

Koncepcja środowiska symulacyjnego do oceny samoorganizacji trasowania w sieci sensorycznej

Streszczenie

Techniki Internetu Rzeczy (IoT – Internet of Things) oraz komunikacji bezpośredniej Maszyna do Maszyny (M2M - Machine to Machine) coraz mocniej wpływają na strukturę i funkcjonalność systemów sterowania stosowanych w maszynach, kształtując przy tym ideę Przemysłu 4.0 (Industry 4.0). Systemy sterowania zgodne z IoT wykorzystują sieci komunikacyjne, często o dużym stopniu komplikacji, łącząc poszczególne podzespoły, moduły, elementy wykonawcze i sensory. W artykule przedstawiono zagadnienie symulacji samoorganizacji ścieżek komunikacyjnych (trasowanie, routing) w złożonej sieci sensorycznej monitorującej działanie krążników przenośnika taśmowego. Poszczególne sensory tworzące sieć są niezależne i wyposażone w elektroniczny układ pomiarowy oraz transmisyjny MTU (Measuring and Transmitting Unit). W celu utworzenia i optymalizacji ścieżek transmisyjnych, w proponowanej strukturze komunikacyjnej, zaproponowano algorytm klasy SA (Swarm Algorithm) bazujący na zachowaniu roju.

Summary

Assumptions of an IoT (Internet of Things) and direct communication M2M (Machine to Machine) got strong influence on the structure and functionality of the control systems of machines, shaping at once an idea of the Industry 4.0 (Industry 4.0). All control systems, in accordance with the IoT, use communication networks, often with a high degree of complexity, combining the various components, modules, actuators and sensors. The paper presents the simulation problem of self-organization communication paths in a complex network of sensory monitoring of operation of the conveyor belt rollers, in which each sensor is equipped with an independent, electronic measuring and transmission unit (MTU). In order to create and optimize the communication structure an algorithm of class SA (Swarm Algorithm), based on the behavior of the swarm, was proposed.

Słowa kluczowe: trasowanie, samoorganizacja, algorytm roju, sieć sensoryczna

Keywords: routing, self-organization, swarm algorithm, sensors network

1. Wstęp

Możliwość zastosowania opisywanej w artykule klasy algorytmów może być powiązana z problemami dotyczącymi sterowania przebiegiem procesów technologicznych i maszyn przy minimalnym udziale ludzi. Koordynacja działań [11], ochrona mienia [15], określanie celów bieżących [4], hierarchizacja funkcji w zrobotyzowanych systemach wydobywczych jest obecnie koniecznością. Prowadzone prace mają także na celu automatyzację działania systemów służących wydobywaniu [1, 2, 6, 7, 8, 9, 13, 14]. W wyniku tych działań otrzymywane są znaczne ilości danych, które utrudniają prowadzenie przez ludzi złożonego procesu wnioskowania. Nawet jeśli wnioskowanie jest możliwe, często jest nieefektywne. Stąd metody i techniki sztucznej inteligencji, bazujące na wiedzy i samoorganizacji (także adaptacji) powinny być coraz szerzej wdrażane w celu dalszego rozwoju technologii stosowanych w górnictwie.

Jednym z ważniejszych zagadnień, związanych z bezpieczeństwem pracy w kopalniach węgla, jest

monitoring lokalizacji ludzi i środków transportu oraz monitoring środowiskowy. Istotnym problemem bezpieczeństwa w górnictwie jest zagrożenie pożarowe na przenośnikach taśmowych do transportu urobku. Krążniki przenośników taśmowych nie są monitorowane w sposób ciągły. Każdy z nich, w sytuacji zatarcia i przegrzania, może stanowić źródło zapłonu pyłu węglowego. Wdrożenie systemu monitoringu krążników, z niezależnym źródłem zasilania i wewnętrzną diagnostyką, wielopoziomą redundancją połączeń komunikacyjnych oraz samoorganizacją ścieżek komunikacyjnych, może poprawić poziom bezpieczeństwa użytkowego i pozwolić na podjęcie działań zapobiegających pożarom. System, dzięki liczbie zainstalowanych w krążnikach jednostek monitorujących MTU (Measuring and Transmitting Unit), transmitujących dane traktowane jako elementy roju (agenci), nie będzie wymagał ustanawiania zaplanowanej redundancji połączeń w zakresie komunikacji M2M. Sam utworzy sieć komunikacyjną wewnętrzną multiredundantną, na kilku poziomach funkcjonowania (sprzętowym, programowym, algorytmicznym).

