



## METODY ZWIĘKSZANIA DOSTĘPNOŚCI I EFEKTYWNOŚCI INFORMATYCZNEJ INFRASTRUKTURY W INTELIGENTNYM MIEŚCIE

**Jerzy Balicki, Piotr Dryja, Waldemar Korlub, Maciej Tyszka**

### Streszczenie

W pracy omówiono metody zwiększania dostępności i efektywności informatycznej infrastruktury w inteligentnym mieście. Sformułowano dwa kryteria do oceny rozmieszczenia kluczowych zasobów w systemie *smart city*. Zobrazowano proces wyznaczania rozwiązań kompromisowych spośród rozwiązań Pareto-optimalnych. Omówiono metaheurystyki inteligencji zbiorowej, w tym roju cząstek, kolonii mrówek, roju pszczoł oraz ewolucji różnicowej, za pomocą których można wyznaczać kompromisową infrastrukturę *smart city*. Wskazano także na inne zastosowania metaheurystyk w *smart city*.

**Słowa kluczowe:** smart city, dostępność infrastruktury informatycznej

### Wstęp

Dostępność i efektywność informatycznej infrastruktury w inteligentnym mieście odgrywa kluczową rolę w kształtowaniu zadowolenia mieszkańców. Wpływa również na rozwój transportu, edukacji, bankowości, handlu czy ochrony zdrowia. Zwiększa także poziom bezpieczeństwa. Możliwe jest systematyczne monitorowanie zanieczyszczeń, w tym smogu. W wypadku portów morskich pojawia się dylemat skrócenia załadunku kontenerów. Z kolei telemedycyna może być bardzo pomocna w doraźnym leczeniu, a nawet operowaniu marynarzy i pasażerów przebywających na statkach na morzu.

Poprawienie jakości życia i poziomu gospodarki można zatem osiągnąć za pomocą efektywniejszej informatyzacji kluczowych procesów związanych z życiem mieszkańców inteligentnych miast. Wymagana jest jednak dostępna i efektywnie działająca infrastruktura. Należy zatem zaprojektować nową wydajniejszą infrastrukturę, z której skorzystamy za kilka lat. W tym czasie powinno się jednak sprawnie zreorganizaować już istniejącą infrastrukturę w celu

poprawienia jej działania, w tym zwiększenia dostępności do 24 godzin na dobę i 7 dni w tygodniu. Wiąże się to z uwzględnieniem niezbędnych modernizacji, konserwacji technicznych oraz aktualizacji oprogramowania. Ponadto ataki na serwery oferujące usługi miejskie są i prawdopodobnie będą, niestety, coraz bardziej rozpowszechnione. Na dodatek najsłabszym ogniwem jest zawsze człowiek: użytkownik, pracownik czy urzędnik, który może udostępnić dane wrażliwe lub hasła, a w gorszym wypadku wykorzystać swoje uprawnienia do własnych celów.

Remedium na szereg z powyższych dylematów jest niezawodny i wiarygodny system informatyczny wspierający funkcjonowanie miasta. Artykuł zorganizowany jest następująco. W sekcji 1 omówiono model informatycznej infrastruktury miejskiej, a w sekcji 2 kryterium dostępności oraz w sekcji 3 prawdopodobieństwo realizacji zadań w terminach. W sekcji 4 scharakteryzowano algorytmy inteligencji zbiorowej do zwiększania niezawodności infrastruktury. Sekcja 5 poświęcona jest prezentacji algorytmu roju cząstek, sekcja 6 – algorytmowi roju pszczół, a sekcja 7 – algorytmowi kolonii mrówek.

## 1. Informatyczna infrastruktura miejska

Z roku na rok zwiększa się liczba i rozszerza zakres projektów dotyczących *inteligentnych miast* (ang. *Smart Cities*). Wiodące w tym względzie miasta Europy takie, jak Dublin czy Amsterdam wprowadzają od dziesięciu już lat nowoczesne rozwiązania w wielu pozornie niezwiązanych ze sobą sferach życia miejskiego. *Smart City* to termin odnoszący się do miasta, które wykorzystuje dostępne rozwiązania informacyjno-komunikacyjne w celu poprawy jakości życia mieszkańców.<sup>1</sup> Mogą to być usprawnienia infrastrukturalne, jak w Trójmieście system zarządzania ruchem „Tristar”, a nawet elektroniczne tablice na przystankach autobusowych. W takich przypadkach „inteligencja” miasta jest zauważalna przez mieszkańców, ponieważ wpływa ona bezpośrednio na poprawę komfortu ich życia.<sup>2</sup>

W *smart city* stosowane są zaawansowane rozwiązania ICT (ang. *Information and Communication Technology*) do ochrony środowiska naturalnego, np. poprzez monitorowanie stanu zanieczyszczenia powietrza. Niekiedy w definicji *smart city* uwzględnia się aspekty związane z działalnością e-urzędów, co pozwala mieszkańcom na sprawne wykonywanie administracyjnych czynności przez Internet. Rolą inteligentnych miast jest wspieranie zrównoważonego rozwoju, poprawa jakości życia ich mieszkańców oraz usprawnienie procesów administracyjnych. Adekwatnym przykładem jest Dublin, który często uznawany jest za lidera w dziedzinie rozwiązań *smart city*. W Dublinie wprowadzono kilka lat temu mobilną aplikację dla mieszkańców informującą o ważnych wydarzeniach w okolicy. Mogą to być przerwy w dostawie wody, zamknięcie drogi, czy wydarzenia z życia lokalnej społeczności.<sup>3</sup> Innym rozwiązaniem są „inteligentne śmietniki”, które za pomocą bezprzewodowego połączenia z Internetem wysyłają maile, gdy tylko poziom ich wykorzystania osiągnie 85%.

