoleń by poprawić o 0.000768 rozwiązanie $Q = -0.195629$ wyznaczone w pokoleniu 473. Startując z rozwiązania dostarczonego przez algorytm ewolucyjny w 473 pokoleniu, algorytm gradientowy (opisanej w punkcie 4.1) musiał tylko 4 razy wyznaczyć rzut sterowania i okresu, by znaleźć rozwiązanie zapewniające wskaźnik jakości $\hat{Q} = -0.196492$. Jak zatem łatwo zauważyć, zastosowanie algorytmu hybrydowego (ewolucyjno-gradientowego) pozwoliło na wyznaczenie lepszego rozwiązani w znacznie krótszym czasie.

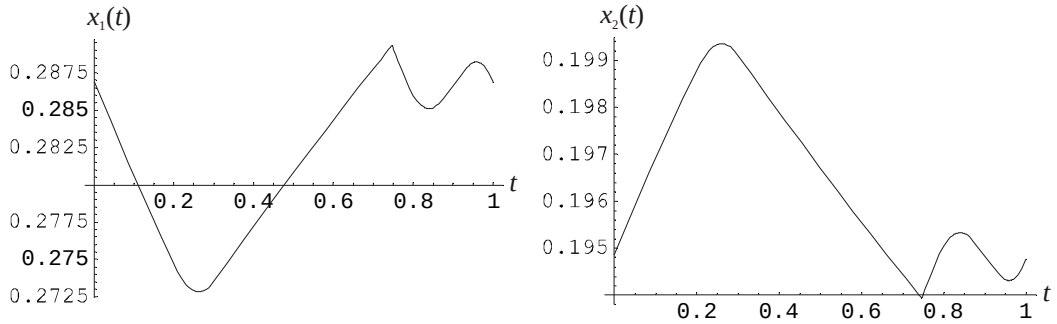
Zadanie z przykładu 1 zostało także rozwiązane z wykorzystaniem metody rzutowanego gradientu startującej z optymalnego rozwiązania statycznego. Otrzymane (w 32 kroku) rozwiązanie, które przedstawione jest na rysunkach 51-52, dostarczyło wskaźnik jakości równy $Q = -0.195360$. Zastosowanie jedynie metody rzutowanego gradientu nie pozwoliło na wyznaczenie rozwiązania lepszego od rozwiązania dostarczonego przez algorytm ewolucyjny (nie mówiąc już o rozwiązaniu otrzymanym po użyciu hybrydowego algorytmu optymalizacji).



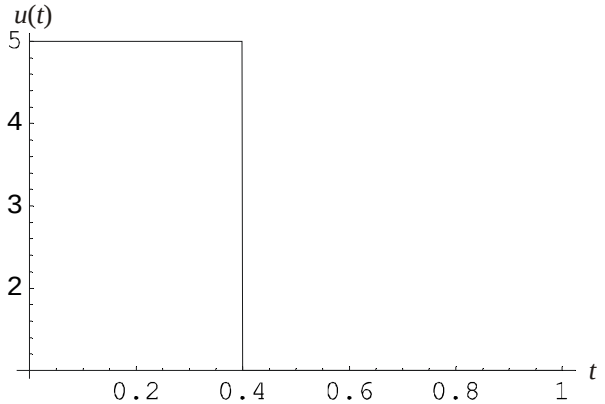
Rysunek 48 Wartość wskaźnika jakości (odpowiadająca najlepszemu osobnikowi w populacji) w funkcji generowanych pokoleń, dla rozwiązania z przykładu 1 zapewniającego najmniejszy wskaźnik jakości (punkt startowy metody rzutowanego gradientu)



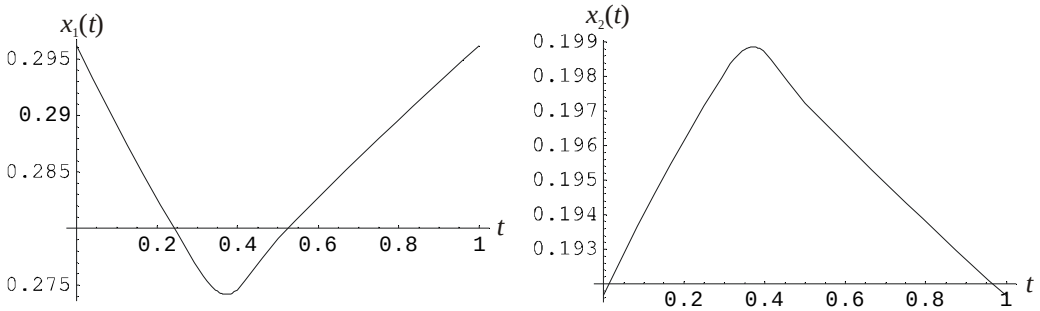
Rysunek 49 Sterowanie okresowe wyznaczone metodą rzutowanego gradientu w algorytmie hybrydowym dla zadania z przykładu 1



Rysunek 50 Stan procesu, który ze sterowaniem z rysunku 49 oraz okresem $\hat{\tau} = 0.1$ zapewnia dla problemu z przykładu 1 wskaźnik jakości $\hat{Q} = -0.196492$



Rysunek 51 Sterowanie okresowe wyznaczone metodą gradientową (startującą z optymalnego statycznego rozwiązania) dla problemu z przykładu 1

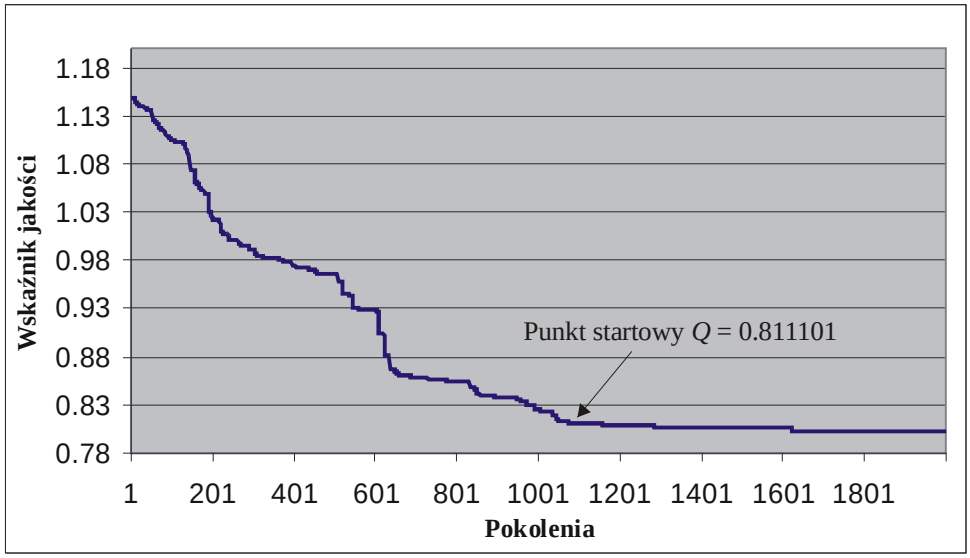


Rysunek 52 Stan procesu, który ze sterowaniem z rysunku 51 oraz okresem $\tau = 0.1$ zapewnia dla problemu z przykładu 1 wskaźnik jakości $Q = -0195360$

