

## INTERFEJS DO ZDALNEGO STEROWANIA PLOTEREM Z FUNKCJĄ ROZPOZNAWANIA OBRAZÓW

Mateusz KAMIŃSKI, Marcin MORAWIEC

Politechnika Gdańska, Wydział Elektrotechniki i Automatyki  
e-mail: matkaminsky@gmail.com, marcin.morawiec@pg.edu.pl

**Streszczenie:** Maszyny numeryczne takie jak obrabiarki CNC, plotery czy drukarki 3D są coraz powszechniejsze w użytku. Na Politechnice Gdańskiej przygotowano pracę magisterską [12], której rezultaty przedstawiono w niniejszym artykule. Ze względu na objętość referatu, przedstawiono jedynie wybrane aspekty budowy plotera, aplikacji na urządzenie mobilne oraz przegląd zastosowanych algorytmów optymalizacji pod kątem szybkości rysowania. Aplikacja funkcjonuje w systemie Android, a komunikuje się z maszyną za pomocą interfejsu bluetooth. Aplikacja oferuje także możliwość optymalizacji tras, zarówno tych rysowanych, jak i nierysowanych, pokonywanych przez ploter. Jest to problem komiwojażera bez powrotu, który rozwiązany może być poprzez algorytmy: zachłanny, genetyczny, wspinaczkowy lub symulowanego wyżarzania.

**Słowa kluczowe:** CNC, silnik krokowy, ploter, problem komiwojażera, algorytm wspinaczkowy, algorytm symulowanego wyżarzania, algorytm genetyczny, algorytm zachłanny, przetwarzanie obrazów, Arduino, GRBL, Open CV, Bluetooth, Android.

### 1. WSTĘP

Urządzenia numeryczne, takie jak obrabiarki CNC czy plotery, dają możliwość wykonywania dowolnych zadań w przestrzeni roboczej [1]. Biorąc pod uwagę szybki rozwój algorytmów przetwarzania obrazów oraz duży wpływ urządzeń mobilnych na rynku powstał pomysł na projekt połączenia tych zagadnień. Niewielka ilość oprogramowania w tej dziedzinie była wystarczającą motywacją do stworzenia aplikacji na system operacyjny Android [3]. Funkcje przygotowane w języku C, zaprezentowane w poniższych podrozdziałach, opracowano na podstawie [9]. W rozdziale 3 przedstawiono projekt oraz budowę plotera, w rozdziale 4 opisano sposób komunikacji pomiędzy interfejsem sterującym, a ploterem, natomiast w rozdziałach 5 i 6 omówiono problematykę rozpoznawania obrazów oraz algorytmy optymalizacji.

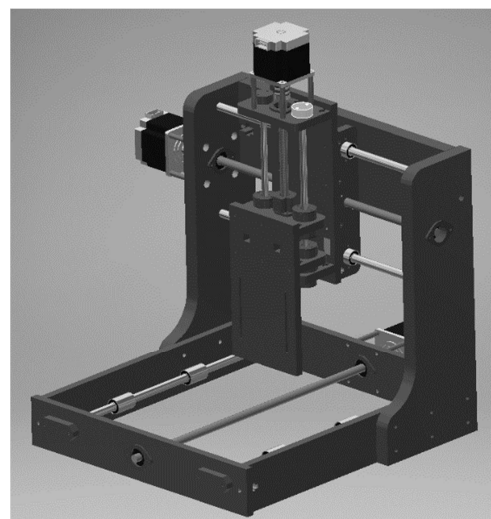
### 2. ZAŁOŻENIA DO PROJEKTU

Ważnym aspektem, który założono na początku, była odpowiednia dokładność rysowania przez urządzenie. Minimalizacja sprężystości konstrukcji oraz błędu obrotu przez silniki były kluczowe do zwiększenia precyzji [1, 2]. Kolejnym aspektem była bezproblemowa komunikacja bezprzewodowa, która nie traciłaby żadnej informacji, co może prowadzić do powstania błędów w sposób lawinowy. Ostatnie założenie to uzyskanie jak najszybszego rysowania

z zachowaniem odpowiedniej dokładności, co wykonane zostało za pomocą algorytmów minimalizujących długość trasy

