

ALGORYTM WIELU KOLONI MRÓWEK DLA OPTYMALNEGO DOPASOWANIA W WAŻONYCH GRAFACH DWUDZIELNYCH

STRESZCZENIE

Praca prezentuje algorytm wykorzystujący metodę optymalizacji różnymi typami kolonii mrówek dla problemu maksymalnego i minimalnego dopasowania w ważonych grafach dwudzielnych. Algorytm ten wyznacza optymalne dopasowanie, bazując na wyznaczaniu rozdzielných ścieżek w grafie między wierzchołkami s - t , które stanowią rozwiązanie dla problemu optymalnego dopasowania w ważonych grafach dwudzielnych. Opracowany algorytm został porównany z algorytmem węgierskim i algorytmem mrówkowym o jednym typie kolonii mrówek i omówione zostały wyniki tego porównania.

Słowa kluczowe: optymalizacja koloniami mrówek wielu typów, minimalny problem dopasowania w grafach dwudzielnych, maksymalny problem dopasowania w grafach dwudzielnych

MULTI-TYPE ANT COLONY ALGORITHM FOR OPTIMAL MATCHING PROBLEM IN WEIGHTED BIPARTITE GRAPH

In this paper algorithm for optimal matching problem in weighted bipartite graph is presented, which is based on multi-type ant colony optimization. Matching problem is modeled as disjoint-paths problem between s - t vertices. Multi-type ants was used in order to find these disjoint paths between s - t vertices which are the solution for optimal matching problem in weighted bipartite graph. The algorithm was compared with Hungarian algorithm and ACO algorithm for optimal matching problem in weighted bipartite graph and results of this comparison was discussed.

Keywords: Multi-type Ant Colony Optimization, Maximum weighted bipartite matching, Minimum weighted bipartite matching

1. WPROWADZENIE

Problem maksymalnego dopasowania w grafach dwudzielnych jest problemem wielomianowym, co oznacza, że czas potrzebny na uzyskanie jego rozwiązania jest wielomianowo zależny od liczby danych wejściowych. Najbardziej znanym dokładnym algorytmem rozwiązującym ten problem jest tak zwany algorytm węgierski o złożoności obliczeniowej rzędu $O(V^3)$ [2], gdzie V oznacza liczbę wierzchołków w grafie.

Wiele jednak problemów grafowych przynależy do klasy NP-trudnych, czy nawet NP-zupełnych, dla których rozwiązanie jest znajdowane w czasie ponadwielomianowym. Dla tych problemów w wielu przypadkach istnieje potrzeba posługiwania się algorytmami heurystycznymi, które wyznaczają rozwiązanie optymalne lub nieznacznie oddalone od optymalnego w czasie znacznie krótszym niż dokonuje tego algorytm dokładny. Wśród algorytmów heurystycznych znajdują się między innymi te inspirowane biologicznie, a pośród nich – oparte na zachowaniu się kolonii mrówek [1].

Dla problemu maksymalnego i minimalnego dopasowania w grafach dwudzielnych zostały opracowane algorytmy oparte na metodzie optymalizacji kolonią mrówek [3] bazujące na algorytmie mrówkowym dla problemu komiwojażera [6] (ACO – *Ant colony Optimization*). Optymalizacja wie-

loma typami kolonii mrówek wraz z pseudokodem algorytmu jest prezentowana w pracy [4] (MACO – *Multi-type Ant Colony Optimization*). Algorytm zaproponowany w tym artykule wykorzystuje do rozwiązania tego problemu wiele typów kolonii mrówek i bazuje na wyznaczaniu wielu ścieżek rozłącznych w grafie między wierzchołkami s - t [5, 6].

2. PROBLEM OPTYMALNEGO DOPASOWANIA W WAŻONYCH GRAFACH DWUDZIELNYCH

Grafem $G = (V, E)$ nazywamy parę zbiorów wierzchołków V i krawędzi E między tymi wierzchołkami. Graf dwudzielny $G = (U, P, E)$ to graf o dwu rozdzielných podzbiorach U i P zbioru wierzchołków V takich, że każda krawędź ze zbioru E łączy dwa wierzchołki z dwóch rozdzielných podzbiorów U i P . Ważonym grafem dwudzielnym nazywamy taki graf dwudzielny, którego krawędziom przypisano wagi w_{ij} .

