

Piotr Ratuszniak

Łukasz Gątnicki

Wydział Elektroniki i Informatyki

Politechnika Koszalińska

ratusz@ie.tu.koszalin.pl

lukasz.gatnicki@gmail.com

Aplikacja wyszukiwania i wizualizacji trasy dla przewoźników samochodowych

Słowa kluczowe: nawigacja, GPS, optymalizacja trasy, problem komiwojażera, algorytmy genetyczne.

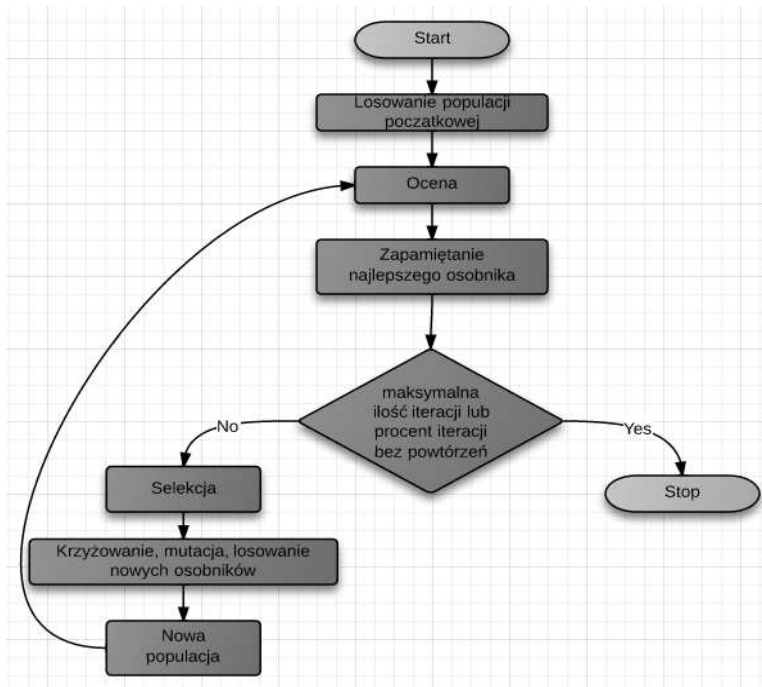
1. Wstęp

Jednym z przełomowych wydarzeń w dziedzinie transportu i nawigacji było udostępnienie w latach osiemdziesiątych przez kongres amerykański systemu GPS (ang. *Global Positioning System*). Amerykański wojskowy system został udostępniony do wykorzystania cywilnego bez jakichkolwiek opłat co zapewne miało decydujący wpływ na jego szybkie upowszechnienie. Obecnie system ten jest wykorzystywany w wielu dziedzinach życia codziennego, np.: w nawigacji samochodowej, systemach nawigacji dla pieszych i rowerzystów, w pozycjonowaniu do określenia miejsca bieżącego pobytu oraz w szeroko pojętym określeniu czasu [1]. Z drugiej strony bardzo dynamiczny rozwój urządzeń mobilnych wyposażonych w odbiorniki GPS spowodował wykorzystanie tego systemu na masową skalę. Pojawiło się wiele różnych aplikacji wykorzystujących ten system, jednak jedną z największych popularności cieszą się aplikacje do nawigacji samochodowej. Obecnie na rynku istnieje wiele aplikacji do nawigacji samochodowej, sprzedawanych często razem z odbiornikami systemu GPS. W Polsce jedną z najbardziej popularnych i najbardziej zaawansowanych aplikacji do nawigacji samochodowej jest AutoMapa [2]. W aplikacji tej mamy kilka różnych scenariuszy wyszukiwania trasy przejazdu. Pomimo ponad dziesięcioletniego pobytu na rynku aplikacja ta, podobnie jak inne popularne aplikacje dostępne na rynku, nie zawiera zaawansowanego algorytmu minimalizacji trasy przejazdu dla wielu zdefiniowanych punktów przejazdu w sposób analogiczny dla problemu komiwojażera. Opcja wyszukiwania trasy analogicznie do tego problemu będzie z pewnością miała szerokie zastosowanie w wielu firmach kurierskich i przewozowych, jak również dla wszystkich przewoźników, dla których kursy realizowane są według scenariusza „baza - wiele punktów docelowych- baza”. Brak tego rodzaju optymalizacji wyszukiwanej trasy

