

Modelowanie systemów złożonych w teorii i praktyce

Jarosław Wąs , Dariusz Pałka , Grzegorz Bazior ,
Robert Lubaś , Michał Turek , Paweł Renc ,
Marek Szlachetka , Jędrzej Byrski , Janusz Miller ,
Filip Kamiński , Adam Domagała , Andrzej Bielecki ,
Maciej Gierdziewicz , Marek Zachara , Dariusz Jamróz ,
Miroslaw Gajer , Michał Pikus 

AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Katedra Informatyki Stosowanej, Kraków

Streszczenie: Artykuł poświęcony jest modelowaniu i analizie systemów złożonych (*complex systems*) od strony teoretycznej i praktycznej. W artykule omówiono wybrane tematy badawcze poświęcone systemom złożonym, będące przedmiotem zainteresowania w Katedrze Informatyki Stosowanej WEAIIB AGH w ostatnich latach oraz przedstawiono aktualne trendy w zakresie modelowania tychże systemów. W szczególności uwagę poświęcono modelowaniu dynamiki tłumu, systemom ADAS i modelowaniu ruchu pojazdów, systemom dynamicznym, systemom złożonym w zarządzaniu, systemom biologicznym dotyczącym funkcjonowania mózgu czy wreszcie predykcji i analizie systemów złożonych za pomocą uczenia maszynowego oraz sieci głębokich.

Słowa kluczowe: systemy złożone, automaty komórkowe, uczenie maszynowe

MODELING OF COMPLEX SYSTEMS IN THEORY AND PRACTICE

Abstract: The article is devoted to the modeling and analysis of Complex Systems from the theoretical and practical side. The article discusses selected research topics, devoted to complex systems, carried out in the Department of Applied Computer Science at WEAIIB Faculty of AGH University in recent years, and presents the current trends in the modeling of these systems. In particular, attention was devoted to: crowd dynamics modeling, ADAS systems and vehicle movement modeling, dynamic systems, complex systems in management, biological systems related to the functioning of the brain, and finally the prediction and analysis of complex systems using machine learning and deep networks.

Keywords: complex systems, cellular automata, machine learning

https://doi.org/10.7494/978-83-66727-89-2_7

1. Wprowadzenie

System złożony (*complex system*) zbudowany jest z komponentów, które mogą ze sobą współdziałać. Przykładami złożonych systemów są: ludzki mózg i ogólnie organizmy żywe, pojedyncze żywe komórki, sieci energetyczne, systemy komunikacyjne i transportowe, złożone oprogramowanie i systemy elektroniczne, organizacje społeczne i gospodarcze, przedsiębiorstwa, ekosystemy czy wreszcie klimat Ziemi.

System złożony działa w wyniku zachowań poszczególnych jego elementów, a nie z powodu z góry sterowanego, określonego planu (Liu i Barabási 2016). Systemy złożone to klasa systemów, których zachowanie jest z natury trudne do modelowania ze względu na zależności, relacje lub interakcje między poszczególnymi komponentami (Wąs i in. 2020, Małecki i in. 2022, Renc i in. 2022) lub pomiędzy samym systemem a jego otoczeniem. Stephen Wolfram w swojej książce *A New Kind of Science* zasugerował, że wiodącym narzędziem do opisu systemów złożonych i generalnie opisu świata będą automaty komórkowe. Obecnie obserwujemy jednak, że do opisu stosowane są w praktyce rozmaite techniki różnych typów systemów złożonych (Wolfram 2002). W kolejnych rozdziałach autorzy skupili się na opisach różnych typów systemów złożonych, które są przedmiotem badań i analiz w Katedrze Informatyki Stosowanej na Wydziale EAIiB AGH.

2. Modelowanie dynamiki tłumu oparte na danych

Jednym ze sztandarowych przykładów systemów złożonych są scenariusze dynamiki tłumu ludzi. Zachowania i dynamika tłumu od wielu wieków fascynowały ludzi. Rozwój technik informatycznych na przełomie XX i XXI wieku sprawił, że zaczęły powstawać coraz bardziej złożone modele i symulatory dynamiki tłumu. Modele tłumu opierają się zarówno na metodzie *top-down*, gdzie definiuje się centralny system sterowania symulacją (np. paradygmatu DES – *discret-event system*), jak i metodzie *bottom-up*, gdzie wykorzystując zasadę emergencji, programuje się dynamikę i zachowania poszczególnych osób i na tej podstawie tworzy się obraz całego systemu. Wśród drugiej z wymienionych grup metod należy wyróżnić ciągłe modele oparte na dynamice molekularnej, a w szczególności słynny model Helbinga–Farkasa–Vicseka, który wyznacza kolejne położenia pieszych na zasadzie superpozycji następujących sił: dążenia do celu, odpychania od przeszkód czy interakcji z innymi pieszymi itd. Przeciwnieństwem tej metody są dyskretne modele, zarówno w czasie, jak i przestrzeni, oparte na niehomogenicznych i asynchronicznych automatach komórkowych. W takim przypadku człowiek jest reprezentowany przez stan automatu komórkowego na siatce zajętości, zaś

o dynamice ruchu decyduje funkcja przejścia, która najczęściej opiera się na dodatkowych siatkach, np. statycznych polach potencjału (odpowiedzialnych za kierowanie się do wcześniej określonych punktów docelowych), dynamicznych polach potencjału (odpowiedzialnych najczęściej za naśladowanie bądź unikanie innych osób), a także innych polach potencjalnych, takich jak *proxemic floor field*, *obstacle floor field* czy *interplay floor field* itd. Klasyczny model dynamiki pieszych oparty na automatach komórkowych odnosi się jedynie do operacyjnych i taktycznych zdolności pieszych. Dopiero zastosowanie paradygmatu ABM (*agent-based modeling*) umożliwia definiowanie strategicznych celów pieszych. Wśród prac realizowanych w tym zakresie w KIS AGH można wymienić liczne projekty: modelowanie ewakuacji za pomocą autorskiego modelu dla obiektów sportowych, np. stadionów Allianz Arena w Monachium, Wisły Kraków czy GKS Tychy, optymalizacja przepływu osób w trakcie imprez masowych czy modelowanie ruchu ludzi w symulatorze pojazdów autonomicznych Carla.