### 3. WYKONANIE URZĄDZENIA

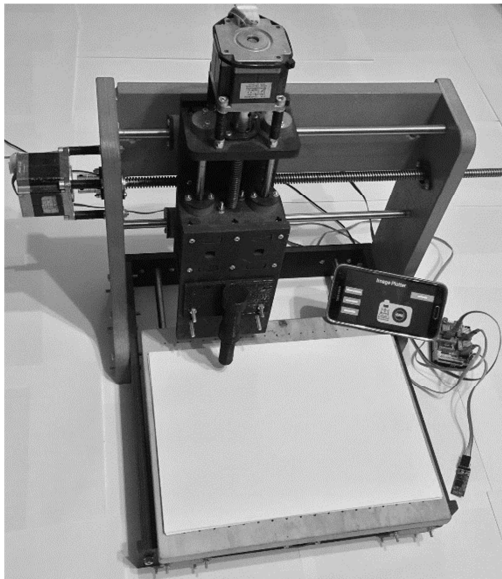
Na potrzeby pracy wykonano najpierw frezarkę CNC, która ze względu na bezpieczeństwo pracy oraz łatwość uzyskania szybkich widocznych rezultatów, wykorzystana została jako ploter. Zamiast narzędzia roboczego w postaci wrzeciona i frezu, posiada zaprojektowany uchwyt na długopis. Maszyna zbudowana została od podstaw, a inspirowana była wieloma projektami dostępnymi m. in. w Internecie [4]. Projekt jest rozszerzalny, gdyż wystarczy wykorzystać inne narzędzia robocze jak np. wrzeciono, laser, aby cała praca nabrała nowych możliwości. Na rys. 1 przedstawiony został projekt maszyny CNC, stworzony w oprogramowaniu AutoCad Inventor, który pozwolił na wizualizację 3D plotera.



Rys. 1. Trójwymiarowy model maszyny CNC

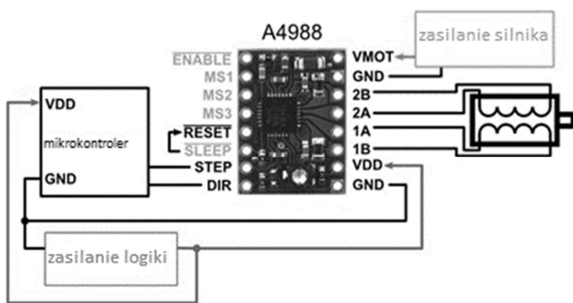
Ze względów ekonomicznych, wszystkie elementy ramy oraz stół, który nie jest pokazany na rysunku, wycięte zostały ze sklejk o grubości 1 cm oraz 2 cm. Reszta elementów mocujących wydrukowana została na drukarce 3D z materiału PLA. Zastosowane zostały łożyska liniowe wraz z prowadnicami ze stali szlifowanej, które prowadzą poszczególne osie i minimalizują niepożądane luzy. Za napęd posłużyły silniki krokowe wraz ze śrubami

trapezowymi o średnicy 12 mm i nakrętkami. Całość konstrukcji po zbudowaniu zaprezentowano na rys. 2.



Rys. 2. Rzeczywiste wykonanie urządzenia

Jako interfejs sterujący wykorzystano Arduino Uno, które jest otwartym projektem [5, 6] z nakładką (z ang. shield) na wyprowadzenia. Osiągnięto przez to sporą miniaturyzację całego sterownika. Pozwala ona w łatwy sposób skonstruować moduł sterujący drukarką 3D lub maszyną CNC. Posiada cztery złącza dla sterowników silników krokowych, które zostaną przedstawione w dalszej części. Płytkę jest kompatybilną z oprogramowaniem interpretującym G-code, wykorzystanym w tej pracy, jakim jest GRBL [4]. Moduły umożliwiają sterowanie silnikiem poprzez wykorzystanie dowolnego urządzenia generującego odpowiednie stany logiczne, w tym przypadku Arduino Uno. Uproszczony schemat podłączenia przedstawiony jest na rys. 3.

