

doi:10.15199/48.2024.10.36

## Zastosowanie biblioteki SFML do modelowania ruchu ulicznego na skrzyżowaniu z sygnalizacją świetlną

**Streszczenie.** W artykule opisano cyfrowy symulator ruchu ulicznego na skrzyżowaniu o ruchu okrężnym umożliwiający sterowanie parametrami funkcjonowania sygnalizacji świetlnej i analizę wpływu tych parametrów na przepustowość skrzyżowania. Odzworowuje on rzeczywiste skrzyżowanie z uwzględnieniem zmierzonego natężenia ruchu. Wykonano go w języku programowania C++ z użyciem biblioteki SFML. Przeprowadzone testy potwierdzają jego dużą dokładność i możliwość praktycznego wykorzystania.

**Abstract.** The article describes a digital traffic simulator at a circular intersection that allows controlling the parameters of traffic lights and analyzing the impact of these parameters on the throughput of the intersection. It reproduces the real intersection taking into account the measured traffic intensity. It was made in the C++ programming language using the SFML library. The tests carried out confirm its high accuracy and possibility of practical use. (Using the SFML library to model traffic at an intersection with traffic lights).

**Słowa kluczowe:** modelowanie ruchu ulicznego, symulator skrzyżowania, sygnalizacja świetlna, biblioteka SFML.

**Keywords:** traffic modeling, intersection simulator, traffic lights, SFML library.

### Wstęp

W ostatnich latach w całej Unii Europejskiej obserwowany jest wzrost liczby samochodów osobowych. Jak podaje Europejski Urząd Statystyczny Eurostat [1] w 2022 r. na 1000 mieszkańców przypadało średnio 560 pojazdów, przy czym w Polsce aż 584 (wobec 492 dekadę wcześniej). Towarzyszy temu tworzenie się korków, a polskie miasta w rankingach najbardziej zakorkowanych znajdują się na wysokich pozycjach. W zestawieniu TomTom traffic index [2] Wrocław znalazł się na 21. miejscu, a Łódź na 39. (w skali światowej).

W rankingu CBS News [3] Łódź była aż na 13. pozycji, Kraków na 20., a Wrocław na 22.

Wśród współczesnych sposobów radzenia sobie z zatorami ulicznymi jest stosowanie sygnalizacji świetlnej: stałoczasowych (o jednym lub kilku programach) i akomodacyjnych (dostosowujących czasy nadawania poszczególnych sygnałów do bieżącego natężenia ruchu). Te drugie mogą być częścią tzw. inteligentnych systemów sterowania ruchem (ITS, ang. Intelligent transportation system) [4, 5], które mogą sterować działaniem wielu sygnalizacji na określonym obszarze.

W Polsce sposób wykonania dróg, ich oznakowanie znakami poziomymi i pionowymi, a także funkcjonowanie sygnalizacji świetlnej musi być zgodne z ustawą Prawo o ruchu drogowym [6] oraz wynikającymi z niej rozporządzeniami.

W szczególności są to rozporządzenia: w sprawie znaków i sygnałów drogowych [7] i w sprawie szczegółowych warunków technicznych dla znaków i sygnałów drogowych oraz urządzeń bezpieczeństwa ruchu drogowego i warunków ich umieszczania na drogach [8]. Funkcjonujące algorytmy sterowania sygnalizacją świetlną muszą zatem uwzględniać określone w powyższych dokumentach m.in. minimalne i maksymalne czasy nadawania poszczególnych sygnałów, czasy międzyczasowe (między zakończeniem i rozpoczęciem sygnałów zielonych dla dwóch wzajemnie kolizyjnych kierunków ruchu) oraz szybkości poruszania się poszczególnych grup uczestników ruchu.

Algorytm sterujący pracą sygnalizacji świetlnej powinien tak ustawiać kolejność i czasy nadawania sygnałów zielonych (pionowych – w przypadku kierujących pojazdami wykonującymi odpłatny przewóz osób na regularnych liniach) dla poszczególnych grup użytkowników, by zapewnić możliwie dużą przepustowość skrzyżowania przy jednoczesnym minimalnym czasie oczekiwania na podanie sygnału zielonego.