w popularnych aplikacjach do nawigacji samochodowej był głównym motywem powstania opisywanej aplikacji. W artykule opisano utworzoną aplikację do optymalizacji trasy przejazdu według opisanego scenariusza, bez konieczności określania kolejności punktów pośrednich, jak ma to miejsce w powszechnie dostępnych aplikacjach do nawigacji samochodowej. Utworzona aplikacja pobiera powszechnie dostępne dane na temat niezbędnych miejscowości oraz odległości pomiędzy nimi za pomocą Internetu oraz dokonuje optymalizacji trasy przejazdu poprzez odpowiedni dobór kolejności punktów pośrednich. Wynikiem działania aplikacji jest wizualizacja wyznaczonej trasy przejazdu na mapie, wygenerowanie wskazówek dojazdu w postaci listy z kolejnymi manewrami. Aplikacja posiada również możliwość wygenerowania pliku z opisem trasy, umożliwiającą jej wczytanie do popularnej aplikacji nawigującej – AutoMapa [2], co z pewnością podnosi jej walory użytkowe.

2. Algorytm genetyczny do optymalizacji trasy

Przedstawiony powyżej problem optymalizacji trasy w algorytmice znany jest pod nazwą Problemu Komiwojażera (z ang. TSP - Travelling Salesman Problem). Problem komiwojażera zaliczany jest do grupy problemów NP-trudnych [3, 4, 5] i jest to zagadnienie natury optymalizacyjnej należące do działy matematyki i informatyki zwanego teorią grafów. Problem ten dotyczy odnalezienia minimalnego cyklu Hamiltona w grafie pełnym ważonym. Innymi słowy chodzi o znalezienie ścieżki po krawędziach grafu o najmniejszej sumie wag, która wychodząc z zadanego punktu będzie przebiegać przez każdy z pozostałych wierzchołków tylko raz i wróci do wierzchołka startowego. W powyższym opisie również łatwo dostrzec podobieństwo teorii do praktycznej pracy wykonywanej przez kuriera. Do jego obowiązków należy bowiem wyruszenie z miasta bazy do wszystkich odbiorców rozlokowanych w różnych miejscowościach i powrót do punktu startu. Obecnie znanych jest wiele metod rozwiązywania problemu komiwojażera wykorzystujących np.: algorytmy genetyczne [6, 7], mrówkowe [8] i memetyczne [9]. W opisywanej aplikacji, na jej bieżącym etapie rozwoju, do wyszukiwania trasy w aplikacji został wykorzystany algorytm genetyczny. W stosunku do klasycznego algorytmu genetycznego zostały wprowadzone pewne modyfikacje. Zmiany te pozwalają na zmniejszenie zbieżności algorytmu do ekstremum lokalnego oraz zmniejszenie czasu jego działania z minimalnym wpływem na jakość wyników. Ogólny schemat działania opracowanego algorytmu przedstawiony jest na rys. 1.