Istotnym kierunkiem obowiązującym w modelowaniu dynamiki tłumu jest symulacja sterowana danymi (*data-driven simulation*). Polega ona na tym, że przy budowaniu symulacji używa się rzeczywistych danych do tworzenia reguł systemu. Do pozyskiwania danych o rzeczywistych parametrach ruchu poszczególnych osób (takich jak prędkości, trajektorie itp.) mogą być wykorzystywane dane pochodzące z różnego rodzaju sensorów. Do najczęściej używanych sensorów stosowanych wewnątrz budynków można zaliczyć kamery wizyjne, w tym kamery stereoskopowe, kamery z oświetleniem strukturalnym i kamery mierzące czas przelotu światła (*time-of-flight cameras*). Druga ważna kategoria to czujniki wykorzystujące fale radiowe. Do tej kategorii należą m.in. czujniki bazujące na określaniu pozycji sensora względem nadajników znajdujących się w znanych lokalizacjach (*beacon*). Czujniki tego typu mogą wykorzystywać nadajniki Wi-Fi, Bluetooth oraz UWB (*ultra-wideband*). Do kategorii czujników działających z wykorzystaniem fal radiowych zalicza się również radary mikrofalowe, w tym radary dopplerowskie. Kolejna kategoria to czujniki do nawigacji inercyjnej INS (*Inertial Navigation System*), których działanie opiera się na scalonych żyroskopach, akcelerometrach i magnetometrach. Do pozyskiwania danych o parametrach ruchu w przestrzeni otwartej (na zewnątrz budynku) można dodatkowo wykorzystywać dane pochodzące z systemów do określania pozycji takich jak GPS (*Global Positioning System*) czy GLONASS. Wszystkie przedstawione powyżej typy sensorów były wykorzystywane w badaniach związanych z dynamiką osób, w szczególności do symulacji sterowanych danymi (*data-driven simulation*). Prowadzone są również badania nad łączeniem danych pochodzących z różnych typów sensorów w celu zwiększenia dokładności i szybkości odczytów; jest to tzw. fuzja czujników (*sensor fusion*). Dobrze rokującym podejściem w ramach tej techniki jest zastosowanie algorytmów sztucznej inteligencji do dynamicznego określania sposobu łączenia danych pochodzących z poszczególnych sensorów.

Ważnym elementem symulacji dynamiki tłumy jest towarzysząca jej wysokowydajna i funkcjonalna wizualizacja wyników. Zamieszczanie w systemie wizualizującym dużej liczby mobilnych i trudno rozróżnialnych obiektów stawia systemowi wizualizacji duże wymagania. Konieczne jest utrzymanie właściwego poziomu przejrzystości takiej wizualizacji. Dlatego przy weryfikacji wyników badań dotyczących dynamiki tłumy warto postawić na wykorzystanie zaawansowanych trójwymiarowych stereoskopowych systemów renderowania, które pozwolą wprowadzić zarówno klasyczną trójwymiarową wizualizację wyników w wirtualnej rzeczywistości VR (*virtual reality*), jak i umożliwią użycie systemu rozszerzonej rzeczywistości AR (*augmented reality*). Wtedy dodatkowo gogle AR lub inny system renderowania AR pozwolą na umieszczenie odbiorcy fizycznie w obrębie podlegającego analizom obiektu (wypełnionego tłumem), umożliwiając mu porównanie wyników symulacji (naniesionych w tym otoczeniu trójwymiarowo w miejscach przebywania osób) ze stanem faktycznym. Istnieje wiele technik wytwarzania sygnału stereoskopowego dla systemów VR i AR, które stosuje się w zależności od rodzaju instalacji sprzętowej planowanej do użycia w systemie wizualizacji. Wspólną cechą stereoskopowych trójwymiarowych systemów renderujących jest generowanie sygnału wizyjnego podwójnie – osobno dla prawego i lewego oka, z korektami geometrii kadru klatki animacji trójwymiarowej uwzględniającymi poziomy rozstaw oczu odbiorcy-observatora. System sprzętowy ma na zadanie prezentację każdej z dwóch sprzężonych ze sobą klatek animacji przed odpowiednim okiem odbiorcy (prawym lub lewym). W przypadku wykorzystania dwóch niezależnych wyświetlaczy konieczne jest wyrenderowanie dwóch klatek przeznaczonych do eksponowania przed oczami odbiorcy w tym samym czasie i bez dalszych modyfikacji. Taki sygnał można podać do gogli VR lub do systemu projekcji wykorzystującego dwa niezależne projektory i techniki rozdzielania obrazu bazujące na polaryzacji światła. System projekcji może jednak bazować na naprzemiennej ekspozycji sygnału wizyjnego (klatki animacji). Wówczas projekcja prowadzona jest z wykorzystaniem jednego wyświetlacza lub projektora, a klatka animacji przeznaczona dla danego oka odbiorcy prezentowana jest w chwili, gdy drugie oko jest zasłonięte (na przykład przy użyciu maski LCD). Częstotliwość naprzemiennego przesłaniania i ekspozycji klatki animacji jest tak duża (około 120 Hz), iż odbiorca nie zauważa migotania klatek. W niektórych sytuacjach składniki sygnału wizyjnego dla prawego i lewego oka miksowane są ze sobą w jednym wizyjnym sygnale (w jednej klatce animacji). Wtedy przykładowo komponowane są one naprzemiennie liniami poziomymi w wynikowej klatce z dwóch klatek wejściowych. Sygnał taki podaje się na wyświetlacze wyposażone w nakładki zawierające pionowe liniowe soczewki Fresnela powodujące odchylenie promieni światła z odpowiednich linii pionowych wyświetlacza w kierunku odpowiednio prawego lub lewego oka odbiorcy. Innym wariantem miksowania jest łączenie wybranych składowych różnych kolorów z klatek prawej i lewej w jednej klatce wynikowej – na użytek anaglifowych technik projekcji stereoskopowej.

Tak przygotowany sygnał odbiorca obserwuje z użyciem okularów filtrujących składowe kolorystyczne. Aktualnie powstają coraz bardziej zaawansowane prototypy uniwersalnych silników graficznych obsługujących wszystkie opisane techniki z możliwością ich szybkiego przełączania. W przypadku symulacji dynamiki tłumy umożliwi to jednocześnie i zsynchronizowaną prezentację wizyjną zarówno dla odbiorców fizycznie znajdujących się w podlegającym analizom obiekcie (dzięki technikom AR), jak i prowadzących studia teoretyczne.

3. Systemy ADAS – modelowanie i walidacja

Zaawansowane systemy wspomagania kierowcy (ADAS) to złożone systemy elektroniczne i mechatroniczne. Zostały one opracowane w celu zwiększenia komfortu kierowcy oraz zapewnienia bezpieczeństwa na drogach dzięki ograniczeniu ludzkich błędów. Systemy te integrują wiele specjalistycznych urządzeń: od czujników (kamery, radary, lidar), przez elektroniczne jednostki sterujące (ECU), po elementy elektromechaniczne. Główną funkcją tych systemów jest unikanie potencjalnie ryzykownych sytuacji drogowych, m.in. kolizji, wraz z możliwością przejścia kontroli przez pojazd w sytuacjach zagrażających zdrowiu i życiu (m.in. automatyczne hamowanie).

