

Joanna KARBOWSKA-CHILIŃSKA, Paweł ZABIELSKI

Politechnika Białostocka, Wydział Informatyki

ul. Wiejska 45 A, 15-351 Białystok

E-mail: j.karbowska@pb.edu.pl, p.zabielski@pb.edu.pl

Optymalizacja trasy podróży turystycznej samochodem elektrycznym

1 Wstęp

W ostatnich latach Komisja Europejska w krajach członkowskich Unii Europejskiej (UE) opracowała kilka dyrektyw mających na celu rozwój ekologicznego transportu [13, 15]. Chodzi o możliwie największe ograniczenie emisji gazów cieplarnianych poprzez wykorzystanie między innymi pojazdów hybrydowych i elektrycznych. Samochody elektryczne są zeroemisyjne, bardziej ciche i tańsze w eksploatacji (koszt zużycia energii to ok. 8 zł/ 100 km). Poziom rozwoju elektromobilności jest różny w poszczególnych krajach UE. Norwegia jest liderem użytkowania samochodów elektrycznych (5-milionowa populacja ma zarejestrowanych około 21 tys. pojazdów) [17]. Rząd Norwegii w trosce o środowisko i jakość życia obywateli wprowadził szereg udogodnień i przywilejów: pojazdy elektryczne są zwolnione z 25% podatku vat, mogą korzystać z buspasów, darmowego parkowania, są zwolnione z podatków drogowych i opłat za promy i tunele. Dodatkowo kierowcy mają darmowy dostęp do rozległej sieci punktów zasilania, dzięki czemu w większych miastach podróżują praktycznie za darmo. Sprzedaż samochodów elektrycznych w Norwegii w pierwszym półroczu 2014 r. wzrosła o około 300% w stosunku do podobnego okresu w 2013 r. [12]. Na uwagę zasługują również działania na rzecz elektromobilności w Estonii. Rozbudowano ogólnonarodową sieć szybkiego ładowania baterii, lokalizując stacje przy głównych drogach w miejscowościach większych niż 5 tys. mieszkańców, odległość między stacjami jest nie większa niż 60 km [19]. Estończycy mogą ubiegać się o dofinansowanie 18 tys. euro przy zakupie nowego samochodu elektrycznego i 1 tys. euro przy zakupie przydomowej stacji ładowania baterii. W Polsce co roku średnio jest rejestrowanych ponad 30 samochodów elektrycznych. Dostępnych jest około 200 (publicznych i prywatnych) punktów ładowania [14, 18]. Właściciele samochodów elektrycznych w Polsce mogą jedynie liczyć na zwolnienia z opłat parkingowych na przykład w Gdańsku, Katowicach, Szczecinie. Coraz częściej samorządy większych miast (np. Gdańska) inwestują w zakup samochodów z napędem elektrycznym. Kilka gmin z województwa lubelskiego napisało projekt o dofinansowanie ze środków UE powstania wypożyczalni samochodów i rowerów elektrycznych oraz stacji ładowania na terenach turystycznych tych gmin [16]. Taka inwestycja pozwoliłaby na jeszcze większe zainteresowanie turystów tym regionem bez zwiększenia ilości spalin i zanieczyszczeń środowiska.

W literaturze naukowej dynamicznie rozwija się trend zwany ekologiczną logistyką (ang. *green logistic*) [4]. Celem ekologicznej logistyki jest między innymi maksymalne wykorzystanie środków transportu, wykorzystanie alternatywnych źródeł energii do napędu pojazdów, zwiększenie wydajności transportu przez dobór pojazdów

z niskim spalaniem. Ważnym zagadnieniem w literaturze jest problem ekologicznej optymalizacji tras transportowych [3, 7] (ang. *green logistic vehicle routing problem*), uwzględniający między innymi omijanie korków lub dużych aglomeracji na trasie czy ograniczenie punktów przeładunkowych podczas transportu. W literaturze naukowej odnajdujemy również publikacje na temat optymalizacji tras pojazdów elektrycznych [8, 10, 11]. Usytuowanie stacji ładowania lub wymiany baterii pojazdu elektrycznego jest jednym z problemów optymalizacyjnych omawianych w tych publikacjach.

W artykule przedstawiamy propozycję optymalizacji planowania trasy podróży turystycznej, gdy środkiem transportu jest cichy, ekonomiczny samochód elektryczny. Wycieczkę najkorzystniej jest zaplanować w taki sposób, żeby zobaczyć jak największą liczbę ciekawych obiektów. Pewnym ograniczeniem podczas podróży samochodem elektrycznym jest ilość kilometrów, które może on przejechać na jednym ładowaniu baterii. Minimalny zasięg w kilometrach dla dostępnych w Polsce marek samochodów został przedstawiony w tabeli 1. Zasięg ten jest wykorzystywany przy testach zaproponowanego w pracy algorytmu ewolucyjnego.

Zakładamy, że każda dana atrakcja turystyczna ma przyporządkowany profit (priorytet), który może na przykład wynikać z zainteresowania innych turystów tym obiektem i ich oceny na turystycznych portalach internetowych. Przedstawiony w pracy algorytm ewolucyjny wyznacza trasę zaczynającą się w punkcie, gdzie można naładować baterie samochodu. Wygenerowana trasa przechodzi przez punkty cieszące się sumarycznie największym zainteresowaniem (priorytetem), a jej długość nie przekracza minimalnego zasięgu baterii samochodu elektrycznego. Punkt końcowy wygenerowanej trasy to punkt, gdzie znów można naładować baterie.