Rys. 1. Schemat blokowy zaimplementowanego algorytmu genetycznego

2.1. Reprezentacja osobników

Jednym z najważniejszych etapów tworzenia algorytmu genetycznego jest określenie reprezentacji danych. W aplikacji wyszukiwania i wizualizacji trasy ważne było znalezienie reprezentacji, która nie będzie znacząco komplikować losowania populacji oraz działania operatorów genetycznych krzyżowania i mutacji. Reprezentacja w postaci prostej listy z nazwami miast mogłaby powodować generowanie osobników niespełniających ograniczeń trasy, w taki sposób, że po operacji krzyżowania i mutacji należałoby sprawdzić poprawność powstałych osobników (tras) w celu wykluczenia zduplikowanych miast. Istnieje co prawda możliwość stosowania mechanizmów naprawczych [10] po zastosowaniu operatorów algorytmu genetycznego, jednak stosowanie tego typu mechanizmów powoduje zwiększenie złożoności obliczeniowej każdej generacji algorytmu, co w konsekwencji może powodować znaczny spadek wydajności całego algorytmu. Z tego powodu ważnym zagadnieniem jest odpowiedni dobór reprezentacji danych. W opisywanej aplikacji zastosowano reprezentację danych w postaci listy odniesień [10]. Polega ona na utworzeniu listy odniesień do listy wszystkich miast w kolejności ich pobierania. Punktem odniesienia w tej reprezentacji jest wprowadzona przez użytkownika do programu lista wszystkich miast, które należy odwiedzić. Każdy osobnik jest odzwierciedleniem tego, w jakiej kolejności są

odwiedzane miasta z głównej listy. Po dopisaniu do listy referencji kolejnej wartości z listy głównej usuwany jest jej odpowiednik. W wyniku tego na i -tej pozycji dla $i = 0..n - 1$, gdzie: n - liczba miast do odwiedzenia, znajdzie się zawsze liczba całkowita z przedziału $[0, n - 1 - i]$. Tworzenie przykładowej listy odniesień przedstawiono na rys. 2.



Rys. 2. Przykładowe kroki w tworzeniu listy odniesień

Taka reprezentacja sprawia jednak pewne trudności w opracowaniu funkcji celu oceny. Aby możliwe było wyliczenie długości tras do ocenienia rozwiązań niezbędne jest przejście do standardowej reprezentacji w postaci listy miast w kolejności odwiedzania. Wiąże się to z koniecznością wykonania dodatkowych operacji i zwiększenia złożoności obliczeniowej funkcji celu, jednak ta dodatkowa złożoność obliczeniowa jest rekompensowana podczas wykonywania operacji krzyżowania i mutacji. Biorąc pod uwagę wzór na i -ty element listy referencji, zarówno podczas operacji standardowego jednopunktowego krzyżowania oraz standardowej operacji mutacji wybranej pozycji, za każdym razem generowany jest nowy osobnik reprezentujący dopuszczalne rozwiązanie problemu. Podczas programowej implementacji w aplikacji do przechowywania informacji o osobniku została zaprojektowana dodatkowa pomocnicza struktura. Poza tablicą zawierającą referencje w postaci indeksów do listy miast dodatkowa struktura posiada tablicę zawierającą odległości cząstkowe trasy oraz zmienną przechowującą całkowitą jej długość.

2.2. Funkcja oceny i operatory algorytmu

Operatory algorytmu

Ponieważ aplikacja wyszukiwania i wizualizacji trasy oparta jest na problemie komiwojażera w zaimplementowanym algorytmie genetycznym wartością funkcji oceny każdego osobnika jest długość trasy, którą on reprezentuje. Po eksperymentalnych doświadczeniach zaobserwowano fakt, że trasy reprezentowane przez poszczególnych osobników różnią w sposób znaczący. Aby wyeliminować możliwość powstawania superosobników, co oznaczałoby zwiększenie prawdopodobieństwa utknięcia algorytmu w minimum lokalnym, zastosowano selekcję metodą rankingu liniowego [11]. Po wyliczeniu odległości całkowitej każdego osobnika cała populacja jest sortowana malejąco. Wartość prawdopodobieństwa przejścia do kolejnej generacji jest wyliczana na podstawie ilorazu położenia osobnika w posortowanej liście przez sumę położen wszystkich osobników. Przykładowo mając trzy osobniki posortowane malejąco 3,2,1

pierwszy osobnik na liście będzie posiadał prawdopodobieństwo $\frac{1}{6}$, drugi $\frac{2}{6}$, trzeci $\frac{3}{6}$.