Koncepcję samoorganizującej sieci komunikacyjnej, w oparciu o którą powstanie technika będąca przedmiotem implementacji w strukturze systemu monitoringu, o cechach samoorganizującego roju, przedstawiono w pracy [12].

2. Samoorganizacja systemu z wykorzystaniem roju cząstek

Algorytm optymalizacji rojem cząstek (PSO) [5], nazywany także algorytmem ptasim, opracowany został przez Kennedy'ego i Eberharta w 1995 roku [3, 11, 18]. Algorytm oparto na obserwacji zachowania wybranej populacji (roju, stada), z uwzględnieniem możliwości komunikowania się między poszczególnymi osobnikami oraz dzielenia informacjami. Każdy osobnik należący do danej populacji traktowany jest jako cząstka. Cząstki przemieszczają się do nowych położeń, poszukując optimum określonej funkcji (np. najkrótszej drogi). Podobnie jak stado ptaków, rój podąża za przywódcą (najlepszym rozwiązaniem) przyspieszając i zmieniając kierunek (przeszukując przestrzeń stanów), do momentu aż lepsze rozwiązanie zostanie znalezione. Przywódcą roju zostaje osobnik o najlepszym dotychczas znanym położeniu. Osobniki podejmują decyzje na podstawie oceny najbliższego otoczenia, jednak dzięki przesyłaniu informacji stado zachowuje swoją dynamikę. Każda cząstka (cząstką jest ramka danych pojawiająca się w sieci komunikacyjnej) posiada określone:

- położenie,
- prędkość,
- zwrot,

znane w odniesieniu do swoich sąsiadów. Każda cząstka pamięta również najlepsze położenia, jakie udało jej się osiągnąć. Zarówno położenie oraz prędkość i-tej cząstki w d-wymiarowej przestrzeni można przedstawić jako wektory: $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ oraz $v_i = [v_{i1}, v_{i2}, \dots, v_{id}]$. Każda cząstka ma swoją własną najlepszą pozycję $p_i = [p_{i1}, p_{i2}, \dots, p_{id}]$, odpowiadającą najlepszej uzyskanej dotychczas wartości funkcji celu, natomiast najlepsza pozycja cząstki-przywódcy w całym roju określona jest jako $p_d = [p_{d1}, p_{d2}, \dots, p_{dd}]$.

Podstawowy algorytm optymalizacji rojem cząstek przebiega w następujący sposób:

- inicjalizacja losowej pozycji cząstek i ich prędkości początkowych,
- ocena położenia cząstek za pomocą funkcji dopasowania,
- porównanie zachowania każdej cząstki z jej najlepszym (do tej pory) zachowaniem,
- wybór przywódcy stada,
- uaktualnienie prędkości każdej cząstki w każdej iteracji k opisaną równaniem (1):

$$v_i(k) = \omega v_i(k-1) + c_1 r_1 [p_i(k-1) - x_i(k-1)] + c_2 r_2 [p_d(k-1) - x_i(k-1)] \quad (1)$$

gdzie:

ω - współczynnik inercji ruchu cząstki,

v_i - wektor prędkości i-tej cząstki,

x_i - wektor położenia i-tej cząstki,

p_i - najlepsze położenie i-tej cząstki,

p_d - najlepsze położenie dowolnej cząstki,

c_1 - stała dodatnia, tzw. wskaźnik samooceny,

c_2 - stała dodatnia, wskaźnik społecznościowy (zaufanie położeniu sąsiadów),

r_1, r_2 - losowe liczby o rozkładzie równomiernym w przedziale [0, 1],

k - kolejny krok iteracji

f) uaktualnienie położenia każdej cząstki opisane równaniem (2):

$$x_i(k) = x_i(k-1) + v_i(k) \quad (2)$$

gdzie:

k - kolejny krok iteracji,

v_i - wektor prędkości i-tej cząstki,

x_i - wektor położenia i-tej cząstki.