Systemy ADAS wykorzystują złożone modele matematyczne do opisu środowiska otaczającego pojazd. Rozróżniamy dwa dopełniające się typy modeli: modele obiektów dynamicznych oraz stacjonarnych. Modele obiektów dynamicznych opisują obiekty (samochody, piesi, rowerzyści itd.) poruszające się (niekoniecznie w danej chwili czasu). Obiekty te są opisane za pomocą modeli ruchu, takich jak CA (*constant acceleration*), CV (*constant velocity*), CTRA (*constant turn rate and acceleration*) i wiele innych. Natomiast w przestrzeni są one reprezentowane przez prymitywne kształty (np. prostokąt). Do pełnego opisu otaczającego środowiska potrzebny jest jeszcze opis stacjonarnych elementów (barierki, krawężniki itp.) Do ich opisywania wykorzystywana jest szeroka gama modeli. Można je podzielić w następujący sposób (Szlachetka i in. 2020): prymitywne kształty, siatki zajętości, sektory oraz kontury. W praktyce modele dobiera się stosownie do wymagań stawianych systemowi oraz z uwzględnieniem ograniczeń sprzętowych, czasu obliczeń, wykorzystania pamięci czy rodzaju użytych czujników.

Wszystkie podsystemy, ze względu na normy bezpieczeństwa, muszą działać niezależnie i ze stuprocentową pewnością, ponieważ wpływają na inne elementy bezpieczeństwa czynnego i podstawowe funkcje pojazdu. Proces walidacji i weryfikacji tych systemów jest bardzo czasochłonny i kosztowny oraz wymaga dużych zasobów ludzkich. W związku z tym konieczne jest prowadzenie niezależnych badań mających na celu opracowanie nowych algorytmów weryfikacji zarówno komponentów programowych, jak i sprzętowych. Symulacja komputerowa jest najlepszym rozwiązaniem do walidacji

i oceny systemów ADAS pod względem kosztów badań i rozwoju, ponownego wykorzystania i ewentualnych modyfikacji. W tym celu opracowywane są wirtualne modele poszczególnych elementów składowych całego systemu ADAS, jak również całych komponentów. Do weryfikacji oprogramowania używa się metodologii SIL (*software in the loop*) opartej na testowaniu w pętli zamkniętej. Główną częścią przygotowania i wdrożenia modelu symulacyjnego w pętli zamkniętej jest emulacja rzeczywistego środowiska wokół pojazdu. Odbywa się to za pomocą specjalnie stworzonych modeli wirtualnych. Aby to było możliwe, niezbędny jest podział implementacji na część symulującą warstwę sprzętową i część symulującą warstwę logiczną (algorytmiczną) rozwiązania. Pierwszy to model warstwy abstrakcji czujnika, drugi to model abstrakcji logicznej. Głównym zadaniem modelu czujnika jest pozyskiwanie informacji ze środowiska wirtualnego otrzymanej z symulacji środowiska wirtualnego oraz tworzenie listy wykrytych obiektów. Model ten zawiera wszystkie informacje przechwycone z urządzenia, podstawowe algorytmy (przykładowo dla kamery algorytmy okluzji czy podstawowy algorytm konfiguracji usuwający zniekształcenia obrazu kamery). Model logiczny przetwarza wiele danych wejściowych z każdego z czujników. Celem tak przeprowadzanej symulacji jest stworzenie projekcji prawdziwego otoczenia samochodu na podstawie rzeczywistych scenariuszy drogowych. Do wizualizacji oraz testowania poszczególnych modeli stosuje się symulatory typu MATLAB Driving Scenario Designer albo Carla. Oprogramowanie to może dostarczyć pełne informacje o środowisku, od danych symulowanych czujników po dynamikę, ruch aktorów, a na końcu wszystkie sceny 3D wirtualnego świata.

4. Wybrane problemy identyfikacji i symulacji systemów dynamicznych

4.1. Identyfikacja parametryczna równań różniczkowych liniowych

Opis dynamiki wielu występujących w przyrodzie zjawisk gwarantują ciągłe modele nieliniowe. Wynika to z podstawowych praw fizyki, np. hydrodynamiki, aerodynamiki, sprężystości czy praw zachowania energii i masy. Ze względu na brak ogólnej teorii rozwiązywania nieliniowych równań różniczkowych powszechnie stosuje się znaną teorię liniowych równań różniczkowych. Modele zlinearyzowane wzorem Taylora wykorzystywane są wtedy tylko do wybranego obszaru zmienności.

Liniowe modele dynamiki są opisywane równaniem różniczkowym zwyczajnym, z parametrami mającymi interpretację fizyczną, z danymi warunkami początkowymi:

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_0 y(t) = b_m u^{(m)}(t) + \dots + b_0 u(t) \quad (1)$$

Mierzonymi sygnałami są $y(t)$ i $u(t)$, nie jest znany przebieg ich pochodnych oraz nie są znane współczynniki a_i i b_j . Często stosuje się wtedy zastąpienie pochodnych ich

ilorazami różnicowymi, co w wyniku identyfikacji daje inne parametry niż te w (1). Istnieje metodologia, która pozwala zidentyfikować bezpośrednio parametry a_i i b_j z (1), wykorzystująca transformację równania (1) za pomocą iloczynu spłotowego do równania algebraicznego, w którym wszystkie funkcje są już znane, a następnie dokonywana jest identyfikacja za pomocą ciągłej wersji MNK (Rao i Unbehauen 2006). Transformacja spłotowa na przesuwym oknie $[t-h, t]$ wykorzystuje własność całkowego iloczynu spłotowego nieznanymi pochodnych $y^{(i)}(t)$ i $u^{(i)}(t)$ ze znaną, specjalnie dobraną funkcją $\varphi(t)$, niezerową tylko na przedziale $[0, h]$ i o znanych pochodnych $\varphi^{(i)}(t)$ na tym przedziale (Shinbrot 1957). Transformacja zachowuje wartość całki przy zamianie rzędów pochodnych między funkcjami, co umożliwi wyliczenie przebiegu tak otrzymanych funkcji spłotowych $y_i(t)$ i $u_i(t)$:

$$u_i(t) = \int_{t-h}^t u^{(i)}(\tau) \varphi(t-\tau) d\tau = \int_{t-h}^t u(\tau) \varphi^{(i)}(t-\tau) d\tau \quad (2)$$

Analogicznie otrzymujemy funkcję $y_i(t)$. Po transformacji (2) zastosowanej do obu stron równania (1) model różniczkowy (1) staje się modelem algebraicznym (3), z już znanymi na przedziale $[h, T]$ funkcjami $y_i(t)$ i $u_i(t)$ i nieznanymi parametrami a_i i b_j :

$$\sum_{i=0}^n a_i y_i(t) = \sum_{j=0}^m b_j u_j(t) \quad (3)$$

Powyższa metoda wykazuje całkowitą niezależność od warunków początkowych, a transformacja całkowa spełnia dodatkowo rolę filtra uśredniającego zakłócenia (Byrski 2021). Metoda działa bardzo dobrze i została sprawdzona na danych przemysłowych procesu kondycjonowania szkła (Byrski i in. 2021). Może być stosowana w odniesieniu do procesów wysokiego rzędu, a obliczenia mogą być zrównoleżone z wykorzystaniem GPGPU. Dokładność numeryczna zależy od zastosowanej metody całkowania.