Po operacji selekcji, w kolejnym kroku algorytmu, wykonywane są operatory krzyżowania, mutacji oraz wszczepiania do populacji nowych losowych osobników. Po szeregu doświadczeń eksperymentalnych, realizowanych dla założonych zestawów punktów pośrednich trasy, określono następujące parametry wymienionych operatorów:

- Krzyżowanie – dzieli osobniki rodzicielskie w okolicach połowy trasy na dwie części i zamienia je krzyżowo. Operacja ta zachodzi z prawdopodobieństwem 50%.
- Mutacja – maksymalnie trzykrotnie wybierana jest losowa wartość genu osobnika i następnie zamieniana ją z inną losową wartością z odpowiedniego przedziału. Operacja zachodzi każdorazowo z prawdopodobieństwem 20%.
- Zamiana osobnika na losowego – w miejsce osobnika rodzicielskiego generowany w sposób losowy nowy osobnik. Operacja zachodzi z prawdopodobieństwem 30%.

Operacja zamiana osobnika na losowego nie jest standardowym operatorem genetycznym, jednak została wykorzystana w celu zmniejszenia zbieżności algorytmu. Losowa zmiana na zupełnie nowego osobnika wyklucza w jeszcze większej mierze możliwość powstania superosobników i utknięcia algorytmu w minimum lokalnym. Operacja ta miała widoczny wpływ na jakość uzyskiwanych rozwiązań.

2.3. Warunki zakończenia pracy algorytmu

Z przedstawionego ogólnego schematu blokowego działania opracowanego algorytmu genetycznego można wywnioskować, że zakończenie pracy algorytmu genetycznego następowało po obliczeniu założonych, eksperymentalnie ustalonych dla

danego przedziału punktów pośrednich trasy obliczonych generacji lub po uzyskaniu procentowej wartości krytycznej bez poprawy jakości uzyskanego rozwiązania.

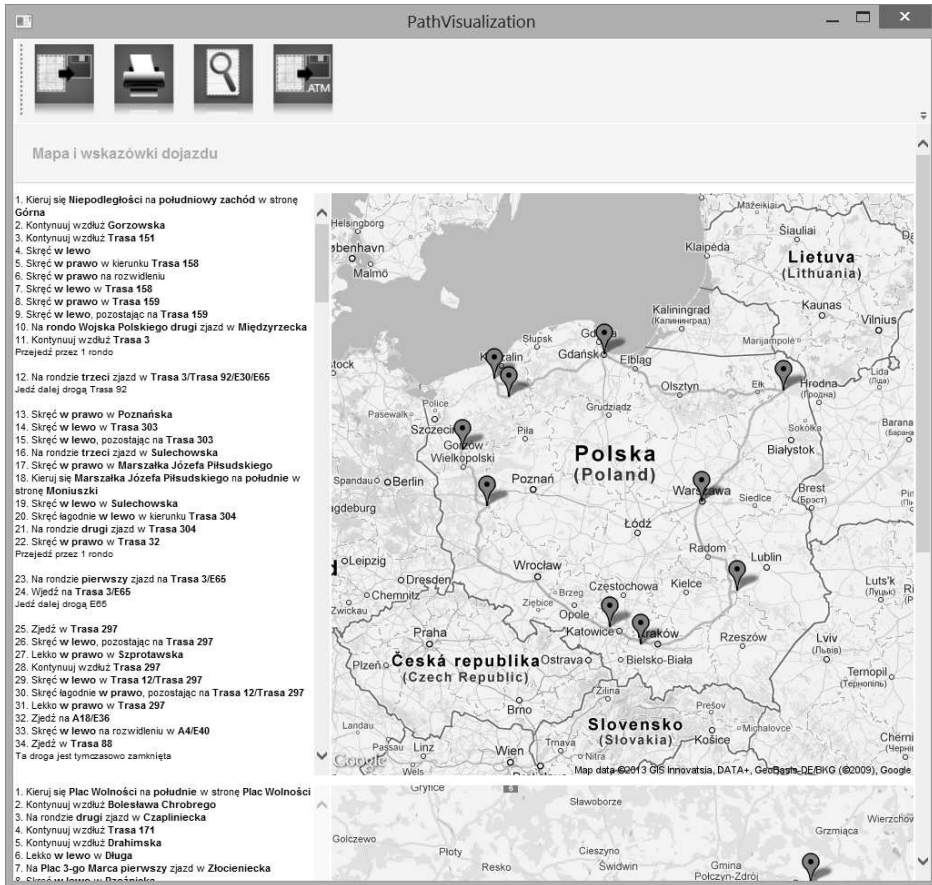
3. Aplikacja wyszukiwania i wizualizacji trasy