Kroki b-f wykonywane są do momentu spełnienia określonego kryterium stopu (warunku zatrzymania algorytmu). Współczynnik inercji ruchu cząstki określa, w jakim stopniu aktualna prędkość cząstki jest uzależniona od wartości poprzedniej (duża wartość ukierunkowuje cząstki na globalne przeszukiwania przestrzeni stanów). Z kolei wskaźnik samooceny określa, jak bardzo dana cząstka ufa kierunkowi prowadzącemu do swojego najlepszego położenia. Dobór współczynników i parametrów algorytmu ma krytyczny wpływ na jego zbieżność.

3. Środowisko symulacyjne

Środowisko symulacyjne, przygotowane w formie programu komputerowego, niezbędne jest do weryfikacji i oceny algorytmów trasujących ścieżki komunikacyjnie pomiędzy węzłami sieci (MTU), zgodnie z zaproponowaną koncepcją [12], bazującą na zachowaniu roju. Środowisko symulacyjne umożliwia wyznaczenie:

- odległości określającej przynależność określonej cząstki (ramki danych) do grupy sąsiadujących ze sobą MTU (wykrycie sąsiedztwa agentów),
- wirtualnego kąta widzenia w odniesieniu do ramki pochodzącej z określonego MTU,

- wirtualnych kątów wektora prędkości ramki w odniesieniu do ramek z sąsiadujących MTU, celem określenia dopasowania bieżącej ścieżki komunikacyjnej do optymalnej,
- dopasowania prędkości ramek z sąsiadujących MTU, celem optymalizacji wydajności transmisji w aspekcie rzeczywistych ograniczeń przesyłu w węzłach sieci,
- priorytetów ścieżek transmisyjnych, celem wyboru optymalnych tras przepływu danych w aspekcie zadanych kryteriów (np. najkrótszej drogi).

Rdzeń środowiska opiera się na implementacji pseudokodu bazującego na algorytmie opisanym w punkcie 2:

```

PSO() {
  swarm.initializeRandomlyParticlesLocationAndVelocity();
  for i from 1 to maxIterations {
    swarm.updateBestLocation();
    for each MTU in swarm {
      Dframe.updateVelocity();
      Dframe.updateLocation();
    }
  }
}

Dframe {
  updateVelocity() {
    for (i from 1 to dimensions) {
      this.v[i] =
        random.uniform(0,g)*(swarm.
        best[i] - this.x[i]) +
        random.uniform(0,l)*(this.best
        [i] - this.x[i]) +
        random.uniform(0,r)*(this.x[i] -
        swarm.random().x[i]) +
        a * this.v[i] +
        y * random.normal(0,1);
    }
  }
  updateLocation() {
    for (i from 1 to dimensions) {
      this.x[i] = this.x[i] + this.v[i];
    }
    if (f(this.best) > f(this.x)) {
      this.best = this.x;
    }
  }
}

```

Powyższy pseudokod opisuje podstawowe kroki charakteryzujące proces optymalizacji z wykorzystaniem PSO. Aby poprawić jego efektywność wprowadzono do kodu roju, bazując na [10], cząstki wynalazcy (*Inventors*) i cząstki odkrywcy (*Explorers*), które poszukują nowych nieeksploatowanych aktualnie rozwiązań:

```

InventorsDframe: Dframe {
  updateVelocity() {
    for (i from 1 to dimensions) {
      this.v[i] =
        random.uniform(0,r)*(swarm.
        best[i] - this.x[i]) +
        random.uniform(0,l)*(this.best
        [i] - this.x[i]) +
        random.uniform(0,g)*(this.x[i]
        - swarm.random.x[i]) +
        a * this.v[i] +
        y * random.normal(0,1);
    }
  }
}

```

Cząsteczki odkrywcy początkowo zachowują się tak samo jak cząsteczki wynalazcy. Jednak cząsteczka odkrywca, w odróżnieniu od wynalazcy, może:

- zmienić strategię działania i zacząć poszukiwać maksimum określonej funkcji celu,
- poszukując maksimum, ulec "zmęczeniu" i zacząć poszukiwać minimum określonej funkcji celu.

