

AKADEMIA GÓRNICZO-HUTNICZA

GŁĘBOKIE UCZENIE I INTELIGENCJA OBLICZENIOWA

AUTOMATYKA I ROBOTYKA II STOPIEŃ

2019/2020

---

# Optymalizacja fabryki z wykorzystaniem algorytmu immunologicznego (selekcji klonalnej)

---

*Skład zespołu:*

Artur Bauer  
Kamil Szostek  
Sławomir Goździewski  
Wiktor Filipiak

*Opiekun:*

dr hab. inż. Joanna Kwiecień

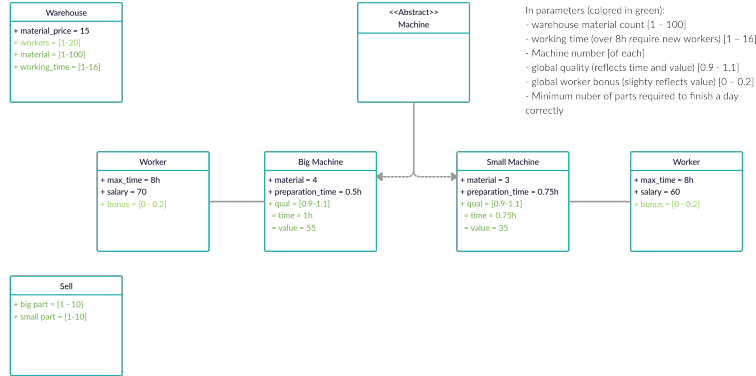
Złożono: 14 maja 2020

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Model fabryki . . . . .	2
1.1.1	Funkcja celu fabryki: . . . . .	2
1.1.2	Kara . . . . .	2
1.1.3	Liczba pracowników . . . . .	3
1.1.4	Maksymalna ilość elementów . . . . .	3
1.1.5	Rzeczywisty czas pracy maszyny na 1 produkt . . . . .	3
1.2	Parametry modelu . . . . .	3
<b>2</b>	<b>Badany problem</b>	<b>3</b>
2.1	Przegląd literatury . . . . .	3
<b>3</b>	<b>Diagram UML fabryki</b>	<b>4</b>
<b>4</b>	<b>Algorytm</b>	<b>5</b>
4.1	Pseudokod . . . . .	5
<b>5</b>	<b>Testy</b>	<b>5</b>
5.1	Wykresy . . . . .	7
5.1.1	Wpływ populacji . . . . .	7
5.1.2	Wpływ iteracji . . . . .	9
5.1.3	Wpływ współczynnika klonowania . . . . .	11
5.1.4	Wpływ watchdog'a . . . . .	13
5.1.5	Wpływ współczynnika wybieranych komórek . . . . .	15
5.2	Wnioski . . . . .	17
5.2.1	Wpływ zmian rozmiaru populacji . . . . .	17
5.2.2	Wpływ zmian liczby iteracji . . . . .	17
5.2.3	Wpływ zmian współczynnika klonowania . . . . .	17
5.2.4	Wpływ zmian watchdog-a . . . . .	18
5.2.5	Wpływ zmian współczynnika wybranych komórek . . . . .	18
<b>6</b>	<b>Podsumowanie</b>	<b>18</b>

# 1 Wstęp

## 1.1 Model fabryki



Rysunek 1: Factory scheme

### 1.1.1 Funkcja celu fabryki:

$$Income = \sum_{i=1}^{n_p} (p_i * (v_i - m_i * m_p)) - (1 + b_i) * \sum_{i=1}^{n_w} (w_i * s_i * t_{wi}) - m_r * m_p - punish$$

Gdzie:

- $n_p$  – ilość rodzajów części
- $p_i(n_m)$  – ilość wyprodukowanych części i-tego typu
- $v_i(v_{bi}, t_{wi}, t_{bi}, w_q)$  – wartość części i-tego typu
- $m_i$  – liczba surowca potrzebna do wytworzenia elementu i-tego typu
- $m_p$  – cena surowca
- $n_w$  – liczba rodzajów pracowników
- $w_i$  – liczba pracowników i-tego rodzaju
- $s_i$  – wypłata pracownika i-tego rodzaju
- $b$  – premia pracownicza
- $m_r(p_i, n_m)$  – pozostały materiał
- $p_{i_{min}}$  – minimalna ilość elementów do wytworzenia i uniknięcia kary
- $p_{i_{max}}$  – maksymalna ilość wytworzonych elementów
- $n_m$  – liczba surowca na początek dnia

### 1.1.2 Kara

$$punish = p_{un} * \sum_{i=1}^{n_p} (p_{num_i}) * v_i$$

$$p_{num_i} = \begin{cases} 0 & \text{if } p_{i_{min}} - p_i \leq 0 \\ p_{i_{min}} - p_i & \text{if } p_{i_{min}} - p_i > 0 \end{cases}$$

Gdzie:

- $p_{un}$  – współczynnik kary
- $p_{num_i}(p_{i_{min}}, p_i)$  – liczba elementów i-tego typu dla których naliczana jest kara

### 1.1.3 Liczba pracowników

Liczba pracowników i-tego typu jest równa ilości maszyn i-tego typu:

$$n_p = n_w$$

### 1.1.4 Maksymalna ilość elementów

Niezbędna ilość elementów i-tego typu:

$$\sum_{i=1}^{n_p} p_{i_{max}} * m_i < n_m$$

### 1.1.5 Rzeczywisty czas pracy maszyny na 1 produkt

$$t_{wi} = t_{pi} + p_i * t_{bi}$$

## 1.2 Parametry modelu

Parametr	oznaczenie	wartość
Ilość surowców	$n_m$	$[x - 100 \ x]$
Koszt surowca	$m_p$	4
Czas pracy	$t_f$	[1 - 16]
Minimalna ilość dużych części	$p_{0_{min}}$	[0 - 10]
Minimalna ilość małych części	$p_{1_{min}}$	[0 - 10]
Wypłata operatora dużej maszyny	$s_0$	19
Wymagana ilość materiału na duży element	$m_0$	6
Czas przygotowania dużej maszyny	$t_{p0}$	1h 45 min
Wartość dużego elementu	$v_{b0}$	70
Podstawowy czas pracy na duży element	$t_{b0}$	1h
Liczba dużych maszyn	$c_0$	[0 - 30]
Wypłata operatora małej maszyny	$s_1$	17
Ilość surowca na mały element	$m_1$	4
Czas przygotowania małej maszyny	$t_{p1}$	1h 25 min
Wartość małego elementu	$v_{b1}$	50
Czas wytworzenia małego elementu	$t_{b1}$	1h 25 min
Ilość małych maszyn	$c_1$	[0 - 30]
Maksymalny czas pracy pracownika	$t_w$	8h
Bonus pracowniczy	b	[0.0 - 0.5]
Współczynnik kary	$p_{un}$	[0 - 1]