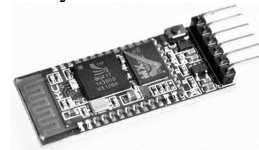


Rys. 3. Uproszczony schemat połączenia

#### 4. KOMUNIKACJA

Do skomunikowania plotera z interfejsem stworzonym na platformie Android wykorzystany został moduł Bluetooth HC-05, który to pracuje w ustandaryzowanej technologii, co eliminuje problem kompatybilności. Wykorzystuje on protokół Bluetooth v2.0 + EDR. Moduł pozwala łączyć się na odległość dziesięciu metrów. Posiada on możliwość pracy jako slave lub master, czyli potrafi inicjować połączenie, jak i podporządkować się do innego urządzenia inicjalizującego. Podłączony jest on przez port UART. Maksymalna prędkość transmisji to 115200. Wielkość wysyłanej ramki to 8 bitów danych oraz 1 bit stopu. Pobierany prąd to około 80 mA

w trakcie parowania i około 8 mA w trakcie transmisji. Wymiary tego modułu to zaledwie 38 mm na 16 mm, co czyni go bardzo przydatnym, małym urządzeniem komunikacyjnym o małym poborze prądu. Moduł przedstawiony jest na rys. 4.



Rys. 4. Moduł Bluetooth

Przykładowy kod programu do obsługi komunikacji z urządzeniem Bluetooth przedstawiono poniżej:

```
myBluetoothAdapter=
BluetoothAdapter.getDefaultAdapter();
if (myBluetoothAdapter == null) {
    if (toast != null) toast.cancel();
    toast = Toast.makeText(getApplicationContext(),
"Device does not support Bluetooth",
Toast.LENGTH_LONG);
    toast.show();
} else if (!myBluetoothAdapter.isEnabled()) {
    Intent activeBluetooth = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    startActivityForResult(activeBluetooth,
IS_ACTIVE);
}
```

W metodzie *onActivityResult* sprawdzany jest przypadek opisany jako *IS\_ACTIVE*, czyli sprawdzane jest, czy moduł jest aktywny, czy nie. Jeśli użytkownik wyraził zgodę to powiadomiony zostanie on stosownym komunikatem, iż bluetooth został aktywowany.

#### 5. PRZETWARZANIE OBRAZÓW

W automatyzacji stanowisk pracy bazuje się na wizualizacji, poprzez wykorzystanie systemów wizyjnych, które są elektronicznym odpowiednikiem wzroku ludzkiego. Coraz więcej dziedzin nauki czerpie korzyści wynikające z korzystania z takich systemów wraz z przetwarzaniem obrazu. Open CV (Open Source Computer Vision Library) [7, 8] jest to otwarte źródło wizji komputerowej i biblioteka oprogramowania do uczenia maszynowego. Open CV zostało stworzone, aby wspomóc infrastrukturę aplikacji wizyjnych. Biblioteka ta posiada ponad dwa tysiące pięćset algorytmów optymalizacyjnych, obejmujących obszerny zestaw klasycznych i nowoczesnych algorytmów komputerowych oraz algorytmów uczenia maszynowego. Te algorytmy mogą służyć do wykrywania i rozpoznawania twarzy, identyfikowania obiektów, klasyfikowania działań ludzkich w filmach, śledzenia ruchu kamer, śledzenia poruszających się obiektów, wyciągania modeli obiektów 3D, łączenia zdjęć w celu uzyskania wysokiej rozdzielczości, znajdowania podobnych obrazów z bazy danych, usuwania czerwonych oczu z obrazów wykonanych przy użyciu lampy błyskowej, obserwowania ruchu oczu itp.

Poza funkcjami wyświetlania obiektów na ekranie jedną z ważniejszych w tej pracy jest Canny Edge Detector, czyli metoda wykrywania krawędzi opracowana przez Johna F. Canny'ego. Wykrywanie krawędzi polega na takiej zmianie obrazu, aby pokazywał on krawędzie wszystkich obiektów znajdujących się na nim. Detekcja odbywa się poprzez wyliczenie pierwszej oraz drugiej pochodnej funkcji, która opisuje zmiany jasności każdego piksela obrazu. Filtry krawędziowe opierające się na pierwszej pochodnej generują najlepsze wyniki dla obrazów posiadających obiekty z dużą różnicą poziomów jasności pikseli w porównaniu do tła. Dla obrazów, gdzie różnica

między tymi jasnościami jest mocno rozmyta stosuje się filtry krawędziowe oparte o drugą pochodną. Dużą wadą stosowania tych filtrów jest wyostanie małych detali i tworzenie szumów impulsowych, co powoduje pogorszenie jakości wynikowego obrazu, a także błędy w detekcji brzegów, które nie istnieją. Na rys. 5 przedstawiony został przykład wykrycia krawędzi.



