Informatyka w Edukacji, XV
UMK Toruń, 2018

PROBLEM KOMIWOJAŻERA W ŚWIETLE KSZTAŁTOWANIA MYŚLENIA KOMPUTACYJNEGO NA WYBRANYCH ETAPACH EDUKACYJNYCH

Sławomir Herma, Włodzimierz Raczek, Bartłomiej Żywczak
V Liceum Ogólnokształcące, Bielsko-Biała, ul. Lompy 10,
Akademia Techniczno-Humanistyczna, ul. Willowa 2
43-300 Bielsko-Biała
slawomir.herma@lo5.bielsko.pl, sherma@ath.bielsko.pl,
wlodzimierz.raczek@lo5.bielsko.pl,
bartlomiej.zywczak@lo5.bielsko.pl

Abstract. The paper presents the travelling salesman problem (TSP) as an example of a problem with a significant didactic potential, giving a wide range of possibilities to shape IT competences and computational thinking. The authors present a set of didactic situations in which the travelling salesman problem is the starting point for multidirectional intellectual development. The work is a result of many years of didactic and scientific activities, carried out both in educational and academic institutions.

1 Wstęp

Stawiając sobie za cel poszukiwanie zagadnień, których dydaktyczny potencjał daje nauczycielowi szerokie możliwości w zakresie stosowania różnych narzędzi, metod i środków służących asymilacji wiedzy przez uczniów i studentów, Autorzy postanowili dokonać kilku przykładowych adaptacji **problemu komiwojażera** do realizacji zajęć na różnych etapach edukacyjnych. Korzystając z doświadczeń będących rezultatem szeregu aktywności podejmowanych na wielu poziomach kształcenia, można zaryzykować stwierdzenie, że rozwijanie zwłaszcza myślenia komputacyjnego, może być znakomicie wzmacniane angażowaniem uczniów nie tylko do uczestnictwa w wartościowych grach dydaktycznych, ale również inspirowaniem ich to tego, by sami próbowali je projektować i programować. Problem komiwojażera, jako zagadnienie o charakterze optymalizacyjnym i decyzyjnym wydaje się stanowić dobre podłoże do uprawiania wspomnianych aktywności nie tylko przez świadomych nauczycieli, lecz przez uczniów oraz studentów, zwłaszcza tych szczególnie uzdolnionych.

2 Problem komiwojażera i zagadnienia pochodne

Jednym z najstarszych problemów optymalizacyjnych jest tzw. problem komiwojażera (lub zadanie komiwojażera, w skrócie TSP, od angielskiej nazwy *travelling salesman problem*). Sformułowany jest on w następujący sposób:

„Komiwojażer ma odwiedzić dokładnie raz każdą z wybranych miejscowości i powrócić do tej, z której rozpoczął swą podróż. Dla znanych kosztów przejazdu między każdą parą miejscowości należy zaplanować komiwojażerowi drogę przejazdu tak, by mógł odwiedzić każdą miejscowość dokładnie raz i całkowity koszt podróży był najmniejszy." [4] Problem komiwojażera jest ściśle związany z teorią grafów, czyli działem matematyki zajmującym się badaniem własności obiektów zwanych grafami. Występuje więc symetryczny problem komiwojażera, w przypadku, gdy modelem jest graf nieskierowany oraz asymetryczny problem komiwojażera, kiedy operuje się na grafie skierowanym.

Szczególnym przypadkiem symetrycznego problemu komiwojażera jest **euklidesowe zadanie komiwojażera**, w którym wierzchołkom grafu przyporządkowane są punkty w przestrzeni trójwymiarowej, a odległości krawędzi są odległościami euklidesowymi między tymi punktami. Dla przestrzeni 2D mówi się wówczas o euklidesowym zadaniu komiwojażera na płaszczyźnie [3].

Istnieje także odmiana problemu komiwojażera, w której po zakończeniu podróży nie musi on wracać do miasta początkowego, mówi się wówczas o problemie **drogi Hamiltona**. Do problemu drogi Hamiltona sprowadza się wiele praktycznych problemów dotyczących objazdu tras.

Należy również nadmienić, że w przypadku niektórych grafów problem komiwojażera nie ma rozwiązania, lub optymalna droga komiwojażera nie pokrywa się z najkrótszym cyklem Hamiltona w grafie. Warunkiem na to, aby przynajmniej jeden z cykli Hamiltona był rozwiązaniem problemu komiwojażera jest, aby wagi w sieci spełniały tzw. warunek trójkąta:

$$w_{i,j} \leq w_{i,k} + w_{k,j} \quad \text{dla wszystkich } i, j, k, \ldots, n,$$

gdzie $w_{i,j}$ jest wagą odpowiedniej krawędzi grafu [4].