Gdzie:

- $x$  – ilość wymaganych elementów \* koszt części
- Parametry wejściowe podane są w kwadratowych nawiasach
- Pracownik jest zatrudniony na pełen etat (8h płacone z góry)
- Pierwsza i druga zmiana są identyczne w ilość i rodzaj maszyn i pracowników
- Rezerwujemy surowce na wymagane elementy
- Wszystkie elementy ponad wymaganą liczbę są ekstra dochodem

## 2 Badany problem

### 2.1 Przegląd literatury

Artykuł dotyczy algorytmu selekcji klonalnej stosowanej do optymalizacji w elektromagnetyce. Autorzy prezentują ich własną koncepcję kodowanego algorytmu selekcji klonalnej, który może zostać użyty w elektromagnetycznej optymalizacji projektu, a także sposób działania algorytmu dla problemu "The TEAM Workshop problem 22".[1].

Artykuł przedstawia zastosowanie sztucznego systemu immunologicznego w aplikacji przemysłowej. Na podstawie parametrów obróbki (siła, moment, itp.) oraz zakłócenia (wibracje, itp.) autorzy wykrywają uszkodzenie narzędzia. Wykorzystywany jest algorytm sztucznego systemu immunologicznego wykorzystuje do działania algorytm selekcji negatywnej [2].

Artykuł przedstawia użycie algorytmów sztucznych systemów immunologicznych w przemyśle. Porównuje on algorytmy sztucznej inteligencji z algorytmem klonowania do algorytmu z mechanizmem uczenia społecznego. Zmieniając wzmocnienie, czas zdwojenia oraz czas wyprzedzenia dobierają one nastawy regulatora PID [6].

Artykuł dotyczy algorytmu selekcji klonalnej opartego na algorytmach memetycznych stosowanego do problemów planowania zadań. Autorzy skupili się na poprawie eksploracji i eksploatacji przy użyciu algorytmu selekcji klonalnej. W artykule przedstawiono użycie selekcji klonalnej do skonstruowania ewolucyjnego mechanizmu wyszukiwania wykorzystywanego do eksploracji.[8].

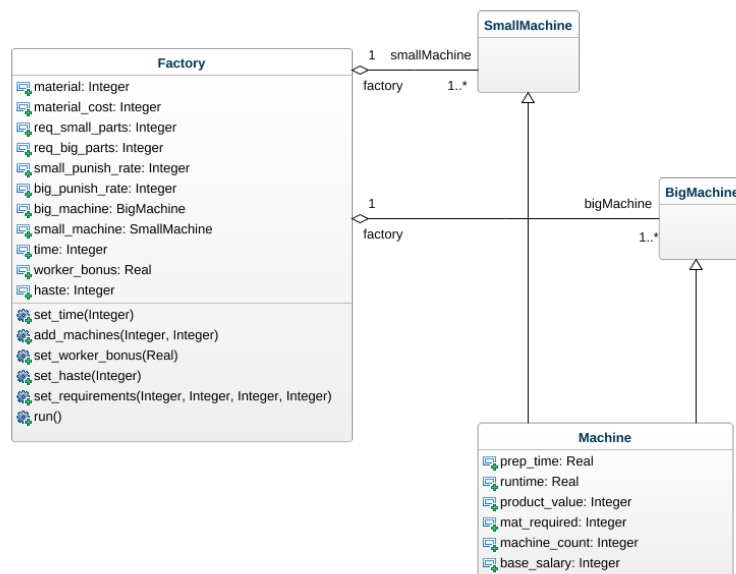
Artykuł przedstawia użycie Algorytmu Selekcji Klonalnej w zastosowaniach inżynierskich. Opisane w nim jest działanie algorytmu od strony teoretycznej, a także działanie zaimplementowanego przez autorów algorytmu przy rozwiązywaniu trzech różnych problemów: binarnego rozpoznawania znaków, wielomodalnej optymalizacji funkcji -  $f(x, y) = x \cdot \sin 4\pi x - y \cdot \sin 4\pi y + \pi + 1$  i problemu komiwojażera dla 30 miast [3].

Artykuł przedstawia użycie Algorytmu Selekcji Klonalnej do optymalizacji ułożenia terenu budowy. Zaprezentowany algorytm minimalizuje koszty produkcji i dystans przebyty pomiędzy n obiektami zaprezentowanymi za pomocą macierzy permutacji o wymiarach  $n \times n$  [7].

Artykuł przedstawia działanie sztucznego systemu immunologicznego (AIS) w przypadku rozwiązania pojemnościowego problemu marszrutyzacji. Celem było znalezienie odpowiedniego zestawienia parametrów algorytmu selekcji klonalnej w celu rozwiązania problemu poprzez podejście eksperymentalne. W artykule oprócz działania AIS, opisano także działanie innych metod rozwiązujących dwadzieścia instancji problemu i przedstawiono wyniki pod względem jakości rozwiązań oraz wykorzystanego czasu obliczeniowego [5].

Artykuł dotyczy zastosowania algorytmu selekcji klonalnej w celu określenia optymalnych punktów pracy w niskonapięciowych, hybrydowych mikrosieciach AC/DC. Celem było zminimalizowanie strat mocy czynnej, kosztów eksploatacji oraz optymalizacja napięcia węzłowego [4].

### 3 Diagram UML fabryki



Rysunek 2: Diagram UML

## 4 Algorytm

Do optymalizacji fabryki zdecydowano się na wybór algorytmu immunologicznego z wykorzystaniem selekcji klonalnej.

Algorytm ten wykorzystuje populację przeciwciał, które są odpowiednikami punktów w wielowymiarowej przestrzeni rozwiązań. Najpierw w sposób losowy tworzona jest populacja przeciwciał które zostają zoptymalizowane. Proces optymalizacji polega na poszukiwaniu przeciwciał maksymalizujących (lub minimalizujących) funkcję celu. Z populacji wybierane są najlepsze przeciwciała które następnie są klonowane. Spośród wybranych przeciwciała o najlepszym dopasowaniu (wyniku) są klonowane wielokrotnie, natomiast ilość klonów przeciwciał o niższym dopasowaniu jest niższa. Następnie klony są poddawane mutacjom których wielkość jest odwrotnie proporcjonalna do ich jakości. Klony o największym dopasowaniu mutowane są delikatnie, natomiast klony o mniejszym dopasowaniu podlegają większym mutacjom. Następnie każde oryginalne przeciwciało jest z pewnym prawdopodobieństwem zastępowane klonem (o ile ten jest lepszej jakości). Proces jest powtarzany do momentu w którym w kolejnych populacjach nie będą dostrzegalne zmiany lub osiągnięta zostanie odpowiednia liczba iteracji. Pseudokod algorytmu został przedstawiony w algorytmie 1.