Tab. 1. Minimalny zasięg samochodu elektrycznego po naładowaniu baterii

Tab. 1. Minimum range of electric car after charging the battery

Marka samochodu	Minimalny zasięg samochodu (w km)
BMW i3	130
Citroën C-Zero	130
Fiat Panda EV	150
Ford Mondeo EV	120
Nissan Leaf	135
Peugeot iOn	150
Peugeot Electric Partner OriginVentur	120
Renault Fluence	185
Renault Kangoo	170
Renault Zoe	210
Smart for two electric drive	109
Tesla Model S	426

W literaturze anglojęzycznej przedstawiony problem jest znany pod nazwą *orienteeering problem* (OP) [1, 2]. OP może być modelowany za pomocą grafu z wagami, który ma profity przy wierzchołkach. Ponadto w grafie dane są obiekty i odległości między nimi. Rozwiązaniem problemu OP jest trasa z danego punktu startowego do danego punktu końcowego, która obejmuje obiekty o największym sumarycznym proficie i nie

przekracza ograniczenia z góry nałożonego na długość trasy. W literaturze OP ma zastosowania między innymi w logistyce (wyznaczania optymalnych tras cystern rozwożących paliwo) [2] czy w planerach tras turystycznych [9].

W rozdziale 2 przedstawimy szczegółowo definicję problemu OP i opiszemy przejście z tego modelu do poruszanego w pracy problemu optymalizacji trasy samochodu elektrycznego. W rozdziale 3 zaprezentujemy szczegóły algorytmu ewolucyjnego. Wyniki testów na rzeczywistych danych obiektów turystycznych i stacji ładowania baterii zostaną przedstawione w kolejnym rozdziale. W ostatnim rozdziale oprócz podsumowania przedstawimy perspektywy dalszych badań.

2 Zdefiniowanie problemu

Optymalizacja trasy wycieczki samochodu elektrycznego jest w pracy modelowana przez wspomniany problem OP. Nazwa problemu wywodzi się od sportowej dyscypliny zwanej biegiem na orientację [2]. Uczestnicy biegu mają do dyspozycji mapy z zaznaczonymi punktami kontrolnymi, w których zdobywają określoną liczbę punktów oraz informację, jaki czas mogą biegać. Wszyscy wyruszają z tego samego punktu. Zwycięza ten uczestnik, który sumarycznie zdobędzie najwięcej punktów w odwiedzonych punktach kontrolnych (nie ma wymogu, że ma odwiedzić wszystkie) i jednocześnie dobiegnie do mety przed znanym na początku biegu limitem czasowym.

Formalnie problem OP jest modelowany przez graf o zadanej liczbie wierzchołków i oznaczonych odległościach między nimi. Każdy wierzchołek i ma przyporządkowany profit p_i , który można interpretować jako zysk naliczany po odwiedzeniu tego wierzchołka. Ponadto dane jest ograniczenie na długość trasy oznaczone jako t_{max} . Rozwiązaniem problemu OP jest trasa pomiędzy danym punktem startowym s i końcowym e , która daje największy sumaryczny profit odwiedzonych obiektów, a długość trasy nie przekracza danego ograniczenia.

W przypadku optymalizacji trasy wycieczki samochodu elektrycznego dany jest zbiór A obiektów turystycznych oraz zbiór B stacji ładowania baterii. Moc tych zbiorów wynosi odpowiednio $|A|=n$ oraz $|B|=m$. Znane są odległości (bezpośrednie połączenia) pomiędzy obiektami turystycznymi oraz obiektami turystycznymi i stacjami ładowania baterii oznaczone jako t_{ij} dla $i, j \in 1, \dots, n+m$. Każdy obiekt turystyczny ma przyporządkowany profit p_i większy lub równy zero interpretowany jako jego atrakcyjność, zaś stacje ładowania baterii mają profity równe zero. Znane jest również ograniczenie liczby kilometrów, jakie rozważany samochód może przejechać na jednym ładowaniu baterii, oznaczone jako t_{max} . Wśród punktów ze zbioru B wybierana jest stacja startowa s i końcowa e , których odległość nie może przekroczyć ograniczenia t_{max} . Celem przedstawionej w tej pracy optymalizacji jest wyznaczenie trasy obejmującej atrakcje turystyczne o największym sumarycznym proficie pomiędzy wybranym wcześniej punktem startowym s i końcowym e . Ponadto długość wyznaczonej trasy nie może przekroczyć ograniczenia t_{max} oraz każda atrakcja na trasie może wystąpić tylko raz. Poniżej problem został zdefiniowany formalnie.

