

KRZYSZTOF SCHIFF\*

**ZASTOSOWANIE ALGORYTMU MRÓWKOWEGO  
DO WYZNACZANIA MAKSYMALNEJ GRUPY  
WZAJEMNIE POŁĄCZONYCH ELEMENTÓW****ANT ALGORITHMS FOR DETERMINING MAXIMUM  
GROUP OF INTERCONNECTED ELEMENTS****Streszczenie**

W artykule zaprezentowano algorytmy mrówkowe wyznaczające największą klikę w grafie, za pomocą której modeluje się problem wyznaczania największego ze skupień wzajemnie połączonych elementów elektronicznych na płycie drukowanej w celu minimalizacji długości połączeń między nimi, a w konsekwencji minimalizacji ilości materiału zużytego na ich wytworzenie. W artykule zaprezentowano algorytm oparty na odmiennych aspektach zachowania się mrówek w porównaniu z dotychczas opracowanymi algorytmami. Główną różnicą między algorytmami jest faza eksploracji, która została wprowadzona w prezentowanym algorytmie. Opracowany algorytm porównano z Algorytmem 457 pod względem wyznaczanego wymiaru klik. Dokonano również porównania procedur lokalnego przeszukiwania (2,1)-wymiany i procedury opartej na metaheurystyce kolonii mrówek.

*Słowa kluczowe: klika maksymalna, algorytm mrówkowy, optymalizacja kombinatoryczna, teoria grafów, algorytmy komputerowe*

**Abstract**

In this paper an ANT algorithm, which is used to find a maximum group of mutually connected electronic elements in order to minimize the total length of connections, is presented. The new algorithm differs from algorithms which have been presented in scientific papers until now. The main difference is a phase of ANT exploration which is absent in other ANT algorithms. Sizes of maximum clique indicated by ANT algorithm and the Algorithm 457 are compared. The influence of the local search was presented also and the (2,1)-exchange local procedure and the ANT procedure of local search was compared.

*Keywords: maximum clique, ANT algorithm, combinatorial optimization, graph theory, computer algorithms*

\* Dr inż. Krzysztof Schiff, Katedra Automatyki, Wydział Inżynierii Elektrycznej i Komputerowej, Politechnika Krakowska.

## 1. Wstęp

W procesie produkcyjnym płytek drukowanych istotnym zagadnieniem jest takie wyznaczenie pozycji elementów elektronicznych na płytce, aby odległości między nimi były jak najmniejsze. Ma to na celu obniżenie kosztów produkcji przez zmniejszenie zużycia materiału na poprowadzenie takich połączeń. Zagadnienie to może być zamodelowane za pomocą grafu, w którym węzły odpowiadają łączonym elementom elektronicznym, a łuki połączeniom między tymi elementami. Rozwiązanie tego problemu uzyskuje się przez znalezienie największych skupisk elementów wzajemnie połączonych, co w grafie sprowadza się do znalezienia klik o maksymalnym wymiarze. W celu przedstawienia algorytmu mrówkowego znajdującego klikę o maksymalnym wymiarze należy najpierw przytoczyć użyteczne przy jego opracowaniu podstawowe pojęcia z obszaru teorii grafów, które znajdują się poniżej.

Zbiór wierzchołków  $V$  i zbiór krawędzi  $E$  łączących wierzchołki ze zbioru  $V$  tworzą graf  $G = (V, E)$ . Zbiór wierzchołków  $C$ , w którym każda dowolna para wierzchołków jest połączona krawędzią, nazywany jest kliką. W zbiorze wierzchołków  $V$  może być wiele podzbiorów tworzących kliki. Kliką maksymalną grafu nazywamy taką klikę, która będąc podzbiorem zbioru wierzchołków  $V$ , składa się z największej liczby wierzchołków spośród wszystkich klik grafu  $G$ , innymi słowami – klika maksymalna grafu jest to podzbiór  $C$  wierzchołków grafu  $G$  o największej liczbie kardynalnej, w którym każda dowolna para wierzchołków jest połączona ze sobą krawędzią  $e \in E$ .

Sąsiadami wierzchołka  $v$  są wszystkie wierzchołki, które są połączone krawędzią z wierzchołkiem  $v$ . Sąsiedztwem wierzchołka  $v$  nazywać będziemy wszystkich sąsiadów wierzchołka  $v$ .