### 4.1 Pseudokod

---

**Algorithm 1:** Pseudokod algorytmu immunologicznego z wykorzystaniem selekcji klonalnej

---

**Data:** population\_size, selection\_rate, clone\_rate

**Result:** bestSolution

**begin**

    population  $\leftarrow$  generate\_population(population\_size);

**while** stop criteria is not met **do**

        selected  $\leftarrow$  select(population, math.ceil(selection\_rate \* population\_size));

        clones  $\leftarrow$  clone(selected, clone\_rate);

        matured  $\leftarrow$  hypermutate(clones);

        population  $\leftarrow$  replace(population, matured);

**end**

    bestSolution  $\leftarrow$  selectBest(population);

**end**

---

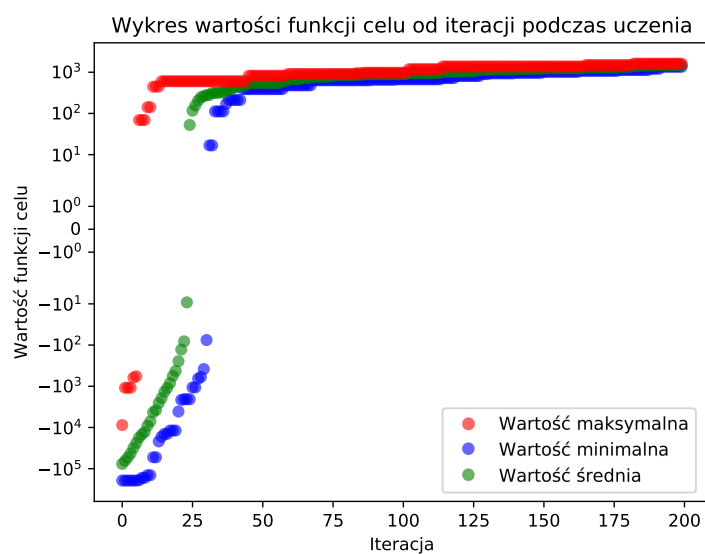
## 5 Testy

Wykonano szereg testów dla różnych parametrów algorytmu immunologicznego. Jako podstawowe parametry przyjęto: rozmiar populacji = 100, ilość iteracji = 200, współczynnik wybranych komórek = 20% współczynnik klonowania = 50% i watchdog = 50. Najlepszym wynikiem osiągniętym było 1612.46 w 183 iteracji. Wykres uczenia został przedstawiony na rys. 3.

Tablica 2: Wpływ parametrów na proces optymalizacji

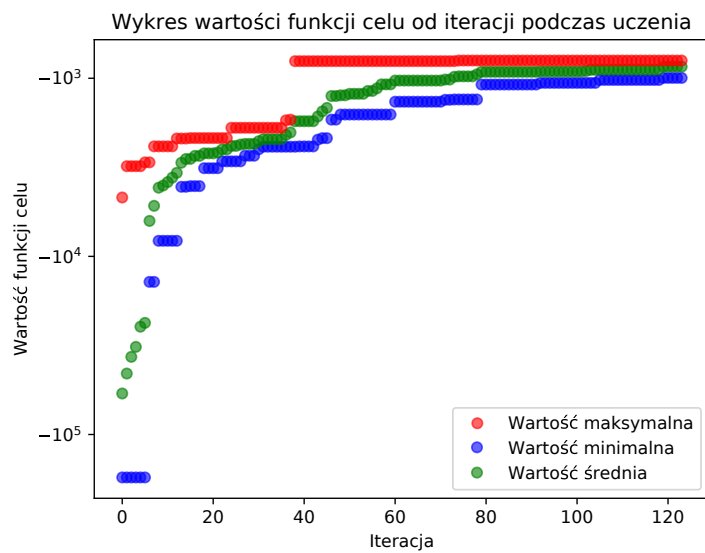
<b>Wpływ liczby populacji</b>			
Liczba populacji	Rysunek	Najlepsza wartość	Iteracja z najlepszą wartością
10	rys. 4	−795.31	74
40	rys. 5	971.91	20
200	rys. 6	1739.31	55
500	rys. 7	1825.85	94
<b>Wpływ liczby iteracji</b>			
Liczba iteracji	Rysunek	Najlepsza wartość	Iteracja z najlepszą wartością
10	rys. 8	−167.0	10
40	rys. 9	1216.45	37
100	rys. 10	844.96	73
500	rys. 11	1680.4	91
<b>Wpływ współczynnika klonowania</b>			
Współczynnik klonowania	Rysunek	Najlepsza wartość	Iteracja z najlepszą wartością
0.25	rys. 12	819.73	26
0.40	rys. 13	1547.17	113
0.60	rys. 14	1615.86	172
0.80	rys. 15	865.72	74
<b>Wpływ watchdog'a</b>			
Watchdog	Rysunek	Najlepsza wartość	Iteracja z najlepszą wartością
5	rys. 16	−101.07	19
40	rys. 17	843.33	20
60	rys. 18	948.33	59
80	rys. 19	1767.93	151
<b>Wpływ współczynnika wybranych komórek</b>			
Współczynnik wybranych komórek	Rysunek	Najlepsza wartość	Iteracja z najlepszą wartością
0.05	rys. 20	1178.58	75
0.10	rys. 21	558.54	24
0.40	rys. 22	1255.00	57
0.50	rys. 23	807.50	46

## 5.1 Wykresy



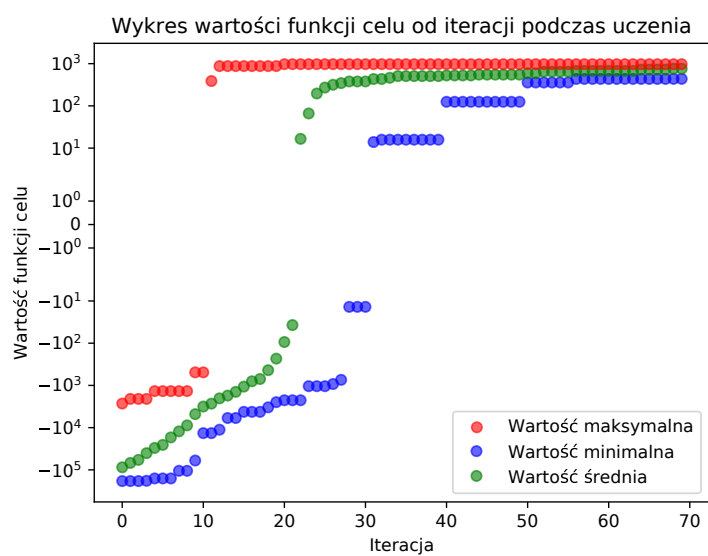
Rysunek 3: Przebieg na podstawowych parametrach