Niech x_{ij} oznacza zmienną decyzyjną oznaczoną następująco:

$$x_{ij} = \begin{cases} 1, & \text{jeśli bezpośrednie połączenie pomiędzy} \\ & \text{punktami } i \text{ oraz } j \text{ należy do trasy optymalnej,} \\ 0 & \text{w przeciwnym przypadku.} \end{cases} \quad (1)$$

Poszukujemy maksimum funkcji celu w postaci (2):

$$\max \sum_{i=1}^{n+m-1} \sum_{j=2}^{n+m} p_i x_{ij}. \quad (2)$$

Zależność (3) to założenie o początku trasy w punkcie 1 i końcu trasy w punkcie $n+m$:

$$\sum_{j=2}^{n+m} x_{1j} = \sum_{i=1}^{n+m-1} x_{in+m} = 1. \quad (3)$$

Ograniczenia (4) i (5) gwarantują odpowiednio, że długość trasy nie przekracza t_{max} oraz punkty pojawiające się na trasie występują tylko raz:

$$\sum_{i=1}^{n+m-1} \sum_{j=2}^{n+m} t_{ij} x_{ij} \leq t_{max}, \quad (4)$$

$$\sum_{i=1}^{n+m-1} x_{ik} = \sum_{j=2}^{n+m} x_{kj} \leq 1, k \in \{2, \dots, n+m-1\}. \quad (5)$$

Ograniczenie (6) i (7) zapobiega istnieniu podcykli na trasie:

$$2 \leq u_i \leq n, i \in \{2, \dots, n+m\}, \quad (6)$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}), i, j \in \{2, \dots, n+m\}, \quad (7)$$

gdzie u_i oznacza kolejny numer odwiedzonego punktu na trasie.

3 Algorytm ewolucyjny

Do rozwiązania problemu generowania optymalnych tras wycieczki samochodu elektrycznego została wykorzystana zmodyfikowana wersja algorytmu ewolucyjnego, rozwiązującego problem OP, zaproponowana przez autorów w [6]. Trasy generowane w populacji początkowej mają początek i koniec reprezentowane przez stacje ładowania baterii, a pozostała część trasy to ciąg kolejnych atrakcji turystycznych.

Autorzy zdecydowali się na udoskonalanie algorytmu ewolucyjnego zaproponowanego w [6] ze względu na wcześniejsze badania. Eksperymenty opisane w pracy [5] potwierdziły, że algorytm zaproponowany w [6] generuje trasy o większym proficie niż znane z literatury metody lokalnych poszukiwań, takie jak zachłanna adaptacyjna procedura przeszukiwania z elementami losowości (ang. *Greedy Randomize Adaptive Search Procedure* (GRASP)) i jego udoskonalona wersja z wymianą wierzchołków na trasie pomiędzy dwiema losowymi trasami (ang. *path relinking* (PR)) [1]. Czas wykonania algorytmu zaproponowanego w [6] wynosił kilka sekund dla grafu o około 1000 wierzchołków. Wskazuje to na możliwość jego praktycznego zastosowania.

Modyfikacja prezentowanego tu algorytmu w porównaniu do [6] polega na:

- zastosowaniu selekcji z podziałem na grupy;
- wielokrotnej mutacji tej samej trasy;
- przy mutacji wstawiającej, kiedy długość trasy jest już bliska ograniczeniu t_{max} , punkty do wstawienia wybieramy z określonego promienia, żeby wyeliminować od razu te obiekty, których wstawienie powodowałoby przekroczenie t_{max} .

Niech zmienna N_g oznacza z góry ustaloną liczbę iteracji algorytmu, natomiast P_{size} liczbę tras w populacji początkowej. Pseudokod proponowanego algorytmu jest następujący:

```
wygenerowanie populacji o  $P_{size}$  trasach początkowych;  
iteracje=0; best_profit= 0;  
while iteracje <  $N_g$ ;  
    iteracje++;  
    selekcja tras z podziałem populacji na grupy;  
    krzyżowanie jednopunktowe;  
    wielokrotna mutacja;  
    if najlepszy_profit_populacji > bestprofit  
        bestprofit = najlepszy_profit_populacji;  
end;  
return best_profit;
```

W kolejnych podrozdziałach przedstawiamy szczegóły kolejnych etapów algorytmu.

3.1 Populacja początkowa

Algorytm rozpoczyna się od generowania populacji tras początkowych. W kolejnych etapach algorytmu będą one poddawane ewolucji. W algorytmie zostały uwzględnione dwa sposoby generowania tras.

Pierwszy dotyczy przypadku, kiedy następuje powrót do tego samego punktu, z którego rozpoczęła się trasa. Od startowej stacji ładowania s wybierany jest losowy punkt v (tzn. atrakcja turystyczna) połączony krawędzią z s . Długość trasy zwiększa się o odległość pomiędzy s i v . Ponadto punkt v jest usuwany ze zbioru wierzchołków, które mogą być użyte. Proces jest powtarzany, dopóki długość trasy nie przekracza wartości $t_{max}/2$. Jeśli wartość ta zostanie przekroczona, wtedy ostatni punkt jest usuwany, a kolejne punkty są dołączone do trasy w odwrotnej kolejności (trasa jest symetryczna względem środka i kończy się na punkcie s). Tak utworzone trasy mają powtarzające się punkty, ale w procesie ewolucji symetria zostanie zaburzona.

Drugi sposób generowania tras początkowych obejmuje przypadek, kiedy początkowa stacja ładowania s jest różna do stacji końcowej e . Wtedy pomiędzy stacje ładowania s i e wstawiamy losowo wybraną atrakcję sąsiadującą (połączoną krawędziami) z tymi wierzchołkami, wykluczając ją jednocześnie ze zbioru kandydatów możliwych do wstawienia. Kolejne punkty trasy dostawiamy zawsze pomiędzy stację e i ostatnio wstawiony punkt trasy, powtarzając proces, dopóki długość trasy nie przekroczy t_{max} . Tak wygenerowana trasa nie ma wierzchołków powtarzających się.