Aplikacja wyszukiwania i wizualizacji trasy dla przewoźników samochodowych została wykonana w celu usprawnienia procesu planowania tras. Umożliwia ona planistom lub samym przewoźnikom odnalezienie suboptymalnej trasy przejazdu do zadanych miejscowości, na podstawie ich nieuporządkowanej listy. Na rysunku 3 przedstawiono okno główne aplikacji z listą dodanych punktów trasy.



Rys. 3. Okno główne aplikacji z listą punktów trasy

Aplikacja pozwala także wygenerować i wydrukować mapy wraz ze wskazówkami dojazdu oraz plik do urządzenia nawigacji samochodowej korzystającej z oprogramowania AutoMapa. Program umożliwia modyfikację parametrów algorytmu genetycznego użytego do wyszukiwania trasy oraz zapis map i wskazówek dojazdu w formie pliku HTML. Na rys. 4 przedstawiono okno aplikacji z wygenerowaną mapą z zaznaczoną zoptymalizowaną trasą przejazdu oraz z wygenerowanymi tekstowymi wskazówkami dojazdu.



Rys. 4. Wygenerowana mapa z zaznaczoną trasą przejazdu oraz wygenerowane tekstowe wskazówki dojazdu

3.1. Wykorzystane narzędzia i technologie

Aplikację wyszukiwania i wizualizacji trasy dla przewoźników samochodowych została wykonana w oparciu o technologie firm Microsoft i Google. Platformą, z której skorzystano podczas tworzenia programu jest .NET Framework i język C#. Ponadto podczas realizacji aplikacji wykorzystano następujące narzędzia i technologie:

- Windows Presentation Foundation (WPF) – jako nowoczesny silnik graficzny i API pozwalające na budowanie interfejsu aplikacji korzystając ze znacznikowego języka XAML opartego na formacie XML;

- Task Parallel Library (TPL) – zestaw bibliotek ułatwiający wprowadzenie do aplikacji elementów przetwarzania równoległego i współbieżności, wykorzystanych w mechanizmie pobierania danych o miejscowościach pośrednich trasy;
- ADO.NET – zbiór bibliotek zapewniających dostęp do baz danych;
- Framework MVVM Light - zestaw narzędzi i komponentów ułatwiający i przyspieszający tworzenie aplikacji zgodnych z wzorcem MVVM w technologiach WPF, Silverlight i Windows Phone. Rozwiązania dostępne w tym zestawie pomagają w zachowaniu separacji kodu pomiędzy klasami modelu i widoku;
- serwer bazodanowy Microsoft SQL Server 2012 w darmowej wersji Express do przechowywania danych na temat miejscowości;
- serwisu Google Maps - do dostarczenia danych o odległościach, map i wskazówek dojazdu za pomocą ogólnodostępnego API w postaci Web Services. Pobranie danych z tego serwisu było możliwe poprzez budowanie odpowiednich adresów URL.

3.2. Funkcjonalności aplikacji

Aplikacja składa się z dwóch zasadniczych modułów: moduł odpowiedzialny za wyszukiwanie trasy oraz moduł służący do jej wizualizacji. Dla obu modułów aplikacji poniżej zamieszczono listy ich wybranych funkcjonalności.