Dla urozmaicenia, w niniejszej pracy Autorzy oparli rozwiązanie zagadnienia komiwojażera o model decyzyjny i metodę programowania wieloetapowego, przedstawioną przez Mareckiego w [1].

3 Model zagadnienia

3.1 Dane wejściowe

Dany jest zbiór miast (lokalizacji) przeznaczonych do odwiedzenia przez komiwojażera:

$$\Omega = \{\omega_n\}_{n=1,\dots,N}$$

Znana jest macierz odległości pomiędzy poszczególnymi miastami:

$$D = \left[d_{i,j}\right]_{i,j=1,\dots,N}$$

oraz znane jest miasto startowe: ω_0 .

3.2 Ograniczenia modelu (warunki brzegowe)

Na podstawie definicji zagadnienia, komiwojażer winien odwiedzić każde miasto raz i tylko raz. Jeśli zatem Ω' oznacza zbiór miast odwiedzonych, wówczas:

$$\Omega' = \Omega$$

Ponadto, jeżeli przyjąć że rozpatrywany będzie również wariant asynchroniczny zagadnienia, odległość pomiędzy miastem ω_i oraz ω_j nie musi być równa odległości z ω_j do ω_i, tzn.: $d_{i,j} \neq d_{j,i}$

3.3 Stan procesu decyzyjnego

Zgodnie z propozycją przedstawioną przez Mareckiego w [1], do rozwiązania opisywanego zagadnienia przyjęto model programowania wieloetapowego, polegający na permutacyjnym generowaniu kolejności odwiedzania poszczególnych miast (lokalizacji) ω_n i wyborze spośród nich rozwiązania optymalnego (kierując się kryterium minimalizacji długości przebytej drogi). Skoro ostatecznym rezultatem opisywanego zagadnienia ma być optymalna marszruta komiwojażera, zasadne (i wygodne z punktu widzenia dalszych implementacji informatycznych algorytmu) wydaje się, by stan procesu odzwierciedlony był za pomocą jednowymiarowej struktury tablicowej (tzn. wektora), zawierającej numer kolejny, jaki odwiedzane miasto (lokalizacja) zajmuje na trasie komiwojażera. W przypadku przeciwnym, tzn. gdy miasto nie zostało jeszcze odwiedzone, wpisywane jest 0, na przykład:

$$P = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \\ 0 \\ 0 \\ 2 \end{bmatrix} \begin{array}{l} \rightarrow \omega_1 \text{ nie odwiedzone} \\ \rightarrow \omega_2 \text{ jako pierwsze} \\ \rightarrow \omega_3 \text{ nie odwiedzone} \\ \rightarrow \omega_4 \text{ jako trzecie} \\ \rightarrow \omega_5 \text{ nie odwiedzone} \\ \rightarrow \omega_6 \text{ nie odwiedzone} \\ \rightarrow \omega_7 \text{ jako drugie} \end{array}$$

Według powyższego, przykładowa marszruta komiwojażera przedstawia się następująco:

$$\omega_0 \to \omega_2 \to \omega_7 \to \omega_4$$

Realizacja tego zadania wymaga zatem zdefiniowania pojęcia stanu procesu decyzyjnego, jako następującego wektora:

$$P^{e,l} = \left[p_i^{e,l} \right]_{i=1,\dots,N}$$

gdzie:

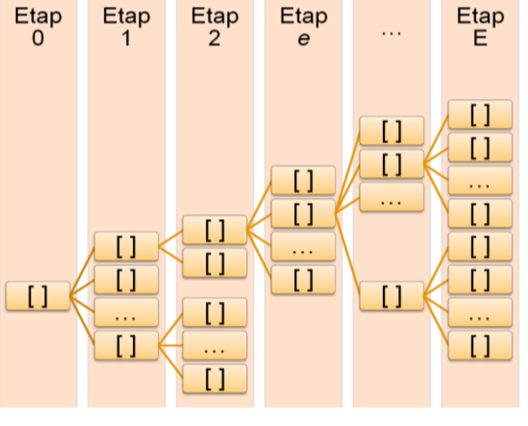
$$p_i^{e,l} = \begin{cases} j \to \text{miasto } \omega_i \text{ jest j-tym na drodze} \\ 0 \to \text{w przeciwnym wypadku} \end{cases}$$

oraz:

e - jest numerem etapu procesu decyzyjnego,

l - jest numerem wektora stanu na danym etapie e .