*Smart city* opiera się na *Internecie Rzeczy* (ang. *Internet of Things* - IoT), który jest systemem obejmującym zbiór urządzeń wytwarzających dane oraz komunikujących się ze sobą. IoT łączy smartfony, kamery, czujniki, czujniki ruchu w ubraniach, jak również lodówki informujące o wyczerpaniu zapasów. Wachlarz zastosowań IoT jest prawie nieograniczony, a jednym

<sup>1</sup> Kanter R., Litow S., *Informed and interconnected: A manifesto for smarter cities*. Harvard Business School, General Management Unit, Working Paper, 2009, p. 223.

<sup>2</sup> Balicka H., Balicki J., Korlub W., Paluszak J., Zadroga M., *Superkomputery do wspomagania procesów gospodarczych ze szczególnym uwzględnieniem sektora bankowego*, „Współczesna Gospodarka” 2014, nr 5, s. 2.

<sup>3</sup> *Smart Dublin Project*, <http://smartdublin.ie/smartstories/unfolding-news-app> (01.07.2017).

z nich może być zastosowanie do wspierania komunikacji w systemach inteligentnych miast. Przyjmuje się, że do 2020 roku aktywnych będzie ok. 50 mld urządzeń w ramach IoT.<sup>4</sup>

Ilość danych koniecznych do przetworzenia rośnie, a obecne rozwiązania technologiczne przestają być wystarczające. Konieczne stało się ponowne zdefiniowanie metod pozwalających w zadawalającym czasie przetworzyć olbrzymie wolumeny danych. Istotną z tego punktu widzenia jest efektywna infrastruktura pozwalająca na szybkie, niezawodne przetwarzanie danych czy też dostarczanie konkretnych usług informatycznych dla mieszkańców. Wydaje się, że infrastruktura ta powinna mieć minimalne przerwy w pracy i być odporna na awarie poszczególnych jej komponentów. Te jednak są nieuniknione w przypadku projektów dużej skali.

## 2. Dostępność systemów informatycznych dla Smart City

Projektując systemy wysokiej dostępności (ang. *High Availability Systems*, HAS), dąży się do minimalizacji czasu, w którym system nie jest w stanie realizować swoich zadań. Miarą dostępności jest często procentowy udział czasu prawidłowej pracy systemu do całkowitego czasu, jak niżej:

$$\text{dostępność} = \frac{\text{całkowity\_czas} - \text{czas\_awarii\_systemu}}{\text{całkowity\_czas}} \cdot 100\% \quad (1)$$

Wartość 100 % oznacza więc, że system jest bezawaryjny. W praktyce, dostępność określa się tzw. „dziewiątkami”. Przykład wartości dostępności infrastruktury w *smart city* dla okresu 30 dni zamieszczono w tabeli 1.

**Tabela 1.** Dostępność infrastruktury *smart city* wyrażona w "dziewiątkach"

Liczba dziewiątek	Procentowa dostępność	Dopuszczalny czas awarii
2 dziewiątki	99%	432 minuty (1% z 43200 minut)
3 dziewiątki	99,9%	43 minuty (0,1% z 43200 minut)
4 dziewiątki	99,99%	4 minuty (0,01% z 43200 minut)

Źródło: Opracowanie własne.

Systemy wysokiej dostępności w *smart city* projektuje się, stosując metody minimalizujące ryzyko awarii całego systemu.<sup>5</sup> Jedną z nich jest wprowadzenie redundancji danego komponentu systemu. W przypadku awarii komponentu wiodącego, inny może go zastąpić do czasu naprawy. Przykładem może być redundantne źródło zasilania serwera czy dodatkowy dostawca usług internetowych dla centrum danych w inteligentnym mieście.

Innym ważnym podejściem jest projektowanie systemów samonaprawiających się (ang. *self-repairing*). System taki jest w stanie samodzielnie, bez interwencji człowieka, dokonać au-

<sup>4</sup> Balicki J., Dryja P., Korłub W., Przybyłek P., Tyszka M., Zadroga M., Zakidalski M., *Metody neuronowe do prognozowania finansowego*, "Współczesna Gospodarka" 2016, nr 2, s. 21.

<sup>5</sup> Winiarski J., *Ryzyko w działalności gospodarczej przedsiębiorstw*, Wyd. Uniwersytet Gdański Instytut Transportu i Handlu Morskiego, Gdańsk 2014, s. 12.

tonomicznie diagnostyki, a następnie naprawy, co nie wpływa na zmniejszenie czasu jego dostępności. Ma to szczególne znaczenie w przypadku, gdy wymagana jest wysoka dostępność. Przykładowo, oczekiwana dostępność „pięciu dziewiątek” oznacza, że system klasy *smart city* może ulec awarii na jedynie 26 sekund w ciągu miesiąca. Jest to czas zdecydowanie zbyt krótki, aby naprawy mógł dokonać człowiek.