Dotychczas zaproponowano bardzo wiele algorytmów sterowania pracą sygnalizacji świetlnej. W [9] przedstawiono wydajny algorytm, który analizuje konkurencyjne potoki ruchu na skrzyżowaniach dróg. Ma na celu zmniejszenie całkowitego zużycia paliwa przez pojazdy, a co za tym idzie redukcję wytwarzanych przez nie gazów. Bierze pod uwagę topologię skrzyżowania, kolizyjne strumienie ruchu i ich charakterystyki w czasie rzeczywistym. Algorytm inteligentnego sterowania pojedynczą sygnalizacją świetlną (ITLC) przedstawiono w [10]. Również uwzględnia on charakterystykę ruchu w czasie rzeczywistym dla każdego strumienia ruchu, mając na celu zwiększenie płynności ruchu poprzez skrócenie czasu oczekiwania na sygnał zielony i zwiększenie liczby pojazdów przejeżdżających przez skrzyżowanie w ciągu sekundy. Z przeprowadzonych przez autorów eksperymentów wynika, że zwiększa płynność ruchu o 25% w porównaniu z poprzednimi programami rozwiązanymi przy użyciu sygnalizacji świetlnej. Obecnie bardzo wiele rozwiązań wykorzystuje algorytmy uczenia maszynowego (ML) i głębokiego uczenia (DL) do przewidywania natężenia ruchu [11, 12]. Bezpośrednie wykorzystanie DL do sterowania działaniem sygnalizacji opisano w [13–15], przy czym w ostatniej z tych prac rozważono także celowość stosowania tzw. zielonej fali, czyli jednoczesnego wyświetlenia sygnału zielonego na kolejnych skrzyżowaniach, by pojazdy nie musiały przed nimi hamować.

W tej pracy przedstawiono cyfrowy symulator działania sygnalizacji świetlnej napisany w języku programowania C++ z wykorzystaniem biblioteki SFML. Pozwala on na testowanie różnych wariantów funkcjonowania sygnalizacji na wybranym, rzeczywistym skrzyżowaniu, na którym regularnie tworzą się zatory uniemożliwiające płynny ruch pojazdów. Wybrano poznańskie Rondo Rataje, które znajduje się u zbiegu głównych arterii komunikacyjnych miasta i po którym kursują tramwaje oraz autobusy komunikacji miejskiej (po bus-pasie wokół ronda i przez jego środek). Opisano także przeprowadzone eksperymenty i wynikające z nich praktyczne wnioski.

### Modelowanie ruchu pojazdów na skrzyżowaniu

Ruch pojazdów można modelować w ujęciu makroskopowym lub mikroskopowym [16]. W ujęciu mikroskopowym każdy pojazd jest postrzegany jako obiekt poruszający się po drodze i dostosowujący swoją prędkość do zachowania innych pojazdów z przodu. Dynamiką regulują różniczkowe dla każdego pojazdu w zależności od odległości pojazdu poprzedzającego i dlatego nazywane są mode-

lami follow-the-leader. Z drugiej strony podejście makroskopowe traktuje ruch jako gęstość w danym miejscu drogi. Zaproponowano także modele hybrydowe, odtwarzające kolejną zjawisk propagacyjnych i zachowań kierowców (także

z uwzględnieniem obecności pojazdów autonomicznych) [17]. Na podstawie tych modeli powstało wiele rozwiązań programistycznych do symulowania ruchu ulicznego, jak np. [18], gdzie w środowisku programistycznym Python zaimplementowano model mikroskopowy.

W tej pracy wykorzystano mikroskopowy model IDM (ang. Intelligent driver model) – ciągły w czasie model podążania za samochodem, służący do symulacji ruchu na autostradach i w miastach [19]. Opiera się on na ustalaniu prędkości każdego pojazdu. Przyspieszenie całkowite jest wyliczane na podstawie aktualnej odległości pojazdu względem obiektu odniesienia. Obiektem tym może być pojazd poprzedzający lub, w szczególnym przypadku, sygnalizator świetlny na skrzyżowaniu.