### 5.1.1 Wpływ populacji

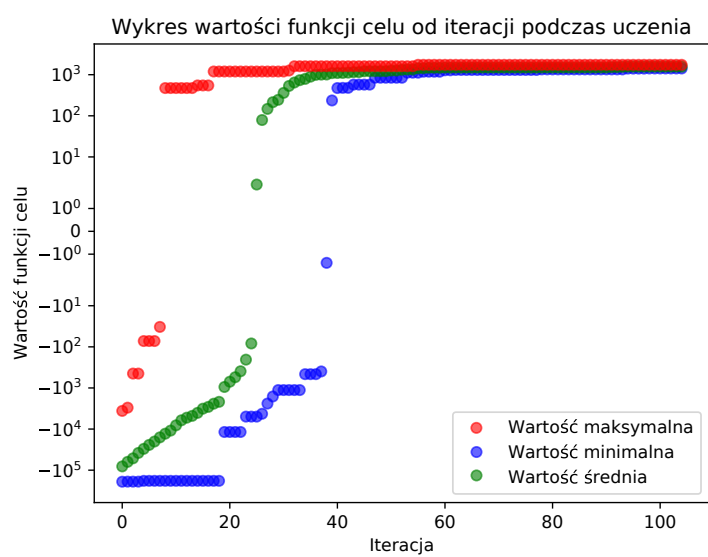


Rysunek 4: Populacja = 10

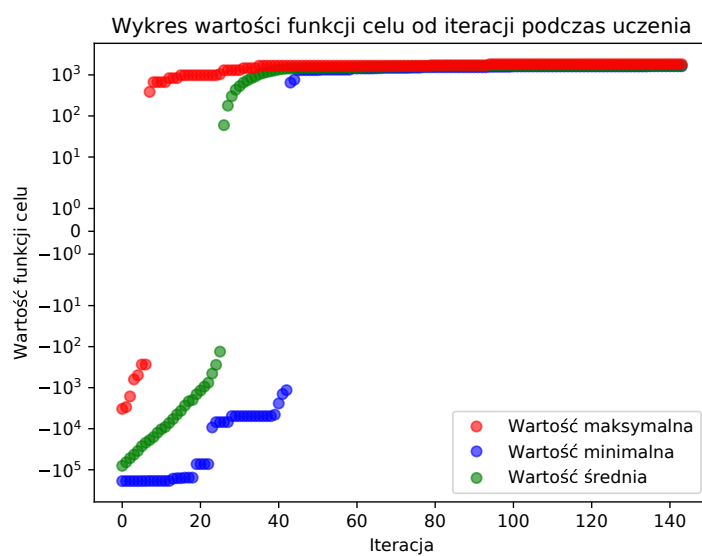




Rysunek 5: Populacja = 40

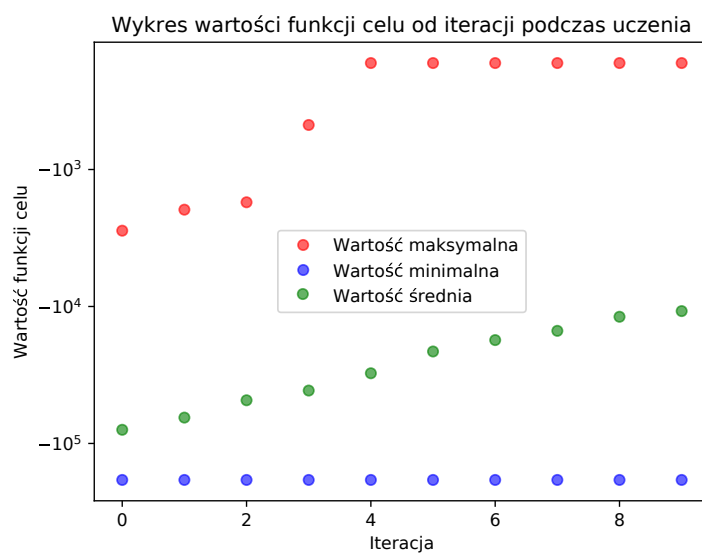


Rysunek 6: Populacja = 200

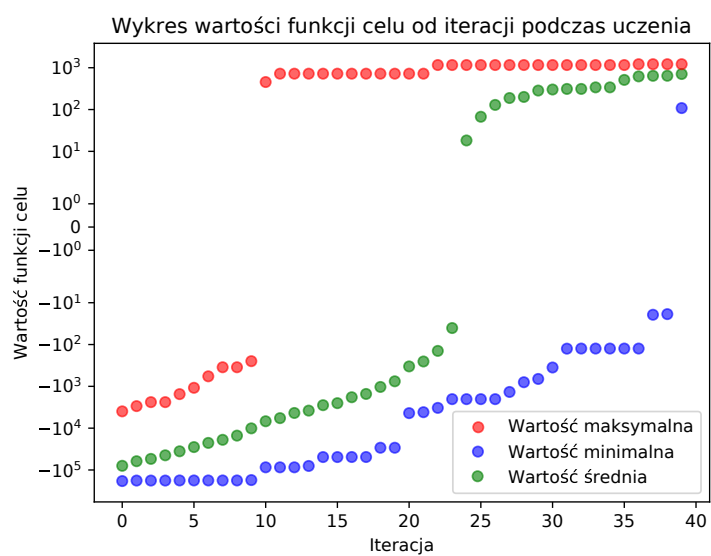


Rysunek 7: Populacja = 500

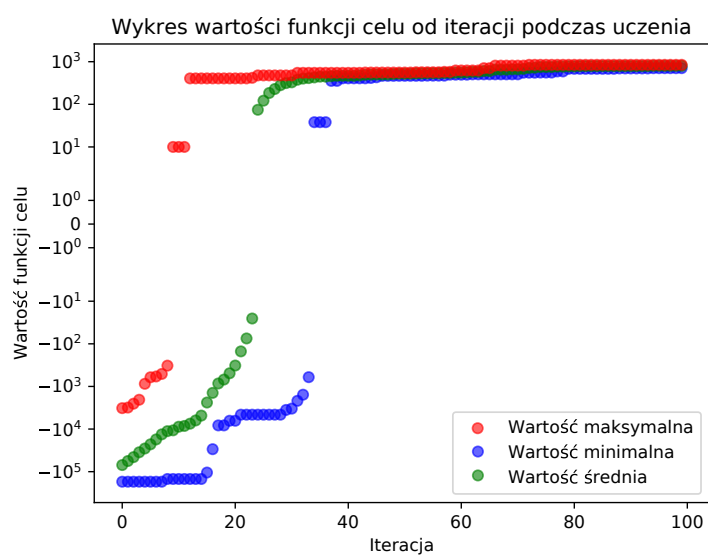
### 5.1.2 Wpływ iteracji



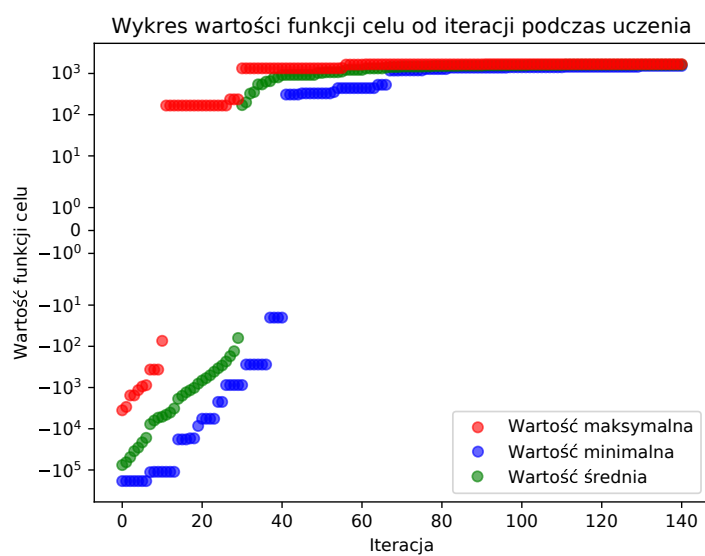
Rysunek 8: Iteracje = 10



Rysunek 9: Iteracje = 40

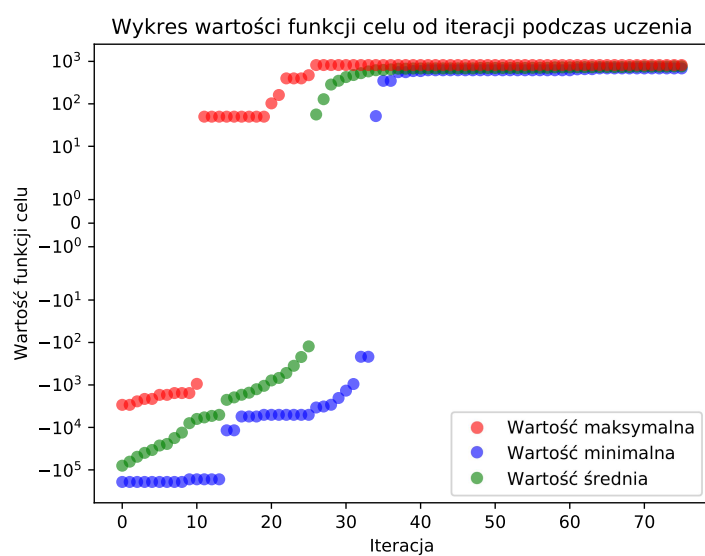


Rysunek 10: Iteracje = 100

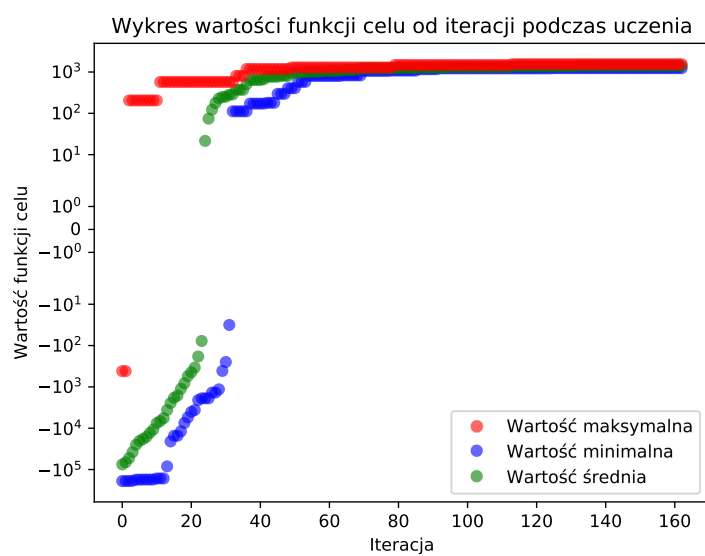


Rysunek 11: Iteracje = 500

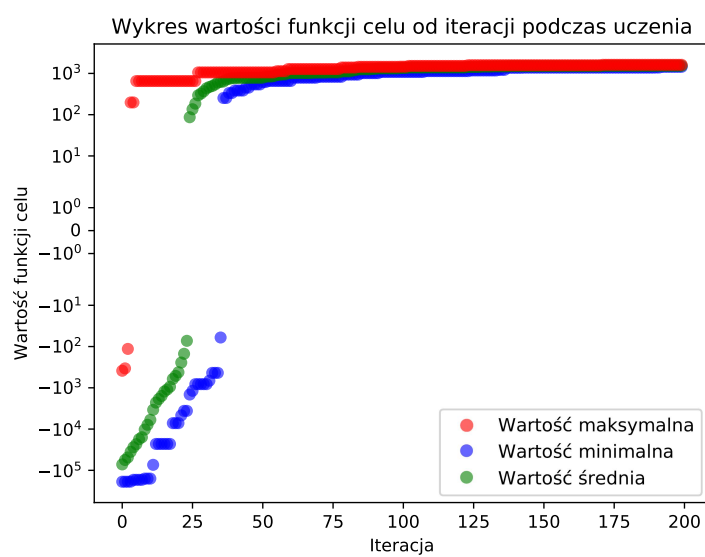
### 5.1.3 Wpływ współczynnika klonowania