Idea algorytmu programowania wieloetapowego, przedstawiona w opracowaniu [1] zakłada, że schemat decyzyjny rozłożony jest na kilka etapów. Etap „0” – początkowy, charakteryzuje się tym, że wektor stanu zawiera wyłącznie wartości „0”. Oznacza to, że do marszruty komiwojażera nie przydzielono jeszcze ani jednego miasta. Kolejny etap „1” zawiera zestaw wektorów stanu, z których każdy posiada już przypisany jeden i tylko jeden, jednak za każdym razem inny element (miasto). Liczba stanów na tym etapie jest równa liczbie miast (lokalizacji) ze zbioru Ω. Na każdym kolejnym etapie procesu decyzyjnego podejmowana jest próba utworzenia zestawu wektorów stanu z kolejnym przypisanym do harmonogramu miastem ω_n, spełniającym podane wcześniej warunki brzegowe.



Rysunek 1 Generowanie kolejnych etapów decyzyjnych

Schemat tego procesu jest więc drzewem decyzyjnym, które, jako byt hierarchiczny, posiada na każdym kolejnym etapie wektory stanu, dziedziczące niejako uzyskane już na etapach poprzednich decyzje. Stąd też, etap końcowy zawierać będzie zestaw rozwiązań problemu, czyli harmonogramów realizujących wszystkie możliwe, dopuszczalne przyporządkowania.

W szczególności stan początkowy jest wektorem zerowym – wszystkie jego elementy p_i mają wartość „0". Stan końcowy zaś, to wektor posiadający na każdej pozycji wartość różną (większą) od zera:

$$\underset{n=1,..,N}{\forall} p_i^{N,l} > 0$$

3.4 Procedura generowania stanu

Kolejny stan procesu decyzyjnego można wygenerować dla każdego elementu ω_n ze zbioru Ω (wszystkich dostępnych miast), jeśli w stanie poprzednim $P^{e-1,\lambda}$ na pozycji odpowiadającej n - temu miastu była wartość „0" (tzn. nie zostało ono dotąd odwiedzone - $p_n^{e-1,\lambda} = 0$). Wówczas wektor stanu kolejnego wyznaczany jest jako suma wektora stanu poprzedniego i wektora Δp tj. ($P^{e,l} = P^{e-1,\lambda} + \Delta p$). Wektor Δp charakteryzuje się tym, że na każdej pozycji ma liczbę „0" z wyjątkiem tej, która odpowiada odwiedzanemu miastu - ω_n :

$$\underset{n=1,..,N}{\forall} \left(p_n^{e-1,\lambda} = 0 \right) \Rightarrow P^{e,l} = P^{e-1,\lambda} + \Delta p$$

gdzie:

$$\Delta p_i = \begin{cases} e \rightarrow i = n \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

Przykład: Realizacja przyporządkowania miasta ω_3 do marszruty komiwojażera przedstawia się następująco:

$$\begin{array}{ccccccc} & P^{e-1,\lambda} & & \Delta p & & P^{e,l} \\ & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \\ 0 \\ 0 \\ 2 \end{bmatrix} & + & \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & = & \begin{bmatrix} 0 \\ 1 \\ 4 \\ 3 \\ 0 \\ 0 \\ 2 \end{bmatrix} \end{array}$$

($\omega_3 \rightarrow$ wskazuje trzecią pozycję wektorów)

3.5 Definicja wartości stanu

Wartość stanu umożliwia ocenę jakości uzyskanego rozwiązania w procesie decyzyjnym i definiowana jest jako długość drogi przebytej według marszruty określonej przez ten stan:

$$V^{e,l} = \sum_{i=1}^{N} d_{N_i^{e,l}, N_{i+1}^{e,l}}$$

gdzie:

$$N_i^{e,l} = n : (p_n^{e,l} = i)$$

jest numerem ostatnio odwiedzonego miasta.

Przykład: Dla następującego wektora stanu:

$$P^{e,l} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

kolejność odwiedzanych miast przedstawia się następująco:

$$\omega_0 \rightarrow \omega_2 \rightarrow \omega_7 \rightarrow \omega_4$$

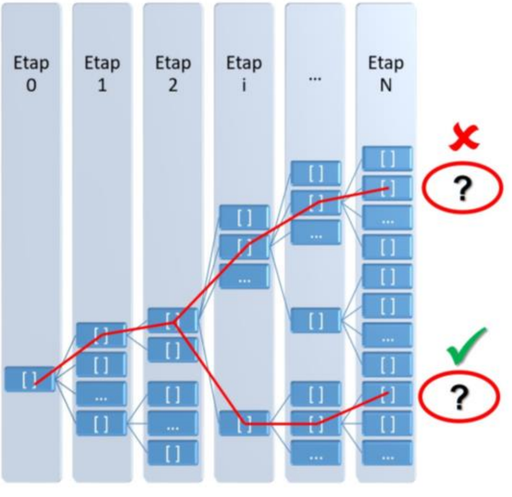
Wobec powyższego:

$$\sum_{i=1}^{N} d_{N_i^{e,l}, N_{i+1}^{e,l}} = d_{0,2} + d_{2,7} + d_{7,4} = V^{e,l}$$