Do implementacji tego modelu użyto język programowania C++ oraz darmową, wieloplatformową i wielojęzyczną bibliotekę programistyczną SFML (Simple and Fast Multimedia Library) [20]. Jest ona zorientowana obiektowo, umożliwia tworzenie gier komputerowych i aplikacji multimedialnych. Składa się z pięciu modułów: systemowego, okienkowego, graficznego, audio i sieciowego. Zintegrowano ją z narzędziem Qt Creator pozwalającym w prosty sposób zaprojektować interfejs graficzny. Dzięki temu było możliwe łatwe zarządzanie wieloma obiektami graficznymi (pojazdami) przy niskim zapotrzebowaniu na zasoby komputera.

### Parametry ruchu na analizowanym skrzyżowaniu

Znajdujące się we wschodniej części Poznania Rondo Rataje zostało przebudowane w latach 2020–2022 kosztem 140 mln zł [21]. Niestety powszechnie krytykowana jest nieczytelna organizacja ruchu [22], jak i tworzące się tam korki, w których tkwią samochody oraz pojazdy komunikacji miejskiej [23].

Do wykonania prezentowanego symulatora przyjęto stan z dnia 24.10.2023 r. Wówczas przez Rondo Rataje kursowało 6 linii tramwajowych z częstotliwością 10-minutową oraz 19 linii autobusowych z różną częstotliwością – część po torowiskach tramwajowych wewnątrz tarczy ronda i korzystających ze wspólnych, podwójnych przystanków poza obrębem ronda. Samo rondo tworzą 4 pasy ruchu, z czego wewnętrzny jest dostępny jedynie dla autobusów miejskich (bus-pas).

W celu zliczenia samochodów przejeżdżających przez rondo posłużono się transmisją prowadzoną przez lokalną

telewizję WTK w godzinach największego szczytu komunikacyjnego (14:30–17:00). Osobno liczone pojazdy jednośladowe, samochody osobowe oraz ciężarowe. Uwzględniano kierunki, z których nadjechały (w tym pasy ruchu) oraz docelowe. Przykładowe dane dla kierunku od ul. B. Krzywoustego zawarto w tabeli 1.

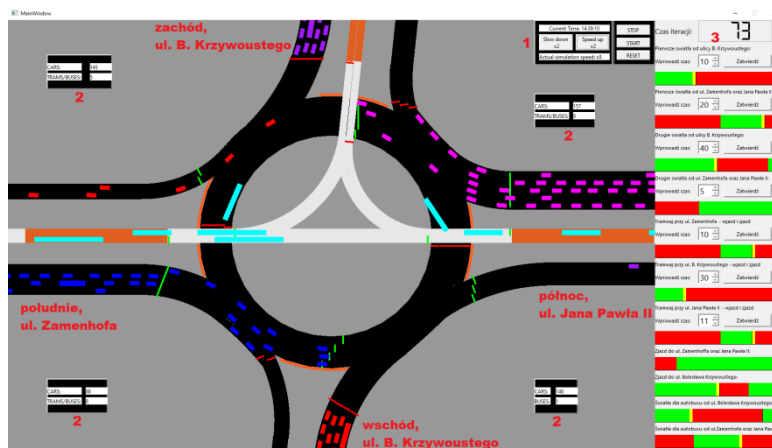
Tabela 1. Natężenie ruchu na rondzie – wjazd od ul. B. Krzywoustego

Kierunek docelowy	Typ pojazdu	Liczba pojazdów	Prawdopodobieństwo jazdy w danym kierunku [%]
Prawo	ciężarowe	6	1,0
	osobowe	578	97,1
	jednośladowe	11	1,8
Prosto (pas prawy)	ciężarowe	38	3,8
	osobowe	953	95,3
	jednośladowe	9	0,9
Prosto (pas lewy)	ciężarowe	6	0,5
	osobowe	1088	99,3
	jednośladowe	2	0,2
Lewo	ciężarowe	15	2,7
	osobowe	540	95,6
	jednośladowe	10	1,8