3.2 Selekcja

Selekcja ma na celu wybór najlepszych osobników do nowej populacji, która również ma P_{size} tras. Stosowana jest selekcja turniejowa z podziałem na grupy. Populacja jest dzielona w kolejności występowania osobników na k grup. Każda grupa zawiera $\frac{P_{size}}{k}$ tras. Spośród osobników grupy wybieranych jest t_{size} losowych tras, a wśród nich dokonywany jest wybór najsilniejszego osobnika. Za najsilniejszego osobnika uważamy tego, który ma największą wartość funkcji celu (2).

Trasa z największą wartością funkcji celu (2) wybierana jest do nowej populacji, natomiast t_{size} poprzednio wybranych tras zwracanych jest do tej samej grupy, z której

zostały wybrane. Proces selekcji w tej samej grupie jest powtarzany $\frac{P_{size}}{k}$ razy. W identyczny sposób selekcja jest przeprowadzana w każdej grupie.

3.3 Krzyżowanie

Krzyżowanie ma na celu połączenie dwóch losowych tras w jedną nową. W algorytmie stosowane jest krzyżowanie jednopunktowe. Po losowym wyborze dwóch tras wyszukiwany jest pierwszy punkt wspólny występujący w jednej, jak i drugiej trasie (oprócz punktu będącego początkiem i końcem trasy) – tzw. punkt przecięcia tych tras. Jeśli taki punkt nie istnieje, krzyżowanie nie zachodzi. W przypadku gdy punkt wspólny istnieje, wymieniane są fragmenty tras od punktu wspólnego do końca tras. Wynikiem takiej wymiany są dwie nowe trasy. Jeśli dwie nowe trasy spełniają ograniczenie t_{max} , to zastępują one rodzicielskie trasy. Jeśli jedna trasa nie spełnia ograniczenia t_{max} , wtedy odpowiednia rodzicielska trasa zastępuje ją w nowej populacji. W przypadku gdy obydwie nowe trasy nie spełniają ograniczenia t_{max} , żadne zmiany w nowej populacji nie zachodzą.

3.4 Mutacja

W algorytmie zastosowano mutację wstawiającą lub usuwającą obiekt z trasy (prawdopodobieństwo każdej mutacji wynosi 0.5 i zostało wyznaczone eksperymentalnie w poprzedniej wersji algorytmu [6]). Losowo wybrana trasa jest wielokrotnie poddawana mutacji. N_m to parametr oznaczający, ile razy wybrana trasa podlega mutacji.

W przypadku mutacji wstawiającej wybierany jest losowy obiekt turystyczny v_j nie należący do trasy. Następnie określamy najdogodniejsze miejsce do wstawienia obiektu v_j na trasie. Rozważmy wstawienie obiektu v_j o proficie p_j pomiędzy v_i oraz v_{i+1} . Następująca zależność (8):

$$\Delta_{ij} = t_{ij} + t_{ji+1} - t_{i+1} \quad (8)$$

wyraża, o ile wzrosła długość trasy po wstawieniu v_j pomiędzy v_i oraz v_{i+1} . Miejsce rozważane do najkorzystniejszego wstawienia obiektu v_j jest wyznaczone na podstawie następnego równania:

$$\hat{s}_{ij} = \frac{p_j^2}{\Delta_{ij}} \quad (9)$$

Jeśli Δ_{ij} jest mniejsze niż 1, wtedy uwzględniana jest w równaniu (9) tylko wartość p_j^2 . Miejsce rozważane do wstawienia obiektu j ma maksymalną wartość \hat{s}_{ij} . Obiekt j jest wstawiany w wyznaczone najkorzystniejsze miejsce trasy, jeśli jej długość po wstawieniu nie przekracza t_{max} oraz zmodyfikowana trasa ma wyższą wartość funkcji celu (2) niż przed modyfikacją. Jeśli długość trasy t jest większa niż $0.95t_{max}$ to wybieramy odpowiedni element do wstawienia. Wybieramy połączenia między wierzchołkami, których odległość jest mniejsza niż $t_{max} - d_t$, gdzie d_t jest sumą odległości kolejnych wierzchołków na trasie. Następnie spośród nich losujemy element do wstawienia.

Mutacja usuwająca ma na celu usunięcie takich obiektów, których profit jest stosunkowo mały względem przyrostu długości trasy. Załóżmy, że rozważamy trasę t o długości d_t i wierzchołek v_i o proficie p_i , który znajduje się pomiędzy wierzchołkami

v_{i-1} oraz v_{i+1} . Dla każdego wierzchołka v_i z trasy (oprócz wierzchołka startowego i końcowego) liczony jest wskaźnik s_i wyznaczony na podstawie (10):

$$s_i = \frac{p_i^2}{d_{t-t_{i-1}+1}}, \quad (10)$$

gdzie t_{i-1} oznacza odległość pomiędzy wierzchołkiem v_{i-1} oraz v_{i+1} .

Ostatecznie usuwany jest z trasy t wierzchołek v_i o najmniejszej wartości s_i .