4.2. Symulacja systemu dynamicznego źle uwarunkowanego

O złożoności systemu decyduje liczba jego komponentów i rodzaj wzajemnych oddziaływań. Przykładem systemu złożonego może być podwójna helisa DNA zbudowana z miliardów nukleotydów. Symulacja ruchu solitonu (fali przesuwnej) wzdłuż helisy wymaga zbudowania wcześniej modelu dynamiki pojedynczego nukleotydu w otoczeniu sąsiednich. W pracy Cadoniego i in. (2007) wykazano, że można tu zastosować wyidealizowany model wahadła podwójnego opisywany dwoma równaniami różniczkowymi 2. rzędu.

Model dynamiki helisy DNA jest wtedy układem dużej liczby równań różniczkowych. Są to równania źle uwarunkowane. Numeryczne rozwiązywanie równań nawet tylko jednego wahadła podwójnego wymaga doboru specjalnych algorytmów. Ich przegląd można znaleźć w publikacji Szymkata (2021). Na przykład zwiększenie dokładności może wymagać rozwiązywania równań sprzężonych. Tu pojawia się konieczność obliczania macierzy Jacobiego funkcji prawych stron równań różniczkowych.

Wydajnym sposobem znajdowania macierzy Jacobiego (a także gradientów i innych form pochodnych) jest wykorzystanie metod różniczkowania automatycznego (Baydin i in. 2018), które łączą w sobie zalety różniczkowania numerycznego i symbolicznego. Metody różniczkowania automatycznego wykorzystują regułę łańcuchową i pozwalają na wydajne obliczanie wartości pochodnych z dokładnością maszynową. Niektóre z gotowych implementacji pozwalają nawet na automatyczne wygenerowanie kodu obliczającego pochodną na podstawie przekazanych im funkcji (co dodatkowo skraca czas potrzebny na implementację rozwiązania).

5. Systemy złożone w zarządzaniu (ERP)

5.1. Przedsiębiorstwo jako system złożony

Organizacja, przedsiębiorstwo, sieci dostawców, jak również procesy decyzyjne uważane są za systemy złożone (Choi i in. 2001, Fabac 2010, McCarthy 2003). Dla tego typu systemów złożonych powstało wiele modeli, symulacji oraz narzędzi wspierających decyzje (Nilsson 2006, Zhou i in. 2003). Analiza narzędzi wspierających podejmowanie decyzji wykazuje, że kluczowymi czynnikami wpływającymi na zmienność wyników oraz wskaźnik niepowodzeń są szeroko rozumiane zagadnienia związane z systemem ERP (*enterprise resource planning*) (Baldwin i in. 2010).

Wyniki badań potwierdzają, że skuteczność procesu wdrożeniowego systemu ERP jest kluczowa dla organizacji, która chciałaby osiągnąć nowy poziom złożoności. Analiza przypadków dotyczących projektów wdrożeniowych wykazuje, że brak zmiany poziomu złożoności w organizacji może wynikać z tego, że proces wdrożeniowy zakończył się niepowodzeniem lub właściwy poziom złożoności był osiągnięty jeszcze przed wdrożeniem (Fontana i Iarozinski Neto 2009). Przykładem osiągnięcia przez przedsiębiorstwo oczekiwanego poziomu złożoności jeszcze przed wdrożeniem jest realizacja projektu systemu ERP w postaci upgrade'u technicznego. Projekt tego typu nie wpływa istotnie na optymalizację procesów technologicznych czy biznesowych, ale jego przeprowadzenie jest ważne ze względów formalnoprawnych i dla utrzymania technicznego (Domagała i in. 2021).

5.2. ERP jako system wieloagentowy

Współczesne aplikacje do zarządzania przedsiębiorstwem to systemy o złożonej architekturze, które mają za zadanie wesprzeć liderów organizacji w podejmowaniu kluczowych decyzji biznesowych (Goldston 2020). Liczne definicje ERP opisują system jako narzędzie integrujące informacje biznesowe, stanowiące wsparcie dla procesów i przepływu informacji oraz dostarczające zestawu gotowych komponentów, których wdrożenie zależne jest od potrzeb klienta (Klaus i in. 2000). Ze względu na wszechstronne możliwości oraz zakres wykorzystania systemu tej klasy można również rozpatrywać go jako system złożony. Jako system złożony, ERP jest przykładem zaawansowanego systemu wieloagentowego, dla którego od ponad dekady istnieją naukowe modele (Lea i in. 2005). Współczesne systemy ERP wykorzystują wbudowanych agentów do podejmowania automatycznych działań na podstawie dostępnych i/lub napływających z innych systemów informacji.

Obsługa procesu zamówienia klienta jest typowym procesem realizowanym często w znacznym stopniu autonomicznie za pomocą wbudowanych agentów (Yu i in. 2008). Uproszczona ścieżka takiego procesu rozpoczyna się od automatycznego wygenerowania informacji na etapie ofertowania, gdzie klient/użytkownik otrzymuje propozycję zgodną z jego historią zamówień, obrotami i innymi warunkami determinującymi uzyskaną cenę oraz możliwy termin realizacji (działanie agenta 1). Zamówienie może być również tworzone w pełni automatycznie na bazie platform pośrednich wspomagających komunikację pomiędzy systemami dostawcy i klienta lub bezpośrednio między nimi (agent 2). Działanie agenta 2 polega na obsłudze zamówienia i weryfikacji danych pod względem dostępności towaru, grupy klientów, warunków realizacji (termin, płatność, dostawa, poziom kompletacji), a następnie przekazaniu informacji do dalszego procesowania przez innych agentów w zależności od wyników działania agenta 2 (López-Ortega i in. 2008). Po zakończeniu weryfikacji stanu zapasów magazynowych i założeń wstępnych klienta/zamówienia agent 3 komunikuje się z magazynem (agent 4), zlecając wykonanie kompletacji dostępnego towaru. Skompletowany towar jest przekazywany do przygotowania wysyłki przez agenta 5, który na podstawie zebranych informacji drukuje dokumenty, komunikuje się ze spedytorem, generując listy przewozowe i udostępnia towar do spakowania (linia pakująca). Po przekazaniu przez spedytora towaru do klienta i przesłaniu automatycznej informacji zwrotnej agent 2 zamyka proces rozliczeniowy, generując finalne dokumenty. Ten sam agent monitoruje nierozliczone zamówienia i dąży do zamknięcia otwartych zamówień, oczekując na wyprodukowanie/dostawę brakujących produktów. Jest to najprostszy realizowany w systemie ERP scenariusz w tym zakresie. Proces ten można znacznie rozbudować

o kolejnych agentów umożliwiających automatyzację innych procesów systemowych w ERP. Są to:

- agent magazynowy monitorujący stany zapasów magazynowych, odpowiedzialny za wsparcie procesu kompletacji zamówienia pod względem optymalizacji jego ścieżki;
- agent produkcyjny monitorujący historyczne i utracone zamówienia, planujący produkcję na podstawie rzeczywistych i prognozowanych potrzeb, a także monitorujący oraz planujący obciążenie maszyn;
- agent obszaru zakupu generujący automatyczne zapotrzebowania na podstawie planów produkcyjnych, dostępnych zapasów magazynowych i innych warunków wejściowych;
- agent finansowy monitorujący i wykonujący rozliczenia finansowe na podstawie plików i danych z systemów zewnętrznych, kontrolujący limity kredytowe klientów, harmonogram płatności oraz inne zdarzenia systemowe i biznesowe;
- agent kadrowy monitorujący i realizujący wsparcie w zakresie automatyzacji procesów kadrowych;
- inni agenci odpowiedzialni za komunikację z innymi systemami.