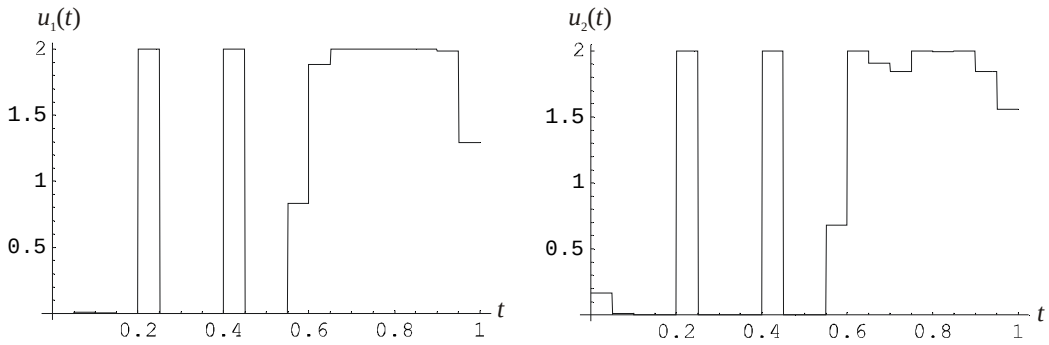
Przykład 14.

Hybrydowy algorytm optymalizacji został przetestowany również na problemie optymalnego sterowania okresowego z przykładu 4. Rysunek 53 przedstawia punkt, w którym zatrzymany został algorytm ewolucyjny i uruchomiony został algorytm gradientowy (opisany w punktu 4.1). Punktem startowym dla algorytmu gradientowego było więc (wyznaczonego przez algorytm ewolucyjny) rozwiązanie, które zapewniało wskaźnik jakości $Q = 0.81101$.

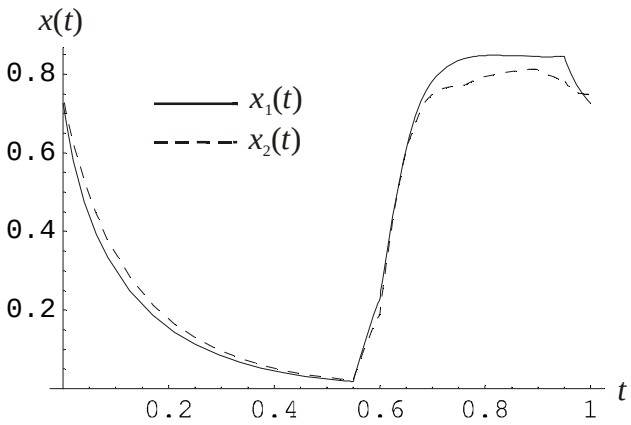
Podobnie jak w przykładzie 13, algorytm hybrydowy w krótszym czasie dostarczył rozwiązanie lepsze od rozwiązania wyznaczonego przez algorytm ewolucyjny. Metoda rzutowanego gradientu zatrzymała się bowiem po 11 krokach dostarczając rozwiązanie, dla którego wskaźnik jakości jest równy $\hat{Q} = 0.799827$ (patrz rysunki 54-56).



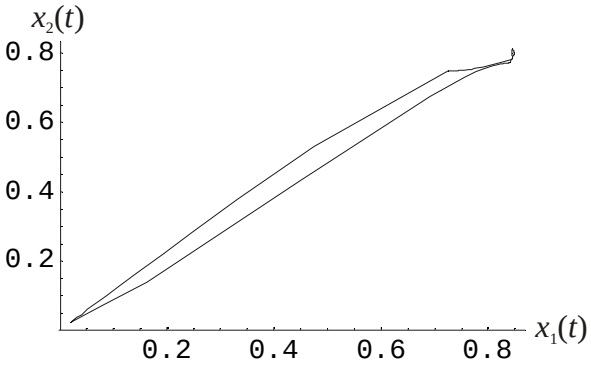
Rysunek 53 Wartość wskaźnika jakości (odpowiadająca najlepszemu osobnikowi w populacji) w funkcji generowanych pokoleń, dla rozwiązania z przykładu 4 zapewniającego najmniejszy wskaźnik jakości (punkt startowy metody rzutowanego gradientu)



Rysunek 54 Optymalne sterowanie okresowe wyznaczone algorytmem hybrydowym, dla problemu z przykładu 4



Rysunek 55 Stan procesu, który ze sterowaniem z rysunku 54 zapewnia dla problemu z przykładu 4 wskaźnik jakości $\hat{Q} = 0.799827$



Rysunek 56 Cykl, dla którego wskaźnik jakości jest równy $\hat{Q} = 0.799827$

5. PODSUMOWANIE

W rozprawie przedstawiony został hybrydowy algorytm optymalizacji, przy pomocy którego można znaleźć globalnie optymalne rozwiązanie problemu okresowego z ograniczeniami zasobowo-technologicznymi oraz (z bardzo trudnymi do uwzględniania) ograniczeniami stabilnościowymi. Wspomniany algorytm składa się: z algorytmu ewolucyjnego oraz z algorytmu gradientowego. Zadaniem algorytmu ewolucyjnego jest wyodrębnienie obszaru, w którym znajduje się cykl globalnie optymalny spełniający narzucone na niego ograniczenia. Celem algorytmu gradientowego (startującego z najlepszego rozwiązania dostarczonego przez algorytm ewolucyjne) jest natomiast wyznaczenie globalnie optymalnego rozwiązania problemu.

Zamieszczone w pracy przykłady potwierdzają, że algorytmy ewolucyjne są bardzo skutecznym „narzędziem" optymalizacyjnym. Algorytmy te pozwalają na uzyskanie rozwiązań zdecydowanie lepszych od rozwiązań dostarczanych przez dotychczas stosowane, klasyczne algorytmy optymalizacji (patrz tabela 3). W przypadku reakcji chemicznej rozważanej w przykładzie 7, algorytm ewolucyjny wyznaczył rozwiązanie okresowe zwiększające wydajność procesu nawet o 111% w porównaniu z wydajnością przy optymalnym sterowaniu wyznaczonym przez Watanabe, Kurimoto, Matsubare i Onogi [138]. Istotnym jest tutaj fakt, że algorytmy ewolucyjne – dzięki zaprojektowanemu algorytmowi rekonstrukcji ograniczeń oraz metodzie kary – dostarczają rozwiązania spełniające narzucone na nie ograniczenia (o ile takie istnieją) – np. w przykładzie 11 algorytm ewolucyjny znalazł stabilny cykl, zapewniający większą wydajność procesu niż optymalny, niestabilny cykl wyznaczony przez Matsurbare i Onogi.