4 Testy algorytmu genetycznego

Algorytm został zaimplementowany w C++ i przetestowany na komputerze z procesorem Intel Core i7, 1.73GHz CPU. Testy przeprowadzono na rzeczywistej bazie obiektów turystycznych Podlasia, która obejmowała 531 obiekty. Współrzędne geograficzne stacji ładowania baterii zostały zaczerpnięte z portalu internetowego ecomoto [14]. Każdy obiekt turystyczny w bazie identyfikowany jest przez współrzędne geograficzne i profit. Profit został wyznaczony na podstawie głosów internautów oddanych na dany obiekt. Odległości pomiędzy każdą parą obiektów oraz pomiędzy obiektami i stacjami ładowania baterii zostały wyznaczone na podstawie zapytań do bazy Bing Map Service (Routes API). Na wejściu podawane były współrzędne geograficzne punktów, pomiędzy którymi chcieliśmy uzyskać odległości tras samochodowych w kilometrach z minimalnym czasem dojazdu.

Testy algorytmu obejmowały trasy, które były cyklami, oraz trasy, w których różny był punkt startowy i końcowy.

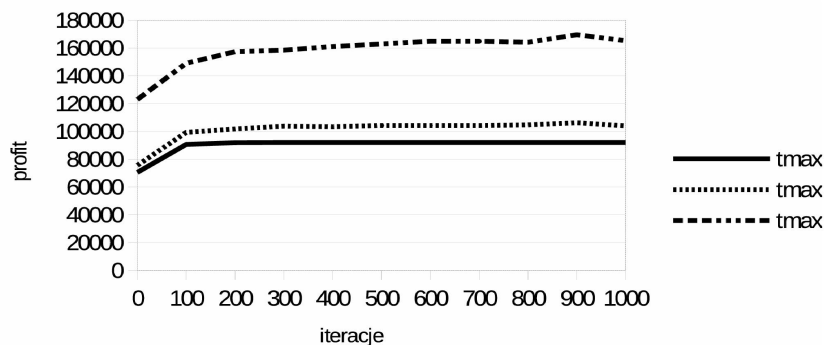
Ponadto w testach były brane pod uwagę różne limity długości tras, m.in.: 120, 130, 135, 145, 150, 170, 185, 190, 210, 220, 500 km. Limit ten to liczba kilometrów możliwych do przejechania po całkowitym naładowaniu baterii.

Podczas licznych testów przy zmiennej liczbie iteracji algorytmu oraz przy zmiennej liczbie ilości grup w selekcji turniejowej ustalono wartości parametrów algorytmu, zamieszczone w tabeli 2.

Tab. 2. Wartości parametrów algorytmu ewolucyjnego

Tab. 2. The parameter values of the evolutionary algorithm

oznaczenie	znaczenie parametru	wartość parametru
P_{size}	liczba tras w populacji początkowej	300
K	liczba grup w selekcji turniejowej	15
T_{size}	liczba tras wybranych do turnieju z grupy	3
N_g	liczba iteracji algorytmu	1000



Rys. 1. Zbieżność algorytmu ewolucyjnego dla wybranych trzech wartości t_{max}

Fig.1. Convergence of the evolutionary algorithm for different values of t_{max} .

Przy wyborze wartości parametru N_g opisującego maksymalną liczbę iteracji algorytmu zauważono (rys. 1), że dla mniejszych wartości t_{max} (do 220) stabilizacja wartości profitu następuje po około 500 iteracjach. Jednak dla wartości $t_{max}=500$ wartość profitu stabilizuje się po około 1000 iteracji. Tabela 3 pokazuje, jak zmienia się profit w zależności od liczby grup (parametr k), na które jest podzielona cała populacja przy selekcji. Okazuje się, że przy mniejszych wartościach t_{max} liczba grup nie wpływa na wartość profitu (np. $t_{max}=145$), natomiast przy większych wartościach wraz z ilością grup profit wzrasta. Zaobserwowano, że wzrost profitu nie jest znaczący po przekroczeniu 15 grup, natomiast coraz bardziej wzrasta czas wykonania algorytmu.

Tab. 3. Wartość profitów dla wybranych t_{max} w zależności od liczby grup w selekcji

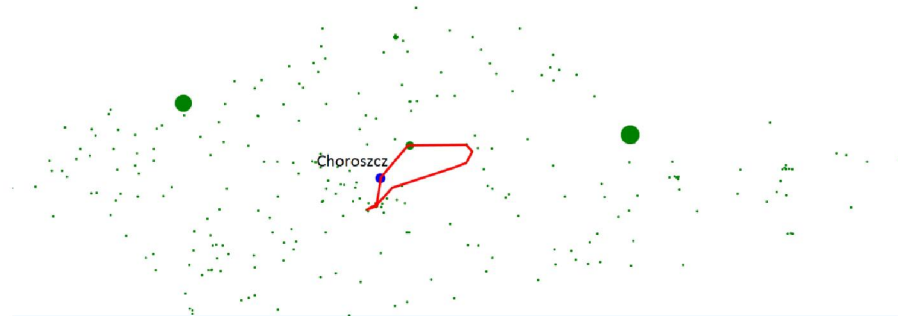
Tab.3. The value of profits for different t_{max} depending on the number of groups in selection