Znajdowanie wszystkich klik w grafie jest ważnym problemem w analizie grupowej [2, 8]. Dokładne algorytmy rozwiązujące ten problem mają ponadwielomianową złożoność obliczeniową [3], dlatego opracowano wiele heurystycznych algorytmów o wielomianowej złożoności obliczeniowej, m.in. bazujące na zachłannej heurystyce sekwencyjnej [4–7] i przeszukiwaniu lokalnym połączonym z metodą Tabu Search [9], Simulated Annealing [10] oraz z selekcją genetyczną [11].

Heurystyka zaprezentowana na podstawie zachowania się kolonii mrówek [12] posłużyła do rozwiązania wielu problemów kombinatorycznych, takich jak problem komiwojażera [13], kwadratowe zagadnienie przydziału [15] czy też problem kolorowania grafów [17] oraz problem plecakowy [19]. Przegląd opracowanych algorytmów mrówkowych dla różnych problemów optymalizacji kombinatorycznej można znaleźć w pracy [18].

W niniejszym artykule zaprezentowano wyniki zastosowania obserwacji zachowania się kolonii mrówek do rozwiązania problemu wyznaczania kliki maksymalnej grafu, a wyznaczone w ten sposób kliki o największych wymiarach porównano z wymiarami klik maksymalnych uzyskanych algorytmem zwanym Algorytmem 457.

## 2. Zachowanie się kolonii mrówek

Najbardziej użytecznym dla rozwiązywania wielu problemów z obszaru optymalizacji kombinatorycznej wynikającym z obserwacji zachowania się kolonii mrówek jest fakt, że każda kolonia mrówek wyznacza najkrótszą ścieżkę między mrowiskiem a miejscem, gdzie

znajduje się pożywienie. Mrówki wykorzystują feromon w celu zaznaczenia tej najkrótszej drogi.

Sztuczne mrówki pozostawiają feromon na wierzchołkach lub krawędziach grafu i wybierają kolejny wierzchołek lub krawędź do przejścia w zależności od ilości feromonu, który został pozostawiony na nich poprzednio. Wybór kolejnego wierzchołka  $j$  sąsiedniego do wierzchołka  $i$  przez sztuczną mrówkę do dowiedzenia odbywa się z prawdopodobieństwem

$$p_{ij}^k = \frac{t_{ij}}{\sum_{j \in N_i} t_{ij}} \quad \text{jeśli } j \text{ jest sąsiadem } i$$

Podejście do rozwiązania problemu wyznaczenia klikli maksymalnej grafu, w którym mrówki poruszały się po wierzchołkach lub krawędziach grafu, tylko i wyłącznie kierując się feromonem, było podstawą opracowania dwóch algorytmów mrówkowych przedstawionych w pracy [16]. W niniejszym artykule podejście do rozwiązania problemu klikli maksymalnej na podstawie kolonii mrówek jest odmienne ze względu na to, że mrówkom pozwolono po wyjściu z mrowiska eksplorować otoczenie mrowiska, a dopiero później – po zadecydowaniu przez nie o powrocie do mrowiska – mogły się one kierować feromonem w celu podążania do mrowiska najkrótszą ścieżką. Opracowane podejście nazwano algorytmem opartym o pracę i eksplorację mrówek. Drugie podejście zwane algorytmem opartym na pracy mrówek jest podobne do podejścia przedstawionego w pracy [16] i różni się tylko w niewielki sposób, a mianowicie tym, że w każdym cyklu pierwszy wierzchołek nie jest wybierany w sposób całkowicie przypadkowy, lecz w zależności od mapy feromonowej, która stanowi pamięć o wcześniejszej pracy algorytmu i o wyznaczonych od początku pracy algorytmu ścieżkach feromonowych. Różni się także tym, że feromon kładziony jest na wszystkich klikach wyznaczonych przez mrówki, a nie tylko na klicie największej spośród klik wyznaczonych przez mrówki w danym cyklu, jak odbywa się to w algorytmie z pracy [16].