Współczesne standardy i praktyki wymuszają coraz częstsze wykorzystanie agentów w systemie ERP, co powoduje jednak coraz większe problemy z wydajnością i realizacją procesów w złożonym systemie. Zdarza się, że niektórzy z agentów realizują bardzo złożone zadania i negatywnie wpływają na efektywność całego systemu. Dlatego też użycie systemowych agentów w ERP wiąże się z koniecznością prowadzenia stałego monitoringu komunikacji i jej efektywności pomiędzy nimi (Berna-Koes i in. 2004).

6. Modelowanie systemów biologicznych – schemat kolbki presynaptycznej

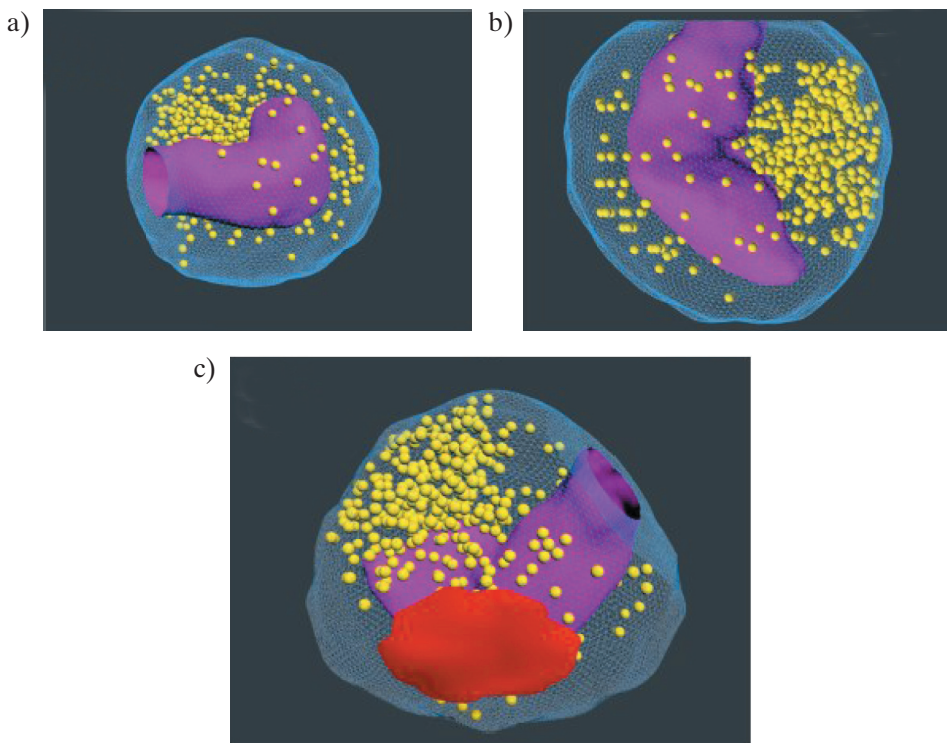
Każdy typ komórki biologicznej, łącznie z najprostszymi komórkami prokariotycznymi, otrzymuje sygnały ze środowiska i przetwarza je. Jednak tylko tkankowce właściwe (*eumetazoa*) wykształciły neurony – komórki odpowiedzialne za złożone przetwarzanie sygnałów, zdolne przesyłać sygnały na duże odległości dzięki strukturom sieciowym utworzonym z połączonych neuronów i komórek gębowych. Układy neuronalne tworzą struktury o bardzo dużym stopniu złożoności. Jednym z kluczowych etapów przetwarzania sygnałów w sieciach neuronowych jest przekazywanie sygnałów między neuronami. Połączeniem między dwoma neuronami jest synapsa, składająca się z kolbki presynaptycznej, szczeliny synaptycznej i receptora neuronu postsynaptycznego.

Przekazywanie impulsów nerwowych, a zwłaszcza zaburzenia tego procesu (depresja synaptyczna) były od dawna przedmiotem badań (Hughes 1958). W synapsach

chemicznych przekazywanie impulsów polega na dostarczaniu, transporcie i uwalnianiu neuroprzekaźnika. Działalność zespołu części współautorów niniejszego artykułu dotyczyła m.in. modelowania tych zjawisk w kolbce presynaptycznej neuronu.

Kolbka presynaptyczna była reprezentowana przez modele 3D o różnym stopniu złożoności. Prosty model, wzorowany na modelu 2D (Bielecki i in. 2010), tworzyła sfera zewnętrzna i jedna lub dwie sfery wewnętrzne (obszar pojawiania się neuroprzekaźnika). Wyróżniony obszar błony zewnętrznej (strefa aktywna) jest miejscem uwalniania neuroprzekaźnika do szczeliny synaptycznej (Bielecki i in. 2020). Kolejne modele były bardziej złożone (rys. 1). Jeden z nich, zbudowany z wykorzystaniem danych z literatury (Wilhelm i in. 2014), przypomina kształtem prawdziwą kolbkę (Gierdziewicz 2021). Obecne jest w nim mitochondrium zajmujące częściowo obszar kolbki oraz mikrotubule; na ich końcach jest uwalniany neuroprzekaźnik.

Przepływ neuroprzekaźnika modelowano za pomocą nieliniowego równania różniczkowego cząstkowego; trwają prace nad wykorzystaniem do tego celu automatu komórkowego.



Rys. 1. Trzy widoki jednego z opracowanych modeli kolbki presynaptycznej neuronu. Sztuczna granica między kolbką a aksonem znajduje się: z lewej (a), u góry (b), z prawej u góry (c). Znaczenie kolorów: szary – ściana kolbki, fioletowy – mitochondrium, żółty – pęcherzyki synaptyczne, czerwony – obszar uwalniania neuroprzekaźnika

7. Predykcja i analiza systemów złożonych z wykorzystaniem uczenia maszynowego i sieci głębokich

Sieci rekurencyjne (*reccurent neural networks*) zaliczane są do modeli uczenia głębokiego wykorzystywanych typowo do rozwiązywania problemów z wykorzystaniem danych sekwencyjnych (takich jak np. szeregi czasowe). W architekturze tego rodzaju sieci pojawiają się połączenia pomiędzy poszczególnymi komórkami, pozwalające uwydatnić i zachować wcześniej występujące informacje i wykorzystać je do przeprowadzania predykcji kolejnych wartości. Jest to możliwe dzięki obecności pętli w ich architekturze, przez co dane wyjściowe z jednego kroku są wykorzystywane jako dane wejściowe w kroku następnym.