Problem maksymalnego lub minimalnego dopasowania w ważonym grafie dwudzielnym polega na znalezieniu takiego dopasowania M , to jest takiego podzbioru zbioru krawędzi E , że żadne dwie krawędzie ze zbioru M nie mają wspólnego wierzchołka, a każda para wierzchołków ($u \in U, p \in P$) jest połączona krawędzią i suma wag im przypisanych ma maksymalną lub minimalną wartość.

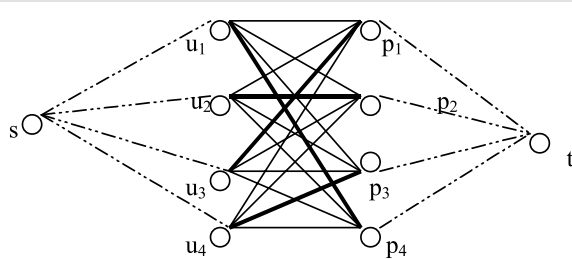
* Politechnika Krakowska, Katedra Automatyki

Problem ten można sformułować następująco:

$$\begin{aligned} & \max/\min \sum_{(u,t) \in E} w_{ut} x_{ut} \\ & \sum_{u=1}^{|U|} x_{up} = 1, \quad \forall p=1, \dots, |P| \\ & \sum_{t=1}^{|P|} x_{ut} = 1, \quad \forall u=1, \dots, |U| \\ & x_{up} \in \{0, 1\} \end{aligned} \quad (1)$$

gdzie $x_{up} = 1$ oznacza, że krawędź (u, p) wchodzi w skład optymalnego dopasowania M .

Problem optymalnego dopasowania w ważonym grafie dwudzielnym można przedstawić w sposób graficzny tak jak na rysunku 1.



Rys. 1. Graf dwudzielnny $G = (U, P, E)$ z dwoma dodatkowymi wierzchołkami s i t

Na rysunku 1 optymalne dopasowanie między wierzchołkami z podzbiorów U i P jest oznaczone jako pogrubione krawędzie. Wierzchołki s i t zostały dodane w celu wyznaczenia między tymi wierzchołkami czterech ścieżek rozłącznych $\{s, p_4, u_1, t\}$, $\{s, p_3, u_4, t\}$, $\{s, p_2, u_2, t\}$ i $\{s, p_1, u_3, t\}$ zawierających w sobie optymalne dopasowanie $M = \{(p_4, u_1), (p_3, u_4), (p_2, u_2), (p_1, u_3)\}$.

3. ALGORYTM MRÓWKOWY O WIELU TYPACH KOLONII

Algorytm mrówkowy o wielu typach kolonii działa podobnie jak zwykły algorytm mrówkowy z tą jednak różnicą, że oprócz mrówek o tym samym typie, które współpracują ze sobą ze względu na to, że mrówki są przyciągane przez feromon pozostawiony przez mrówki tego samego typu, występują również mrówki o odmiennych typach, które współzawodniczą ze sobą poprzez to, że mrówki jednego typu są odpychane zapachem feromonu pozostawionego przez mrówki innego typu. Takie zachowanie polegające na współpracy i współzawodniczeniu jest w rzeczywistości wykorzystywane do rozwiązywania pewnych problemów i w takim również celu zostało wykorzystane w prezentowanym w tej pracy algorytmie, gdzie w kolejnych iteracjach mrówki o odmiennych typach wyznaczają inne krawędzie optymalnego dopasowania w grafie dwudzielnym poprzez wyznaczenie rozdzielnych ścieżek między źródłem s i spływem t , zaś mrówki tego samego typu pracują w tym celu, aby te rozdzielne ścieżki miały jak najkrótszą długość.

W grafie dwudzielnym krawędzie (s, u_i) a więc wierzchołki ze zbioru U , są wybierane poprzez mrówki m losowo zgodnie z prawdopodobieństwem wyrażonym wzorem (2).

$$\begin{aligned} P_m(s, u_i) &= \\ &= \{t_m(u_i)^\alpha * [n - \phi(u_i)]^\beta\} / \\ & / \{\sum_{i=1}^n (t_m(u_i)^\alpha * [n - \phi(u_i)]^\beta)\} \end{aligned} \quad (2)$$

gdzie:

- $t_m(u_i)$ – oznacza ilość feromonu mrówki m na wierzchołku u_i ,
- $\phi(u_i)$ – oznacza ilość feromonu pozostawionego na wierzchołku u_i przez inne mrówki,
- n – liczba wierzchołków w zbiorze U , liczba par wierzchołków (u, p) .