Mrówki, które poruszają się, kierując zapachem feromonu, zwane będą mrówkami-pracującymi. Pozostawiony feromon jest informacją dla wszystkich mrówek o najkrótszej ścieżce do mrowiska. Wykorzystanie tylko i wyłącznie sztucznych mrówek-pracujących ma miejsce w opracowanych algorytmach [16], w których mrówki pozostawiały feromon w jednym algorytmie na wierzchołkach, a w drugim na krawędziach grafu. W prezentowanych w tym artykule algorytmach mrówki pozostawiają feromon tylko na wierzchołkach grafu i wykorzystywane są w jednym z nich tylko mrówki-pracujące, a w drugim nie tylko mrówki-pracujące, ale i również mrówki-eksplorujące.

Skoro pozostawiony feromon jest informacją dla wszystkich mrówek o najkrótszej ścieżce do mrowiska, to brak tego feromonu jest również pewną informacją dla wszystkich mrówek. Mrówki wiedzą, że gdy nie ma feromonu, oznacza to, że są na ścieżce, która prowadzi do nieznanego mrówkom miejsca. Ta informacja jest wykorzystywana przez mrówki-zwiadowców. Ta grupa mrówek poszukuje nowych miejsc, gdzie mrówki mogą znaleźć pożywienie, zaś ich sztuczne odpowiedniki – wierzchołki, z których mogą rozpocząć wyznaczanie kolejnych rozwiązań cząstkowych problemów.

Sztuczne mrówki-zwiadowcy poruszają się wzdłuż tych wierzchołków lub krawędzi, gdzie feromonu nie ma lub jego ilość jest minimalna. Sztuczne mrówki-zwiadowcy wybierają nowy wierzchołek i jego sąsiedztwo w celu jego przebadania pod kątem znalezienia

kliki maksymalnej. Ten nowo wybrany wierzchołek zwany będzie dalej wierzchołkiem-mrowiskiem. Za mrówkami-zwiadowcami podążają mrówki-eksplorujące, które badają otoczenie wierzchołka-mrowiska.

Ścieżki feromonowe utworzone przez mrówki-eksplorujące mogą prowadzić pomiędzy miejscami niebezpiecznymi dla mrówek lub też może być tak, że z miejsc, do których one prowadzą, całe pożywienie zostało już przetransportowane do mrowiska. Informacja o takich przypadkach również rozchodzi się w jakiś sposób między mrówkami. Mrówki nie podążają już do miejsca, gdzie znajdowało się pożywienie i którego już tam nie ma, omijają też miejsca niebezpieczne. Pozostawiony feromon na ścieżce wiodącej z mrowiska do miejsca, gdzie znajdowało się pożywienie jeszcze nie wyparował całkowicie, więc mrówki wciąż podążałyby tą ścieżką. W celu uniknięcia powyższych sytuacji sztuczne mrówki pozostawiają tak zwany feromon zakazu na wierzchołkach. Są to miejsca wykluczone z obszaru przeszukiwania i mrówki powinny ich unikać zarówno podczas eksploracji, jak i podczas tworzenia rozwiązania cząstkowego. Rozwiązanie cząstkowe, a w gruncie rzeczy klika, jest tworzona podczas powrotu mrówki do wierzchołka-mrowiska przez sąsiadujące z nim wierzchołki grafu.

Wszystkie mrówki współpracują ze sobą, więc również współpracują z mrówkami-zwiadowcami – za nimi wobec tego podążają mrówki-eksplorujące. Mrówki-zwiadowcy poruszają się wzdłuż ścieżek, na których nie ma pozostawionego feromonu lub jego ilość jest minimalna. Mrówki-eksplorujące rozchodzą się we wszystkich kierunkach i nie zwracają uwagi na ilość pozostawionego na wierzchołkach feromonu. Sztuczne mrówki-eksplorujące poruszają się więc po wierzchołkach i krawędziach z i bez feromonu i eksplorują w swobodny sposób sąsiedztwo wierzchołka-mrowiska. Poruszając się, mrówki pozostawiają cały czas za sobą feromon.

Mrówki-eksplorujące mogą znaleźć pożywienie w nowym miejscu. Oczywiście, by nie męczyć się zbytnio, podążą z nim do mrowiska po najkrótszej drodze, kierując się przy tym intensywnością zapachu feromonu. Sztuczne mrówki-pracujące, rozpoczynając powrót po eksploracji do mrowiska, tworzą konstrukcję rozwiązania cząstkowego, czyli konstrukcję kliki, którą kończą, włączając na koniec do rozwiązania wierzchołek będący wierzchołkiem-mrowiskiem. Sztuczne mrówki-pracujące rozpoczynające tworzenie rozwiązania cząstkowego od wierzchołka początkowego będącego mrowiskiem są wykorzystywane w pracy [16] i algorytmie opartym na pracujących mrówkach w tym artykule. Sztuczne mrówki-pracujące nie mają więc fazy eksploracji i natychmiast przystępują do wyznaczania kliki, najkrótszej możliwej drogi wiodącej do uzyskania rozwiązania cząstkowego. Poniżej opisany system kolonii mrówek i komunikacji został zestawiony w tabeli 1 z ogólną strategią przeszukiwania stosowaną w celu znalezienia rozwiązania problemu.