Rys. 5. Wykrycie krawędzi obrazu

Wykrywanie krawędzi jest techniką wyodrębniania użytecznych informacji strukturalnych z różnych obiektów widzenia i znacznego zmniejszenia ilości przetwarzanych danych. Jest szeroko stosowany w różnych komputerowych systemach wizyjnych. Rozwiązanie do wykrywania krawędzi w celu spełnienia tych wymagań może być realizowane w szerokim zakresie sytuacji. Ogólne kryteria wykrywania krawędzi obejmują:

- Wykrywanie krawędzi z dużą dokładnością, co oznacza, że wykrywanie powinno uchwycić jak najwięcej krawędzi widocznych na zdjęciu.
- Punkt krawędzi wykryty przez operatora powinien dokładnie zlokalizować się na środku krawędzi.
- Dana krawędź obrazu powinna być oznaczona tylko raz, a jeśli to możliwe, szum obrazu nie powinien tworzyć fałszywych krawędzi.

Aby spełnić te wymagania wykorzystano rachunek wariacji – technikę, która znajduje funkcję optymalizującą. Optymalna funkcja w detektorze jest opisana sumą czterech wykładniczych składników, ale może być przybliżona przez pierwszą pochodną funkcji Gaussa.

Wśród opracowanych dotychczas metod wykrywania krawędzi, algorytm wykrywania krawędzi Canny'ego jest jedną z metod zapewniającą dobrą i niezawodną detekcję.

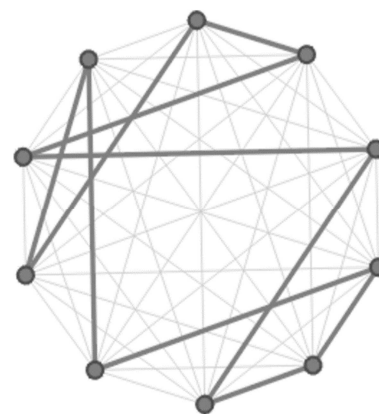
Algorytm wykrywania krawędzi Canny'ego może być podzielony na pięć etapów [11]:

- Zastosowanie filtru Gaussa (do usunięcia szumów).
- Wyznaczenie gradientu intensywności obrazu.
- Zastosowanie niemaxymalnego tłumienia, aby pozbyć się fałszywej odpowiedzi na wykrywanie krawędzi.
- Zastosowanie podwójnego progu w celu określenia potencjalnych krawędzi.
- Śledzenie krawędzi przez histerezę: zakończenie wykrywania krawędzi, tłumiąc wszystkie pozostałe

krawędzie, które są słabe i niepołączone z silnymi krawędziami.

## 6. PROBLEM KOMIWOJAŻERA

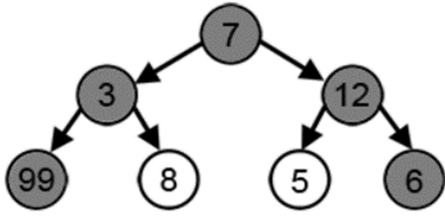
Określenie komiwojażera wywodzi się ze zwrotu *commis-voyageur*, który znaczy wędrowny sprzedawca podróżujący po kraju. Głównym celem komiwojażera jest odbycie swojej podróży zaliczając odpowiednią ilość miast. Problemem w tej sytuacji staje się wybór trasy, aby wyruszając odwiedzić każde z miast tylko raz i wrócić. Najlepszym planem w takiej sytuacji jest plan gwarantujący najkrótszą drogę. Rozumie się przez to najkrótszy dystans, czas podróży lub koszt. Problem ten określony skrótem TSP (ang. TSP – Travelling Salesman Problem) [10] jest jednym z najbardziej studiowanych tematów optymalizacji na świecie. TSP jest ściśle związany z cyklem Hamiltona, gdyż rozwiązanie tego problemu sprowadza się do odnalezienia minimalnego cyklu Hamiltona w pełnym, ważonym, nieskierowanym grafie, gdzie wierzchołki reprezentują miasta. Każdy z tych wierzchołków musi być połączony z pozostałymi poprzez krawędź. Jej długość odpowiada czasowi podróży. Rozwiązaniem optymalnym problemu komiwojażera jest więc cykl w grafie, który posiada pojedyncze przejście przez każdy wierzchołek, a suma długości krawędzi jest minimalna. Opis ten dotyczy symetrycznego problemu, oznaczanego skrótem STSP (ang. Symmetric TSP). Jest to problem, gdzie koszt podróży od miasta A do B jest taki sam jak z miasta B do A. W przypadku, w którym czas podróży w dwie strony się różni, problem jest asymetryczny i należy wykorzystać graf skierowany. Przykładowy cykl Hamiltona pokazany jest na rys. 6.