Działanie cząsteczek odkrywców można opisać następującym pseudokodem:

```

ExplorersParticle: InventorsParticle {
  updateLocation() {
    for (i from 1 to dimensions) {
      this.x[i] = this.x[i] + this.v[i];
    }
    if (f(this.best[t]) > f(this.x) &&
    this.isSerchingForMinimum) {
      this.best[t] = this.x;
    }
    if (f(this.best[t]) < f(this.x) &&
    !this.isSerchingForMinimum) {
      this.best[t] = this.x;
    }
    if (!this.isSerchingForMinimum) {
      if (fatigue++ > 2 *
      this.samples_to_switch) {
        fatigue = 0;
        this.isSerchingForMinimum = true;
        this.samples_to_switch *= 2;
      }
    }
    if (!this.best[t-samples_to_switch] - this.best[t]
    < precision_to_switch) {
      fatigue = 0;
      this.isSerchingForMinimum =
      !this.isSerchingForMinimum;
    }
  }
}

```

Przyjęto następujące założenia opisujące i odnoszące się do działania programowego środowiska symulacyjnego:

- 1) Kryteria stopu (zatrzymania symulacji):
Wybrano 2 warunki, które prowadzą do zakończenia algorytmu symulacji. Przy wyborze kierowano się ich niskim zapotrzebowaniem na moc obliczeniową komputera:
 - Kryterium 1: „Powolny rój” – należy porównywać prędkości wszystkich cząsteczek z ustaloną wartością progową. W przypadku, gdy cząsteczki w roju poruszają się poniżej dobranej eksperymentalnie wartości program kończy działanie.
 - Kryterium 2: „Brak poprawy od długiego czasu” – jeżeli nie następuje poprawa wyniku optymalizacji ścieżki komunikacyjnej (np. liczby skoków ramek komunikacyjnych przez węzły sieci), w zakresie ustalonej eksperymentalnie liczby iteracji, program kończy działanie.
- 2) Złożoność obliczeniowa
Złożoność obliczeniowa jest zależna od liczby cząsteczek roju oraz kryterium stopu. Z tego powodu jest trudna do oszacowania, gdyż liczba iteracji głównej pętli algorytmu symulacji może być zmienna. Przyjmując:
 - i - liczba iteracji głównej pętli algorytmu,
 - n - liczba cząsteczek,można określić $O(in)$, przy czym liczba iteracji będzie zazwyczaj znacznie większa od liczby MTU.
- 3) Wejście/wyjście programu
Dane wejściowe mogą zostać utworzone na dwa sposoby, wczytane z pliku lub wygenerowane losowo. Na wyjściu otrzymuje się długość (określoną w skokach) najkrótszej ścieżki komunikacyjnej, osiągniętą wydajność (prędkość transmisji), czas wykonywania obliczeń oraz liczbę punktów wejściowych. Opcjonalnie możliwe jest wypisanie macierzy odległości pomiędzy MTU oraz porównanie wyniku z wynikami działania innych algorytmów samoorganizacji struktur transmisji danych.
- 4) Parametry
W odniesieniu do algorytmu symulacji możliwe jest ustalenie liczby MTU i maksymalnej liczby iteracji nieznajdujących lepszego rozwiązania funkcji celu.
- 5) Struktura danych
Do reprezentacji grafu zawierającego długości i wydajności ścieżek transmisyjnych wyznaczonych pomiędzy MTU, ze względu na to, że jest to graf pełny (możliwe jest połączenie dowolnego punktu z innym), wybrano macierzową reprezentację grafu.