Liczby i czasy przyjazdu poszczególnych pojazdów komunikacji miejskiej pobrano z ich rozkładów jazdy, a czasy postojów na przystankach są losowane ze zmierzonego zakresu 14,3–28,4s. Na podstawie zdjęć satelitarnych skrzyżowania wyliczono średnią odległość pomiędzy stojącymi pojazdami, a średnie prędkości poszczególnych grup pojazdów wyliczono na podstawie ww. transmisji telewizyjnej. Średnie przyspieszenia oraz opóźnienia autobusów i tramwajów oszacowano na podstawie odczytów z ich prędkościomierzy i pomiarów stoperem, natomiast pozostałych pojazdów – na podstawie danych z literatury, korygując je na podstawie własnych obserwacji. Zebrano je w tabeli 2., gdzie „a” oznacza przyspieszenie, „b” – opóźnienie, „T” – minimalny czas dojazdu do pozycji poprzedzającego pojazdu, „S<sub>min</sub>” – minimalną odległość pojazdu do obiektu odniesienia, a „V<sub>max</sub>” – maksymalną prędkość pojazdu.

Tabela 2. Parametry ruchu pojazdów przyjęte do symulacji

Parametr	Typ pojazdu				
	Osobowy	Ciężar.	Jednośl.	Tramwaj	Autobus
a [m/s <sup>2</sup> ]	1,70	1,70	1,70	0,72	0,95
b [m/s <sup>2</sup> ]	2,80	2,80	2,80	0,92	0,86
T [s]	1,40	1,40	1,40	2,50	2,50
S <sub>min</sub> [m]	(1,7; 3,7)	(1,7; 3,7)	(1,7; 3,7)	(0,8; 1,5)	(0,8; 1,5)
V <sub>max</sub> [m/s]	(9,2; 10,5)	(7,2; 8,9)	(8,8; 12,2)	5,56	5,56



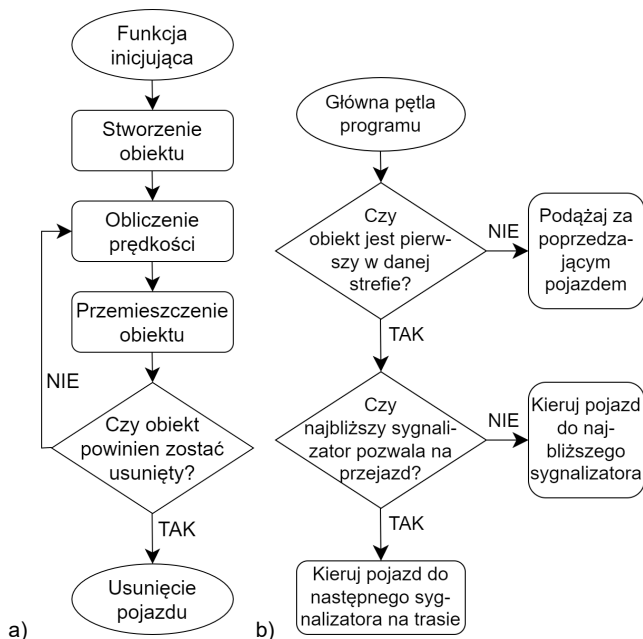
Rys. 1. Widok uruchomionego symulatora: 1 – panel sterujący, 2 – liczniki pojazdów oczekujących na wjazd z poszczególnych kierunków, 3 – menu programu sygnalizacji świetlnej

Wymiary ronda odczytano korzystając z map zamieszczonych w serwisie internetowym [geoportal.gov.pl](http://geoportal.gov.pl) i przyjęto, że rondo jest idealnym kołem, co jest wystarczającym przybliżeniem.

### Opracowany symulator

Symulator umożliwia zbadanie przepustowości Ronda Rataje w zależności od ustawionego programu działania sygnalizacji świetlnej generując 60 klatek obrazu na sekundę. Uwzględnia jedynie pojazdy, a ruch pieszy i rowerowy pomija jako niekolidujący z dopuszczalnymi parametrami pracy sygnalizacji. Ma postać aplikacji graficznej, która może być uruchomiona na dowolnym komputerze z systemem MS Windows bez konieczności instalacji dodatkowego oprogramowania. Konieczny jest ekran o rozdzielczości co najmniej 1920×1080 pikseli. Wówczas 1 metr w terenie odpowiada 5 pikselom. Wygląd symulatora przedstawiono na rysunku 1.