Istnieje kilka metod, dzięki którym można osiągnąć opisywaną charakterystykę systemu. Dobrym przykładem wydaje się być usługa typu *load-balancer*, która przekierowuje otrzymane żądania do tych komponentów, które funkcjonują poprawnie. W ten sposób eliminuje się w znacznym stopniu liczbę wysłanych żądań zakończonych niepowodzeniem na skutek awarii elementu systemu. Technikę tę autorzy zastosowali w eksperymentalnym gridzie *Comcute PG* na Politechnice Gdańskiej, który jest dedykowany do wspierania wybranych sytuacji kryzysowych w modelu aglomeracji klasy *smart city*.<sup>6</sup>

Innym przykładem jest zastosowanie modułu *VMWare HA* w ramach oprogramowania *VMWare* wspierającego wirtualizację. Program obserwuje działanie węzłów w klastrze, na którym umieszczone są maszyny wirtualne. W przypadku stwierdzenia awarii, maszyny wirtualne restartowane są na poprawnie działającym węźle w sposób automatyczny. Dzięki temu znacząco zmniejsza się czas, w którym maszyna wirtualna nie ma możliwości pracy z powodu awarii komponentu, od którego była dotąd zależna.

Warto pamiętać, że wysoka dostępność systemu niesie ze sobą zwiększenie złożoności systemu, a przez to większe koszty jego wytworzenia i utrzymania. Hipoteza, że niezawodność na poziomie 99,999% jest lepsza niż ta 99% wydaje się być uzasadniona tylko w określonych warunkach. Nie każdy system wymaga takiej samej niezawodności, a konkretne decyzje powinny zapadać w oparciu o wymagania przedsięwzięcia. Niekiedy dążenie do jak najwyższej niezawodności może być kontrproduktywne i bardzo kosztowne.

Dwa komputery, które mogą przetwarzać dane do zadań w ramach *smart city* zazwyczaj cechują się różnymi charakterystykami w zakresie dostępności. Niech procedura obliczeniowa  $m_v$  może być wykonana na komputerze klasy  $\pi_j \in \Pi = \{\pi_1, \dots, \pi_j, \dots, \pi_J\}$  w czasie  $t_{vj}$ . W modelu zakłada się, że komputer klasy  $\pi_j$  ulega awarii zgodnie z rozkładem wykładniczym z parametrem  $\lambda_j$ . Zadania powinny być wykonywane na tych komputerach, na których maksymalizowana jest funkcja dostępności całej infrastruktury *smart city*  $R$ , jak niżej<sup>7</sup>:

$$R(x) = \prod_{v=1}^V \prod_{i=1}^I \prod_{j=1}^J \exp(-\lambda_j t_{vj} x_{vi}^m x_{ij}^\pi), \quad (2)$$

gdzie:

$$x_{ij}^\pi = \begin{cases} 1 & \text{gdy komputer klasy } \pi_j \text{ przydzielono do } w_i, \\ 0 & \text{w przeciwnym razie.} \end{cases}$$

$$x_{vi}^m = \begin{cases} 1 & \text{gdy zadanie } T_v \text{ przydzielono do } w_i, \\ 0 & \text{w przeciwnym razie,} \end{cases}$$

W celu maksymalizacji dostępności należy wyznaczyć wektor zmiennych decyzyjnych, za pomocą którego dokonuje się rozmieszczenia kluczowych zasobów w węzłach *smart city*:

$$x = (x^m, x^\pi) = [x_{11}^m, \dots, x_{1I}^m, \dots, x_{v1}^m, \dots, x_{vI}^m, x_{11}^\pi, \dots, x_{1J}^\pi, \dots, x_{ij}^\pi, \dots, x_{I1}^\pi, \dots, x_{Ij}^\pi, \dots, x_{IJ}^\pi]^T. \quad (3)$$

Skonstruowanie zintegrowanego systemu informatycznego w *smart city* o wysokiej dostępności HAS jest bardzo trudne. Na rysunku 1 przedstawiono wartości dostępności trzech

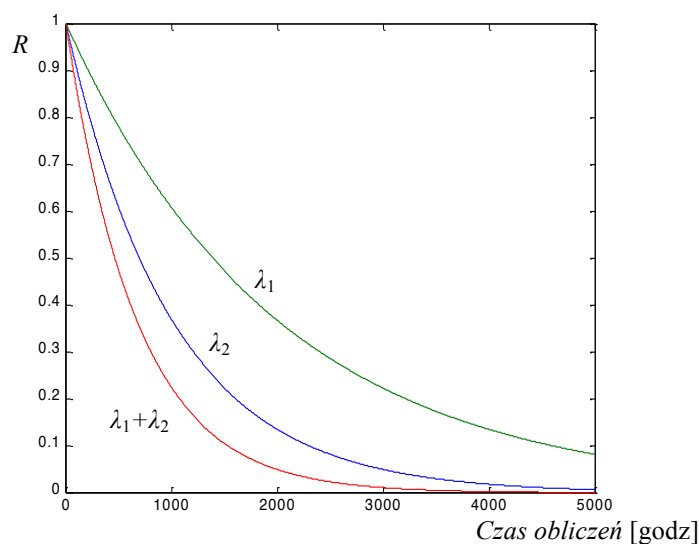
<sup>6</sup> Balicki J., Korlub W., *Uczenie maszynowe do samoorganizacji systemów rozproszonych w zastosowaniach gospodarczych*, „Współczesna Gospodarka” 2017, nr 1, s. 10.