Kolejny wierzchołek ścieżki rozłącznej, a więc wierzchołek p_i ze zbioru P , wybierany jest również w sposób losowy zgodnie z prawdopodobieństwem wyrażonym wzorem (3).

$$\begin{aligned} P_m(u, p_i) &= \\ &= \{t_m(p_i)^\alpha * [n - \phi(p_i)]^\beta * \eta(u, p_i)^\gamma\} / \\ & / \{\sum_{i=1}^n (t_m(p_i)^\alpha * [n - \phi(p_i)]^\beta * \eta(u, p_i)^\gamma)\} \end{aligned} \quad (3)$$

gdzie:

- $t_m(p_i)$ – oznacza ilość feromonu mrówki m na wierzchołku p_i ,
- $\phi(p_i)$ – oznacza ilość feromonu pozostawionego na wierzchołku p_i przez inne mrówki,
- $\eta(u, p_i)^\gamma$ – oznacza pożądanie krawędzi (u, p_i) w rozwiązaniu (jest to waga krawędzi).

Wybrana w powyższy sposób krawędź wchodzi w skład zbioru krawędzi tworzących optymalne dopasowanie w ważonym grafie dwudzielnym, powiększając poprzez swoją wagę sumaryczną wagę cząstkowego rozwiązania problemu $waga_M$. To czy mamy do czynienia z problemem minimalnego czy maksymalnego dopasowania, zależy od tego, w jaki sposób przyjmiemy funkcję pożądaną krawędzi. Jeśli wyraża się ona poprzez $\eta(u, p_i) = w[i][j]$, to mamy do czynienia z problemem maksymalnego dopasowania, a jeśli $\eta(u, p_i) = (1/w[i][j])$, to mamy do czynienia z minimalnym dopasowaniem w ważonym grafie dwudzielnym.

Oczywiście ilość feromonu jest uaktualniana zgodnie ze wzorem (4), gdzie w każdej iteracji spośród pewnej liczby prób określonych poprzez lp_m wybiera się najlepsze rozwiązanie w celu określenia przyrostu feromonu dla problemu maksymalnego dopasowania $dt = (1/(1+(n-waga_M)))$ i kładzie się go na wierzchołki z uwzględnieniem typu mrówki m , która wyznaczyła daną krawędź do optymalnego dopasowania M .

$$\begin{aligned} & \text{jeśli } (i, j) \in M, \text{ to } t[i][m] = (1-r) * t[i][m] + dt \\ & \text{oraz } t[j][m] = (1-r) * t[j][m] + dt \end{aligned} \quad (4)$$

Po tak uaktualnionej ilości feromonu na wierzchołkach przy kolejnej wykonywanej iteracji następuje ponowne obliczenie sumarycznych ilości feromonu pozostawionych na wierzchołkach przez wszystkich mrówki niezależnie od typu i ponownie zostają obliczone prawdopodobieństwa wyboru wierzchołków przez mrówki w zależności od ilości pozostawionego na nich feromonu przez określone typy mrówek. W celu zapobieżenia powstawania ścieżek przebiegających przez wspólne wierzchołki zaraz po włączeniu do tworzonego dopasowania M wierzchołkom nadaje się status niedostępnych. Pseudokod opracowanego algorytmu został przedstawiony na rysunku 2.