Rys. 6. Przykładowy cykl Hamiltona

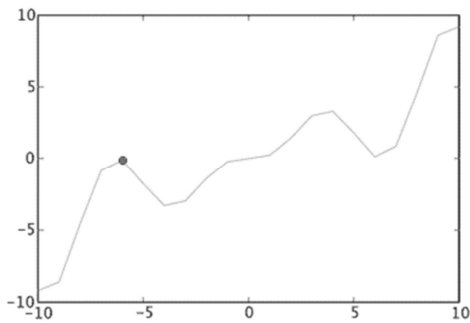
Algorytmy służące do rozwiązania tego problemu dzielą się na dokładne oraz heurystyczne. Te pierwsze działają w taki sposób, iż znajdują zawsze rozwiązanie optymalne. Porównują one wszystkie możliwe permutacje wierzchołków, a następnie wybierają najmniejszą sumę wag odległości między nimi. Algorytmy heurystyczne nie mogą zagwarantować znalezienia rozwiązania optymalnego, lecz ich szybkość w porównaniu do metod dokładnych jest o wiele większa. Dzięki temu są powszechnie stosowane do tego typu zadań. Algorytm zachłanny to algorytm, który w każdej swojej kolejnej fazie znajduje najlepsze dla danego momentu rozwiązanie, jednakże dokonuje decyzji lokalnie optymalnej. W wielu przypadkach metoda ta nie generuje dobrego wyniku, lecz daje pewne przybliżenie optymalnego rozwiązania w bardzo krótkim czasie. Strategia dla problemu komiwojażera polega na tym, żeby w każdym kolejnym

roku „odwiedzić nieodwiedzony” punkt, który znajduje się jak najbliżej. Na rys. 7 można zobaczyć prosty przypadek, w którym algorytm ten wybiera gorsze rozwiązanie. Nie daje on żadnej gwarancji znalezienia optymalnego rozwiązania. Zwykle są one około 25% gorsze od najlepszego. Istnieje możliwość znalezienia przez ten algorytm rozwiązania optymalnego, lecz istnieją również dane, dla których znajdzie on najgorsze możliwe rozwiązanie.



Rys. 7. Proces wykonywania algorytmu zachłannego (dwie ścieżki)

Algorytm wspinaczkowy jest prostą metodą stochastyczną poszukiwania optimum, która działa na zasadzie, znajdź punkt początkowy, znajdź jakiegokolwiek lepsze rozwiązanie, przejdź do niego, sprawdzaj warunek stopu. Metaforycznie można powiedzieć, że algorytm ciągle wspina się ku górze. Jest to algorytm zachłanny na lepsze rozwiązanie, czyli jeśli znajdzie on jakiegokolwiek lepsze rozwiązanie, zmierza ku niemu. Metoda ta zapewnia bardzo szybkie dotarcie do maksimum, lecz niekoniecznie jest to maksimum globalne, co przedstawione jest na rys. 8. Można zauważyć, że rozwiązanie zaznaczone kropką może zejść już tylko w dół, co nie oznacza, że jest to najlepsze rozwiązanie tego problemu. Standardowy algorytm wspinaczkowy jest bardzo podatny na natrafianie na ekstrema lokalne, co można zredukować poprzez wielokrotne uruchamianie algorytmu z różnymi punktami startowymi.



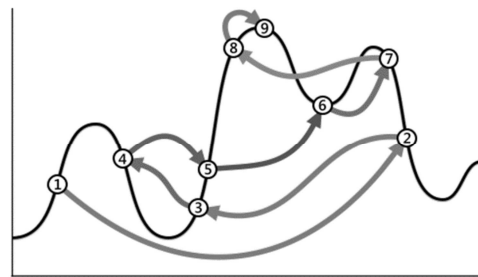
Rys. 8. Działanie algorytmu wspinaczkowego