<sup>7</sup> Balicki J., Przybyłek P., Zadroga M., Zakidalski M., *Sztuczne sieci neuronowe oraz metoda wektorów wspierających w bankowych systemach informatycznych*, „Współczesna Gospodarka” 2013, nr 4, s. 4.

systemów. Wraz z upływem czasu wyrażonego w godzinach miara dostępności  $R$  maleje wykładniczo zgodnie z zależnością (2). Niech pierwszy komputer ulega awarii średnio raz na tysiąc godzin. Wówczas  $\lambda_1=0,001$  [godz<sup>-1</sup>]. Natomiast w wypadku drugiego komputera średni czas bezawaryjnej pracy to 500 godzin, a zatem  $\lambda_2=0,002$  [godz<sup>-1</sup>]. Jeśli połączymy oba komputery na pomocą niezawodnego kanału transmisji danych, to dostępność systemu maleje znacznie szybciej, gdyż system ulega awarii średnio trzykrotnie w ciągu tysiąca godzin pracy.

Sytuację komplikuje fakt, że w systemach wspierających infrastrukturę komputerową *smart city* połączonych są tysiące komponentów, w tym komputerów, czujników czy aktuatorów. Przykładowo w systemie *SmartSantander* liczba sensorów przekracza 20 000 umieszczonych w czterech miastach: *Santander*, *Belgrad*, *Guildford* oraz *Lübeck*. Podobną złożonością cechuje się brytyjski projekt *FixMyStreet*, który przeznaczony jest do uwzględniania skarg mieszkańców na utrudnienia w ich codziennym życiu: hałas, zanieczyszczenia, dziury w drogach, czy korki.<sup>8</sup>

Z tego powodu w Japonii zbudowano miasto od podstaw z nową wysokodostępną infrastrukturą informatyczną. Eksperymenty z modelem *smart city* w *Tsukuba Science City* pozwalają na znacznie głębszą analizę i predykcję procesów zachodzących w inteligentnej aglomeracji. Wnioski z tych eksperymentów wskazują na kluczową rolę mieszkańców, jakości życia, edukacji, badań, środowiska, sposobu zarządzania miastem, gospodarki, opieki zdrowotnej, możliwości zatrudnienia, mobilności, energetyki, budownictwa, infrastruktury oraz technologii. Te kluczowe obszary powinny być wspierane przez wykształconych ludzi, społeczności lokalne oraz infrastrukturę ICT.



**Rysunek 1.** Redukcja dostępności w zależności od czasu obliczeń dla trzech wybranych systemów komputerowych w *smart city*

Źródło: Opracowanie własne.

### 3. Prawdopodobieństwo realizacji zadań w terminach

Niech w systemie *smart city* realizowanych jest  $N$  zadań obliczeniowych w cyklu dobowym. Zadanie  $T_n$  rozpoczyna się o godzinie  $\alpha_n$  i powinno się zakończyć o godzinie  $\omega_n$ . Na rysunku 2 przedstawiono przykład realizacji dwóch zadań. Zadanie pierwsze realizuje

<sup>8</sup> Balicki J.: *Wybrane paradygmaty sztucznej inteligencji w informatycznych systemach finansowych*, „Współczesna Gospodarka” 2015, nr 4, s. 12.

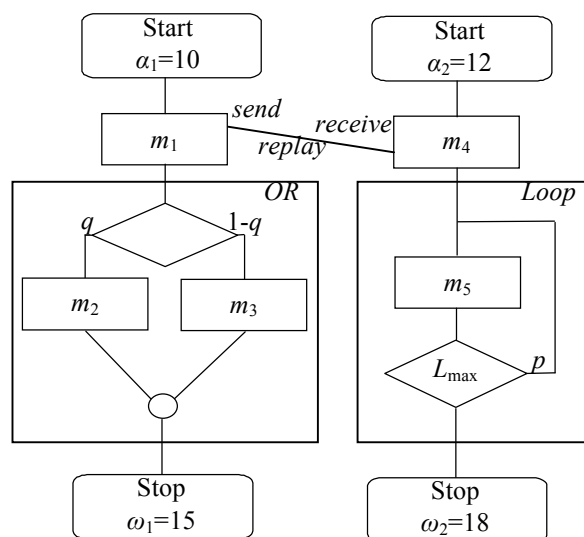
obliczenia wspierające regulację światła na skrzyżowaniach, a zadanie drugie - obliczenia do monitorowania zanieczyszczenia powietrza. W zadaniu  $T_1$  po zrealizowaniu procedury  $m_1$  wykonywana jest operacja  $m_2$  z prawdopodobieństwem  $q$  (poważny wypadek na skrzyżowaniu) w podgrafie decyzyjnym typu *OR* (rys. 2) lub operacja  $m_3$  - z prawdopodobieństwem  $(1-q)$ . Natomiast w zadaniu  $T_2$  procedura  $m_4$  może być wykonana co najwyżej  $L_{\max}$  razy w podgrafie pętli *Loop*, przy czym każda iteracja realizacji procedury odbywa się z prawdopodobieństwem  $p$  (wykrycie zanieczyszczenia przekraczającego bezpieczny poziom przez czujnik).