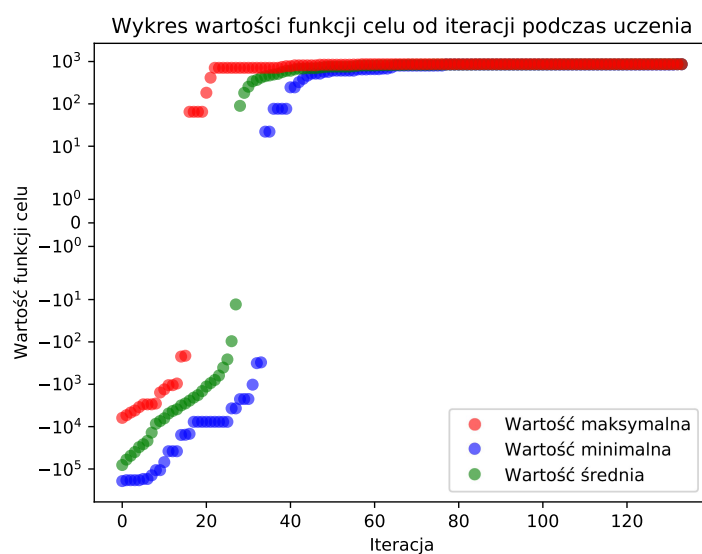
Rysunek 12: Współczynnik klonowania = 25%