- 6) Ocena wyników symulacji
Program umożliwia dwie kategorie testów – jakościowych oraz ilościowych (wydajnościowych). Z tego powodu konieczne było umożliwienie implementacji innych algorytmów samoorganizacji ścieżek komunikacyjnych, z którymi porównywane są uzyskane wyniki. Porównywana jest długość najkrótszej znalezionej ścieżki oraz czas poszukiwań w odniesieniu do różnych grafów.

4. Zastosowanie PSO do optymalizacji trasowania wielowymiarowego modelu systemu komunikacyjnego

Samoorganizacja systemu komunikacyjnego stanowi główny problem wielowymiarowej optymalizacji [16, 17]. Polega on na utworzeniu ścieżek komunikacyjnych o jak najmniejszej długości, nie przekraczając przy tym ich ograniczonej przepływności. Może być rozważany jako problem alokacji m zasobów do n obiektów. Każdy zasób ma bowiem określony budżet M_i , a obiekt przynosi zysk p_j oraz zużywa w_{ij} zasobu i . Maksymalizowana funkcja celu (3) zdefiniowana jest następująco:

$$f(x) = \sum_{j=1}^n p_j x_j \quad (3)$$

przy ograniczeniach:

$$\sum_{j=1}^n w_{ij} x_j \leq M_i \quad i=1, \dots, m$$

$$x_j \in \{0, 1\}, \quad p_j > 0, \quad w_{ij} \geq 0, \quad M_i \geq 0$$

gdzie:

w_{ij} – część przepustowości i -tej ścieżki, zużywana przez MTU,

M_i – maksymalna przepływność i -tej ścieżki komunikacyjnej,

p_j – liczba przesłanych danych określoną ścieżką komunikacyjną przez MTU,

m – liczba ścieżek komunikacyjnych w sieci,

n – liczba MTU

5. Podsumowanie

Algorytm optymalizacji rojem cząstek może być stosowany do rozwiązania wielu celów optymalizacji. Jednym z nich jest trasowanie optymalnych ścieżek komunikacyjnych w sieci sensorycznej. Uzyskane, początkowe wyniki symulacji, prowadzone w odniesieniu do wybranych instancji testowych, wskazują na duży potencjał algorytmu PSO. Dzięki wymianie informacji między cząstkami roju w trakcie

wykonywania algorytmu, a tym samym ograniczeniu liczby wywołań funkcji dopasowania, metodę można stosować do rozwiązywania wielu problemów wymagających dużej mocy obliczeniowych. Interesującym obszarem mogą być zagadnienia związane z wprowadzeniem nowych formuł aktualizacji położenia i prędkości cząstek. Opracowana metoda samoorganizacji i środowisko symulacyjne pozwalają na zaimplementowanie nowoczesnej i wydajnej techniki monitoringu prowadzonego w wyrobiskach podziemnych, szczególnie w celu:

- zabezpieczenia ppoż. przenośników (krążniki i bębny napędowe z zainstalowanymi MTU mogą być traktowane jak elementy sieci o strukturze kraty),
- robotyzacji i automatyzacji wydobywania surowców w skrajnie trudnych warunkach, w których praca ludzi może być obciążona bardzo dużym ryzykiem i narażeniem ich zdrowia i życia,
- prowadzenia akcji ratowniczych z wykorzystaniem mikrorobotów do poszukiwania uszkodzonych, przekazywania energii i komunikacji z nimi,
- monitoringu atmosfery,
- monitoringu kondycji i ruchu ludzi,
- monitoringu położenia i planowania tras środków transportu,
- inteligentnej komunikacji.

W toku dalszych prac zaplanowano testy porównawcze z wykorzystaniem przedstawionego środowiska symulacyjnego, mające na celu doskonalenie algorytmu trasowania i potwierdzenie przydatności opracowanej metody do ww. zastosowań.