	$t_{max}=145$	$t_{max}=220$	$t_{max}=500$
$k=3$	92059	103433	165868
$k=5$	92059	103999	165901
$k=15$	92059	104242	165954

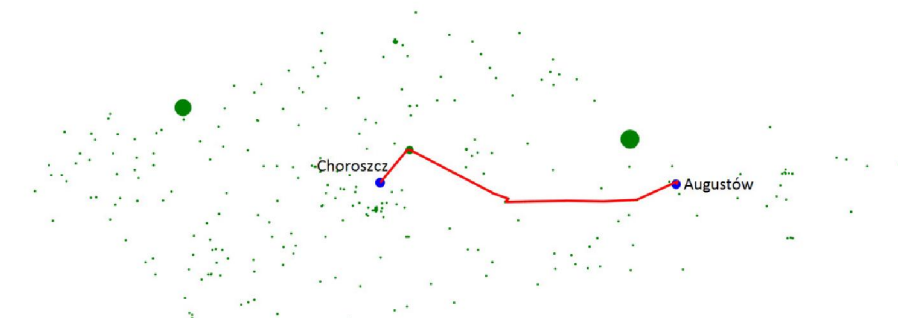
W czasie testów wynikowych algorytmu uwzględniana była zmienna liczba mutacji jednej lub kilku tras. Szczegółowe wyniki zostały uwzględnione w tabeli 4 oraz 5. Jeden test to piętnaście uruchomień programu (jedno uruchomienie to 1000 iteracji algorytmu). Wynikiem jest trasa o najlepszym proficie z tych piętnastu uruchomień. Czas wygenerowania tej trasy to czas związany z piętnastoma uruchomieniami programu.

Na rysunku 2 i 3 zostały zaprezentowane przykładowe trasy, które zostały wygenerowane przy wartościach parametrów z tabeli 2. Obiekty turystyczne Podlasia zostały oznaczone za pomocą punktów. Atrakcyjność każdego obiektu zróżnicowana

została przez wielkość odpowiadającego punktu. Linia ciągłą została oznaczona przykładowa trasa. Punkty początkowe i końcowe trasy zostały podpisane.



Rys. 2. Przykładowa trasa – punkt startowy i końcowy jest taki sam oraz $t_{max}=120$ km
Fig. 2. Example route – start and end point is the same and $t_{max}=120$ km



Rys. 3. Przykładowa trasa – punkt startowy i końcowy są różne oraz $t_{max}=120$ km
Fig. 3. Example route – start and end point are different and $t_{max}=120$ km

W tabeli 4 zostały przedstawione wyniki profitów wygenerowanych tras oraz czas działania algorytmu dla różnych limitów długości tras przy pięciokrotnej i dziesięciokrotnej mutacji trasy w czasie jednej iteracji algorytmu (punkt startowy - Choroszcz, końcowy - Augustów). Procentowa różnica pomiędzy wynikami kolumny (b) i (a) została wyliczona w kolumnie (c) z zależności $\%gap(b)(a)=100*(1-\frac{b}{a})$.

Tab. 4. Profit trasy i czas działania algorytmu w zależności od limitu trasy i liczby mutacji w jednej iteracji algorytmu. Porównanie wyników poprzedniej wersji algorytmu

Tab. 4. Profit and execution time of the algorithm depending on the limit of the route and the number of mutations in a single iteration of the algorithm. Comparison the results of the previous version of the algorithm

Limit trasy	5-krotna mutacja		10-krotna mutacja		% gap(b)(a)	poprzednia wersja algorytmu [6]		% gap(a)(d)
	Profit (a)	Czas (sek.)	Profit (b)	Czas (sek.)	(c)	Profit (d)	Czas (sek.)	(e)
120	39576	3.6	39677	3.8	-0.26	39423	4.8	-0.39
130	77870	3.7	77870	4.0	0.00	77870	5.5	0.00
135	82199	3.7	82199	4.1	0.00	82199	5.4	0.00
145	92059	3.9	92059	4.1	0.00	92059	5.1	0.00
150	93438	4.2	93438	4.3	0.00	93143	4.4	-0.32
170	96909	4.7	96760	4.4	0.15	96301	4.6	-0.63
185	99979	5.8	99289	5.8	0.69	96014	4.4	-4.13
190	100493	5.3	100152	6.3	0.34	97489	5.0	-3.08
210	104526	6.2	102948	6.3	1,51	99120	4.5	-5.45
220	104041	6.2	104242	6.7	-0.19	97893	4.8	-6.28
500	165295	11.6	165532	13.4	-0.14	150658	8.0	-9.72

Ponadto w tabeli 4 umieszczono porównanie wyników z poprzednią wersją algorytmu [6]. W kolumnie (e) została przedstawiona procentowa różnica pomiędzy wynikami kolumny (a) i (d), która została wyliczona z zależności $\%gap(a)(d) = 100 * (1 - \frac{a}{d})$. Z tabeli 4 wynika, że zwielokrotnienie liczby mutacji na jednym osobniku w jednej iteracji algorytmu tylko w przypadku $t_{max}=120, 220, 500$ wpływa na zwiększenie profitu trasy. Wzrost jednak nie przekracza 0.3%. Dla tras o limicie od 130 do 150 nie ma różnic w profitach. Dla tras o limicie od 170 do 210 zwielokrotnienie liczby mutacji nie wpływa na zwiększenie profitu. Zaproponowany w tej pracy algorytm w porównaniu z poprzednią wersją algorytmu [6] generuje lepsze trasy (nawet o ok. 10%) dla dużych wartości, na przykład $t_{max}=500$, przy porównywalnych czasach wykonania algorytmów. Dla tras o limicie od 130 do 145 nie ma różnic w profitach oraz znaczących różnic pomiędzy czasem wykonania zaproponowanego algorytmu a wersją zaproponowaną w [6]. Podjęto próby wzmocnienia algorytmu poprzez mutowanie nie tylko jednej trasy ale i większej liczby tras podczas jednej iteracji algorytmu.