Przykładowy kod algorytmu wspinaczkowego jest następujący [9]:

```
public City[] GreedyAlgorithm(City[] cities){
    this.totalDistance=0;
    int numberOfCities = cities.length;
    int[] tagTab = new int[numberOfCities];
    int n = 1;
    int index = 0;
    int popIndex = 0;
    City[] newCities = new
    City[numberOfCities];
    double distance;
    double distanceTemp;
    tagTab[0] = 1;
    newCities[0] = cities[0];
    while(n<numberOfCities) {
        distance = Double.MAX_VALUE;
        for (int j = 0; j < numberOfCities; j++) {
            if (tagTab[j] != 1) {
```

```
                distanceTemp =
                cities[popIndex].distanceTo(cities[j]);
                if (distanceTemp < distance) {
                    distance = distanceTemp;
                    index = j;
                    newCities[n] =
                    cities[index];}}}
                totalDistance += distance;
                tagTab[index] = 1;
                popIndex = index;
                n++;
            }
            this.city = newCities;
            return newCities;
        }
    }
```

Algorytm wyżarzania opiera się na procesie wyżarzania w metalurgii. W trakcie wyżarzania następuje podgrzewanie i ochładzanie materiału w celu zmiany parametrów fizycznych, co spowodowane jest zmianami w strukturze wewnętrznej. Po ochłodzeniu metalu jego nowa struktura staje się stała, co powoduje, iż materiał ten zachowuje nowe właściwości. W algorytmie wyżarzania kontroluje się temperaturę ustawiając ją na wysokim poziomie, a następnie powolnie ją zmniejszając. W trakcie, gdy temperatura jest wysoka algorytm może w większym stopniu akceptować rozwiązania gorsze niż aktualnie otrzymane. Daje to mu zdolność do wychodzenia z optimum lokalnych. Wraz z obniżaniem się temperatury zmniejsza się także możliwość akceptowania gorszych rozwiązań, w związku z tym algorytm może skupić się na obszarze, w którym jest szansa na znalezienie optimum globalnego. Ten stopniowy proces ochładzania sprawia, iż algorytm ten jest bardzo dobry w problemach zawierających dużą ilość optimum lokalnych. Bardzo ważną rolę odgrywa funkcja akceptacji. Determinuje ona, w jaki sposób algorytm decyduje, które rozwiązanie można zaakceptować.



Rys. 9. Działanie algorytmu symulowanego wyżarzania

Na początku sprawdza się, czy sąsiednie rozwiązanie jest lepsze. Jeżeli nie, wtedy sprawdza się o ile gorsze jest to rozwiązanie, a także jak wysoka jest temperatura, gdzie im ona wyższa, tym łatwiej zaakceptować gorsze rozwiązanie. Na rys. 9 przedstawione jest działanie algorytmu symulowanego wyżarzania.

Przykładowy kod program przedstawiono poniżej [9]:

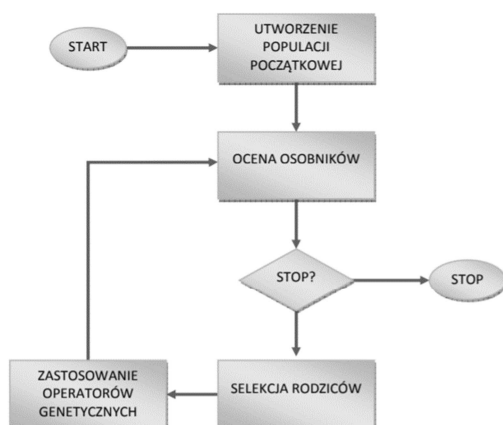
```
public static Genetics.Tour makeAnnealing(double
temp, double coolingRate) {
    Genetics.Tour currentSolution = new
    Genetics.Tour ();
    currentSolution.generateIndividual ();
    Genetics.Tour best = new
    Genetics.Tour(currentSolution.getTour ());
    while (temp > 1) {
        Tour newSolution = new
        Tour(currentSolution.getTour ());
        int tourPos1 = (int) (newSolution.tourSize () *
        Math.random ());
```

```

int tourPos2 = (int) (newSolution.tourSize() *
Math.random());
Genetics.City citySwap1 =
newSolution.getCity(tourPos1);
Genetics.City citySwap2 =
newSolution.getCity(tourPos2);
newSolution.setCity(tourPos2, citySwap1);
newSolution.setCity(tourPos1, citySwap2);
double currentEnergy =
currentSolution.getDistance();
double neighbourEnergy = newSolution.getDistance();
if (acceptanceProbability(currentEnergy,
neighbourEnergy, temp) > Math.random()) {
currentSolution = new
Genetics.Tour(newSolution.getTour());
}
if (currentSolution.getDistance() <
best.getDistance()) {best =
newGenetics.Tour(currentSolution.getTour());
}
Log.d(TAG, "cooling: " + coolingRate);
Log.d(TAG, "temp: " + temp);
temp *= 1-coolingRate;
}
return best;
}