W ramach pracy badawczej zostały wykorzystane między innymi warstwy sieci LSTM (*long short term memory*), których działanie opiera się na dwóch wektorach stanu. Sieci te idealnie sprawdzają się w sektorach działań związanych z analizą tekstu, mowy czy nagrań dźwiękowych, a także analizach szeregów czasowych.

Komórka GRU (*gated recurrent unit*) to okrojona wersja LSTM, w tej architekturze oba wektory stanu są scalone w pojedynczy wektor.

Sieci konwolucyjne hybrydowe CNN + LSTM/GRU złożone są z warstw sieci splotowych i sieci rekurencyjnych. Warstwa CNN ma za zadanie redukcję przestrzeni problemu. Sieci splotowe są nastawione na znajdowanie zależności między dwiema funkcjami, co pozwala sieciom rekurencyjnym na lepsze nauczenie się poszczególnych zależności. Taka architektura sprawdza się w analizie dźwięku i mowy, gdzie mamy do czynienia z dużą ilością danych uzależnionych od siebie czasem, jakim po sobie występują. W najczęściej spotykanych rozwiązaniach warstwy sieci splotowych występują na początku modelu, a ich wyjście jest wprowadzane do warstwy LSTM/GRU. Wada tego typu rozwiązania wynika z dużej ilości danych, która zazwyczaj jest wymagana do nauki sieci splotowych.

Model TFT (*temporal fusion transformer*) to zaawansowana architektura umożliwiająca interpretację i wielohoryzontową prognozę szeregów czasowych. Model został po raz pierwszy opracowany i wdrożony przez Google we współpracy z Uniwersytetem Oksfordzkim.

Model TFT jest architekturą hybrydową łączącą kodowanie LSTM szeregów czasowych i interpretowalność warstw uwagi transformatora. Predykcja opiera się na trzech typach zmiennych: statycznych (stałych dla danego szeregu czasowego), znanych (znanych z góry w odniesieniu do całej historii i przyszłości) i obserwowanych (znanych tylko w odniesieniu do danych historycznych).

DeepAR jest jednym z najbardziej wydajnych i najnowocześniejszych modeli prognostycznych. Wydany przez Amazon i zintegrowany z platformą ML SageMaker, DeepAR wyróżnia się zdolnością uczenia się na „skali” przy użyciu wielu współzmienn-

nych. Składa się z metodologii prognozowania opartej na RNN AR, która uczy się globalnego modelu na podstawie danych historycznych ze wszystkich szeregów czasowych w zbiorze danych i tworzy dokładne prognozy probabilistyczne.

Niezwykle interesującym obszarem zastosowań głębokich sieci neuronowych jest dziedzina przekładu komputerowego. Początki badań nad automatyzacją przekładu sięgają późnych lat 40. XX wieku, gdy zaczęto rozważać możliwości innych zastosowań maszyn cyfrowych niż domena obliczeń numerycznych. W 1954 roku w USA odbył się pierwszy publiczny pokaz przekładu komputerowego, który w owym czasie wywołał wielką sensację. Niektórzy wieszczyli, że za kilkanaście lat zwód tłumacza całkowicie zaniknie, ponieważ tego rodzaju zadania będą w całości realizowane za pomocą komputerów. Niestety przez kolejne pół wieku w dziedzinie komputerowego przekładu nie nastąpił zbyt wielki postęp, a prace badawcze – wydawało się – tkwiły w miejscu. Przełom dokonał się dopiero w drugiej dekadzie XXI wieku, gdy do realizacji przekładu komputerowego zaprzęgnięto sztuczne sieci neuronowe o architekturze głębokiej. Obecnie zdecydowanym liderem w tej dziedzinie jest program DeepL, który tłumaczy pomiędzy ponad dwudziestoma językami, w tym również: węgierskim, fińskim, estońskim, japońskim i chińskim. Jakość uzyskiwanego przekładu w przypadku programu DeepL jest bardzo wysoka, w związku z czym z jednej strony zawodowi tłumacze mogą się w pewnym sensie czuć zagrożeni, chociaż z drugiej – sami masowo z tego programu już korzystają. Działanie programu DeepL oparte jest na konwolucyjnych sieciach głębokich, do których wytrenowania wykorzystano zgromadzony przez kilkanaście lat potężny korpus równoległy zawierający ponad miliard zdań w języku angielskim wraz z ich przekładami na pozostałe języki. Do wytrenowania sieci głębokich potrzebna była potężna moc obliczeniowa, którą zapewnił zainstalowany na Islandii superkomputer zasilany z lokalnej elektrowni wodnej. Program DeepL podczas tłumaczenia pomiędzy językami innymi niż język angielski wykorzystuje ten język jako język pośredniczący. Co ciekawe, takie dwuetapowe tłumaczenie nie powoduje widocznego obniżenia jakości uzyskiwanych przekładów, które są bardzo dobre nawet w przypadku języków „egzotycznych”, czyli nienależących do indoeuropejskiej rodziny językowej.

Zarówno przy projektowaniu, jak i realizacji systemów złożonych pomocna może być analiza posiadanych złożonych wielowymiarowych danych, z których może wynikać zarówno sposób działania, jak i architektura konstruowanego systemu. Dlatego ważne może być zwrócenie uwagi na pewne jakościowe cechy takich danych. Dane te często składają się z setek, a nawet tysięcy wymiarów oraz wielu tysięcy próbek, więc ich właściwa analiza może stanowić problem. Jedną z metod pozwalających na analizę jakościową takich danych jest metoda wykorzystująca autoasocjacyjne sieci neuronowe. Są one przykładem głębokiej sieci neuronowej uczonej w ten sposób, by dla danych n -wymiarowych podawanych na jej n wejściach na jej n wyjściach pojawił się sygnał jak najbardziej zbliżony do tego, który został podany na jej n wejściach. Jeśli jedna z warstw

pośrednich składa się z dwóch neuronów, to następuje kompresja przestrzeni n -wymiarowej do przestrzeni dwuwymiarowej, a następnie dekompresja z przestrzeni dwuwymiarowej do przestrzeni n -wymiarowej. Jeśli taką sieć udało się prawidłowo nauczyć, oznacza to, że w wyniku kompresji w przestrzeni dwuwymiarowej zostały uchwycone wszystkie istotne cechy oraz struktura analizowanych danych, pozwalające na w miarę wierną dekompresję tych danych. Jednocześnie każda próbka reprezentowana jest w warstwie złożonej z dwóch neuronów przez dwie wartości, więc tych dwóch wartości można bezpośrednio użyć jako współrzędnych obrazu analizowanej próbki na stworzonym rysunku. W ten sposób możemy na dwuwymiarowym rysunku zobaczyć obraz wielowymiarowych złożonych danych, który zachowuje istotne cechy oraz strukturę tych danych. W celu analizy wielowymiarowych danych poza stosowaniem kryterium uczenia autoasocjacyjnej sieci neuronowej stosuje się również odmienne kryteria uczenia sieci neuronowych (Jamróz 2020). W wyniku prowadzonych badań nad różnymi kryteriami wykazano, że użycie jednego z zaproponowanych kryteriów pozwala uzyskać skuteczność dwukrotnie wyższą niż przy zastosowaniu autoasocjacyjnej sieci neuronowej.