4. WYNIKI TESTÓW

Opracowany algorytm o wielu typach mrówek został porównany ze zwykłym algorytmem o jednym typie mrówek oraz z dokładnym algorytmem węgierskim. Wyniki porównania zwykłego algorytmu mrówkowego i algorytmu węgierskiego zostały zaprezentowane w pracy [3], z których wynika, że zwykły algorytm mrówkowy o jednym typie mrówek winien działać szybciej niż algorytm węgierski przy bardzo dużych wymiarach problemu, choć nie jest tak dokładny, jak wynika z porównania jego rezultatów działań z wynikami uzyskiwanymi przy uruchomieniu algorytmu

```

w każdej iteracji  $l\_iter$ 
  {wykonaj  $lp\_m$  powtórzeń
  {
  dla każdego wierzchołka  $i$  z  $U$  zlicz  $suma\_feromonu1[i]$  pozostawiony na przyległych krawędziach
  dla każdego wierzchołka  $p$  z  $P$  zlicz  $suma\_feromonu2[i]$  pozostawiony na przyległych krawędziach
  dla każdej mrówki  $m$ 
  {
  wybierz dostępny wierzchołek  $i$  z  $U$  według prawdopodobieństwa
  {
  dla każdego dostępnego wierzchołka  $i$  liczone jest prawdopodobieństwo
  {
  oblicz po wszystkich wierzchołkach dostępnych  $i$ 

$$suma1 = \sum_{i=1}^n \{n - (suma\_feromonu1[i] - feromon1[m][i])\}^{\alpha} * \{feromon1[m][i]\}^{\beta}$$


$$p1[i] = \{n - (suma\_feromonu1[i] - feromon1[m][i])\}^{\alpha} * \{feromon1[m][i]\}^{\beta} / suma1$$

  }
  po wybraniu wierzchołek staje się niedostępny
  }
  wybierz dostępny wierzchołek  $i$  z  $P$  według prawdopodobieństwa
  {
  dla każdego dostępnego wierzchołka  $i$  liczone jest prawdopodobieństwo
  {
  oblicz po wszystkich wierzchołkach dostępnych  $j$ 

$$suma2 = \sum_{i=1}^n \{n - (suma\_feromonu2[i] - feromon2[m][i])\}^{\alpha} * \{feromon2[m][i]\}^{\beta} * a[i][j]^{\gamma}$$


$$p2[i] = \{n - (suma\_feromonu2[i] - feromon2[m][i])\}^{\alpha} * \{feromon2[m][i]\}^{\beta} * a[i][j]^{\gamma} / suma2$$

  }
  po wybraniu wierzchołek staje się niedostępny
  zapamiętaj krawędź w dopasowaniu  $M\_tymczas$  oraz mrówkę, która wybrała krawędź
  dodaj wagę wybranej krawędzi do sumy wag krawędzi z dopasowania  $M\_tymczas$ 
  gdy nie wybrano wierzchołka  $i$  z  $P$  do pary do wierzchołka  $i$  z  $U$  to dodaj wagę kary
  } //koniec dla każdej mrówki
  oblicz pozostawiony feromon w powtórzeniu  $dt[lp\_m] = (1 / (1 + (n - waga)))$ 
  zapamiętaj najlepsze rozwiązanie z uzyskanych w trakcie powtórzeń wagę i  $dt$ 
  } //  $lp\_m$  koniec powtórzeń
  uaktualnij ilość feromonu na wszystkich krawędziach według wzoru  $t = (1 - r) t$ 
  dodaj  $dt[lp\_m]$  do ilości feromonu na krawędziach, które tworzyły rozwiązanie w powtórzeniach  $lp\_m$ 
  z uwzględnieniem, która to mrówka składała feromon
  } // koniec iteracji

```

Rys. 2. Pseudokod algorytmu mrówkowego o wielu typach kolonii dla problemu optymalnego dopasowania w ważonym grafie dwudzielnym

węgierskiego. To spodziewane pożądanie szybsze działanie algorytmu mrówkowego oparte jest na spostrzeżeniu, że przyrost czasu potrzebnego na uzyskanie rozwiązania jest znacznie większy w algorytmie węgierskim niż w algorytmie mrówkowym. Prezentowane wyniki przeprowadzanych testów w tym podrozdziale mają dać odpowiedź na pytania, czy algorytm o wielu typach mrówek daje lepsze rozwiązania i czy działa szybciej niż zwykły algorytm mrówkowy.