Wszystkie pojazdy poruszają się po zdefiniowanych na stałe ścieżkach (dla samochodów nie uwzględniono zawracania). W sytuacjach przecinania się ścieżek pojazdy oczekują, aby nie najeżdżały na siebie. Tramwaje i autobusy są generowane zgodnie z rzeczywistym rozkładem jazdy, a pozostałe pojazdy – w sposób losowy zgodny z opisanymi wcześniej pomiarami. Każdy generowany pojazd jest zliczany, a ich liczba dla każdego kierunku jest wyświetlana na licznikach (nr 2 na rysunku 1.). Osobno są wyświetlane tramwaje wraz z autobusami oraz pozostałe pojazdy. Parametry ruchu każdego pojazdu są obliczane zgodnie z modelem IDM dla powyżej opisanych parametrów. Po przejechaniu skrzyżowania pojazdy zostają usunięte. Algorytm głównej pętli obsługującej wyświetlanie pojazdów przedstawiono na rysunku 2a.



Rys. 2. Program symulatora: a) główna pętla obsługująca wyświetlanie pojazdów, b) algorytm sterujący ruchem pojazdów

Przy wyznaczaniu kolejnych pozycji pojazdów (tramwaje są wizualizowane jako podwójne, dwuczłonowe pojazdy) określa się, czy mają one podążać za poprzednim pojazdem, czy do najbliższego sygnalizatora, jeśli ten jest bliżej. Dlatego od strony logicznej rondo jest podzielone na 4 strefy, a ruch pojazdów odbywa się w nich zgodnie z algorytmem pokazanym na rysunku 2b.

Panel sterujący użytkownika (nr 1 na rysunku 1.) umożliwia uruchomienie symulacji, jej zatrzymanie i zresetowanie

po zmianie ustawionego programu sygnalizacji. Pozwala także na przyspieszenie przebiegu symulacji nawet 8-krotnie. Z kolei menu programu sygnalizacji (nr 3 na rysunku 1.) umożliwia w przejrzysty sposób wprowadzenie czasów nadawania sygnałów zielonych dla poszczególnych grup sygnalizatorów. Uwzględniono rzeczywiste ograniczenia czasów trwania poszczególnych sygnałów wynikające z przepisów oraz wytycznych stosowanych przez Zarząd Dróg Miejskich w Poznaniu. Nieistotnym, z punktu widzenia działania symulatora, jest także uproszczenie polegające na braku rozdzielania na kierunkowe sygnalizatorów dla pojazdów komunikacji miejskiej na wewnętrznym bus-pasie.

### Eksperyment

W ramach testów sprawdzono 4 różne programy sygnalizacji w opisanych warunkach popołudniowego szczytu komunikacyjnego.

Pierwszym z nich był program rzeczywiście funkcjonujący na tym skrzyżowaniu. Stwierdzono, że symulator z bardzo dużą dokładnością odwzorowywał miejsca tworzenia się zatorów. Wykazano, że sygnalizatory zjazdowe w ul. L. Zamenhofska utrudniają przejazd na wprost samochodom jadącym na wschód ul. B. Krzywoustego. Kolejnym źródłem zatorów były autobusy, które, pomimo znajdowania się na bus-pasie, blokowały przejazd samochodom swoimi tyłami. Również pojazdy komunikacji miejskiej nie mogąc opuścić tarczy ronda powodowały chwilowe własne zatory. Przyczyną tego zjawiska były niewystarczające czasy na przejazd przez rondo uniemożliwiające pokonanie go w jednym cyklu, o co wielokrotnie postulowali motorniczowie tramwajów. Łącznie po godzinie obserwacji deficyt przepustowości skrzyżowania wynosił przeszło 1100 samochodów (najwięcej od ul. Jana Pawła II) i 4 pojazdów komunikacji miejskiej.