8. Podsumowanie

W artykule omówione zostały wybrane zagadnienia związane z badaniami nad systemami złożonymi (*complex systems*), które były przedmiotem prac badawczych pracowników i doktorantów Katedry Informatyki Stosowanej WEAIIB w ostatnich latach. W roku 2022 mija dwadzieścia lat od wydania książki Stephana Wolframa *A New Kind of Science*. Z perspektywy czasu, na podstawie analizy literatury i przeprowadzonych prac badawczych możemy stwierdzić, że metody dyskretne modelowania działające w schemacie *bottom-up*, takie jak automaty komórkowe, znacząco się rozwinęły i zyskały na popularności. Tendencją, którą obserwujemy aktualnie, jest hybrydyzacja, czyli łączenie automatów z innymi technikami modelowania w celu tworzenia efektywnych funkcji przejścia: łączenie z algorytmami genetycznymi, metodą elementów skończonych (MES) czy systemami rozmytymi. Analiza literatury potwierdza, że wciąż rozwijają się kolejne pola zastosowań automatów komórkowych, natomiast w wielu dziedzinach nadal zastosowanie mają metody tradycyjne, oparte na równaniach różniczkowych. Istotnym trendem w opisie i analizie systemów złożonych są aktualnie również metody uczenia maszynowego czy głębokich sieci neuronowych.

Literatura

Baldwin J., Allen P., Ridgway K., 2010, *An evolutionary complex systems decision-support tool for the management of operations*, International Journal of Operations & Production Management, vol. 30(7), s. 700–720.

- Baydin A., Pearlmutter B., Radul A., Siskind J., 2018, *Automatic differentiation in machine learning: a survey*, The Journal of Machine Learning Research, vol. 18(153), s. 1–43.
- Belbachir A., Smal J.C., Blosseville J.M., Gruyer D., 2012, *Simulation-Driven Validation of Advanced Driving Assistance Systems*, Procedia – Social and Behavioral Sciences, vol. 48, s. 1205–1214. <https://doi.org/10.1016/j.sbspro.2012.06.1096>.
- Berna-Koes M., Nourbakhsh I., Sycara K., 2004, *Communication efficiency in multi-agent systems*, [w:] *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 – May 1, 2004, New Orleans, LA, USA*, vol. 3, IEEE, Piscataway, s. 2129–2134. <https://doi.org/10.1109/ROBOT.2004.1307377>.
- Bielecki A., Kalita P., Lewandowski M., Siwek B., 2010, *Numerical simulation for a neurotransmitter transport model in the axon terminal of a presynaptic neuron*, Biological Cybernetics, vol. 102, s. 489–502. <https://doi.org/10.1007/s00422-010-0380-z>.
- Bielecki A., Gierdziewicz M., Kalita P., Szostek K., 2016, *Construction of a 3D geometric model of a presynaptic bouton for use in modeling of neurotransmitter flow*, [w:] Chmielewski L., Datta A., Kozera R., Wojciechowski K. (eds.), *Computer Vision and Graphics: International Conference: ICCVG 2016: Warsaw, Poland, September 19–21, 2016: proceedings*, Lecture Notes in Computer Science, vol. 9972, Springer, Cham, s. 377–386. https://doi.org/10.1007/978-3-319-46418-3_33.
- Bielecki A., Gierdziewicz M., Kalita P., 2020, *A study on efficiency of 3D partial differential diffusive model of presynaptic processes*, Biocybernetics and Biomedical Engineering, vol. 40, s. 100–118. <https://doi.org/10.1016/j.bbe.2019.02.001>.
- Blue V.J., Adler J.L., 2002, *Flow capacities from cellular automata modeling of proportional splits of pedestrians by direction*, [w:] Schreckenberg M., Sarma S. (eds.), *Pedestrian and Evacuation Dynamics*, Springer, Berlin, Heidelberg, s. 115–122.
- Bringmann E., Krämer A., 2008, *Model-based Testing of Automotive Systems*, [w:] *First International Conference on Software Testing, Verification, and Validation, ICST 2008, Lillehammer, Norway, April 9–11, 2008*, IEEE, Piscataway, s. 485–493. <https://doi.org/10.1109/ICST.2008.45>.
- Byrski J., 2021, *Identyfikacja współczynników liniowego równania różniczkowego*, [w:] Wąs J. (red.), *Wybrane zagadnienia numeryczne: analiza i dobór algorytmów*, Wydawnictwa AGH, Kraków, s. 64–91.
- Byrski W., Drapała M., Byrski J., 2021, *New on-line algorithms for modelling, identification and simulation of dynamic systems using modulating functions and non-asymptotic state estimators: case study for a chosen physical process*, [w:] Paszynski M., Kranzlmüller D., Krzhizhanovskaya V.V., Dongarra J.J., Sloot P.M. (eds.), *Computational Science – ICCS 2021: 21st International Conference: Krakow, Poland, June 16–18, 2021: proceedings, Pt. 4*, Lecture Notes in Computer Science, vol. 12745, Springer, Cham, s. 284–297. https://doi.org/10.1007/978-3-030-77970-2_22.

- Cadoni M., De Leo R., Gaeta G., 2007, *Solitons in a double pendulums chain model, and DNA roto-torsional dynamics*. <https://www.atlantis-press.com/article/1122.pdf> [dostęp: 4.04.2022].
- Choi T.Y., Dooley K.J., Rungtusanatham M., 2001, *Supply networks and complex adaptive systems: control versus emergence*, *Journal of Operations Management*, vol. 19, no. 3, s. 351–66. [https://doi.org/10.1016/S0272-6963\(00\)00068-1](https://doi.org/10.1016/S0272-6963(00)00068-1).
- Domagała A., Grobler-Dębska K., Wąs J., Kucharska E., 2021, *Post-Implementation ERP Software Development: Upgrade or Reimplementation*, *Applied Sciences*, vol. 11(11), 4937. <https://doi.org/10.3390/app11114937>.
- Fabac R., 2010, *Complexity in organizations and environment – adaptive changes and adaptive decision-making*, *Interdisciplinary Description of Complex Systems*, vol. 8(1), s. 34–48.
- Fontana R.M., Iarozinski Neto A., 2009, *ERP Systems implementation in complex organizations*, *Journal of Information Systems and Technology Management*, vol. 6, no. 1, s. 61–92. <https://doi.org/10.4301/S1807-17752009000100004>.
- Gierdziewicz M., 2021, *Relations between geometric parameters and numerical simulation accuracy in modeling signal transmission in the presynaptic bouton*, *Applied Sciences (Basel)*, vol. 11(6), 2811. <https://doi.org/10.3390/app11062811>.
- Gietelink O., Ploeg J., De Schutter B., Verhaegen M., 2006, *Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations*, *Vehicle System Dynamics*, vol. 44(7), s. 569–590. <https://doi.org/10.1080/00423110600563338>.
- Goldston J., 2020, *The Evolution of ERP Systems: A Literature Review*, *International Journal of Research Publications (IJRP)*, vol. 50(1).
- Hanke T., Hirsenkorn N., van-Driesten C., Garcia-Ramos P., Schiementz M., Schneider S., Biebl E., 2017, *A generic interface for the environment perception of automated driving functions in virtual scenarios*. <https://www.hot.ei.tum.de/forschung/automotive-veroeffentlichungen/> [dostęp: 5.04.2022].
- Hansen N., Wiechowski N., Kugler A., Kowalewski S., Rambow S., Busch R., 2017, *Model-in-the-Loop and Software-in-the-Loop Testing of Closed-Loop Automotive Software with Artest*, [w:] Eibl M., Gaedke M. (Hrsg.), *Informatik 2017*, Gesellschaft für Informatik, Bonn, s. 1537–1549. http://doi.org/10.18420/in2017_154.
- Helbing, D., Molnár P., 1995, *Social force model for pedestrian dynamics*, *Physical Review E*, vol. 51, s.4282–4287. <https://doi.org/10.48550/arXiv.cond-mat/9805244>.
- Hughes J.R., 1958, *Post-tetanic potentiation*, *Physiological Reviews*, vol. 38, s. 91–113. <https://doi.org/10.1152/physrev.1958.38.1.91>.
- Jamróz D., 2020, *The examination of the effect of the criterion for neural network’s learning on the effectiveness of the qualitative analysis of multidimensional data*, *Knowledge and Information Systems*, vol. 62, s. 3263–3289. <https://doi.org/10.1007/s10115-020-01441-8>.