3.6 Eliminacja stanów nieperspektywicznych

Problem komiwojażera, rozwiązywany metodą programowania wieloetapowego stanowi problem klasy O(n!). Można jednak dokonać próby skutecznego ograniczenia liczby generowanych stanów w procesie decyzyjnym, stosując następujące dwie heurystyki:

- **metodę generowania trajektoriami** – która sprowadza się do wyznaczenia pewnego stanu końcowego $P_1^{E,l}$ w dosyć specyficzny sposób. Należy mianowicie uzupełniać wektor stanu na każdym etapie procesu decyzyjnego o kolejny element (miasto) oraz na każdym etapie generować tylko jeden stan. Po otrzymaniu pierwszego stanu końcowego, należy obliczyć jego wartość i porównać z wartością innego stanu końcowego $P_2^{E,l}$, wygenerowanego wg odmiennej trajektorii. Jeżeli wartości obu stanów są różne, wówczas perspektywiczne okażą się stany pochodne trajektorii, dla której wartość ostatniego stanu była mniejsza. Pod pojęciem **stanu perspektywicznego** należy rozumieć taki, który rokuje w dalszej perspektywie procesu decyzyjnego otrzymanie rozwiązania optymalnego lub jemu bliskiego.



Rysunek 2 Idea generowania trajektoriami i eliminacji stanów nieperspektywicznych

- **regułę dominacji**, według której na każdym z kolejno generowanych etapów porównuje się wartości stanów, z których tylko ten o najmniejszej wartości stanowi podstawę dalszego postępowania decyzyjnego. Odpowiada to sytuacji, w której marszruta komiwojażera zawiera miasta najbliżej od siebie położone (tzw. heurystyka najbliższego sąsiada).

3.7 Rozszerzenie modelu

Zaproponowany powyżej model matematyczny można uzupełnić o dodatkowe aspekty związane z:

- występowaniem wiaduktów o znanej wysokości na określonych odcinkach dróg,
- występowaniem mostów o znanym dopuszczalnym udźwigu,
- istnieniem ograniczeń prędkości na wybranych trasach.

W związku z powyższym, definiuje się ponadto:

- macierz wiaduktów:

$$W = \left[w_{i,j} \right]_{i,j=1,\dots,N}$$

gdzie:

$$w_{i,j} = \begin{cases} 1 \rightarrow \text{na drodze i do j jest wiadukt} \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

lub też:

$$w_{i,j} = \begin{cases} h \rightarrow \text{na drodze i do j jest wiadukt o wys. h} \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

- macierz mostów:

$$M = \left[m_{i,j} \right]_{i,j=1,\dots,N}$$

gdzie:

$$m_{i,j} = \begin{cases} 1 \rightarrow \text{na drodze i do j jest most} \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

lub też:

$$m_{i,j} = \begin{cases} q \rightarrow \text{ most na drodze i do j ma dop. udźwig q} \\ 0 \rightarrow \text{w przeciwnym wypadku} \end{cases}$$

- całkowita wysokość pojazdu komiwojażera h_0 oraz ciężar c_0

Wobec powyższego procedura generowania stanu przyjmuje postać:

$$\underset{n=1,..,N}{\forall} \left(p_n^{e-1,\lambda} = 0 \right) \wedge \left(w_{N^{e-1,\lambda},n} = 0 \right) \wedge \left(m_{N^{e-1,\lambda},n} = 0 \right) \Rightarrow P^{e,l} = P^{e-1,\lambda} + \Delta p$$

lub:

$$\underset{n=1,..,N}{\forall} \left[\left(p_n^{e-1,\lambda} = 0 \right) \wedge \left(w_{N^{e-1,\lambda},n} \geq h_0 \right) \wedge \left(m_{N^{e-1,\lambda},n} \geq q_0 \right) \right] \Rightarrow P^{e,l} = P^{e-1,\lambda} + \Delta p$$

gdzie:

$N^{e-1,\lambda}$ jest numerem miasta odwiedzonego jako ostatnie na etapie $(e-1,\lambda)$