Rysunek 13: Współczynnik klonowania = 40%

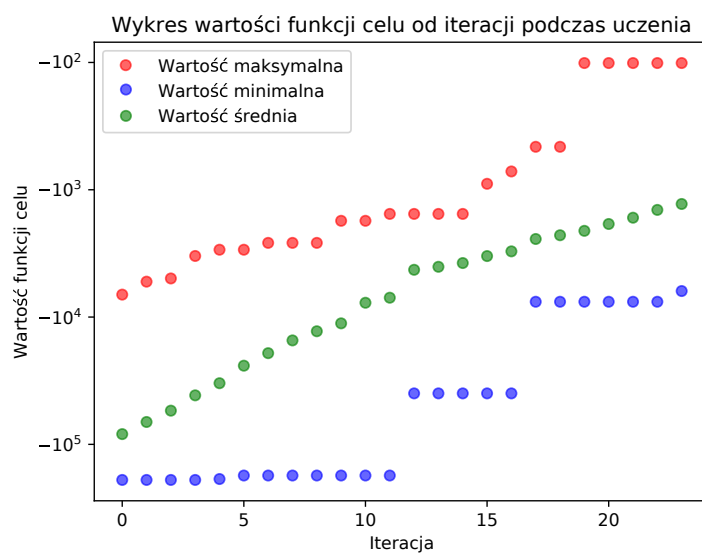


Rysunek 14: Współczynnik klonowania = 60%

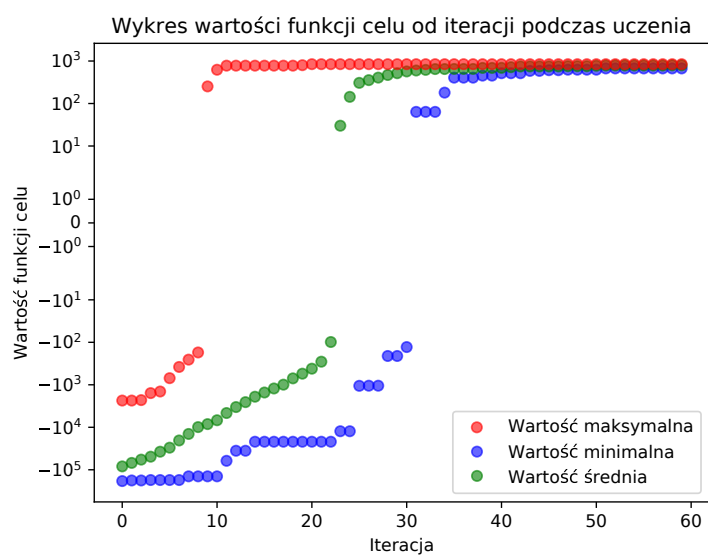


Rysunek 15: Współczynnik klonowania = 80%

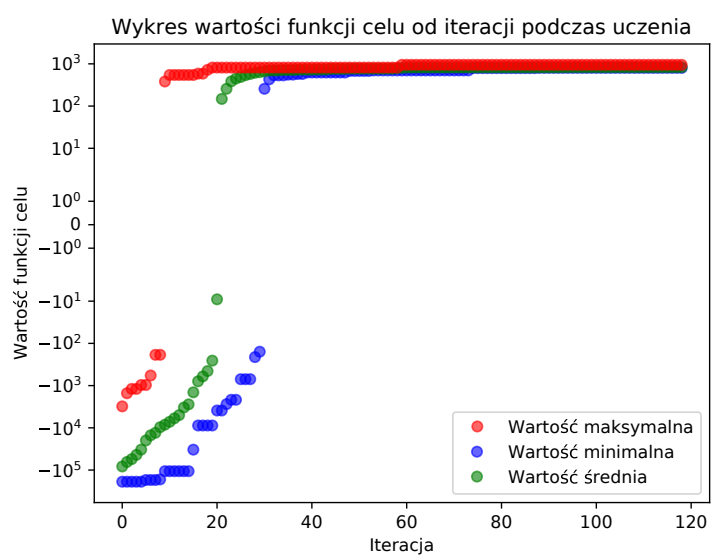
#### 5.1.4 Wpływ watchdog'a



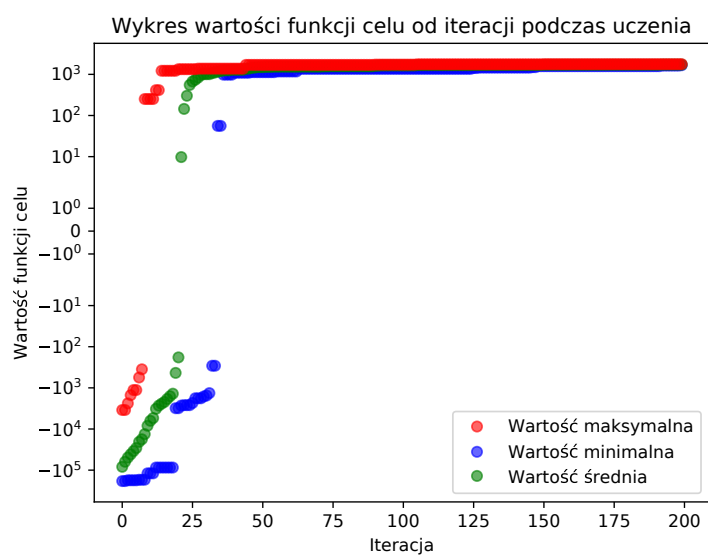
Rysunek 16: Watchdog = 5



Rysunek 17: Watchdog = 40

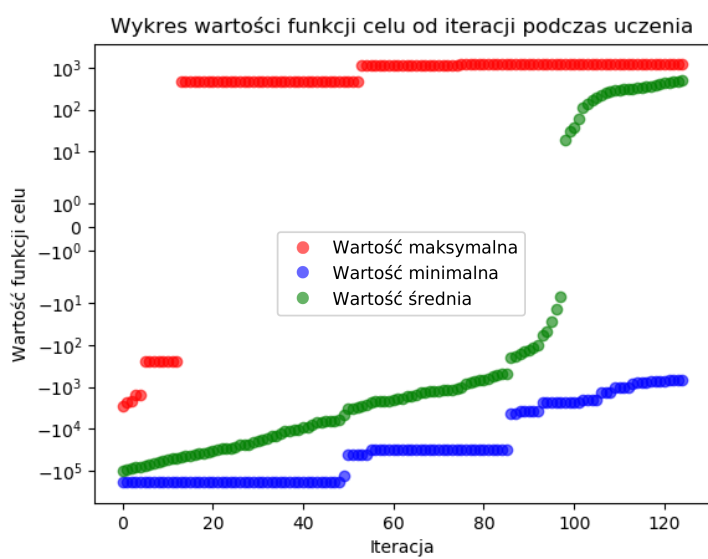


Rysunek 18: Watchdog = 60



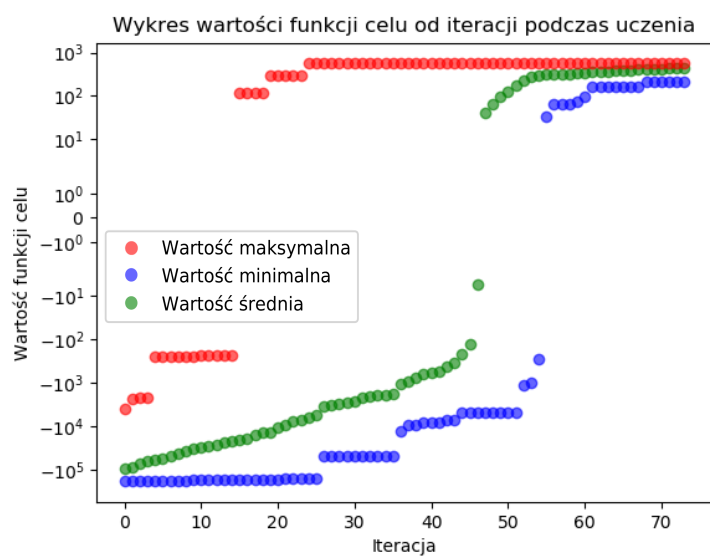
Rysunek 19: Watchdog = 80

### 5.1.5 Wpływ współczynnika wybieranych komórek

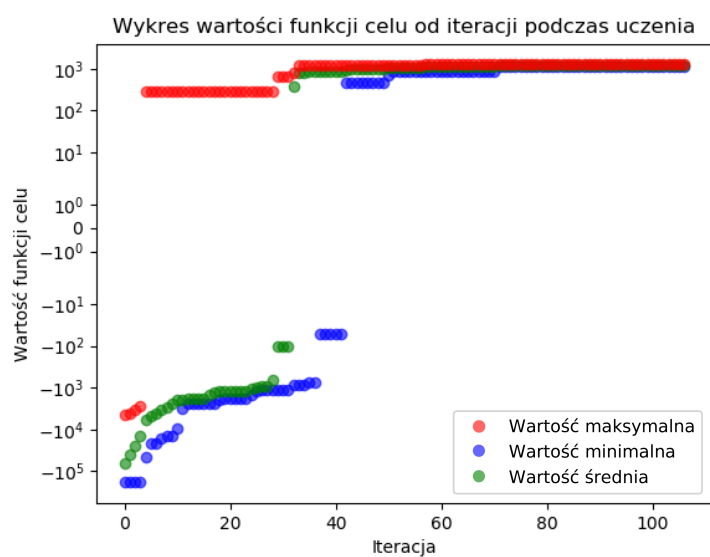


Rysunek 20: Współczynnik wybieranych komórek = 5%

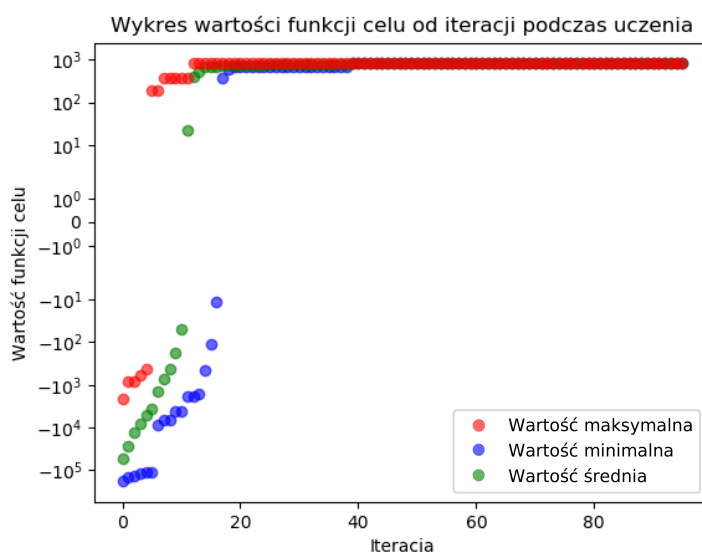




Rysunek 21: Współczynnik wybieranych komórek = 10%



Rysunek 22: Współczynnik wybieranych komórek = 40%



Rysunek 23: Współczynnik wybieranych komórek = 50%

## 5.2 Wnioski

Wnioski wyciągnięte na podstawie analizy wykonanych testów z podziałem na poszczególne parametry.

### 5.2.1 Wpływ zmian rozmiaru populacji

Na podstawie przeprowadzonych testów można stwierdzić że rozmiar populacji mocno wpływa na wyniki symulacji. Wraz z wzrostem rozmiaru diametralnie poprawia się najlepszy osiągalny wynik. Im większa populacja tym więcej rozwiązań problemu porównujemy jednocześnie co, przy braku zmian pozostałych parametrów, naturalnie wiąże się z większą szansą na znalezienie lepszego rozwiązania. Dodatkowo można zauważyć że dla niewielkich populacji algorytm kończył się przed 200 (standardową) iteracją. Może być to powodowane tym że cała populacja "utyka" w lokalnym minimum i nie jest w stanie się z niego wydostać (ponieważ na podstawie innych testów wiemy że lepsze rozwiązania istnieją). Aby w przyszłości uniknąć takich sytuacji należało by lepiej dopasować współczynnik mutacji.