Łatwo pokazać, że możliwych jest  $2L_{\max}$  instancji diagramu z rysunku 2. Instancja, w której procedura  $m_2$  jest realizowana raz, a procedura  $m_5$  -  $k$  razy, występuje z następującym prawdopodobieństwem:

$$p_i = q(1-p)p^{k-1} \quad (4)$$

Na podstawie decyzji przydziału procedur do komputerów  $(x^m, x^\pi)$  można oszacować czasy zakończenia obliczeń procedur  $C_1, \dots, C_v, \dots, C_V$  zgodnie z przyjętym algorytmem harmonogramowania zadań. Niech  $d_v$  reprezentuje wymagany czas zakończenia  $v$ -tej procedury. Jeśli  $C_v \leq d_v$ , to *deadline* nie jest przekroczony i stan spełnienia tego ograniczenia zapisuje się jako  $\xi(d_v - C_v) = 1$ . Stan ograniczeń dotyczący *deadline*ów procedur uwzględnianych w  $i$ -tej instancji diagramu realizacji zadań ze zbiorem procedur  $M_i$  można zapisać, jak niżej:

$$S_i = \prod_{m_v \in M_i} \xi(d_v - C_v(x)). \quad (5)$$



Rysunek 2. Przykładowy diagram realizacji dwóch zadań w systemie klasy *smart city*

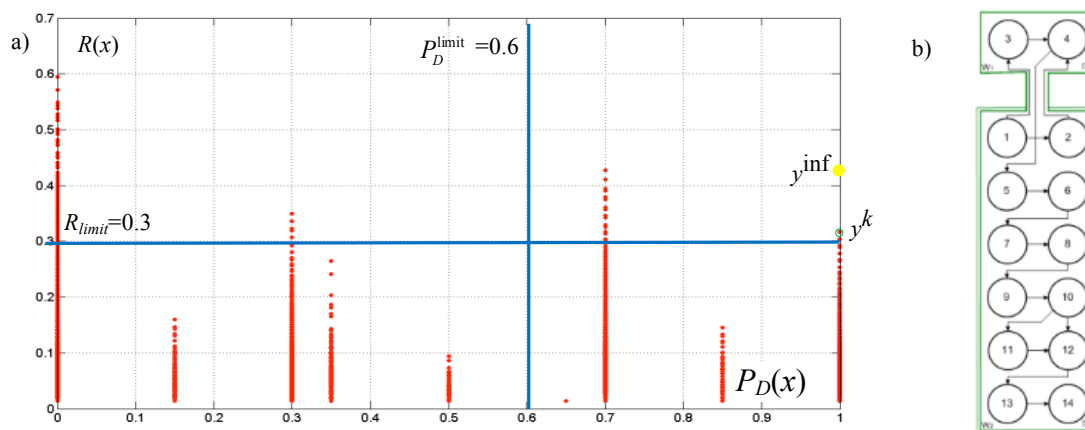
Źródło: Opracowanie własne.

W konsekwencji wyznaczamy prawdopodobieństwo nieprzekroczenia *deadline*ów przez procedury dla wszystkich  $K$  instancji diagramu realizacji zadań w systemie *smart city*, jak niżej:

$$P_D(x) = \sum_{i=1}^K p_i \prod_{m_v \in M_i} \xi(d_v - C_v(x)). \quad (6)$$

Na rysunku 3 zobrazowano wielokryterialną przestrzeń ocen infrastruktur informatycznych klasy *smart city*. Zaznaczono oceny będące parami wartości  $(P_D, R)$  oraz ograniczenia nałożone

na kryteria optymalizacji. Ponadto przedstawiono infrastrukturę kompromisową rozmieszczenia 14 procedur na 2 komputerach, której odpowiada ocena  $y^k$  najbliższa w sensie odległości Euklidesa do punktu idealnego  $y^{inf}$ . Przyjęto, że  $P_D \geq 0,6$ , a  $R \geq 0,3$ . Warto podkreślić, że wartości nałożonych ograniczeń na kryteria można zmieniać, a wpływ na to mają interesariusze: administratorzy systemów, projektanci, urzędnicy, samorząd, czy nawet mieszkańcy. Na podstawie ograniczeń zawężany jest zbiór struktur dopuszczalnych, z którego można wybrać rozwiązania Pareto-optymalne, a w szczególności infrastrukturę kompromisową. Warto jednak podkreślić, że problem jest NP-trudny ze względu na złożoność obliczeniową i do jego rozwiązania wymagane są metaheurystyki.



Rysunek 3. Wielokryterialna przestrzeń ocen infrastruktur informatycznych klasy *smart city*:  
a) oceny z ograniczeniami na kryteria optymalizacji b) infrastruktura kompromisowa

Źródło: Opracowanie własne

#### 4. Algorytmy inteligencji zbiorowej

Do wyboru infrastruktur informatycznych w *smart city* użyteczne są metaheurystyki. Pewne metaheurystyki, takie jak *Particle Swarm Optimization*, *Ant Colony Optimization* i *Artificial Bee Colony* można określić, jako należące do algorytmów inteligencji zbiorowej (ang. *swarm intelligence*). Algorytmy te, chociaż znacząco się różnią, posiadają wspólną cechę, jaką jest działanie poszczególnych agentów bez wcześniej określonego planu oraz bez nadrzędnej jednostki zarządzającej. Ponadto, typowym dla nich jest korzystanie ze zbiorowej wiedzy gromadzonej podczas ewaluacji algorytmu. Każdy agent działa w sposób autonomiczny, ale uwzględnia zbiorczą wiedzę całej populacji. W przypadku PSO mówić możemy o inspiracji zachowaniem ptaków, ACO – mrówek, a ABC – pszczół.

Warto także wspomnieć o algorytmie ewolucji różnicowej. Algorytm ten został przedstawiony przez Pricea i Storma w 1995 roku jako algorytm optymalizacji z niewielką liczbą parametrów. Populacja składa się z określonej liczby wektorów zawierających wariacje rozwiązania problemu. Cechą charakterystyczną tego algorytmu jest metoda krzyżowania, która wykorzystuje trzy wektory wylosowane z populacji rodziców. Wybór najlepszego wektora do populacji potomnej następuje między wektorami z populacji rodziców a wektorami z populacji powstałej przy wykorzystaniu metod krzyżowania i mutacji. W wielu problemach optymalizacji algorytm okazał się lepszy w porównaniu do innych algorytmów ewolucyjnych.<sup>9</sup>

<sup>9</sup> J. Huibin, L. Mingguang. *An Improved Differential Evolution Algorithm for Optimization*. IITA International Conference on Control, Automation and Systems Engineering 2009, p. 659.