W programie 2. próbowano wydłużyć cykl zielonego światła od zachodniej części ul. B. Krzywoustego, by zniwelować tworzące się tam zatory kosztem skrócenia go w dalszej części ul. Krzywoustego i w relacji północ-południe. Nie spowodowało to oczekiwanej poprawy, pojawił się problem ze zjazdem autobusów z wewnętrznego bus-pasa i tworzeniem większych zatorów w relacji północ-południe (także dla tramwajów). Tym razem deficyt wzrósł do 1332 samochodów i aż 41 pojazdów komunikacji miejskiej.

Kompromis pomiędzy założeniami programu 1. i 2. spróbowano zawrzeć w programie nr 3, w którym nieco ułatwiono przejazd w relacji północ-południe. Deficyt przepustowości był jednak analogiczny jak dla programu 1. Z kolei w programie 4. spróbowano maksymalnie wydłużyć czas sygnalizatorów umożliwiających okrążanie ronda, by nie tworzyły się na nim zatory. Dzięki temu nie dochodziło do blokowania pojazdów komunikacji miejskiej wewnątrz ronda, jednak jego przepustowość dla samochodów jeszcze bardziej spadła (deficyt 1408 pojazdów).

Z przeprowadzonych symulacji (w tym także tu nieopisanych) wynika, że obecny program sygnalizacji świetlnej jest najbardziej optymalny i nie ma sensu go zmieniać. To prowadziłoby do dalszego zwiększenia zatorów, którego – dzięki przeprowadzonym symulacjom – można uniknąć. Na zmianach, takich jak dla programu 4., zyskałyby jedynie pojazdy komunikacji miejskiej kosztem samochodów.

Niezależnie sprawdzono również wydajność działania symulatora pod kątem koniecznych zasobów komputera. Na testowym laptopie wyposażonym w procesor AMD Athlon Gold 3150U (2,40 GHz) ze zintegrowaną kartą graficzną i 8 GB pamięci RAM (dostępne niecałe 6 GB) wykonywał on procesor równomiernie w 8% potrzebując jedynie 42 MB pamięci operacyjnej. Pomimo tego działał płynnie, bez żadnych zauważalnych spowolnień.

Po dłuższym czasie pracy (kilkanaście minut) stwierdzano, że sporadycznie ruch niektórych pojazdów nie był

aktualizowany. W związku z tym zdarzało się, że niektóre z nich nakładały się na siebie. Nie utrudniało to jednak obserwacji miejsc, w których tworzą się zatory. Miejsca tworzenia się zatorów odpowiadały zaś rzeczywistym miejscom na prawdziwym skrzyżowaniu, w których zbierają się pojazdy. To dowodzi wysokiego stopnia realizmu opracowanego symulatora i możliwości jego praktycznego wykorzystania do testowania różnych programów sygnalizacji świetlnej, także w warunkach innych natężeń ruchu.

## Podsumowanie

W pracy tej zaproponowano symulator rzeczywistego skrzyżowania (poznajskie Rondo Rataje), na którym codziennie tworzą się zatory drogowe. Wiernie odwzorowuje on ruch pojazdów umożliwiając symulowanie działania różnych programów sygnalizacji świetlnej.

Wykorzystana biblioteka SFML okazała się optymalna do wizualizacji ruchu bardzo wielu pojazdów na skrzyżowaniu pomimo konieczności niezależnego obliczania ich parametrów ruchu. Prostota jej użycia, płynne działanie napisanego z jej użyciem symulatora i niewielkie zasoby sprzętowe wymagane do jego uruchomienia były bardzo istotne.

Przeprowadzone eksperymenty dowiodły, że bez zmiany oznaczeń pasów ruchu w obrębie tego skrzyżowania oraz sposobu kursowania autobusów komunikacji miejskiej przez wewnętrzną tarczę ronda nie jest możliwe uzyskanie poprawy przepustowości. Ponadto symulacje potwierdziły, że obecnie działający program sygnalizacji świetlnej jest optymalny i jego zmiany nie są wskazane.