### 5.2.2 Wpływ zmian liczby iteracji

Na podstawie testów możemy stwierdzić iż liczba iteracji nie wpływa w znaczący sposób na rozwiązanie. Co było widoczne w wykonanych testach po pewnej liczbie iteracji wartość najlepszych rozwiązań staje w miejscu". Wystarczy więc by ilość iteracji była na tyle duża by algorytm dotarł do miejsca zatrzymania się znaczącej poprawy. W związku z tym prawdopodobnie najlepiej było by uzależnić liczbę iteracji od innych parametrów. To że liczba iteracji nie wpływa bezpośrednio na najlepszą wartość uzyskaną przez algorytm można potwierdzić porównując test dla 40 i 100 iteracji. Przy 40 iteracjach rozwiązania "wpadły" w lepsze minimum lokalne i najlepszy wynik wyniósł około 1200, natomiast przy 100 iteracjach rozwiązania "wpadły" w gorsze minimum lokalne i po osiągnięciu pułapu 800 mimo dużej ilości iteracji nie były w stanie ulec znacznej poprawie.

### 5.2.3 Wpływ zmian współczynnika klonowania

Z testów wynika że współczynnik klonowania wpływa na jakość rozwiązań do pewnego stopnia. Naturalnie im większa liczba klonów, tym więcej mutacji, tym większa szansa że pojawi się rozwiązanie lepsze od obecnego. Jednak wraz z wzrostem parametru mocno zmienia się złożoność obliczeniowa. Wobec tego parametr należy dobrać optymalnie pod względem poprawy jakości rozwiązania i czasu wykonania algorytmu. Dodatkowo poprawa jakości wraz z wzrostem współczynnika wydaje się być coraz mniejsza co pozwala nam przypuszczać że odpowiednie optimum istnieje. Najlepsza wartość dla współczynnika o wartości 0.8 która jest mniejsza od najlepszej wartości dla współczynnika

0.6 wydaje się być dziełem przypadku a nie spadkiem jakości rozwiązania w związku z wzrostem współczynnika. Można dojść do takich wniosków na podstawie analizy działania całego algorytmu gdzie wzrost współczynnika nie ma możliwości osłabienia jakości rozwiązania najlepszego.

#### **5.2.4 Wpływ zmian watchdog-a**

Na podstawie testów można określić wpływ parametru watchdog na niewielki. Ma on wpływ na wynik algorytmu jedynie dla jego niewielkich wartości gdy algorytm w kilku krokach nie zdąży poprawić rozwiązań a watchdog jest na tyle mały że zakończy algorytm przed dojściem do maksimum wartości. Przy wielu testach nie udało się zaobserwować sytuacji w której najlepsze rozwiązanie nie zmieniałoby się przez ponad 40 iteracji a następnie uległo znacznej poprawie. Wydaje się więc że ustalenie wartości tego parametru na więcej niż 40, przy obecnym algorytmie, mija się z celem i jedynie wydłuża działanie algorytmu.

#### **5.2.5 Wpływ zmian współczynnika wybranych komórek**

Zmiana parametru wydaje się nie mieć większego wpływu na jakość najlepszego rozwiązania. Na podstawie analizy działania algorytmu można dojść do podobnych wniosków. Jako iż algorytm zastępuje rozwiązania tylko lepszymi rozwiązaniami, wybór dodatkowych gorszych rozwiązań do stworzenia klonów które i tak z dużym prawdopodobieństwem zostaną pominięte wydaje się być nieuzasadniony.

## **6 Podsumowanie**

Algorytm immunologiczny selekcji klonalnej został wykorzystany w celu maksymalizacji funkcji celu utworzonego modelu fabryki. Przeprowadzone testy pokazują że w zależności od różnych ustawień parametrów, algorytm dąży do znalezienia najlepszego rezultatu. Uzyskane wartości funkcji celu są zadowalające. Należy mieć na uwadze, że ze względu na swój sposób działania, algorytm nie zawsze uzyska globalne optimum w każdym przypadku.

Dla ustalonych niedużych wartości parametrów liczby iteracji i rozmiaru populacji algorytm jest stosunkowo wydajny obliczeniowo, jednak niekiedy przekłada się to na gorszą jakość uzyskanych wyników. Z kolei dla większych wartości tych parametrów można wyraźnie odczuć, że zapotrzebowanie obliczeniowe jest większe, jednakże skutkuje to uzyskaniem lepszych rezultatów.

Reasumując, algorytm immunologiczny wykorzystujący zasadę selekcji klonalnej okazał się zdolny do zoptymalizowania postawionego problemu, niemniej jednak z powodzeniem może być stosowany do rozwiązywania różnych, innych problemów optymalizacyjnych.

## Bibliografia

- [1] F. Campelo i in. “A clonal selection algorithm for optimization in electromagnetics”. W: *IEEE Transactions on Magnetics* 41.5 (2005), s. 1736–1739.
- [2] Dipankar Dasgupta i Stephanie Forrest. “Artificial immune systems in industrial applications”. W: *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. IPMM’99 (Cat. No. 99EX296)*. T. 1. IEEE. 1999, s. 257–267.
- [3] Leandro De Castro i Fernando Von Zuben. “The Clonal Selection Algorithm with Engineering Applications 1”. W: (sty. 2000).
- [4] Łukasz Rokicki. “The application of the CLONALG algorithm in the process of optimal operation control of hybrid AC/DC low voltage microgrid”. W: *E3S Web of Conferences* 84 (sty. 2019), s. 02011. DOI: 10.1051/e3sconf/20198402011.
- [5] Warattapop Thapatsuwan, Pupong Pongcharoen i Peeraya Thapatsuwan. “Clonal Selection of Artificial Immune System for Solving the Capacitated Vehicle Routing Problem”. W: *Journal of Next Generation Information Technology* 4 (maj 2013), s. 167–179. DOI: 10.4156/jnit.vol4.issue3.20.
- [6] Mingan Wang i in. “An Artificial Immune System Algorithm with Social Learning and Its Application in Industrial PID Controller Design”. en. W: *Mathematical Problems in Engineering* 2017 (2017), s. 1–13. ISSN: 1024-123X, 1563-5147. DOI: 10.1155/2017/3959474. URL: <https://www.hindawi.com/journals/mpe/2017/3959474/> (term. wiz. 25.03.2020).
- [7] Xi Wang i in. “Application of Clonal Selection Algorithm in Construction Site Utilization Planning Optimization”. W: *Procedia Engineering* 145 (2016). ICSDEC 2016 – Integrating Data Science, Construction and Sustainability, s. 267 –273. ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2016.04.073>. URL: <http://www.sciencedirect.com/science/article/pii/S1877705816300789>.
- [8] Jin-hui Yang i in. “Clonal selection based memetic algorithm for job shop scheduling problems”. W: *Journal of Bionic Engineering* 5.2 (2008), s. 111–119.