Tabela 3 Porównanie najlepszych wyników uzyskanych algorytmem ewolucyjnym z wynikami uzyskanymi algorytmami optymalizacji lokalnej (klasycznymi)

Przykład	Wartość wskaźnika jakości		Poprawa rozwiązania klasycznego [%]
	Rozwiązanie wyznaczone alg. klasycznym	Rozwiązanie wyznaczone alg. ewolucyjnym	
1	-0.195	-0.19697	0.7
2	-0.0923	-0.09680	4.9
3	-0.7433	-0.771062	3.7
4	0.9522	0.802253	15.7
5	0.3234	0.241378	25.4
6	-0.225	-0.272907	21.3
7	-0.0692	-0.145961	111
9	-0.127	-0.187652	47.8
11	-0.1946	-0.205982	5.8

Także omówione w pracy gradientowe algorytmy lokalnej optymalizacji (bazujące na metodzie λ-rzutowanego gradientu) dostarczyły bardzo dobre wyniki. Przy pomocy jednego z nich zostało wyznaczone (w niewielkiej liczbie iteracji) okresowe rozwiązanie optymalne, które poprawia optymalne rozwiązanie statyczne o blisko 37% (przykład 12). Zastosowanie algorytmu gradientowego w hybrydowym algorytmie optymalizacji spełniło także wcześniejsze oczekiwania. Algorytm gradientowy zdecydowanie przyspieszył wyznaczanie rozwiązania globalnie optymalnego, a co więcej poprawił rozwiązanie dostarczone przez algorytm ewolucyjny ($Q = 0.81101$) o 1.4 % (algorytm gradientowy rozwiązywał okresowe równanie stanu 149 razy i dostarczył rozwiązanie zapewniające wskaźnik jakości $\hat{Q} = 0.799827$, podczas gdy algorytm ewolucyjny by dostarczyć rozwiązanie $\hat{Q} = 0.802253$ z rozwiązania $Q = 0.81101$ – musiał rozwiązać okresowe równanie stanu aż 11560 razy.

Mimo iż przedstawione w pracy algorytmy ewolucyjne i gradientowe – w przytoczonych przykładach – dostarczyły rozwiązań lepszych od rozwiązań wygenerowanych przez inne metody, nie są one niezawodne. Jakość rozwiązań dostarczanych przez algorytmy ewolucyjne dosyć mocno zależała od użytych operatorów genetycznych; jednak z uwagi na dużą czasochłonność algorytmów ewolucyjnych, nie wykonano tylu porównań operatorów genetycznych, by można było wyciągnąć jednoznaczne wnioski co do tego, który z zastosowanych operatorów jest lepszy. Otrzymane wyniki wskazują jednak, że algorytmy ewolucyjne z nierównomiernymi operatorami mutacji oraz z krzyżowaniem arytmetycznym dostarczają o wiele gorsze rozwiązania niż algorytmy, w których użyte zostały równomierny operator mutacji oraz równomierny operator krzyżowania. W przypadku problemu z przykładu 4 algorytmy ewolucyjne z mutacją nierównomierną i z krzyżowaniem arytmetycznym dostarczyły rozwiązania gorsze nawet od rozwiązań dostarczanych przez tradycyjne metody optymalizacji lokalnej (podobna sytuacji miała miejsce również w przykładzie 3). Z przeprowadzonych badań wynika również, że w – przypadku gdy rozwiązanie optymalne jest typu bang-bang – lepiej jest stosować mutację brzegową i krzyżowanie wymieniające. Algorytmy, w których wykorzystano takie właśnie operatory genetyczne były zbieżne do rozwiązania optymalnego szybciej, niż algorytmy z innymi operatorami genetycznymi.

Słabością opisanych algorytmów gradientowych jest natomiast fakt, że nie potrafią one, w trakcie poszukiwania optimum, uwzględniać ograniczeń stabilnościowych. Ponadto w algorytmach tych, w przypadku nietrywialnych ograniczeń uśrednionych narzuconych na sterowanie, mogą wystąpić trudności z wyznaczaniem rozwiązania tzw. λ-funkcji (184), a co za tym idzie – także rzutu sterowania. Ponieważ jednak w problemach optymalnego sterowania okresowego najczęściej występują uśrednione ograniczenia zasobowe w postaci rozpatrywanej w tej pracy, wspomniana wada nie dyskwalifikuje algorytmów gradientowych, lecz jedynie ogranicza możliwości ich wykorzystania.

Jak już wspomniano przedstawione w pracy algorytmy – choć są skutecznym narzędziem optymalizacji problemów okresowych – nie są doskonałe i można je jeszcze udoskonalać.

W przypadku algorytmów ewolucyjnych należy rozważyć możliwość wprowadzenia innego sposobu kodowania, który np. będzie zapobiegał dostarczaniu przez algorytm rozwiązań trudnych do realizacji w praktyce (patrz np. rozwiązanie z przykładu 1). Problem okresowy mógłby np. być zakodowany za pomocą następującego wektora liczb rzeczywistych

$$z = (\tau, x(0), u_K, (t_{1,} ..., t_{K-1}))$$

gdzie t_i $(i = 1,..., K-1)$ są to momenty, w których następuje przełączenie sterowania.

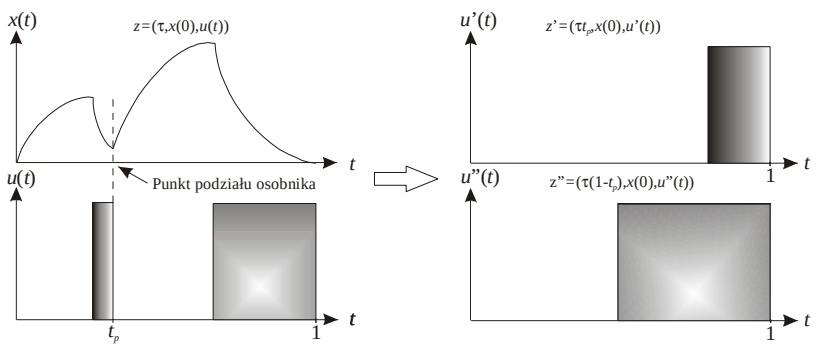
W trakcie optymalizacji, gdy zakres zmian okresu jest zbyt duży, może dochodzić do sytuacji, w której algorytm ewolucyjny będzie wyznaczał rozwiązanie będące multicyklem rozwiązania globalnie optymalnego. Dlatego też rozsądnym wydaje się prowadzenie dalszych badań nad nowymi operatorami genetycznymi, których zadaniem byłoby wyłuskiwanie najlepszego podcyklu z cyklu wielokrotnego. Mogłoby to na przykład być zrealizowane poprzez badanie kierunku pochodnej przebiegu funkcji stanu – gdy zmienia ona dosyć często swój znak, wtedy – w wybranych miejscach jej zmiany – można dokonywać podziału osobnika na kilka nowych osobników; podział ten mógłby się odbywać w sposób przedstawiony na rysunku 57 (proporcjonalnie do podziału sterowania).