4 Propozycje rozwiązań zagadnienia i realizacji zajęć

4.1 Programowanie bez komputera

Przygotowanie zajęć

Nauczyciel przedstawia zagadnienie w postaci dylematu świętego Mikołaja, którego zadaniem jest jak najsprawniejsze dostarczenie prezentów dzieciom pod choinkę. Obszar działania świętego Mikołaja można dowolnie ograniczyć np. do określonych miast danego regionu lub Polski. Warunkiem przeprowadzenia zajęć jest posiadanie odpowiednich map w postaci kserokopii formatu A3, z zaznaczonymi wyraźnie lokalizacjami (miasta, wioski, ewentualnie domy). Niezbędne okażą się również zwykłe tablice korkowe, kolorowe szpilki do tablic oraz cienkie nici lub sznurki. Zbiór miast lub lokalizacji, które winien odwiedzić święty Mikołaj można wskazać poprzez wbicie szpilki na mapie, a zadaniem uczniów jest zaznaczanie za pomocą nici lub sznurka, trasy jaką powinien przebyć, by dotrzeć z prezentami do wszystkich (obowiązuje zasada odwiedzania tylko raz jednej lokalizacji). Nauczyciel, zależnie od rozwoju sytuacji dydaktycznej, może nieco urozmaicić zadanie – wprowadzając np. konieczność powrotu świętego Mikołaja do punktu z którego wyruszył w trasę.

Przebieg zajęć

- Nauczyciel dzieli uczniów na kilka zespołów lub pozwala na swobodny dobór grup o odpowiedniej liczbie osób. Każdy zespół otrzymuje tablicę korkową, mapę, zestaw kolorowych szpilek, sznurek oraz linijkę lub przymiar krawiecki.
- Wspólnie z nauczycielem, wszystkie grupy zaznaczają ten sam zbiór miast do odwiedzenia, wbijając szpilki w odpowiednie lokalizacje na tablicach korkowych z umieszczonymi nań mapami. Również wspólnie ustalane jest miasto startowe, od którego rozpoczyna się zadanie.
- Uczniowie, w ramach każdej z grup, podejmują wspólnie decyzje dotyczące kolejności odwiedzanych miast, zaznaczając trasę sznurkiem wokół szpilek na tablicy.
- Uczniowie bardzo szybko orientują się, że w miarę podejmowania kolejnych decyzji muszą zużyć coraz więcej sznurka by opisać trasę podróżnika. Zdają sobie również sprawę z faktu ścisłej zależności pomiędzy całkowitą długością trasy a kolejnością odwiedzanych miast.
- Gdy każda z grup uzna, że zadanie zostało wykonane, zdejmuje sznurek z tablicy i mierzy jego długość.

Ocena rozwiązań i formułowanie wniosków

Po zakończeniu zasadniczej części zajęć, nauczyciel proponuje przeprowadzenie oceny uzyskanych rozwiązań przez poszczególne grupy. Zależnie od etapu edukacyjnego, na którym przeprowadzono zajęcia, można dodatkowo zachęcić uczniów by dokonali próby werbalizacji przyjętej metody rozwiązania zadania. Uczniom bardziej ambitnym można pozostawić sformułowanie rozwiązania w postaci listy kroków (czyli algorytmu).

Scenariusz alternatywny

Scenariusz alternatywny może przewidywać wykorzystanie tablicy interaktywnej, wyposażonej w szereg udogodnień w postaci gotowych map (np. Polski lub określonego województwa). Korzystając z darmowych narzędzi typu *Google Maps* lub *Google Earth*, omawiane zagadnienie można przenieść na dowolny obszar geograficzny, włączając w to również obszar nieba i gwiazd. Dokonywanie pomiarów może się wówczas odbyć gotowymi narzędziami programowymi, w które są wyposażone wspomniane aplikacje.

4.2 Prosty model symulacyjny w środowisku MsExcel

Przedstawiony punkcie 3. model problemu komiwojażera korzysta ze struktur danych, które po przeniesieniu do arkusza kalkulacyjnego np. MsExcel lub analogicznego, dają podstawę utworzenia interesującej przestrzeni symulacyjnej. Bez konieczności stosowania języka VBA lub innego (np. Open Office Basic, Libre Office Basic) uczniowie (studenci) mają możliwość zbudowania arkusza, w którym rozwiązanie problemu będzie odbywać się według **algorytmu zachłannego**, zaproponowanego w pkt. 3.6 jako reguła dominacji. Istotne jest by nauczyciel przedstawił zadanie jako **grę**, którą uczniowie samodzielnie mogą utworzyć, a następnie w nią zagrać. Ramowy scenariusz zajęć mógłby przedstawiać się następująco:

- Nauczyciel przedstawia zagadnienie, wyjaśniając istotę problemu i obowiązujące ograniczenia.
- Uczniowie tworzą arkusz, rozpoczynając od przyjęcia liczby miast (np. 10 lub 15), a następnie ograniczają zakres komórek z przeznaczeniem na macierz odległości pomiędzy nimi. Tworzona macierz jest kwadratowa, z wyłączeniem przekątnej głównej. Dosyć żmudne prowadzanie danych numerycznych do macierzy można przyspieszyć, wykorzystując wbudowaną funkcję *LOS* lub *LOS.ZAKR* (MsExcel).
- Następnie, w tym lub nowym arkuszu, należy w analogiczny sposób utworzyć macierze wiaduktów oraz mostów, wprowadzając odpowiednie wartości liczbowe. Należy jednak pamiętać, by podawane wielkości nie ograniczyły możliwości prowadzenia procesu decyzyjnego w dalszej części zajęć.

- Odrębny zakres komórek (np. jedną z kolumn) należy przeznaczyć na wektor stanu, w którym grający będzie mieć możliwość decydowania o kolejności odwiedzanych miast. Obok, za pomocą wbudowanej funkcji *INDEKS* należy zadbać o prawidłowe pobieranie wartości odległości pomiędzy miastami oraz wysokości wiaduktów i udźwigu mostów.
- Sumowanie odległości da odpowiedź o całkowitej trasie, a zastosowanie odpowiedniego formatowania warunkowego winno zapewnić wizualną kontrolę poprawności trasy z punktu widzenia istniejących ograniczeń.

	A	B	C	D	E	F	G	H	I	J	K
1		1	2	3	4	5	6	7	8	9	10
2	1		6	12	2	68	43	30	20	11	4
3	2	6		5	8	12	45	2	25	44	6
4	3	12	5		40	1	52	22	20	4	8
5	4	1	8	43		30	8	25	13	27	10
6	5	69	11	1	30		10	8	34	16	14
7	6	44	35	53	8	10		3	22	8	18
8	7	26	2	26	25	7	2		27	40	22
9	8	22	25	16	12	33	26	30		4	26
10	9	10	43	3	27	17	6	36	2		30
11	10	5	6	8	11	15	17	22	26	31	
12				Macierz odległości							
13											
14											
15											

Macierz odległości | Macierz Mostów | Macierz wiaduktów

Rysunek 3. Fragment arkusza danych wejściowych

M	N	O	P	Q
Hmax	4		Umax	30
Nr miasta	Odległość	Mosty	Wiadukty	
5				
1	69	0	7,2	
4	2	0	0	
6	8	40	0	
3	53	0	3	
9	4	0	3,8	
7	36	0	0	
8	27	0	0	
2	25	15	0	
10	6	0	0	
5	15	0	0	
	245			

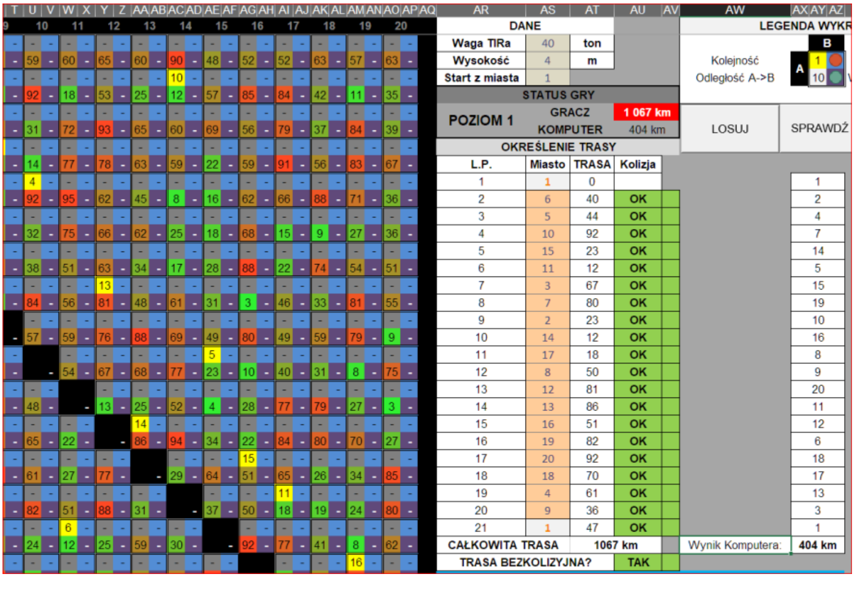
Rysunek 4. Fragment arkusza symulacji zagadnienia komiwojażera

Problem komiwojażera stanowi również interesującą propozycję w zakresie realizacji zajęć według jeszcze innego scenariusza, w którym uczniowie tworzą swoje aplikacje w wybranym języku np. C++.