W ramach dalszych prac celowe będzie zasymulowanie różnych sposobów prowadzenia tras pojazdów komunikacji miejskiej przez to skrzyżowanie oraz różnego układu pasów ruchu, z likwidacją wewnętrznych bus-pasów włącznie.

*Badania sfinansowano z subwencji badawczej 0211/SBAD/0224.*

**Autorzy:** dr inż. Adam Konieczka, Politechnika Poznańska, Wydział Automatyki, Robotyki i Elektrotechniki, ul. Piotrowo 3A, 60-965 Poznań, E-mail: [adam.konieczka@put.poznan.pl](mailto:adam.konieczka@put.poznan.pl); inż. Hubert Antczak, Politechnika Poznańska, Wydział Automatyki, Robotyki i Elektrotechniki, ul. Piotrowo 3A, 60-965 Poznań, E-mail: [hubert.antczak@o2.pl](mailto:hubert.antczak@o2.pl); inż. Patryk Kaczmarek, Politechnika Poznańska, Wydział Automatyki, Robotyki i Elektrotechniki, ul. Piotrowo 3A, 60-965 Poznań, E-mail: [patryk.kaczmarek@op.pl](mailto:patryk.kaczmarek@op.pl); inż. Dawid Szwarc, Politechnika Poznańska, Wydział Automatyki, Robotyki i Elektro-techniki, ul. Piotrowo 3A, 60-965 Poznań, E-mail: [dawid-szwarc2001@gmail.com](mailto:dawid-szwarc2001@gmail.com).