Zarówno przedstawione algorytmy ewolucyjne jak i algorytmy gradientowe wykorzystują metodę Newtona-Raphsona do wyznaczania okresowego rozwiązania równań stanu. Choć metoda ta jest metodą skuteczną, to jest ona również bardzo czasochłonna i nie zawsze zbieżna; dlatego należałoby opracować nową metodę wyznaczania rozwią-

zania okresowego. Pewnym rozwiązaniem, pozwalającym uniknąć stosowania metody Newtona-Raphsona, może być zastosowanie aproksymacji równań stanu za pomocą kardynalnej bazy interpolacyjnej Lagrange'a. W metodzie tej, różniczkowe równanie stanu $\dot{x}(t) = \tau\ f(x(t), u(t))$ opisujące proces okresowy z $x(0) = x(1)$ może być zapisane w następującej równoważnej, całkowej postaci:

$$x_k = x_{k_r} + \tau \int_0^{t_k} f\left(\sum_{k=0}^{k_r} x_k e_k(\theta), \sum_{k=0}^{k_r - 1} u_k d_k(\theta)\right) d\theta\ , \quad (k = 0,1,...,k_r) \qquad (211)$$

gdzie $k_r = h_r(l_r + 1) + 1$ jest liczbą momentów siatki czasowej $\{t_k\}_{k=0}^{k_r}$ określonej za pomocą równomiernej dużej siatki $t_{h(l_r+1)} = h / h_r$ (h=0,1,...,h_r) i za pomocą nierównomiernej małej siatki z przesuniętymi węzłami Czebyszewa $t_{h(l_r+1)+l} = \gamma \cos(0.5(2l - l)\pi / l_r) + \gamma_h$, $\gamma = 0.5 / h_r$, $\gamma_h = 0.5(2h + 1) / h_r$ ($h = 0,1,...,h_r - 1$; $l = 1,2,...,l_r$); $x_k \in R^{n_x}$ jest aproksymacją stanu w chwili t_k ($x = (x_k)_{k=0}^{k_r}$); $u_k \in R^{n_u}$ jest aproksymacją sterowania w chwili t_k ($u = (u_k)_{k=0}^{k_r}$); $\{e_k(t)\}_{k=0}^{k_r}$ jest kardynalną bazą interpolacyjną Lagrange'a stopnia $l_r + 1$ stosowaną do aproksymacji równań stanu. Powyższy zapis ułatwia wprowadzenie równania (211) do funkcji kary w metodzie ewolucyjnej lub do przesuwanej funkcji kary w metodzie gradientowej, dzięki czemu nie trzeba będzie używać metody Newtona-Raphsona w celu znalezienia rozwiązania okresowego. Jako, że większość optymalnych rozwiązań okresowych ma charakter segmentowy (dla których trajektoria stanu jest przedziałami gładka z możliwymi przedziałami stałości stanu osiąganymi na brzegu obszaru dopuszczalnego), zastosowanie kardynalnej bazy interpolacyjnej Lagrange'a do aproksymacji równań stanu może przyczynić się do zwiększenia szybkości zbieżności algorytmów do rozwiązania optymalnego.



Rysunek 57 Schemat podział osobnika

6. LITERATURA

[1] E. M. Abulesz, G. Lyberatos: Periodic optimization of continuous microbial growth processes, *Biotechnology and Bioengineering*, vol. 29, 1059-1065, 1987.

[2] A. H. Aguirre, S. B. Rionda, C. A. C. Coello, G. L. Lizárraga, E. M. Montes: Handling constraints using multiobjective optimization concepts, *International Journal for Numerical Methods in Engineering*, vol. 59, 1989-2017, 2004.

[3] J. Arabas, Z. Michalewicz, J. Mulawka: GAVaPS – a genetic algorithm with varying population size, *In: IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on Evolutionary Computation*, Vol. 1. 73-78, 1994.

[4] J. Arabas: *Wykłady z Algorytmów Ewolucyjnych*, Wydawnictwa Naukowo – Techniczne, Warszawa 2001.

[5] P. Arcara, S. Bittanti: Periodic control of helicopter rotors for attenuation of vibrations in forward flight, *IEEE Transactions on Control Systems Technology*, vol. 8, no. 6, 2000.

[6] J. E. Bailey: Necessary conditions for optimality in a general class of non-linear mixed boundary value control problems, *International Journal of Control*, vol. 16, 311-320, 1972.

[7] J. E. Bailey: Periodic operation of chemical reactors: A review, *Chemical Engineering Communication*, vol. 1, pp. 111-124, 1973.

[8] T. Back, and F. Hoffmeister: Extended selection mechanism in genetic algorithms, *In Proceedings of the fourth International Conference on Genetic Algorithms*, 92-99, 1991.

[9] J. E. Bailey, F. J. M. Horn: Comparison between two sufficient conditions for improvement of an optimal steady-state process by periodic operation, *Journal of Optimization Theory and Applications*, vol. 67, 378-384, 1971.

[10] J. E. Bailey, F. J. M. Horn, R. C. Lin: Cyclic operation of reaction systems: Effects of heat and mass transfer resistance, *AIChE Journal*, vol 17, 818-825, 1971.

[11] V. Barbu: Optimal control of linear periodic resonant systems in Hilbert space, *SIAM Journal on Control and Optimization*, vol. 35, 2137-2156, 1997.

[12] V. Barbu, N. H. Pavel: Periodic optimal control in Hilbert space, *Applied Mathematics and Optimization*, vol. 33, 169-188, 1996.

[13] D. S. Bernstein: Control constraints, abnormality, and improved performance by periodic control, *IEEE Transactions on Automatic Control*, AC-30, 367-376, 1985.

[14] D. S. Bernstein, E. G. Gilbert: Optimal periodic control: the π test revisited, *IEEE Transactions on Automatic Control*, vol. 25, 673-684, 1980.

[15] S. Bittanti, G. Fronza, G. Guardabassi: Periodic control: A frequency domain approach, *IEEE Transactions on Automatic Control*, AC-18, 33-38, 1973.