Tabela 1

**System kolonii mrówek i strategia poszukiwania rozwiązań problemów**

Funkcja	Rodzaj mrówki	Feromon
Przeszukiwanie globalne	mrówka-zwiadowca	minimalna ilość
Przeszukiwanie lokalne	mrówka-eksplorująca	bez znaczenia
Rozwiązanie cząstkowe	mrówka-pracująca	maksymalna ilość
Funkcja kary	każda mrówka	feromon zakazu

### 3. Algorytmy mrówkowe

Opracowane algorytmy mrówkowe różnią się między sobą w związku z różnym sposobem wykorzystywania zachowania się kolonii mrówek. W prezentowanym algorytmie w podrozdziale 3.1 wykorzystuje się dwa aspekty z zachowania mrówek, a mianowicie aspekt ich pracy oraz aspekt zwiadowczy, czyli przemieszczania w nowe miejsca, w otoczeniu których dokonywana jest praca. W podrozdziale 3.2 dołączono etap eksploracji otoczenia, który występuje zawsze przed etapem pracy. W algorytmach prezentowanych w pracy [16] wykorzystano jedynie aspekt pracy mrówek, zaś w pracy [14] dodano procedurę zakazu odwiedzin pewnych wierzchołków w celu lepszego zdywersyfikowania przeszukiwań. Oczywiście, wymienione etapy mogą różnić się implementacją w poszczególnych algorytmach w zależności od inwencji i pomysłowości ich autorów.

#### 3.1. Algorytm eksponujący pracę mrówek

W algorytmie tym losowo wybierany jest pierwszy wierzchołek, począwszy od którego mrówki rozpoczynają natychmiast pracę w celu wyznaczenia kliku. Wierzchołek ten wybierany jest spośród tych wierzchołków grafu, na których znajduje się minimalna ilość feromonu.

Początkowo wyznaczony wierzchołek jest pierwszym wierzchołkiem wchodzącym w skład kliku  $C$ , więc pozostawiany jest na nim feromon  $ASF[s] = ASF[s] + 1$ . Feromon rozpylany jest też na wszystkie wierzchołki sąsiednie do wierzchołka, w którym znajduje się mrówka w wierszu

$$\text{dla każdego } v \text{ if } v \text{ jest sąsiadem do } s \text{ then } ARSF[v] = ARSF[v] + 1$$

a następnie spośród wierzchołków sąsiednich do tworzonej przez mrówkę kliku wybiera się losowo wierzchołek spośród tych o maksymalnej ilości posiadanego feromonu. Wierzchołek ten włączany jest do kliku  $C$  i pozostawiany jest na nim feromon  $ASF[s] = ASF[s] + 1$  i znów wybierany jest losowo wierzchołek spośród wierzchołków sąsiednich do tworzonej kliku. Wybór wierzchołka  $v_i$  dokonuje się z prawdopodobieństwem  $p(v_i)$  określanym jako

$$p(v_i) = \frac{ARSF[v_i]_{\max}}{\sum_{k \in I_{ARSF_{\max}}} ARSF[v_k]_{\max}}$$

Gdy takich wierzchołków nie ma, oznacza to, że dotychczas utworzona klika nie może być powiększona, następuje jej zapamiętanie i uaktualnienie ścieżki feromonowej wyznaczonej przez wszystkie dotychczas pracujące mrówki w danym cyklu

$$\text{dla każdego wierzchołka } v: SF[v] = (r * SF[v]) + ASF[v]$$

gdzie  $r$  to stopień trwałości ścieżki feromonowej.