Bubal i Lee<sup>10</sup> zaproponowali planowanie połączeń transportowych i komunikacyjnych w taki sposób, aby były one optymalnie dopasowane do zapotrzebowania osób w dużych aglomeracjach miejskich za pomocą algorytmu ewolucji różnicowej. Opracowany algorytm uwzględnia nie tylko zadowolenie pasażerów, ale również koszty ponoszone przez zarządców autobusów, tramwajów oraz pociągów, tak aby połączenia były opłacalne.

## 5. Algorytm roju cząstek

Algorytm *Particle Swarm Optimization* skonstruowali Eberhart i Kennedy<sup>11</sup> w 1995 roku. Inspirację stanowiły obserwacje społecznych zachowań ptaków w stadzie podczas poszukiwania przez nie pokarmu. Jeśli członek stada wyczuje zapach pokarmu, to za pomocą wydawanych dźwięków, informuje innych o swoim położeniu. Wówczas reszta stada stara się zmienić kierunek lotu tak, aby się do niego zbliżyć. Jeśli kolejne ptaki wyczuwają jeszcze silniejszy zapach pożywienia, ponownie informują pozostałe ptaki poprzez jeszcze głośniejsze dźwięki. Wpływają w ten sposób na kierunek lotu reszty stada.

W algorytmie PSO, podstawowym komponentem są cząstki (ang. *particles*). Każda z cząstek porusza się po przestrzeni rozwiązań w sposób niezależny, poszukując najlepszego rozwiązania. Na kierunek poszukiwań każdej z nich wpływ ma najlepsze w danej chwili, globalne rozwiązanie wybierane spośród wszystkich cząsteczek. W algorytmie PSO znaleźć można podobieństwa do algorytmów genetycznych. Uważa się jednak, że PSO jest łatwiejsze do implementacji i wdrożenia ze względu na mniejszą liczbę parametrów koniecznych do dopasowania.

Nikam i Mane<sup>12</sup> zaproponowali wykorzystanie algorytmu PSO do optymalizacji bezprzewodowej sieci czujników (ang. *wireless sensor network*, WSN) łączącej elementy infrastruktury *smart city* w zakresie zarządzania odpadami. Sieci WSN składają się z wielu niezależnych urządzeń monitorujących zjawiska fizyczne, takie jak temperatura, ruch, czy obecność przedmiotu w przestrzeni. Każdy z węzłów bezprzewodowo przesyła dane do centrali, gdzie są przetwarzane. Algorytm PSO wykorzystany został do poprawy skalowalności sieci, optymalizacji zużycia energii oraz odporność na uszkodzenia.

## 6. Algorytm kolonii pszczół

Algorytm ABC został opracowany w 2005 roku przez Karabogę<sup>13</sup>. Proces komunikacji pszczół zachodzi w tańcu (ang. *waggle dance*) i polega na wykonywaniu odpowiednich ruchów w celu przekazania informacji o odległości do pokarmu, jego jakości oraz azymucie lotu w odniesieniu do kierunku słońca.

W algorytmie rozważa się trzy rodzaje agentów reaktywnych: aktywne lub oczekujące sztuczne pszczoły (agenty poszukujące rozwiązania), a także sztuczne źródła pokarmu (wyznaczone rozwiązania dla problemu optymalizacyjnego). Zakłada się wymianę informacji między sztucznymi pszczołami, mającą na celu eksplorowanie „sąsiadujących” do wcześniej wyznaczonych rozwiązań, dających większe nadzieje oraz porzucanie gorszych alternatyw. Algorytm

<sup>10</sup> A. T. Bubal, L. S. Lee, *Differential Evolution for Urban Transit Routing Problem*, Journal of Computer and Communications, 2016, 4, 11-25, s. 12.

<sup>11</sup> J. Kennedy, R. Eberhart, *Particle Swarm Optimization*, Proceedings of IEEE International Conference on Neural Networks IV, 1995, p. 1942.

<sup>12</sup> S. Nikam and P. Mane, *Swarm Intelligent WSN for Smart City*, Proceedings of the International Conference on Data Engineering and Communication Technology, 2016, p. 691.

<sup>13</sup> D. Karaboga, *An Idea Based On Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005, p. 223.



ABC zastosowano w problemach optymalizacji kombinatorycznej<sup>14</sup> czy też optymalizacji wielokryterialnej<sup>15</sup>.

Wykorzystanie algorytmu ABC jako elementu systemu zarządzającego ruchem *Personal Rapid Transit* (PTR) zaproponowali *Fatnassi, Chebbi i Siala*<sup>16</sup>. Autorzy zaproponowali algorytm, którego celem jest minimalizacja liczby przejazdów pustych pojazdów. Do głównych zalet algorytmu zalicza się czas działania w stosunku do jakości wyznaczonych rozwiązań. PTR opiera się na koncepcji zindywidualizowanego, publicznego transportu sterowanego automatycznie. Stosuje się małe, kilkusobowe pojazdy poruszające się na przygotowanych trasach. O trasie przejazdu decyduje komputer, przez co wybrana trasa jest optymalna pod względem czasu podróży, jak i zużycia energii. Jednocześnie transport jest bezpieczniejszy, gdyż nie ma możliwości wystąpienia kolizji pojazdów. PTR wdrożono na lotnisku *Heathrow* w *Wielkiej Brytanii*, zastępując częściowo transport autobusowy. Wprowadzenie PTR pozwoliło na zredukowanie czasu podróży pasażerów z kilkudziesięciu do kilku minut.