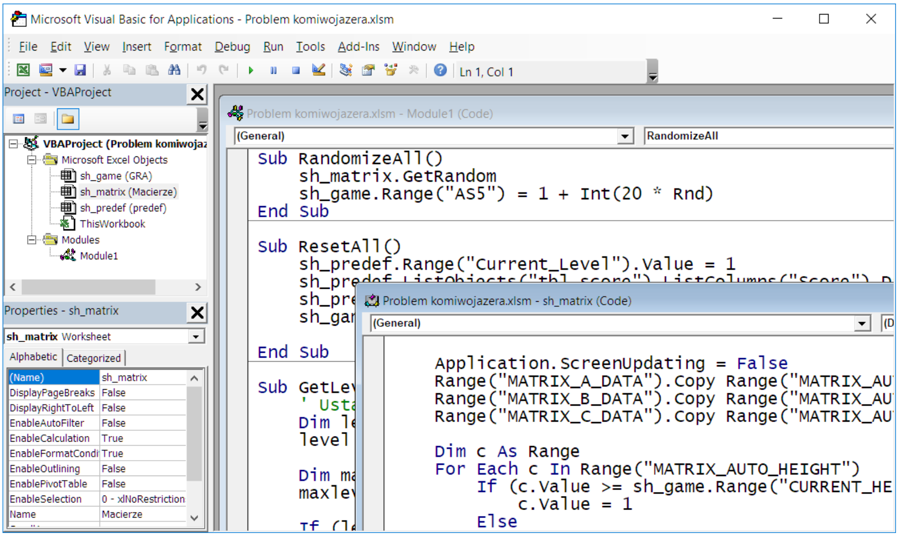
4.3 Gra dydaktyczna z użyciem VBA w środowisku MsExcel

Nieco trudniejszym zadaniem, którym można już obarczyć zdolniejszych uczniów liceów o profilu mat.-inf. lub studentów wybranych kierunków inżynierskich, jest zbudowanie wielopoziomowej gry dydaktycznej o kilku stopniach trudności. Przedsięwzięcie takie jest dobrą okazją do stworzenia przestrzeni programowania zespołowego dla uczniów zapoznanych z programowaniem w języku VBA w środowisku MsExcel.

Przeprowadzony przez Autorów w 2017 roku eksperyment dydaktyczny wykazał, że w oparciu o przedstawiony wcześniej model, wykorzystujący ideę programowania wieloetapowego, uczniowie potrafili utworzyć, przy wsparciu nauczyciela prowadzącego, aplikację realizującą tak sformułowane zadanie, Ponadto, w trakcie prac nad aplikacją zauważono cały szereg aktywności opisywanych w literaturze jako charakterystycznych dla rozwijania myślenia komputacyjnego. Stworzona gra umożliwia prowadzenie rozgrywek grupowych na wspólnym zestawie danych wejściowych, bieżącą weryfikację rozwiązania z punktu widzenia przyjętych ograniczeń oraz weryfikację rozwiązania końcowego względem zaimplementowanego algorytmu wieloetapowego, co znakomicie wzmacnia autorefleksję grającego.



Rysunek 5. Fragment interfejsu gry w środowisku MsExcel



Rysunek 6. Fragment środowiska programistycznego VBA MsExcel

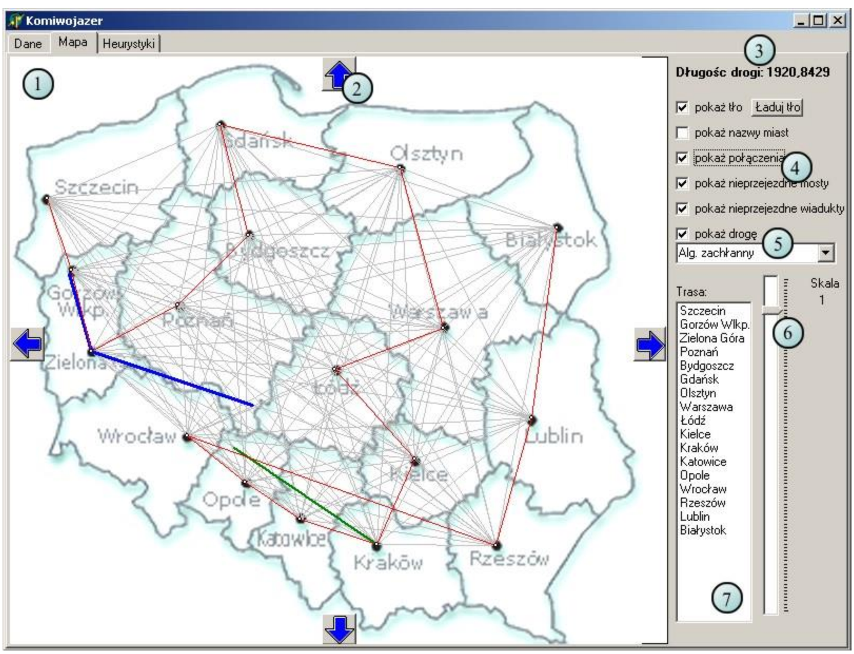
Projekt ten został poddany praktycznej weryfikacji podczas XVIII Beskidzkiego Festiwalu Nauki i Sztuki, i wykorzystany jako element warsztatów informatycznych przeznaczonych dla uczniów szkół ponadpodstawowych regionu Podbeskidzia.