W tabeli 5 znajdują się wyniki badań, gdy dwie lub cztery trasy były mutowane dziesięciokrotnie podczas jednej iteracji algorytmu. Procentowa różnica pomiędzy wynikami kolumny (f) i (g) została wyliczona w kolumnie (h) z zależności $\%gap(g)(f) = 100 * (1 - \frac{g}{f})$. Dla wartości $t_{max}=135, 170, 185, 500$ mutowanie czterech tras dało minimalnie gorszy wynik (do ok. 0.4%) niż mutowanie dwóch tras jednocześnie. W przypadku $t_{max}=120, 130, 145, 150$ różnicy w wynikach nie ma. Dla tras o limicie

od 190 do 500 zwiększenie liczby mutowanych tras nie wpływa znacząco na wynik (różnica jest do ok. 2%), jednocześnie powodując wydłużenie czasu generowania wynikowej trasy.

Tab. 5. Profit trasy i czas działania algorytmu w zależności od limitu trasy i wielokrotnej mutacji dwóch i czterech tras w jednej iteracji algorytmu

Tab. 5. Profit and execution time of the algorithm depending on the limit of the route and the multiple mutation (two and four routes) in a single iteration of the algorithm.

Limit trasy	10-krotna mutacja dwóch tras		10-krotna mutacja czterech tras		% gap(g)(f) (h)
	Profit (f)	Czas (sek.)	Profit (g)	Czas (sek.)	
120	39677	4.3	39677	4.7	0.00
130	77870	3.9	77870	4.8	0.00
135	82242	4.1	82199	5.5	-0.05
145	92059	4.3	92059	5.3	0.00
150	93438	4.4	93438	5.6	0.00
170	97093	5.7	96710	6.8	-0.40
185	100062	6.1	99661	7.5	-0.40
190	100562	5.8	100568	7.8	0.01
210	102535	6.5	104785	8.8	2.15
220	106202	7.7	105547	8.5	0.62
500	166751	15.5	170434	18.2	2.16

5 Wnioski i perspektywa dalszych badań

Większość samochodów elektrycznych dostępnych na rynku ma wbudowane oprogramowanie, które ułatwia i uprzyjemnia jazdę i podróżowanie. Przykładem jest samochód BMW i3, który ma nawigację lokalizującą publiczne stacje ładowania. Funkcja Range Assistant w tym samochodzie jest przydatna w przypadku planowania trasy, jak i w trakcie dojazdu do celu. Jeśli miejsce docelowe leży poza zasięgiem baterii, system sugeruje przejść w tryb oszczędnościowy i przelicza inną wersję trasy dla BMW i3. Zaproponowany w pracy algorytm ewolucyjny może być częścią modułu takiego oprogramowania lub aplikacją na smartfony, która planuje wycieczkę po najbardziej atrakcyjnych punktach, uwzględniając w punkcie początkowym i końcowym trasy ładowanie baterii. Trasa wycieczki przynosi możliwie największą satysfakcję dla podróżujących, bez zbędnego zanieczyszczenia środowiska i bez obawy, że zabraknie energii w czasie podróży.

W dalszych badaniach planowane jest rozszerzenie algorytmu o możliwość generowania optymalnej wycieczki wieloetapowej, gdzie będzie możliwość naładowania baterii po ich wyczerpaniu i kontynuowania podróży. W takim przypadku wynikiem algorytmu będzie trasa najbardziej atrakcyjnej wycieczki kilkudniowej. Ponadto zostaną wyznaczone optymalnie stacje ładowania baterii samochodu elektrycznego, np. ładowanie będzie uwzględniane podczas noclegu, a limit kilometrów do przejechania w czasie jednego dnia będzie wyznaczony pojemnością baterii.

Dotychczas w optymalizacji nie był brany pod uwagę całkowity czas trwania wycieczki. Istnieje możliwość uwzględnienia w wygenerowanej trasie przedziałów czasowych, w których dany obiekt turystyczny można zwiedzać, oraz czasu zwiedzania obiektu, jak również czasu przejazdu między obiektami.