Rysunek 4. Pojazdy PTR na lotnisku *Heathrow*

Źródło: <http://www.ultraglobalprt.com> (11.08.2017)

## 7. Algorytm kolonii mrówek

*Dorigo*<sup>17</sup> zaprezentował algorytm *Ant Colony Optimization* (ACO) w 1992 do wyznaczenia optymalnej ścieżki w grafie. Algorytm stanowi podstawę dla istotnej liczby rozszerzeń takich jak *Ranked-based ant system* czy *Recursive ant colony optimization*. Prace nad ACO zainspirowane były obserwacją zachowań kolonii mrówek. Z biologicznego punktu widzenia zachowanie mrówek świadczy o tym, że przetrwanie kolonii jest celem nadrzędnym w odniesieniu do przetrwania pojedynczej jednostki. Inspirację stanowił proces poszukiwania pokarmu przez mrówki. W początkowej fazie poszukiwań mrówki poruszają się w sposób losowy wokół gniazda. Każda z mrówek pozostawia za sobą ślad feromonów w czasie poszukiwania

<sup>14</sup> Q. Pan, M. Fatih Tasgetiren, P. Suganthan and T. Chua, *A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem*, Elsevier, Information Sciences, vol. 181, no. 12, 2011, p. 2455.

<sup>15</sup> S.N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, S. Gopalakrishnan, *Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures*, Elsevier, Applied Soft Computing, Netherlands, 2010, p. 12.

<sup>16</sup> E. Fatnassi, O. Chebbi, J. Siala, *Bee colony algorithm for the routing of guided automated battery-operated electric vehicles in personal rapid transit systems*, 2014 IEEE Congress on Evolutionary Computation (CEC), 2014

<sup>17</sup> M. Dorigo, L.M. Gambardella, *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Transactions On Evolutionary Computation, Vol. 1, No. 1, 1997, pp. 53–66.

pokarmu. Ślad ten jest wyczuwalny przez inne osobniki i stanowi niejako drogowskaz do źródła pokarmu. Mrówki wybierają bowiem tą ścieżkę, na której ślad feromonów jest najsilniejszy, przyczyniając się jednocześnie do jego ciągłego wzmacniania. Na ścieżce, która wykorzystywana jest rzadziej bądź wcale, ślad zaczyna stopniowo wietrzeć, by ostatecznie całkowicie zniknąć. W ten sposób mrówki stopniowo wybierają krótszą drogę do pożywienia.

W algorytmie ACO wyróżnić można trzy główne kroki. Pierwszym z nich jest generowanie rozwiązań dla każdej sztucznej mrówki. Jest to wygenerowanie ścieżki w zadanym grafie. Następnym krokiem jest uaktualnienie sztucznych śladów feromonowych, czyli symulacja wyparowywania oraz wzmacniania śladów feromonowych na podstawie wygenerowanych poprzednio rozwiązań. Krok ten jest związany z dziedziną problemową, w której algorytm jest wykorzystywany. W ostatnim etapie wykonywane są zazwyczaj akcje dodatkowe oraz takie, które operują na wiedzy globalnej, np. dodatkowe wzmocnienie danego śladu, który okazał się być najbardziej korzystnym.

*Bello i Zeadally*<sup>18</sup> zaproponowali wykorzystanie algorytmu ACO w celu usprawnienia transmisji danych pomiędzy urządzeniami IoT. Sieci IoT może cechować duża różnorodność architektur, jak i znaczna dynamika zachodzących zmian, np. zmiana topologii sieci w przypadku mobilnych urządzeń. ACO może funkcjonować jako istotny element wspomagający routing w takiej sieci. Dzięki wykorzystaniu algorytmu zagadnienia takie, jak samo-organizacja, samo-konfiguracja sieci, częsta zmiana położenia urządzeń w przestrzeni czy odporność na uszkodzenia mogą zostać rozwiązane.

## Zakończenie

W pracy omówiono metody zwiększania dostępności i efektywności informatycznej infrastruktury w inteligentnym mieście. Sformułowano dwa kryteria do oceny rozmieszczenia kluczowych zasobów w systemie *smart city*. Zobrazowano proces wyznaczania rozwiązań kompromisowych spośród rozwiązań Pareto-optimalnych. Omówiono metaheurystyki inteligencji zbiorowej, w tym roju cząstek, kolonii mrówek, roju pszczół oraz ewolucji różnicowej, za pomocą których można wyznaczać kompromisową infrastrukturę *smart city*.

Warto podkreślić, że algorytm roju pszczół zastosowano do optymalizacji systemu zarządzającego ruchem *Personal Rapid Transit* na lotnisku *Heathrow*. Ponadto planowanie połączeń transportowych i komunikacyjnych za pomocą algorytmu ewolucji różnicowej uwzględnia nie tylko zadowolenie pasażerów, ale również koszty ponoszone przez zarządców autobusów, tramwajów oraz pociągów. Natomiast algorytmu roju cząstek zastosowano do optymalizacji bezprzewodowej sieci czujników łączącej elementy infrastruktury *smart city* w zakresie zarządzania odpadami. Algorytm kolonii mrówek stosowany jest do wspomagania routingu w *Inter-necie Rzeczy*, a także do samoorganizacji i samokonfiguracji IoT.