Wybrane funkcjonalności, możliwości i ograniczenia modułu wyszukiwania trasy:

- możliwość wykorzystania algorytmu wyszukiwania trasy dla liczby miast mieszczącej się w przedziale od 5 do 99;
- wykorzystanie przez algorytm wyszukiwania trasy bazy danych odległości opartej na informacjach pozyskanych z serwisu Google Maps;
- autouzupełnianie pola tekstowego używanego do wprowadzania nazw miejscowości na podstawie listy miast znajdującej się w bazie danych;
- wyświetlanie w formie listy w głównym oknie programu miast ułożonych w odpowiedniej kolejności po zakończeniu działania algorytmu genetycznego;
- możliwość skorzystania z predefiniowanych parametrów algorytmu wyszukiwania takich jak liczebność populacji, liczba iteracji i współczynnik iteracji bez zmian;
- możliwość definiowania własnych ustawień algorytmu wyszukiwania, takich jak: liczebność populacji, liczba iteracji i współczynnik iteracji bez zmian;

- informacja o działaniu algorytmu wyszukiwania trasy za pomocą paska postępu pracującego w trybie niekreślonym, w pasku statusu w głównym oknie aplikacji;
- możliwość skorzystania z pomocy dostępnej w aplikacji dotyczącej ustawień i parametrów algorytmu wyszukiwania trasy.

Wybrane funkcjonalności, możliwości i ograniczenia modułu wizualizacji trasy:

- przedstawienie tras za pomocą map z zaznaczoną wyraźną linią i tekstowych wskazówek dojazdu w oknie dialogowym aplikacji za pomocą usług webowych Google Maps;
- przedstawienie na jednej mapie maksymalnie dziesięciu punktów trasy-wskazówki dojazdu osobne dla każdej mapy;
- mapy i wskazówki dojazdu przedstawione oknie dialogowym w formie stosu jedna pod drugą;
- możliwość wydruku map wraz ze wskazówkami dojazdu z możliwością podglądu wydruku;
- możliwość wygenerowania pliku ATM pozwalającego na przeniesienie trasy do odbiornika GPS z oprogramowaniem AutoMapa;
- możliwość zapisu map wraz ze wskazówkami dojazdu do pliku HTML;
- możliwość skorzystania z pomocy dostępnej w aplikacji, dotyczącej korzystania z modułu wizualizacji trasy.

3.3. Schemat działania aplikacji

Głównym celem zaprojektowanej aplikacji jest wyszukiwanie najlepszej możliwej trasy ułożonej na podstawie wprowadzonej przez użytkownika listy, a następnie przedstawienie jej za pomocą graficznych map i tekstowych wskazówek. Nie jest to jednak jedyny dostępny scenariusz wykorzystania, co obrazuje rysunek 5.



Rys. 5. Ogólny schemat działania aplikacji

Aplikacja, poza wyżej wspomnianym scenariuszem, może służyć do pobrania trasy dla dowolnie wprowadzonych i ułożonych przez użytkownika punktów. Nie jest konieczne uruchamianie algorytmu wyszukiwania najlepszej kombinacji. Poza tym po pobraniu i wyświetleniu map i wskazówek istnieje możliwość ich wydruku, zapisu do pliku *.html, możliwego do otwarcia w dowolnej przeglądarce, a także wygenerowania pliku *.atm współpracującego z urządzeniami wyposażonymi w odbiornik GPS z oprogramowaniem AutoMapa.

4. Podsumowanie

Zaprojektowana aplikacja jest potrzebnym rozszerzeniem istniejącego oprogramowania systemów GPS poprzez dodanie możliwości wyszukania drogi na podstawie wielu punktów bez określania ich kolejności. Aplikacja za pomocą algorytmu genetycznego wyznacza kolejność miejscowości wpisanych przez użytkownika. Uporządkowana w ten sposób lista to rozwiązanie zapewne suboptymalne, jednak uzyskane w rozsądnym czasie. Wbudowany moduł wyświetlania tras i wskazówek dojazdu oraz możliwość ich drukowania pozwalają na korzystanie z nich w podróży. Gdyby jednak ich statyczna forma okazała się niewystarczająca dla użytkownika można skorzystać z wyznaczonej trasy na odbiornikach GPS poprzez wygenerowanie odpowiedniego pliku. W obecnej wersji wspierane jest oprogramowanie jednego z najbardziej popularnych tego typu rozwiązań - AutoMapa. Dzięki tym możliwościom aplikacja jest oprogramowaniem dostosowanym do potrzeb przewoźników samochodowych i zapewne znajdzie zastosowanie, gdyż wypełnia dość istotną lukę na rynku aplikacji nawigacji samochodowej.