## LITERATURA

- [1] Eurostat, Passenger cars per 1 000 inhabitants reached 560 in 2022, 17 January 2024, <https://ec.europa.eu/eurostat/web/prod-ucts-eurostat-news/w/ddn-20240117-1> (dostęp: 24.03.2024 r.)
- [2] TomTom traffic index. Ranking 2023, <https://www.tomtom.com/traffic-index/ranking/> (dostęp: 24.03.2024 r.)
- [3] Van Allen F., Cities with the worst traffic in the world, ranked, *CBS News*, 22.08.2022 r., <https://www.cbsnews.com/pictures/worst-traffic-cities-in-the-world/> (dostęp: 24.03.2024 r.)
- [4] Zarząd Dróg Miejskich w Poznaniu, Sygnalizacja świetlna, 22.08.2018 r., <https://zdm.poznan.pl/pl/sygnalizacja-swietlna> (dostęp: 24.03.2024 r.)
- [5] Perallos A., Hernandez-Jayo U. i in., *Intelligent Transport Systems: Technologies and Applications*, 2016 John Wiley & Sons, Ltd.
- [6] Ustawa z dnia 20 czerwca 1997 r. - Prawo o ruchu drogowym, Dz.U. 1997 nr 98 poz. 602 z późn. zmian., <https://isap.sejm.gov.pl/isap.nsf/DocDetails.xsp?id=wdu19970980602> (dostęp: 24.03.2024 r.)
- [7] Obwieszczenie Ministra Infrastruktury oraz Ministra Spraw Wewnętrznych i Administracji z dnia 31 października 2019 r. w sprawie ogłoszenia jednolitego tekstu rozporządzenia Ministrów Infrastruktury oraz Spraw Wewnętrznych i Administracji w sprawie znaków i sygnałów drogowych, Dz.U. 2019 poz. 2310 z późn. zmian., <https://isap.sejm.gov.pl/isap.nsf/DocDetails.xsp?id=WDU20190002310> (dostęp: 24.03.2024 r.)
- [8] Obwieszczenie Ministra Infrastruktury z dnia 9 września 2019 r. w sprawie ogłoszenia jednolitego tekstu rozporządzenia Ministra Infrastruktury w sprawie szczegółowych warunków technicznych dla znaków i sygnałów drogowych oraz urządzeń bezpieczeństwa ruchu drogowego i warunków ich umieszczania na drogach, Dz.U. 2019 poz. 2311 z późn. zmian., <https://isap.sejm.gov.pl/isap.nsf/DocDetails.xsp?id=WDU20190002311> (dostęp: 24.03.2024 r.)
- [9] Younes M.B., Boukerche A., De Rango F., SmartLight: A smart efficient traffic light scheduling algorithm for green road intersections, *Ad Hoc Networks*, Vol 140, 2023, 103061, <https://doi.org/10.1016/j.adhoc.2022.103061>.
- [10] Younes M.B., Boukerche A., An Intelligent Traffic Light scheduling algorithm through VANETs, *39th Annual IEEE Conference on Local Computer Networks Workshops*, Edmonton, AB, Canada, 2014, 637-642, doi: 10.1109/LCNW.2014.6927714.
- [11] Navarro-Espinoza A., López-Bonilla O.R., García-Guerrero i in., Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms, *Technologies*, 2022, 10, 5. <https://doi.org/10.3390/technologies10010005>
- [12] Moumen I., Abouchabaka J., Rafalia N., Adaptive traffic lights based on traffic flow prediction using machine learning models, *Int. J. of Electrical and Comp. Eng.*, Vol. 13, No. 5, 10.2023, 5813-5823, DOI: 10.11591/ijece.v13i5.pp5813-5823
- [13] Wang B., He Z., Sheng J., Chen Y. Deep Reinforcement Learning for Traffic Light Timing Optimization. *Processes*, 2022, 10, 2458, <https://doi.org/10.3390/pr10112458>
- [14] Hu X., Zhao C., Wang G., A Traffic Light Dynamic Control Algorithm with Deep Reinforcement Learning Based on GNN Prediction, *arXiv:2009.14627*, <https://doi.org/10.48550/arXiv.2009.14627>
- [15] Liu X.-Y., Zhu M., Borst S., Walid A., Deep Reinforcement Learning for Traffic Light Control in Intelligent Transportation Systems, *arXiv:2302.03669*, <https://doi.org/10.48550/arXiv.2302.03669>
- [16] Göttlich S., Schillinger T., Microscopic and Macroscopic Traffic Flow Models including Random Accidents, *arXiv:2008.11987*, <https://doi.org/10.48550/arXiv.2008.11987>
- [17] Storani, F., Di Pace, R., Bruno, F. et al. Analysis and comparison of traffic flow models: a new hybrid traffic flow model vs benchmark models, *Eur. Transp. Res. Rev.*, 13, 58 (2021). <https://doi.org/10.1186/s12544-021-00515-0>
- [18] Himite B., Simulating Traffic Flow in Python, *Medium*, 5.08.2021 r., <https://towardsdatascience.com/simulating-traffic-flow-in-python-ee1eab4dd20f> (dostęp: 24.03.2024 r.)
- [19] Treiber M., The Intelligent-Driver Model and its Variants, [https://traffic-simulation.de/info/info\\_IDM.html](https://traffic-simulation.de/info/info_IDM.html) (dostęp: 24.03.2024 r.)
- [20] Simple and Fast Multimedia Library, <https://www.sfml-dev.org/index.php> (dostęp: 10.10.2023 r.)
- [21] Poznań. Rondo Rataje oficjalnie oddano do użytku, *Transport Publiczny*, 13.07.2022 r., <https://www.transport-publiczny.pl/wiadomosci/poznan-rondo-rataje-oficjalnie-oddano-do-uzytku-74395.html> (dostęp: 24.03.2024 r.)
- [22] KaT, Wiemy, ile zdarzeń drogowych odnotowano na rondzie Rataje. Co dalej z organizacją ruchu w tym miejscu?, *epoznan.pl*, 7.03.2024 r., <https://epoznan.pl/news-news-148610-wiemy-ile-zdarzen-drogowych-odnotowano-na-rondzie-rataje-co-dalej-z-organizacja-ruchu-w-tym-miejscu> (dostęp: 24.03.2024 r.)
- [23] Korki autobusowo-tramwajowe przed rondem Rataje, *RTK*, 7.03.2023 r., <https://rtk.poznan.pl/korki-autobusowo-tramwajowe-przed-rondem-rataje/> (dostęp: 24.03.2024 r.)