4.4 Implementacja informatyczna pozostałych, wybranych rozwiązań

Problem komiwojażera doczekał się szeregu wartościowych opracowań naukowych, prezentujących możliwy do wykorzystania wachlarz algorytmów rozwiązujących to zagadnienie. W ramach kolejnego autorskiego eksperymentu dydaktycznego, przeprowadzonego tym razem wśród studentów kierunku „Informatyka" z użyciem środowiska programistycznego Embarcadero Delphi, zaprojektowana i wykonana została aplikacja umożliwiająca dokonywanie szczegółowych analiz złożoności obliczeniowej i czasowej następujących algorytmów:

- dokładnego – umożliwiającego przeszukiwanie całej dostępnej przestrzeni rozwiązań, dającego gwarancję uzyskania rozwiązania globalnie optymalnego, wymagającego jednak bardzo długiego czasu obliczeniowego (nadającego się jedynie do rozwiązywania zadań ograniczonych do dziesięciu miast),
- zachłannego – jako przykładu heurystyki tradycyjnej, dającej dość dobre rozwiązania w bardzo krótkim czasie,

- genetycznego – jako przykładu metaheurystyki, charakteryzującej się dużą złożonością obliczeniową, która z uwagi na specyfikę zadania, nie przełożyła się jednak na jakość uzyskiwanych rozwiązań,
- symulowanego wyżarzania – jako odmiany algorytmu ewolucyjnego, osiągającego – jak się okazało – znacznie lepsze wyniki od algorytmu genetycznego (i w dużo krótszym czasie),
- losowego – nie będącego co prawda metodą optymalizacyjną, lecz generującego rozwiązania losowe, stanowiąc swoistą przeciwwagę dla algorytmu dokładnego i pomagając tym samym obiektywnie ocenić skuteczność pozostałych metod,
- użytkownika – jako rozwiązania samodzielnego, odzwierciedlającego absolutnie indywidualny, subiektywny proces decyzyjny.



Rysunek 7. Fragment interfejsu użytkownika aplikacji „Komiwojażer"

5 Podsumowanie

Zadanie komiwojażera to klasyczny problem optymalizacji kombinatorycznej. Ma on bogatą literaturę oraz liczne zastosowania. Wiele realnych zadań optymalizacyjnych sprowadza się do niego. Przykładem może być opracowanie trasy jazdy autobusów, określenie najkrótszej drogi, po której będzie poruszało się ramię robota, instalującego nity na skrzydle samolotu, wyznaczenie drogi przejazdu w systemie nawigacji samochodowej lub zaprojektowanie optymalnej marszruty środków transportu wewnątrzmagazynowego itd. [2] Jest klasą abstrakcji dla szeregu zagadnień praktycznych.

Jest zatem znakomitym polem do rozwijania i kształtowania **myślenia komputacyjnego**. Wymaga bowiem umiejętności modelowania zagadnień, dekompozycji złożonych problemów, poszukiwania rozwiązań heurystycznych, podejmowania działań symulacyjnych, logicznej organizacji danych, algorytmicznego podejścia do poszukiwania rozwiązań, analizy i oceny efektywności algorytmów i jakości otrzymywanych wyników oraz jak najlepszego wykorzystania możliwości i zasobów komputera.

Literatura

1. Bucki R., Marecki F., *Digital Simulation of Discreete Processes,* Network Integrators Associates, 2006.
2. Christofides N., Mingozzi A., Toth P., Scandi C.: *Combinatorial Optimization*, Wiley, New York 1979.
3. Nahorski Z., Chudy M., Straszak A.: *Optymalizacja: Zadania, metody, algorytmy*, PTBOiS, Warszawa 1991.
4. Sysło M.M., Deo N., Kowalik J.S.: *Algorytmy optymalizacji dyskretnej*, PWN, Warszawa 1999.