[16] S. Bittanti, G. Guardabassi: Optimal periodic control and periodic systems analysis: an overview, *Proceedings of 25th Conference on Decision and Control*, Athens, Greece, 1417-1423, 1986.

[17] S. Bittanti, A. Locatelli, C. Maffezzoni: Second-variation methods in periodic optimization, *Journal of Optimization Theory and Applications*, vol. 14, 31-49, 1974.

[18] W. L. Chan, S. K. Sg: Normality and proper performance improvement in periodic control, *Journal of Optimization Theory and Applications*, vol. 29, 215-229, 1979.

[19] C. Chen, C. Hwang, R. Yang: Optimal periodic forcing of nonlinear chemical processes for performance improvement, *The Canadian Journal of Chemical Engineering*, vol. 72, 672-682, 1994.

[20] C. Chen, C. Hwang, R. Yang: Optimal periodic forcing of nonlinear chemical processes for performance improvements, *Canadian Journal of Chemical Engineering*, vol. 72, 672-682, 1994.

[21] A. Chmiel: *Biotechnologia*, PWN, Warszawa, 1991.

[22] S. B. Choi, H. Hwang: Dynamic optimization of production planning problem with periodic operation, *International Journal of Systems Science*, vol. 17, 1163-1174, 1986.

[23] C. H. Chuang, H. Morimoto: Periodic optimal cruise for a hypersonic vehicle with constraints, *Journal of Spacecraft and Rockets*, vol. 34, 165-171, 1997.

[24] C. H. Chuang, J. L. Speyer: Periodic optimal hypersonic scramjet cruise, *Optimal Control Applied Methods*, vol. 8, 231-242, 1987.

[25] F. H. Clarke: *Optimization and Nonsmooth Analysis*, Wiely, New York, 1983.

[26] C. A. C. Coello: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a Survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering*, vol. 191, 1245-1287, 2002.

[27] C. A. C. Coello: Treating constraints as objectives for single-objective evolutionary optimization, *Engineering Optimization*, vol. 32, 275-308, 2000.

[28] E. A. Coddington, N. Levinson: *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955.

[29] F. Colonius: On optimal periodic control and nested optimization problems, *Journal of Optimization Theory and Applications*, vol. 50, 525-533, 1986.

[30] F. Colonius: Optimality for periodic control of functional differential systems, *Journal of Mathematical Analysis and Applications*, vol 120, 119-149, 1986.

[31] F. Colonius: *Optimal Periodic Control*, Springer-Verlag, New York, 1988.

[32] F. Colonius: Optimal periodic control: a scenario for local properness, *SIAM Journal on Control and Optimization*, vol. 26, 452-464, 1988.

[33] F. Colonius, W. Kliemann: Infinite time optimal control and periodicity, *Applied Mathematics and Optimization*, vol. 20, 113-130, 1989.

[34] M. Deklerk, M. Gatto: Some remarks on periodic harvesting of a fish population, *Mathematical Biosciences*, vol. 56, 47-69, 1981.

[35] B. P. Demidowicz: *Matematyczna Teoria Stabilności*, Wydawnictwo Naukowo-Techniczne, Warszawa 1972.

[36] V. F. Demyanov, A. M. Rubinov: *Approximate Methods in Optimization Problems*, American Elsevier, New York, 1970.

[37] J. E. Dennis, R. B. Schnabel: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[38] L. D. Dewell, J. L. Speyer: Fuel-optimal periodic control and regulation in constrained hypersonic flight, *Journal of Guidance Control and Dynamics*, vol. 20, 923-932, 1997.

[39] P. Dorato, H. K. Knudsen: Periodic optimization with applications to solar energy control, *Automatica*, vo. 15, 673-676, 1979.

[40] J. M. Douglas: *Process Dynamics and Control*, Englewood Cliffs, New Jersey, Prentice Hall, 1972.

[41] J. C. Dunn: Global and asymptotic convergence rate estimates for a class of projected gradient processes, *SIAM Journal on Control Optimization*, vol. 19, 368-400, 1981.

[42] W. R. Emanuel, R. J. Mulholland: Linear periodic control with applications to environmental systems, *International Journal of Control*, vol. 24, 807-820, 1976.

[43] R. T. Evans, J. L. Speyer, C. H. Chuang: Solution of a periodic optimal control problem by asymptotic series, *Journal of optimization Theory and Applications*, vol. 52, 343-364, 1987.

[44] F. A. Farhadpour: Optimal lumped-parameter periodic processes, *International Journal of Control*, vol. 44, 91-115, 1986.

[45] F. A. Farhadpour, L. G. Gibilaro: On the optimal unsteady state operation of a continuous stirred tank reactor, *Chemical Engineering Science*, vol. 36, 143-147, 1981.

[46] M. Farkas: *Periodic Motions.* Springer, New York, 1994.

[47] R Farmani, J. A. Wright: Self-adaptive fitness formulation for constrained optimization, *IEEE Trans. Evolutionary Computation*, vol. 7, 445-455, 2003.

[48] M. Fjeld: On periodic processes, Report 67-54-D, *Division of Automatic Control, The Technical University of Norway*, 1967.

[49] M. Fjeld: Optimal control of multivariable periodic processes, *Automatica*, vol. 5, 487-506, 1969.

[50] L. J. Fogel, A. J. Owens, M. J. Walsh: *Artificial Intelligence through Simulated Evolution*, John Wiley, New York, 1966.

[51] A. Friedman, S. Huang, J. Young: Optimal periodic control for the two-phase Stefan problem, *SIAM Journal Control and Optimization*, vol. 26, 23-41, 1988.

[52] E. G. Gilbert: Vehicle cruise: improved fuel economy by periodic control, *Automatica*, vol. 12, 159-166, 1976.

[53] E. G. Gilbert: Optimal periodic control: a general theory of necessary conditions, *SIAM Journal on Control Optimization*, vol. 15, 717-746, 1977.

[54] E. G. Gilbert, M. G. Parsons: Periodic control and the optimality of aircraft cruise, *Journal of Aircraft*, vol. 13, 828-830, 1978.

[55] D. E. Goldberg: Sizing Populations for serial and parallel genetic algorithms, *Proc. Third International Conf. on Genetic Algorithms*, Morgan Kaufmann, 1989.

[56] D. E. Goldberg: Real-coded genetic algorithms, virtual alphabets, and blocking, *University of Illinois at Urbana-Champaign*, Technical Report No. 90001, September 1990.

[57] D. E. Goldberg, K. Deb: A comparative analysis of selection used in genetic algorithms, *Gregory Rawlins, editor. Foundations of Genetic Algorithms, San Mateo, CA: Morgan Kaufmann Publishers*, 69-93, 1991.

[58] D. E. Goldberg: *Algorytmy Genetyczne i Ich Zastosowania*, WNT, Warszawa 1995.