Interesującym kierunkiem dalszych badań jest skonstruowanie modelu i metody obejmującej szerszą perspektywę optymalizacji zasobów w *smart city*.

## Literatura

1. Balicka H., Balicki J., Korlub W., Paluszak J., Zadroga M., *Superkomputery do wspomaganie procesów gospodarczych ze szczególnym uwzględnieniem sektora bankowego*. "Współczesna Gospodarka" 2014, nr 4

<sup>18</sup> O. Bello and S. Zeadally, *Intelligent Device-to-Device Communication in the Internet of Things*, IEEE Systems Journal, vol. 10, no. 3, 2016, pp. 1172-1182.

2. Balicki J., Dryja P., Korłub W., Przybyłek P., Tyszka M., Zadroga M., Zakidalski M., *Metody neuronowe do prognozowania finansowego*, "Współczesna Gospodarka" 2016, nr 2
3. Balicki J., Korłub W., *Uczenie maszynowe do samoorganizacji systemów rozproszonych w zastosowaniach gospodarczych*, „Współczesna Gospodarka” 2017, nr 1
4. Balicki J., Przybyłek P., Zadroga M., Zakidalski M., *Sztuczne sieci neuronowe oraz metoda wektorów wspierających w bankowych systemach informatycznych*, "Współczesna Gospodarka" 2013, nr 4
5. Balicki J.: *Wybrane paradygmaty sztucznej inteligencji w informatycznych systemach finansowych*, "Współczesna Gospodarka" 2015, nr 4
6. Bello O., Zeadally S., *Intelligent Device-to-Device Communication in the Internet of Things*, IEEE Systems Journal, vol. 10, no. 3, 2016, pp. 1172-1182
7. Bubal A. T., Lee L. S., *Differential Evolution for Urban Transit Routing Problem*, Journal of Computer and Communications, 2016, no. 4, pp. 12 – 14.
8. Dorigo M, Gambardella LM., *Ant colony system: A cooperative learning approach to the traveling salesman problem*. IEEE Trans Evolutionary Comput 1997, pp. 53–66.
9. Fatnassi E., Chebbi, J. Siala, *Bee colony algorithm for the routing of guided automated battery-operated electric vehicles in personal rapid transit systems*, 2014 IEEE Congress on Evolutionary Computation (CEC), 2014
10. Huibin J., Mingguang L., *An Improved Differential Evolution Alogorithm for Optimization*. IITA International Conference on Control, Automation and Systems Engineering 2009, pp. 659-662
11. Kanter R., Litow S., *Informed and interconnected: A manifesto for smarter cities*. Harvard Business School, General Management Unit, Working Paper, 2009
12. Karaboga D.: *An Idea Based On Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Engineering Department, 2005
13. Kennedy, J.; Eberhart, R.: *Particle Swarm Optimization*, Proceedings of IEEE International Conference on Neural Networks, 1995, pp. 1942–1948
14. Nikam S., Mane P.: *Swarm Intelligent WSN for Smart City*, Proceedings of the International Conference on Data Engineering and Communication Technology, 2016, pp. 691-700
15. Omkar S.N., Senthilnath J., Khandelwal R., Naik G. N., Gopalakrishnan S.: *Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures*, Elsevier, Applied Soft Computing, Netherlands, 2010
16. Pan Q., Tasgetiren M. F., Suganthan P., Chua T.: *A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem*, Elsevier, Information Sciences, vol. 181, no. 12, 2011, pp. 2455-2468
17. *Smart Dublin Project*, <http://smartdublin.ie/smartstories/unfolding-news-app> (01.07.2017)
18. Winiarski J., *Ryzyko w działalności gospodarczej przedsiębiorstw*. Wydawnictwo Uniwersytet Gdański Instytut Transportu i Handlu Morskiego, Gdańsk 2014



## METHODS FOR IMPROVING AVAILABILITY AND EFFICIENCY OF COMPUTER INFRASTRUCTURE IN SMART CITIES

### Summary

This paper discusses methods for increasing the availability and efficiency of information infrastructure in smart cities. Two criteria have been formulated to assign some key resources in smart city system. The process of finding some compromise solutions from Pareto-optimal solutions has been illustrated. Metaheuristics of collective intelligence, including particle swarm optimization PSO, ant colony optimization ACO, algorithm of bee colony ABC, and differential evolution DE have been described due to smart city infrastructure improving. Other application of above metaheuristics in smart city have been also presented.

**Keywords:** smart city, availability of computer infrastructure

dr hab. inż. Jerzy Balicki, prof. PW,  
Politechnika Warszawska, Wydział Matematyki i Nauk Informatycznych  
Zakład Strukturalnych Metod Przetwarzania Wiedzy  
ul. Koszykowa 75  
00-662 Warszawa  
J.Balicki@mini.pw.edu.pl

mgr inż. Piotr Dryja  
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki  
Katedra Architektury Systemów Komputerowych  
ul. G. Narutowicza 11/12  
80-233 Gdańsk  
piotrdryja83@gmail.com

dr inż. Waldemar Korlub  
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki  
Katedra Architektury Systemów Komputerowych  
ul. G. Narutowicza 11/12  
80-233 Gdańsk  
waldemar.korlub@pg.gda.pl

mgr inż. Maciej Tyszka  
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki  
Katedra Architektury Systemów Komputerowych  
ul. G. Narutowicza 11/12  
80-233 Gdańsk  
tyszka.maciej@gmail.com