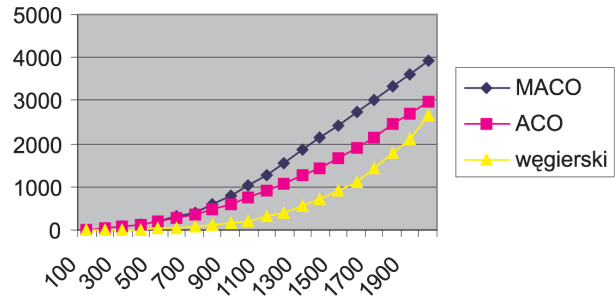
Testy przeprowadzono przy pełnych grafach dwudzielnych o liczbie par wierzchołków (u, p) odpowiednio wynoszącej $n = 100, \dots, 2000$ przy wadze od 0,0 do 1,0, liczbie iteracji wynoszącej $l_iter = 200$, liczbie mrówek $l_mr = 20$ w przypadku algorytmu o jednym typie mrówek i liczbie mrówek $l_mr = n$ oraz liczbie powtórzeń $lp_m = 20$ dla algorytmu o wielu typach mrówek, dla współczynnika parowania feromonu $r = 0,01$ i $r = 0,001$ odpowiednio dla tych algorytmów. Parametry powyższe uzyskano w wyniku przeprowadzonych testów. Dla parametrów tych uzyskano najlepsze zachowanie się badanych algorytmów, ale wyniki tych testów nie są prezentowane w tym artykule ze względu na brak ich wpływu na istotę tego artykułu, którego celem jest odpowiedź, czy i w jakim stopniu algorytm o wielu typach mrówek jest lepszy od algorytmu o jednym typie mrówek przy rozwiązywaniu problemu optymalnego dopasowania w ważonym grafie dwudzielnym.

Tabela 1. Czas działania badanych algorytmów w [ms]

Wymiar n	Czas		
	MACO	ACO	węgierski
100x100	10	12	0
200x200	37	28	1
300x300	70	68	4
400x400	124	120	10
500x500	194	187	20
600x600	304	268	35
700x700	401	363	65
800x800	587	475	106
900x900	783	603	146
1000x1000	1015	742	204
1100x1100	1268	903	300
1200x1200	1564	1069	395
1300x1300	1852	1254	537
1400x1400	2145	1448	709
1500x1500	2438	1668	907
1600x1600	2733	1895	1098
1700x1700	3028	2151	1424
1800x1800	3325	2463	1786
1900x1900	3619	2692	2111
2000x2000	3917	2974	2663

Jak wynika z tabeli 1, czas działania obu algorytmów mrówkowych w porównaniu z algorytmem węgierskim wskazuje na szybsze działanie tego drugiego, ale jak przyglądnijemy się tempu przyrostu czasu działania algorytmów

w zależności od przyrostu rozmiaru rozwiązywanego problemu, to okazuje się, że przy wzroście wymiaru problemu z 200 do 2000 par wierzchołków czas działania wzrasta około 106, 106 i 2663 razy (to jest stosunek czasu działania algorytmu przy 2000 wierzchołkach do czasu działania algorytmu przy 200 wierzchołkach) odpowiednio dla algorytmu o wielu typach mrówek, algorytmu o jednym typie mrówek i algorytmu węgierskiego (rys. 3).



Rys. 3. Czas działania algorytmów (oś – y) w funkcji wymiaru problemu (oś – x)

Tabela 2. Waga optymalnego dopasowania badanych algorytmów

Wymiar n	Waga		
	MACO	ACO	Węgierski
100x100	95.288	93.762	98.563
200x200	193.09	186.75	198.23
300x300	290.372	277.29	298.34
400x400	383.792	370.19	398.35
500x500	485.244	463.16	498.32
600x600	583.932	554.45	598.29
700x700	683.124	644.91	698.27
800x800	780.708	731.77	798.33
900x900	880.011	827.95	898.37
1000x1000	978.443	915.86	998.42
1100x1100	1045.22	1001.69	1098.36
1200x1200	1134.39	1091.74	1198.36
1300x1300	1228.38	1188.24	1298.35
1400x1400	1322.37	1272.00	1398.37
1500x1500	1416.40	1362.46	1498.42
1600x1600	1509.84	1453.66	1598.37
1700x1700	1606.52	1544.75	1698.38
1800x1800	1699.68	1631.94	1798.37
1900x1900	1793.64	1719.49	1898.37
2000x2000	1888.16	1810.81	1998.35

Algorytm węgierski jest algorytmem dokładnym, więc zawsze daje on rozwiązanie optymalne. W porównaniu do niego, jak wynika z tabeli 2 i 3, algorytmy mrówkowe nie wyznaczają rozwiązań optymalnych, lecz wynik osiągany przez nie jest wynikiem zbliżonym. Dla zwykłego algorytmu o jednym typie mrówek otrzymywana wartość wagi optymalnego dopasowania M to wartość, która osiąga ponad

90% wartości optymalnej wagi dopasowania. Dla algorytmu o wielu typach mrówek wartość ta znajduje się w pobliżu 95% i w porównaniu z algorytmem o jednym typie mrówek zawsze wynik ten jest lepszy, co więcej wynik algorytmu o jednym typie mrówek wraz ze wzrostem wymiaru problemu pogarsza się, gdy tymczasem wynik algorytmu o wielu typach mrówek po początkowym pogarszaniu się stabilizuje się w pobliżu 94,5% wartości wagi optymalnego dopasowania uzyskiwanego algorytmem węgierskim.