```

Alгоритмы генетичне сѧ досконѧле до знойдowania optymalnego rozwiŧzania złoŧzonego problemu. Sѧ często używane w wielu obszarach inŧynierskich do tworzenia produktŧw wysokiej jakości. Przykŧadem moŧe być wyszukiwanie idealnej kombinacji materiałŧw z projektem w celu uzyskania najlepszego czy najwytrzymalszego połączenia. Uŧywane sѧ teŧ często w programach komputerowych do zarządzania procesami lub do rozwiązywania problemŧw optymalizacji. Alгоритмы генетичне базую на процесie ewolucyjnym. Naśladują one metodologię, którą ewolucja wykorzystuje do rozwiązywania problemŧw. Inspirowane sѧ one teoriŧ doboru naturalnego oraz dziedziczenia. Zostały one zapoczątkowane przez Johna Hollanda w latach 60' i 70'. Alгоритm bazuje na populacji składającej się ze zbioru osobników, które reprezentują rozwiązanie. Kaŧdy osobnik zbudowany jest z chromosomŧw, które to z kolei jest zbiorem genŧw. Gen jest najmniejszym elementem informacji i reprezentuje on pojedynczŧ cechę. Rzeczywiste cechy osobnika, które określają jego stopień przystosowania to fenotyp. Pokazuje on na ile dany osobnik jest lepszy od pozostałych. Reasumując, w populacji mogą występować osobniki o tym samym fenotypie i różnym genotypie, ale nie możliwe jest występowanie kilku osobników o różnym fenotypie i tym samym genotypie. Ogólne model algorithmu генетичnego przedstawiony jest na rys. 10.



Rys. 10. Model algorithmu генетичnego

Funkcję realizującŧ metodę krzyŧowania osobników przedstawiono poniŧej [9]:

```

public static Tour crossover(Tour parent1, Tour
parent2) {
    Tour child = new Tour();
    int startPos = (int) (Math.random() *
parent1.tourSize());
    int endPos = (int) (Math.random() *
parent1.tourSize());
    for (int i = 0; i < child.tourSize(); i++) {
    if (startPos < endPos && i > startPos && i <
endPos) {
    child.setCity(i, parent1.getCity(i));
    }
    else if (startPos > endPos) {
    if (!(i < startPos && i > endPos)){
    child.setCity(i,
parent1.getCity(i));}}
    for (int i = 0; i < parent2.tourSize(); i++) {
    if (!child.containsCity(parent2.getCity(i))) {
    for (int ii = 0; ii < child.tourSize(); ii++) {
    if (child.getCity(ii) == null) {
    child.setCity(ii, parent2.getCity(i));
    break;}}}}
    return child;}

```

## 7. TESTY EKSPERYMENTALNE

Na rys. 11 przedstawiono narysowany obraz przez ploter. Można zauwaŧzyć wynikowe wykryte krawędzie, które zostały narysowane przez ploter. Rezultat jest bardzo dobry, biorąc pod uwagę, ŧe kartka w trakcie rysowania potrafi się uginać w górę i w dół. Nie widać szczegŧlnych różnic pomiędzy skanem narysowanego obrazu (rys. 11), a cyfrowŧ wersjŧ przedstawianŧ na ekranie smartphone (rys. 12). Narysowany obraz cechuje się duŧŧ dokładnoścŧŧ w porŧwnaniu do obrazu wejściowego (rys. 12).

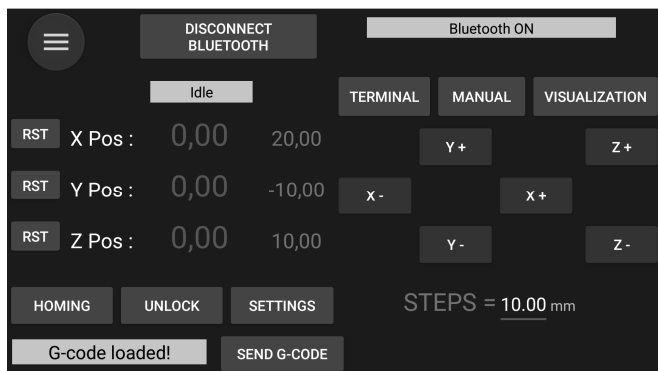


Rys. 11. Skan narysowanego obrazu



Rys. 12. Rastrowy obraz wejściowy

Na rys. 13 zaprezentowano interfejs aplikacji pracującej w systemie Android. W aplikacji teŧ wyświetlana jest aktualna pozycja wrzeczona plotera, jest moŧliwość konfiguracji urządzenia bluetooth, długości kroku w ciągu 1s, wizualizacji pracy oraz ustawienia terminala, jak również wprowadzenia ustawień ręcznych.