[59] H. Grabmüller, U. Hoffmann, K. Schädlich: Prediction of conversion improvements by periodic operation for isothermal plug-flow reactors, *Chemical Engineering Science*, vol. 40, 951-960, 1985.

[60] G. Guardabassi: Optimal steady-state versus periodic optimization: a circle criterion, *Ricerce di Automatica*, vol. 2, 1971.

[61] G. Guardabassi: The optimal periodic control problem, *Journal A.*, vol. 17, 75-83, 1976.

[62] G. Guardabassi, A. Locatelli and S. Rinaldi: Status of periodic optimization of dynamical systems, *Journal of Optimization Theory and Application*, vol. 12, 1-20, 1974.

[63] R. Guthke, W. A. Knorre: Efficiency on the cyclic bath antibiotic fermentation, *Biotechnology and Bioengineering*, vol. 24, 2129-2136, 1982.

[64] M. Han, G. Feichtinger, R. F. Hartl: Nonconcavity and proper optimal periodic control, *Journal of Economic Dynamics and Control*, vol. 975-990, 1994.

[65] E. Heinzle, J. Moes, I. J. Dunn: The influence of cyclic glucose feeding on continuous baker's yeast culture, *Biotechnology Letters*, vol. 7, 235-240, 1985.

[66] J. H. Holland: Outline for a logical theory in adaptive systems, *Journal of the Association for Computing Machinery*, Vol. 3, 297-314, 1962.

[67] J. H. Holland: *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[68] F. J. M. Horn, R. C. Lin: Periodic processes: a variational approach, *I&EC Process Design and Development*, vol. 6, 21-30, 1967.

[69] S. B. Hsu, P. Waltman: Analysis of a model of two competitors in a chemostat with an external inhibitor, *SIAM Journal on Applied Mathematics*, vol. 52, 528-540, 1991.

[70] J. Joines, C. Houck: On the use of non-stationary penalty functions to solve constrained optimization problems with genetic algorithms, *IEEE International Symposium Evolutionary Computation*, Orlando, 579-584, 1994.

[71] K.A. De Jong and W.M. Spears: A formal analysis of the role of multi-point crossover in genetic algorithms, *Artificial Intelligence*, vol. 51-26, 1992.

[72] M. De Klerk, M. Gatto: Some remarks on periodic harvesting of a fish population, *Mathematical Biosciences*, vol. 56, 47-69, 1981.

[73] G. P. Kumar, I. V. Sastry, M. Chidambaram: Periodic operation of a bioreactor with input multiplicities, *Canadian journal of Chemical Engineering*, vol. 71, 766-770, 1993.

[74] C. S. Lee, P. Ang: A simple model for seaweed growth and optimal harvesting strategy, *Ecological Modelling*, vol. 55, 67-74, 1991.

[75] X. J. Li: Maximum principle of optimal periodic control for functional differential systems, *Journal of Optimization Theory and Applications*, vol. 50, 421-429, 1986.

[76] H. S. Liu, S. C. Wu, S. C. Lin: Theoretical analysis of frequency response in continuous bioreactor, *Chemical Engineering Science*, vol. 51, 2471-2477, 1996.

[77] R. Marzollo: Periodic Optimization, Vols. 1 and 2, Springer, New York, 1972.

[78] M. Matsubara, K. Onogi: Structure analysis of periodically controlled nonlinear systems via the stroboscopic approach, *IEEE Transactions on Automatic Control*, vol. 22, 678-680, 1977.

[79] M. Matsubara, K. Onogi: Stabilized suboptimal periodic control of a chemical reactor IEEE Transactions on Automatic Control Vol. 3, 1005-1008, 1978.

[80] M. Matsubara, K. Onogi: Unstable suboptimal periodic control of a certain chemical reactor, *IEEE Transactions on Automatic Control*, vol. 23, 1111-1113, 1978.

[81] H. Maurer, C. Büskens, G. Feichtinger: Solution techniques for periodic control problems: A case study in production planning, *Optimal Control Applications and Methods*, vol. 19, 185-203, 1998.

[82] R. M. May: *Stability and Complexity in Model Ecosystems*, Princeton University Press, 1973

[83] Z. Michalewicz: *Algorytmy Genetyczne + Struktury Danych = Programy Ewolucyjne*, Wydawnictwa Naukowo – Techniczne, Warszawa 1996.

[84] Z. Michalewicz: A survey of constraint handling techniques in evolutionary computation methods, in *Proc. 4th Annual Conference on Evolutionary Programming*, J. R. McDonnel, R. G. Reynolds, and D. B. Fogel, Eds., 135-155, 1995.

[85] Z. Michalewicz and M. Schoenauer: Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation*, vol. 4, 1-32, 1996.

[86] H. Moed, S. Rinaldi: Problem of minimum interception in ecosystems via periodic control, *Proceedings of Seventh Triennial world Congress of IFAC: a Link be-*

tween Science and Applications of Automatic Control, Helsinki, Oxford, Pergamon Press, 2065-2077, 1979.

[87] K. Nitka-Styczeń: Some computational methods for optimal periodic hereditary control problems. *Materiały III krajowej konferencji naukowo-technicznej: Automatyzacji nawigacji i systemów sterowania*, Gdynia Akademia Marynarki Wojennej, vol. B, 133-140, 1991.

[88] K. Nitka-Styczeń: The heredity shift operator in descent approach to the optimal hereditary control problem, *International Journal of Systems Science*, vol. 28, 705-719, 1997.

[89] E. J. Noldus: Periodic optimization of a chemical reactor system using perturbation methods, *Journal of Engineering Mathematics*, vol. 11, 49-66, 1977.

[90] K. Onogi, M. Matsubara: Structure analysis of periodically controlled chemical processes, *Chemical Engineering Science*, vol. 34, 1009-1019, 1980.

[91] F. Özgülsen, R. A. Adomaitis, A. Cinar: A numerical method for determining optimal parameter values in forced periodic operation, *Chemical Engineering Science*, vol. 47, 605-613, 1992.

[92] S. J. Parulekar: Analysis of forced periodic operations of continuous bioprocesses - single input variations, *Chemical Engineering Science*, vol. 53, 2481-2502, 1998.

[93] S. J. Parulekar, R. S. Waghmare, H. Lim: Yield optimization for multiple reactions, *Chemical Engineering Science*, vol. 43, 3077-3091, 1988.

[94] S. Pavlou, L. Kevrekidis, G. Lyberatos: On the coexistence of competing microbial species in a chemostat under cycling, *Biotechnology and Bioengineering*, vol. 35, 224-322, 1990.

[95] M. E. Pittelkau: Optimal periodic control for spacecraft pointing and attitude determination, *Journal of Guidance Control and Dynamics*, vol. 16, 1078-1084, 1993.