Literatura

1. Bartoszek S.: Metoda pozycjonowania górniczych maszyn mobilnych w wyrobiskach korytarzowych. W: Innowacyjne techniki i technologie dla górnictwa. Bezpieczeństwo – Efektywność – niezawodność. KOMTECH 2012, Instytut Techniki Górniczej KOMAG, Gliwice 2012 s. 387-399.
2. Bartoszek S., Jagoda J., Jura J., Latos M.: Systemy wbudowane w zespołach sterowania, diagnostyki oraz wizualizacji dla górnictwa. W: Innowacyjne techniki i technologie dla górnictwa. Bezpieczeństwo – Efektywność – niezawodność. KOMTECH 2014, Instytut Techniki Górniczej KOMAG, Gliwice 2014, ISBN 978-83-60708-83-5
3. Eberhart R., Shi Y., Kennedy J.: Swarm Intelligence. Morgan Kaufman, San Francisco, 2001.
4. Gong T., Tuson A.L.: Particle swarm optimization for quadratic assignment problems - a formal analysis approach. *International Journal of Computational Intelligence Research*, 4, 2008, 177185.
5. Kennedy J., Eberhart R.: Particle Swarm Optimization. *Materiały IEEE International Conference on Neural Networks*, 4, 1995, 1942-1948.
6. Jonak J., Prostański D., Jasiulek D., Rogala-Rojek J., Puchała B.: Koncepcja adaptacyjnego układu sterowania w kombajnach chodnikowych REMAG SA. *Problemy bezpieczeństwa w budowie i eksploatacji maszyn i urządzeń górnictwa podziemnego*. Monografia pod redakcją Krzysztofa Krauze, Centrum Badań i Dozoru Górnictwa Podziemnego sp. z o.o., Łęczyny 2010 s. 115-123.
7. Jonak J., Rogala-Rojek J.: System doradczy wspomagający operatora kombajnu chodnikowego. *Prace Naukowe - Monografie KOMAG nr 38*, Instytut Techniki Górniczej KOMAG, Gliwice 2012.
8. Latos M., Bartoszek S., Rogala-Rojek J.: Diagnostics of underground mining machinery. *Proceedings of the 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, 2014, s. 782-787.
9. Latos M., Stankiewicz K.: Studies on the effectiveness of noise protection for an enclosed industrial area using global active noise reduction systems. *Low Frequency Noise, Vibration and Active Control Journal*, 2015, vol. 34, nr 1, s. 9-20 2.
10. Lian Z., Gu X., Jiao B., A similar particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan. *Applied Mathematics and Computation*, 175, 2006, 773785.
11. Sha D.Y., Hsu Ch.Y., A new particle swarm optimization for the open shop scheduling problem. *Computers & Operations Research*, 35, 2008, 3243-3261.
12. Stankiewicz K.: Koncepcja metody samoorganizacji złożonego systemu komunikacyjnego do zastosowań w górnictwie, W: Innowacyjne techniki i technologie dla górnictwa. Bezpieczeństwo – Efektywność – niezawodność KOMTECH 2012, Instytut Techniki Górniczej KOMAG, Gliwice 2012 s. 329-338

-
13. Świder J., Woszczyński M.: Zastosowanie układu rekuperacji energii w silniku spalinowym maszyny górniczej. Prace Naukowe - Monografie KOMAG nr 43, Instytut Techniki Górniczej KOMAG, Gliwice 2014 s. 1-106; 6,69 ark. wyd., ISBN 978-83-60708-82-8
 14. Świder J., Woszczyński M.: Use of the System for Energy Recuperation and Control in Diesel Machines. Machine Dynamics Research 2014, Vol. 38, No 1
 15. Tłaga J., Tłaga W.: Samoorganizujący się system roju - nowe podejście w projektowaniu systemów zabezpieczeń (cz. 2), Ochrona Mienia i Informacji, nr 6/2007, str. 38-42
 16. Tasgetiren M.F., LiangY.C., Sevkli M., Gencyilmaz G.: A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing. European Journal of Operational Research, 177, 2007, 1930-1947.
 17. Wang L., Wang X., Fu J., Zhen L.: A novel probability binary particle swarm optimization algorithm and its application. Journal of Software, 3, 2008, 28-35.
 18. Zhu Q., Qian L., Li Y., Zhu S.: An improved particle swarm optimization algorithm for vehicle routing problem with time windows. Materiały IEEE Congress on Evolutionary Computation, 2006, 1386-1390.

Artykuł wpłynął do redakcji w maju 2015 r.