Zaprezentowana aplikacja cechuje się dużymi możliwościami rozbudowy. Możliwe jest rozszerzenie bazy danych odległości o kolejne miejscowości niebędące miastami Polski i dzięki temu możliwe jest uwzględnienie obsługi innych państw Europy. Ponadto planowane jest dalsze usprawnianie działania samego algorytmu genetycznego poprzez jego równoległą realizację, np. z wykorzystaniem powszechnie już dostępnych procesorów wielordzeniowych. Równoległa implementacja zapewne pozwoli jeszcze zredukować czas optymalizacji trasy. Usprawnienia mogą dotyczyć również interfejsu użytkownika lub obsługi większej ilości systemów GPS. Ciekawą opcją może być też stworzenie wersji działającej na urządzeniach mobilnych z wbudowanymi modułami GPS.

Bibliografia

1. <http://www.gps.gov/applications/>
2. <http://www.automapa.pl/?PEI=35489&lng=PL>
3. Papadimitriou, Christos H. (1977), "The Euclidean traveling salesman problem is NP-complete", *Theoretical Computer Science* 4 (3): 237–244,

4. Applegate, D. L.; Bixby, R. M.; Chvátal, V.; Cook, W. J. , The Traveling Salesman Problem, 2006, ISBN 0-691-12993-2
5. Gutin, G.; Punnen, A. P. (2006), The Traveling Salesman Problem and Its Variations, Springer, ISBN 0-387-44459-9
6. F. H. Khan, N. Khan, S. Inayatulla, S. Nizami: Solving ISP Problem by Using Genetic Algorithm, International Journal of Basic & Applied Sciences IJBAS-IJENS Vol:09 No:10 ,2009
7. M.Karova, V.Smarkov, S. Penev: Genetic operators crossover and mutation in solving the TSP problem, International Conference on Computer Systems and Technologies, 2005
8. Marco Dorigo, Luca Maria Gambardella: Ant colonies for the travelling salesman problem, Biosystems, Volume 43, Issue 2, Elsevier, July 1997, Pages 73–81
9. Peter Merz, Bernd Freisleben: Memetic Algorithms for the Traveling Salesman Problem, Complex Systems, 13 (2001) 297–345; 2001
10. Z. Michalewicz, „Algorytmy genetyczne + struktury danych = programy ewolucyjne”, Wydawnictwo Naukowo-Techniczne, 1996
11. David E. Goldberg, Kalyanmoy DebA comparative analysis of selection schemes used in genetic algorithms, Foundations of Genetic Algorithms, 1991

Streszczenie

W artykule zaprezentowano praktyczną implementację algorytmu genetycznego do rozwiązywania problemu optymalizacji trasy analogicznego do problemu komiwojażera. Algorytm został zaimplementowany w autorskiej aplikacji do wyznaczania trasy przejazdu dla rzeczywistych danych geograficznych polskich miejscowości pobieranych z serwisu Google Maps. Prezentowana aplikacja generuje wskazówki dojazdu i umożliwia export wyznaczonej trasy do programu Automapa, co stanowi jego doskonale uzupełnienie.

Abstract

The paper presents a practical implementation of a genetic algorithm to solve the problem of route optimization analogous to the traveling salesman problem. The algorithm has been implemented in the author's application for route calculation for the real Polish geographic data retrieved from Google Maps service. Presented application generates travel directions in the text and graphic form and allows to export the computed route to the Automapa program, which is his perfect complement.

Słowa kluczowe: vehicle navigation system, GPS, route optimization, travelling salesman problem, genetic algorithms.