[96] H. Pohlheim: *GEATbx: genetic and evolutionary algorithm toolbox for use with MATLAB*, Ver. 1.92, 1997/1998.

[97] E. Polak: *Optimization. Algorithms and Consistent Approximations*, Springer-Verlag, New York, 1997.

[98] I. Rechenberg: *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Friedrich Fromann Verlag, Stuttgart, 1973.

[99] P. H. Richter, J. Ross: The efficiency of engines operating around a steady-state at finite frequencies, *Journal of Chemical Physics*, vol. 69, 5521-5530, 1978.

[100] S. M. Robinson: Strongly regular generalized equations, *Mathematics of Operations Research*, vol. 5, 43-62, 1980.

[101] L. I. Rozonoer, A. M. Tsirlin: Optimalnoe upravlenie termodinamičeskimi processami, *Avtomatika i Telemechanika*, no. 3, 55-64, 1983.

[102] L. Ruan, X. D. Chen: Comparison of several periodic operations of a continuous fermentation process, *Biotechnol. Progr.*, vol. 12, 286-288, 1993.

[103] D. Rutkowska, M. Piliński, L. Rutkowski: *Sieci Neuronowe, Algorytmy. Genetyczne i Systemy Rozmyte*, Wydawnictwo Naukowe PWN, Warszawa 1997.

[104] K. Shädlich, U. Hoffmann, H. Hofmann: Periodical operation of chemical processes and evaluation of conversion improvements, *Chemical Engineering Science*, vol. 38, 1375-1384, 1983.

[105] N. Z. Shor: *Minimization Methods for Nondiferentiable Functions and its Applications*, Naukova Dumka, Kiev, 1979.

[106] D. Sinčić, J. E. Bailey: Analytical optimization and sensitivity analysis of forced periodic chemical processes, *Chemical Engineering Science*, vol. 35, 1153-1161, 1980.

[107] M. Skowron: Modelowanie algorytmu ewolucyjnego dla problemów optymalnego sterowania okresowego z ograniczeniami energetycznymi, *artykuł zaakceptowany do druku w materiałach IV Krajowej konferencji Modelowanie i symulacja. MiS-4, Kościelisko 19-23 czerwca 2006.*

[108] M. Skowron: Evolutionary algorithm for globally optimal periodic control problems with different kinds of constraints, *Raporty Inst. Inform. Autom. Robot. Pwroc.*, Ser. PRE nr 43, 2005 (artykuł zgłoszony do publikacji w *Computers & Chemical Engineering* – obecnie oczekuje na recenzję).

[109] M. Skowron: Metoda λ-rzutowanego gradientu dla problemów optymalnego sterowania okresowego z ograniczeniami zasobowymi, *V Ogólnopolskie Warsztaty Doktoranckie, OWD Istebna-Zaolzie, 18-21 października 2003, Komitet Organizacyjny Sympozjum PPEE i Seminarium BSE 2003*, Z. 4., vol. 17, 237-241, 2003.

[110] M. Skowron, K. Styczeń: Adaptacja optymalnego sterowania okresowego, *Inżynieria wiedzy i systemy ekspertowe, Wrocław, 11-13 czerwca 2003, Oficyna Wydaw. PWroc.*, T1, 264-270, 2003.

[111] M. Skowron, K. Styczeń: Algorytm ewolucyjny globalnej optymalizacji procesów okresowych z ograniczeniami stabilnościowymi, *XV Krajowa Konferencja Automatyki, Warszawa, 27-30 czerwca 2005, Instytut Badań Systemowych PAN*, T. 2, 21-26, 2005.

[112] M. Skowron, K. Styczeń: *Aproksymacja sklejana dla segmentowych problemów optymalnego sterowania okresowego*. Systemy sterowania. Praca zbiorowa pod red. Włodzimierza Greblickiego i Czesława Smutnickiego, Warszawa: WKŁ, 269-280, 2005.

[113] M. Skowron, K. Styczeń: Evolutionary search for globally optimal constrained stable cycles, *Chemical Engineering Science* (artykuł wstępnie przyjęty do druku).

[114] M. Skowron, K. Styczeń: Evolutionary search for globally optimal ecological process, *Sieci i systemy informatyczne, Teoria, projekty, wdrożenia, aplikacje - XII Konferencja, Łódź, 21-22 października 2004*, T. 2, 543-549, 2004.

[115] M. Skowron, K. Styczeń: Metoda λ-rzutowanego gradientu dla problemów optymalnego sterowania okresowego z ograniczeniami energetycznymi, *Przegląd Elektrotechniczny*, nr 11, 1162-1165, 2004.

[116] M. Skowron, K. Styczeń: Mixed gradient projection and shifted penalty method for constrained optimal periodic control problems, *Raporty Inst. Cybern. PWroc*., Ser. PRE nr 66, 2003.

[117] M. Skowron, K. Styczeń: Modelowanie metody λ-rzutowanego gradientu dla problemów optymalnego sterowania okresowego z ograniczeniami energetycznymi, *III Krajowa konferencja Modelowanie i symulacja. MiS-3, Kościelisko, 21-25 czerwca 2004*. T. 2., 575-582, 2004.

[118] M. Skowron, K. Styczeń: Optimal periodic control with resource constraints via the lambda-projected gradient method, *Raporty Inst. Cybern. PWroc*., Ser. PRE nr 63, 2003 (artykuł zgłoszony do publikacji w *Computational Optimization and Applications* – obecnie oczekuje na recenzję).

[119] M. Skowron, K. Styczeń: Pomiary w układach optymalnego sterowania okresowego ze sprzężeniem zwrotnym, *Raporty Inst. Cybern. PWroc*. Ser. PRE nr 28, 2004.

[120] M. Skowron, K. Styczeń: The λ-projected gradient method for constrained pseudoperiodic control problems, *Proceedings of the 15th International Conference on*

Systems Science, Eds Zdzisław Bubnicki, Adam Grzech. Wrocław, 7-10 September 2004, Wrocław: Oficyna Wydaw. PWroc., vol. 2, 72-79, 2004.

[121] M. Skowron, K. Styczeń: Wpływ opóźnień pomiarowych na strukturę układu sterowania okresowego, *Metrologia w procesie poznania, Kongres Metrologii KM, Materiały kongresowe Pod red. Janusza Mroczki, Wrocław, 6-9.09.2004*, Katedra Metrologii Elektronicznej i Fotonicznej, Wydział Elektroniki PWroc., T. 2, 687-690, 2004.

[122] H. Smith, P. Waltman: *The Theory of the Chemostat. Dynamics of Microbial Competition*, Cambridge: University Press, 1995.

[123] W. M. Spears: *The Role of Mutation and Recombination in Evolutionary Algorithms*, Ph. D. Thesis, Fairfax, VA: George Mason University, 1998.