Po zakończeniu pracy wszystkich wygenerowanych mrówek w danym cyklu następuje uaktualnienie mapy feromonowej

$$\text{dla każdego wierzchołka } v: MF[v] = r * MF[v] + SF[v]$$

Z mapy feromonowej  $MF[v]$  w następnym okresie będzie korzystała mrówka-zwiadowca w celu wyznaczenia mrowiska, czyli kolejnego miejsca, w otoczeniu którego mrówki będą pracować nad wyznaczaniem klik grafu. Wybór losowy wierzchołka  $v_i$  dokonana się z prawdopodobieństwem określonym wzorem

$$p(v_i) = \frac{MP[v_i]_{\min}}{\sum_{k \in MP_{\min}} MP[v_k]_{\min}}$$

Poprzez wybór wierzchołków na mapie feromonowej mających minimalną ilość feromonu będzie realizowane przeszukiwanie globalne grafu, zaś przeszukiwanie lokalne realizowane będzie w otoczeniu tych wierzchołków. Poniżej przedstawiony jest pseudokod omówionego algorytmu.

```

for (okres=1; okres<=max_okres; okres++)
{
  wybierz losowo wierzchołek s spośród v, gdy MP[v]=min_feromon
  dla każdego wierzchołka v: SF[v]=0;
  for(cykl=1; cykl <= max_cykl; cykl++)
  {
    for(mrowka=1; mrowka <= max_mrowka; mrowka++)
    {
      dla każdego v: ASF[v]=0; oraz ARSF[v]=0;
      ASF[s]=ASF[s]+1; C←s; pracuje =1;
      while( pracuje )
      {
        pracuje=0;
        dla każdego v if v jest sąsiadem do s then ARSF[v]=ARSF[v]+1;
        wybierz losowo wierzchołek s, s jest sąsiadem C oraz ARSF[s]=max_feromon
        if s istnieje then
          { C←s; ASF[s]=ASF[s]+1; pracuje=1; }
      }/while
      zapamiętaj największą klikę C
      dla każdego wierzchołka v: SF[v]=((r*SF[v])+ASF[v]);
    }/(mrowka)
  }/( cykl)
  dla każdego wierzchołka v: MF[v]=r*MF[v]+SF[v];
} /( okres)

```

Rys. 1. Pseudokod dla algorytmu z mrówką pracującą

Fig. 1. Pseudo code for algorithm with labour ANT

### 3.2. Algorytm eksponujący eksplorację i pracę mrówek

W każdym „okresie” algorytmu mrówka pełniąca rolę zwiadowcy wybiera w sposób losowy wierzchołek  $v$  spośród wierzchołków o minimalnej ilości feromonu  $min\_feromon$ ,

który to wierzchołek następnie staje się wierzchołkiem-mrowiskiem. Najpierw ustalana jest więc minimalna ilość feromonu pozostawiona na wierzchołkach

$$\forall v \in V \text{ if } MP[v] < \min\_fermon \text{ the } \min\_foromon = MP[v]$$

Informacja o ilości feromonu, który pozostawiony został na wierzchołkach, jest przechowywana w mapie feromonowej M.

Następnie spośród wierzchołków, które mają na sobie minimalną ilość feromonu wybiera się jeden z nich w sposób losowy i z prawdopodobieństwem określonym wzorem

$$p(v_i) = \frac{MP[v_i]_{\min}}{\sum_{k \in I_{MP_{\min}}} ARSF[v_k]_{\min}}$$

Mapa feromonowa MP jest uaktualniana przy końcu każdego „okresu” algorytmu według zależności

$$\text{dla każdego } v: M[v] = M[v] + PT[v]$$

gdzie:

PT[v] – ścieżka feromonowa,  
v – wierzchołek.

Mapa feromonowa reprezentuje intensywność badań wierzchołków i ich sąsiedztwa w poszukiwaniu klikli maksymalnej grafu. Wierzchołki bez lub z minimalną ilością feromonu są wierzchołkami, w sąsiedztwie których poszukiwanie klikli maksymalnej nie było prowadzone lub było prowadzone, ale nie w sposób intensywny. Mrówka-zwiadowca wybiera nowy wierzchołek wraz z jego sąsiedztwem w celu przebadania go w poszukiwaniu klikli maksymalnej i w taki sposób w grafie zrealizowane jest przeszukiwanie globalne.