Literatura

1. Campos V., Marti R., Sanchez-Oro J., Duarte A.: Grasp with Path Relinking for the Orienteering Problem, *Journal of the Operational Research Society*, 1-14, 2013
 2. Chao I.M., Golden B., Wasil E.: A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, 88(3), pp. 475-489, 1996
 3. Ćirović G., Pamučar D., Božanić D.: Green logistic vehicle routing problem: Routing light delivery vehicles in urban areas using a neuro-fuzzy model. *Expert Systems with Applications*, 41(9), pp. 4245-4258, 2014
 4. Dekker R., Bloemhof J., Mallidis I.: Operations Research for green logistics – An overview of aspects, issues, contributions and challenges, *European Journal of Operational Research*, 219(3), pp. 671-679, 2012
 5. Karbowska-Chilińska J., Zabielski P.: A genetic algorithm vs. local search methods for solving the orienteering problem in large networks. *Lecture Notes in Computer Science*, 7828 LNAI, pp. 11-20, 2012
 6. Karbowska-Chilińska J., Koszelew J., Ostrowski K., Zabielski P.: Genetic algorithm solving orienteering problem in large networks. *Frontiers in Artificial Intelligence and Applications*, 243, pp. 28-38, 2012
 7. Lin C., Choy K.L., Ho G.T.S., Chung S.H., Lam H.Y.: Survey of Green Vehicle Routing Problem: Past and future trends, *Expert Systems with Applications*, 41(4, Part 1), pp. 1118–1138, 2014
 8. Sekozawa T., Yamamoto S., Masuda K.: Maximization of EV tour points: The problem and a solution, *IEEJ Transactions on Electronics, Information and Systems*, 134 (6), pp. 773-779, 2014
 9. Vansteenwegen P., Souffriau W., Vanden Berghe G., Van Oudheusden D.: The City Trip Planner: An expert system for tourists. *Expert Systems with Applications*, 38(6), pp. 6540-6546, 2011
 10. Xi X., Sioshans R., Marano V.: Simulation–optimization model for location of a public electric vehicle charging infrastructure. *Transportation Research Part D: Transport and Environment*, 22, pp. 60-69, 2013
 11. Yang J., Sun H.: Battery swap station location-routing problem with capacitated electric vehicles, *Computers & Operations Research*, 55, pp. 217-232, 2015
- Strony internetowe - dostęp w dniu 15.09.2014:
12. <http://www.allinx.eu/content/electric-car-sales-eu-increase-77>
 13. http://www.consilium.europa.eu/uedocs/cms_data/docs/pressdata/en/trans/141906.pdf
 14. <http://www.ecomoto.info/punkty/nowe.html> (stacje ładowania)
 15. <http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//NONSGML+AMD+A7-2013-0444+120-120+DOC+PDF+V0//EN>
 16. <http://gramwzielone.pl/auto-ekologiczne/11372/lubelskie-gminy-stworza-siec-wypożyczalni-samochodow-elektrycznych>
 17. <http://www.mobilnagdynamia.pl/samochod/148-pojazdy-elektryczne-w-norwegii>

18. http://samochodyelektryczne.org/kategorie/display_right2.htm
19. http://samochodyelektryczne.org/prezentacja_na_temat_sieci_szybkiego_ladowani_w_estonii.htm

Streszczenie

Samochód elektryczny jest zeroemisyjny, bardzo cichy i tani w eksploatacji. Może być wykorzystywany zarówno jako samochód miejski, jak i w podróżowaniu turystycznym. W artykule przedstawiamy algorytm, który zaplanuje trasę wycieczki w taki sposób, żeby odwiedzone zostały najatrakcyjniejsze obiekty turystyczne, oraz uwzględni w punkcie początkowym i końcowym trasy ładowanie baterii. Atrakcyjność obiektu jest wyznaczana na podstawie opinii internatów o danym obiekcie. Maksymalna długość wycieczki to liczba kilometrów, jakie samochód może przejechać na jednym ładowaniu baterii. Zaproponowany przez autorów algorytm ewolucyjny został przetestowany na rzeczywistych danych, obejmujących obiekty turystyczne i stacje ładowania baterii na Podlasiu. Czas działania algorytmu oraz wyniki testów wykazują, że opisany algorytm może być częścią modułu oprogramowania stosowanego w samochodach elektrycznych lub aplikacją na smartfony, która ułatwia i uprzyjemnia podróżowanie, a jednocześnie pozwala optymalnie wykorzystać energię samochodu elektrycznego.

Słowa kluczowe: graf ważony z profitami przy wierzchołkach, generowanie trasy z profitami i ograniczeniami, optymalizacja trasy, samochód elektryczny

The optimization of a tourist route of electric vehicle

Summary

Electric vehicle (EV) does not emit harmful gases, it is very quiet and cheap to use. It can be used both as a city car and in the travel tourism. In this paper we present an algorithm that will plan a route of electric vehicle in such a way that the most attractive tourist points of interest are visited and takes into account the starting point and the final point of a route as a EV charging station. Attractiveness of points of interest is determined on the basis of a ranking on the internet. The maximum length of the tour is determined by the number of kilometres that the car can travel on a single battery charge. The evolutionary algorithm proposed by us was tested on realistic database points of interests and EV charging stations in Podlasie region. On the basis of the tests results and execution times of the algorithm we conclude that the proposed algorithm could be a part of a software module in EV or an application for smart phones which makes traveling easier and more comfortable. Moreover EV battery power is used optimally.

Keywords: orienteering problem, route optimisation, evolutionary algorithm, electrical vehicle

Artykuł został sfinansowany z prac badawczych S/WI/1/2014 oraz W/WI/2/2013 realizowanych na Wydziale Informatyki Politechniki Białostockiej.

Autorzy dziękują firmie WMP KONWERSJA Tomasz Płoński za informacje o dostępnych stacjach ładowania i wykorzystaniu samochodów elektrycznych w Polsce.