[124] J. L. Speyer: Periodic optimal flight, *Journal of Guidance, Control, and Dynamics*, vol. 19, 745-755, 1996.

[125] J. L. Speyer, D. Dannemiller, D. Walker: Periodic optimal cruise of an atmospheric vehicle, *Journal of Guidance, Control and Dynamics*, vol. 8, 31-38, 1985.

[126] J. L. Speyer, R. T. Evans: A second variational theory for optimal periodic processes, *IEEE Transactions on Automatic Control*, vol 29, 138-147, 1984.

[127] J. L. Speyer: Nonoptimality of the steady-state cruise for aircraft, *AIAA Journal*, vol. 14, 1604-1610, 1976.

[128] G. Stephanopoulos, A. G. Fredrickson, R. Aris: The growth of competing microbial population in CSTR with periodically varying inputs, *AIChE Journal*, vo. 25, 863-873, 1979.

[129] K. Styczeń: Trigonometric approximation of optimal periodic control problems, *International Journal of Control*, vol. 43, 1531-1542, 1986.

[130] K. Styczeń, K. Nitka-Styczeń: Trigonometric approximation of optimal periodic control problems for systems with inertial controllers, *IEEETransaction on Automatic Control*, vol. 34, 1102-1105, 1989.

[131] G. Syswerda: Uniform crossover in genetic algorithms, *in Proc. 3rd Int. Conf. Genetic Algorithms*, 2-9, 1989.

[132] T. Takahama, S. Sakai: Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations, *IEEE Transactions on Evolutionary Computation*, vol. 9, 437-451, 2005.

[133] J. Timonen, R. P. Hamalainen: Optimal periodic control strategies in dynamic pricing problem, *International Journal of Systems Science*, vol. 10, 197-205, 1979.

[134] S. Varigonda, T. Georgiou, P. Daoutidis: Numerical solution of the optimal periodic control problem using differential flatness, *IEEE Transactions on Automatic Control*, vol. 49, 271-275, 2004.

[135] Q. Wang, J. L. Speyer: Necessary and sufficient conditions for local optimality of a periodic process, *SIAM Journal on Control and Optimization*, vol 28, 482-487, 1990.

[136] N. Watanabe, H. Kurimoto, M. Matsubara: Periodic control of continuous stirred tank reactors - I, *Chemical Engineering Science*, vol. 36, 809-818, 1981.

[137] N. Watanabe, H. Kurimoto, M. Matsubara: Periodic control of continuous stirred tank reactors - III, *Chemical Engineering Science*, vol. 39, 31-36, 1984.

[138] N. Watanabe, H. Kurimoto, M. Matsubara, K. Onogi: Periodic control of continuous stirred tank reactors - II, *Chemical Engineering Science*, vol. 37, 745-752, 1982.

[139] N. Watanabe, K. Onogi, M, Matsubara: Periodic control of continuous stirred tank reactors - I. The criterion and its applications to isothermal cases, *Chemical Engineering Science*, vol. 36, 809-818, 1981.

[140] M. Weill, G. Lutfalla, K. Mogensen, F. Chandre, A. Berthomieu, C. Berticat, N. Pasteur, A. Philips, P. Fort & M. Raymond: Comparative genomics: Insecticide resistance in mosquito vectors, *Nature*, vol. 423, 136–137 (2003).

[141] A. P. Wierzbicki, S. Kurcyusz: Projection on a cone, penalty functionals and duality theory for problems with inequality constraints in Hilbert space, *SIAM Journal on Control and Optimization*, vol. 15, 25-56, 1977.

[142] S. Wolfram: *Mathematica. A System For Doing Mathematics by Computer*, Addison-Wesley, California, 1994.

[143] Q. Yanl, V. Kapilal, and A. G. Sparks: Pulse-Based Periodic Control for Spacecraft Formation Flying, *Proceedings of the American Control Conference Chicago*, Illinois June 2000.

[144] N. R. Zagalsky, R. E. Irons, and R. L. Schultz: The energy state approximation and minimum-fuel fixed-range trajectory, *Journal of Aircraft*, vol. 8, pp. 488-490, 1971.

[145] W. B. Zhang; Economic cycles in an inflationary open economy, *International Journal of Systems Science*, vol. 20, 139-149, 1989.

Najważniejsze oznaczenia i skróty

$|\cdot|$ - norma Euklidesowa,

$\| s \|_\infty = \max_{1 \le i \le n_x} | s_i |$,

$\tilde{\varphi}_+^d = (\max(0, \varphi_i^d))_{i=1}^{n_{\tilde{\varphi}}}$,

$\phi, \tilde{\phi}$ - ograniczenia technologiczne,

$\varphi, \tilde{\varphi}$ - ograniczenia zasobowe,

$\Phi(x, u, \tau)$ - macierz monodromii,

f_x - macierz pierwszych pochodnych cząstkowych funkcji f,

g - numer pokolenia,

$\mathcal{H}^r([0,\tau]; R^n)$ -przestrzeń n wymiarowych funkcji rzeczywistych z r-tą pochodną, całkowalnych z kwadratem w przedziale $[0,\tau]$ i wyposażonych w normę $\| x \| = (\int_0^\tau \sum_{\rho=0}^{r} | x^{(\rho)}(t) |_2^2 \, dt)^{1/2}$,

J - wskaźnik jakości zredukowany do przestrzeni sterowania i okresu,

K - liczba chwil czasowych, w których zdefiniowane jest sterowanie dyskretne u_K

G_{max} - maksymalna liczba pokoleń,

n - rozmiar wektora,

N – liczebność populacji,

N_t - liczba osobników biorących udział w turnieju w turniejowej metodzie selekcji,

p_c - prawdopodobieństwo krzyżowania,

p_m - prawdopodobieństwo mutacji,

P^g - populacja w pokoleniu g,

$P_{\mathcal{U}}$, $P_{\mathcal{T}}$ - rzuty: dyskretnego sterowania i okresu,

q - wskaźniki charakterystyczne procesu

Q - wskaźnik jakości problemu okresowego,

$\overline{Q}$ - wskaźnik jakości problemu statycznego,

Q_Φ - funkcja przystosowania,

s - wartości własne macierzy monodromii,

t - czas,

u - sterowanie,

x - stan procesu,

z - osobnik – zakodowane rozwiązanie okresowe,

$z^{\min}$ - osobnik, dla którego funkcja celu Q przyjmuje najmniejszą wartość,

$z^{\max}$ - osobnik, dla którego funkcja celu Q przyjmuje największą wartość,

z', z'' - nowe chromosomy (osobniki) utworzone w wyniku działania operatorów genetycznych,

problem GOSO - problem globalnie optymalnego sterowania okresowego