- Klaus H., Rosemann M., Gable G.G., 2000, *What is ERP?*, Information Systems Frontiers, vol. 2, s. 141–162. <https://doi.org/10.1023/A:1026543906354>.
- Lea B.-R., Gupta M.C., Yu W.-B., 2005, *A prototype multi-agent ERP system: an integrated architecture and a conceptual framework*, Technovation, vol. 25(4), s. 433–441. [https://doi.org/10.1016/S0166-4972\(03\)00153-6](https://doi.org/10.1016/S0166-4972(03)00153-6).
- Liu Y.-Y., Barabási A.-L., 2016, *Control principles of complex systems*, Reviews of Modern Physics, vol. 88(3), 035006. <https://doi.org/10.1103/RevModPhys.88.035006>.
- López-Ortega O., López-Morales V., Villar-Medina I., 2008, *Intelligent and collaborative Multi-Agent System to generate and schedule production orders*, Journal of Intelligent Manufacturing, vol. 19, s. 677–687. <https://doi.org/10.1007/s10845-008-0119-z>.
- McCarthy I.P., 2003, *Technology management – a complex adaptive systems approach*, International Journal of Technology Management, vol. 25, no. 8, s. 728–745. <https://doi.org/10.1504/IJTM.2003.003134>.
- Małeckı K., Kamiński K., Wąs J., 2022, *A multi-cell Cellular Automata model of traffic flow with emergency vehicles: Effect of a corridor of life and drivers' behavior*, Journal of Computational Science, vol. 61, 101628. <https://doi.org/10.1016/j.jocs.2022.101628>.
- Nilsson F., Darley V., 2006, *On complex adaptive systems and agent-based modelling for improving decision-making in manufacturing and logistics settings – experiences from a packaging company*, International Journal of Operations & Production Management, vol. 26(12), s. 1351–1373. <https://doi.org/10.1108/01443570610710588>.
- Rao G.P., Unbehauen G., 2006, *Identification of continuous-time systems*, IEEE Proceedings: Control Theory and Applications, vol. 153(2), s. 185–220. <https://doi.org/10.1049/ip-cta:20045250>.
- Renc P., Pęczak T., De Rango A., Spataro W., Mendicino G., Wąs J., 2022, *Towards efficient GPGPU Cellular Automata model implementation using persistent active cells*, Journal of Computational Science, vol. 59, 101538. <https://doi.org/10.1016/j.jocs.2021.101538>.
- Shinbrot M., 1957, *On the analysis of linear and nonlinear systems*, Transactions of the American Society of Mechanical Engineers: Journal of Basic Engineering, vol. 79(3), s. 547–552. <https://doi.org/10.1115/1.4013092>.
- Szlachetka M., Borkowski D., Was J., 2020, *Stationary environment models for Advanced Driver Assistance Systems*, [w:] SPA 2020: Signal Processing: Algorithms, Architectures, Arrangements and Applications: Poznan, 23rd–25th September 2020: conference proceedings, Signal Processing: Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, IEEE, Piscataway, s. 116–121. <https://doi.org/10.23919/SPA50552.2020.9241306>.
- Szymkat M., 2021, *Solwery ODE w środowisku MATLAB-a*, [w:] Wąs J. (red.), *Wybrane zagadnienia numeryczne: analiza i dobór algorytmów*, Wydawnictwa AGH, Kraków, s. 119–141.

- Wąs J., Lubaś R., 2014, *Towards realistic and effective agent-based models of crowd dynamics*, Neurocomputing, vol. 146, s. 199–209. <https://doi.org/10.1016/j.neucom.2014.04.057>.
- Wąs J., Karp A., Łukasik S., Pałka D., 2020, *Modeling of fire spread including different heat transfer mechanisms using cellular automata*, [w:] Krzhizhanovskaya V.V., Závodszy G., Lees M.H., Dongarra J.J., Sloot P.M.A., Brissos S., Teixeira J. (eds.), *Computational Science – ICCS 2020: 20th International Conference: Amsterdam, The Netherlands, June 3–5, 2020: proceedings, Pt. 1*, Lecture Notes in Computer Science, vol. 12137, Springer, Cham, s. 445–458. https://doi.org/10.1007/978-3-030-50371-0_33.
- Wilhelm B.G., Mandad S., Truckenbrodt S., Kröhnert K., Schäfer C., Rammner B., Koo S.J., Claßen G.A., Krauss M., Haucke V., Urlaub H., Rizzoli S.O., 2014, *Composition of isolated synaptic boutons reveals the amounts of vesicle trafficking proteins*, Science Reports, vol. 344, s. 1023–1028. <https://doi.org/10.1126/science.1252884>.
- Wolfram S., 2002, *A New Kind of Science*, Wolfram Media, Champaign, IL.
- Yu L., Masabo E., Tan L., He M., 2008, *Multi-Agent Automated Intelligent Shopping System (MAISS)*, [w:] 2008 The 9th International Conference for Young Computer Scientists, IEEE, Piscataway, s. 665–670. <https://doi.org/10.1109/ICYCS.2008.35>.
- Zhou Z.D., Wang H.H., Chen Y.P., Ai W., Ong S.K., Fuh J.Y.H., Nee A.Y.C., 2003, *A multi-agent-based agile scheduling model for a virtual manufacturing environment*, International Journal of Advanced Manufacturing Technology, vol. 21, no. 12, s. 980–984. <https://doi.org/10.1007/s00170-002-1420-1>.