Tabela 3. Waga optymalnego dopasowania badanych algorytmów w [%]

Wymiar n	Waga		
	MACO	ACO	Węgierski
100×100	96,5	95,1	100
200×200	97,0	94,2	100
300×300	97,3	92,9	100
400×400	96,4	92,9	100
500×500	97,4	92,9	100
600×600	97,6	92,7	100
700×700	97,8	92,4	100
800×800	97,8	91,7	100
900×900	98,0	92,2	100
1000×1000	98,0	91,7	100
1100×1100	95,2	91,2	100
1200×1200	94,7	91,1	100
1300×1300	94,6	91,5	100
1400×1400	94,6	91,0	100
1500×1500	94,5	90,9	100
1600×1600	94,5	90,9	100
1700×1700	94,6	91,0	100
1800×1800	94,5	90,7	100
1900×1900	94,5	90,6	100
2000×2000	94,5	90,6	100

5. ZAKOŃCZENIE

Artykuł ten prezentuje algorytm o wielu typach mrówek (MACO), który jak się okazuje pracuje wolniej niż algorytm o jednym typie mrówek (ACO), ale daje znacznie lepszy wynik, co więcej oba algorytmy ze względu na wolniejsze tempo przyrostu czasu obliczeń w porównaniu do algo-

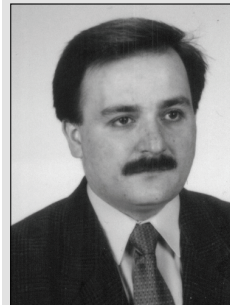
rytmu węgierskiego mogą znacznie szybciej osiągać wynik swojej pracy przy problemach o większych rozmiarach niż algorytm węgierski. Szczególnie cecha ta jest istotna przy dynamicznie zmieniającym się problemie optymalnego dopasowania w ważonym grafie dwudzielnym.

Literatura

- [1] Dorigo M., Di Caro G.: *The Ant Colony Optimization meta-heuristics*. [w:] Corne D., Dorigo M., Glover F. (eds.), *New Ideas in Optimization*, McGraw Hill, UK, 1999, 11–32
- [2] Munkers J.: *Algorithms for the Assignment and Transportation Problems*. *Journal of the Society of Industrial and Applied Mathematics*, vol. 5, no. 1, 1957, 32–38
- [3] Shih H.: *Two Algorithms for the maximum and minimum weighted bipartite matching*. Raport, Department of Computer Science and Information engineering, National Taiwan University 2006
- [4] Nowe A., Verbeeck K., Vrancx P.: *Multi-type Ant colony: The Edge Disjoint Paths Problem* [w:] M. Dorigo i in.: *Ant colony Optimization and Swarm Intelligence*, 4th International Workshops, Belgium, September 2004, Proceedings, Springer 2004
- [5] Blesa M.J., Blum Ch.: *On Solving the Maximum Disjoint Paths Problem with Ant Colony Optimization*. Raport, Departament de Llenguatges i Sistemes Informatics, Universitat Politecnica de Catalunya, Barcelona, Spain 2006
- [6] Stutzle T., Dorigo M.: *ACO Algorithms for the Traveling Salesman Problem*. [w:] K. Miettinen, M. Makela, P. Neittanmaki i J. Peiriaux (Eds), *Evolutionary Algorithms in Engineering and Computer Science*, Wiley 1999

Wpłynęło: 30.11.2008

Krzysztof SCHIFF



Absolwent Wydziału Elektrotechniki, Automatyki i Elektroniki AGH w 1992 roku. W latach 1992–1996 doktorant na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki AGH, na którym w roku 1998 uzyskuje tytuł doktora w dyscyplinie automatyka i robotyka. Wydział Zarządzania AGH kończy w 1995 roku, a w 2003 uzyskuje tytuł Microsoft Certified Professional. Od 1992 roku asystent, a od 1998 adiunkt w Instytucie, następnie Katedrze Automatyki Wydziału Inżynierii Elektrycznej i Komputerowej Politechniki Krakowskiej.

e-mail: kschiff@pk.edu.pl