Rys. 13. Wygląd okna aplikacji dla systemu Android

## 8. PODSUMOWANIE

W referacie scharakteryzowano problem komiwożera, a także dokonano przeglądu metod służących do rozwiązywania tego zagadnienia. Założenia projektu zostały zrealizowane, ponieważ prototyp urządzenia funkcjonuje prawidłowo, rysując obrazy z dużą dokładnością oraz dużą szybkością (ze względu na zastosowane algorytmy optymalizacyjne). Aplikacja została napisana w ten sposób, że użytkowanie jej nie sprawia problemów użytkownikowi, a także szybkość jej działania jest na odpowiednim poziomie. Stworzona aplikacja posiada z jednej strony charakter edukacyjny, gdyż można sprawdzić, w jaki sposób metody optymalizacji radzą sobie z danym problemem. Z drugiej strony jest ona bardzo przydatna osobom, które zajmują się obsługą tego typu urządzeń, gdyż pozwala zdalnie sterować maszyną bez wykorzystania innych urządzeń.

## INTERFACE FOR REMOTE CONTROL OF THE PLOTTER WITH IMAGE RECOGNITION FUNCTION

The topic is to build a CNC machine working as a plotter, and also create an application running on the Android operating system, which processes images, optimizes the code and allows to control the built machine. The examination is an estimation which of the methods used in this thesis is optimal and gives the best results in this type of problem, which is choosing the shortest path of the salesman problem. The purpose of the paper is to obtain the code processed by the numerical machine in the shortest possible time, which will be carried out in a short period of time with the proper accuracy of the work. Another important aspect is the implementation of an intuitive user interface that does not cause problems with support. The optimization methods used produce satisfactory results that are more or less practical depending on the problem. The effect in the form of drawn images is at a high level of accuracy. After many hours of working with the CNC machine using the created application, it can be seen that this is a useful set for both people who want to create their own images as well as for those seeking education in this field. The great advantage is that the design is very easily expandable so that it can acquire new, very useful features for a small amount of extra work in the form of adding new functionality.

**Keywords:** CNC, stepper motor, plotter, travelling salesman problem, hill climbing algorithm, simulated annealing algorithm, genetics algorithm, greedy algorithm, image processing, Arduino, GRBL, Open CV, Bluetooth, Android.

## 9. BIBLIOGRAFIA

1. Wit G., Kiszka P., Niesłony P.: Programowanie obrabiarek CNC, Wydawnictwo Naukowe PWN, Warszawa, 2017.
2. Przepiórkowski J.: Silniki elektryczne w praktyce elektronika. Wydawnictwo BTC, Warszawa, 2007.
3. Wei-Meng L.: Android Poradnik programisty, Wydawnictwo Promise, Warszawa, 2015. Helion, Noga Bogdan: Inventor. Podstawy projektowania.
4. Grbl: <http://www.github.com/grbl/grbl/wiki>, (data dostępu 12-14.08.2017 r.).
5. Monk S., tł. Matuk K.: Arduino dla początkujących. Podstawy i szkice. Wydawnictwo HELION, Gliwice, 2014.
6. Dokumentacja techniczna mikrokontrolera ATMEGA 328P, <http://www.atmel.com>, (data dostępu 10-16.07.2017 r.).
7. Brahmhatt S.: Practical OpenCV. Wydawnictwo Apress, 2013.
8. Dokumentacja Open CV: <http://www.opencv.org>, (data dostępu 18-19.08.2017 r.).
9. <http://www.theprojectspot.com>, (data dostępu 10-16.07.2017 r.).
10. Dantzig G. B., Fulkerson D. R., Johnson S. M.: Solution of large-scale travelling-salesman problem, Journal of the Operations Research Society of America, Nov. 1, 1954.
11. Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
12. Kamiński M.: Interfejs do zdalnego sterowania ploterem z funkcją rozpoznawania obrazów. Praca magisterska, Politechnika Gdańska 2017.