W każdym „cyklu” algorytmu wszystkie mrówki rozpoczynają swoją pracę począwszy od wierzchołka-mrowiska. Mrówki opuszczające mrowisko nazwane zostały mrówkami-eksplorującymi. Mogą one eksplorować tylko wierzchołki sąsiednie do mrowiska. Gdy mrówki opuszczają mrowisko, na żadnym z wierzchołków v grafu nie ma pozostawionego feromonu i aktualna ścieżka feromonowa AFT jest wyczyszczona, tzn. bez feromonu, czyli dla każdego wierzchołka v mamy  $APT[v] = 0$  oraz usuwany jest także z każdego wierzchołka, na którym był pozostawiony, feromon zakazu  $FZ[v] = 0$ . Każda mrówka tworzy własną ścieżkę feromonową zwaną dalej aktualną ścieżką feromonową APT.

Mrówki-eksplorujące odwiedzają wszystkie wierzchołki z sąsiedztwa wierzchołka, w którym zlokalizowane jest mrowisko. Wybierają one przypadkowo wierzchołek z wszystkich sąsiadujących wierzchołków do wierzchołka, w którym aktualnie przebywają. Nie zwracają one uwagi na ilość pozostawionego feromonu na wierzchołkach i wybierają niezależnie od niego kolejny wierzchołek do odwiedzenia. Jeżeli wierzchołek v jest wierzchołkiem, w którym aktualnie znajduje się mrówka, to ustalone są wszystkie sąsiednie wierzchołki do v, które są równocześnie sąsiednie do wierzchołka-mrowiska, gdyż tylko w sąsiedztwie mrowiska następuje eksploracja.

Podczas eksploracji mrówki pozostawiają na każdym odwiedzionym przez siebie wierzchołku feromon. Liczba odwiedzin danego wierzchołka v przechowywana w elemencie  $APT[v]$  jest zwiększana o jeden  $APT[v] = APT[v] + 1$ , gdy wierzchołek v jest odwiedzany przez mrówkę i w taki sposób aktualna ścieżka feromonowa jest uaktualniana.



Mrówki-eksplorujące za każdym razem, gdy odwiedzają nowy wierzchołek, podejmują decyzję dotyczącą tego, czy mają odwiedzić kolejny wierzchołek grafu, czy też mają rozpocząć już drogę powrotną do mrowiska. Prawdopodobieństwo wyboru między pracą a eksploracją może przybierać wartości od 0,0 do 1,0. Liczba takich decyzji jest także ograniczona przez wartość zmiennej *max-krok*, w przypadku gdy prawdopodobieństwo wyboru przez mrówkę dalszej eksploracji otoczenia mrowiska wynosi 1,0.

Odpowiednia wartość zmiennej *max-krok*, wynosząca ok. 4, dobrana została podczas przeprowadzania testów i jej wartość została ustalona w zależności od całkowitej liczby wierzchołków w grafie.

Jeśli mrówki-eksplorujące zdecydowały, że powracają do mrowiska, to przemieniają się one w mrówki-pracujące. Mrówki-pracujące powracające do mrowiska po wierzchołkach, wybierając kolejno wierzchołki o maksymalnej ilości feromonu, i konstruują kliki z tych wierzchołków, tzn. ze wszystkich wierzchołków, które wcześniej same odwiedziły, które znajdują się w aktualnej ścieżce feromonowej i dla których  $APT[v] > 0$ .

Ponieważ eksploracja otoczenia jest dokonywana przez mrówki w sąsiedztwie mrowiska, więc feromon pozostawiony został tylko na wierzchołkach sąsiadujących z wierzchołkiem, w którym znajduje się mrowisko. Wierzchołek, w którym znajduje się mrowisko, może być więc zawsze włączony do konstruowanej kliki przez mrówkę-pracującą i powracającą do mrowiska na końcu.

Za każdym razem mrówka-pracująca, czyli mrówka powracająca do mrowiska, konstruuje klikę przez wybór losowy i włączanie do konstruowanej kliki wierzchołka  $v$  wybranego ze wszystkich wierzchołków, które mrówka wcześniej odwiedziła i włączyła do aktualnej tworzonej ścieżki feromonowej przez wierzchołki, dla których  $APF[v] > 0$ .

Wybór losowy wierzchołka dokonywany jest spośród wierzchołków o maksymalnej ilości złożonego na nich feromonu, czyli spośród wierzchołków o  $APT[v] = \text{max\_feromon}$  i które nie zostały równocześnie naznaczone feromonem zakazu  $FZ[v] = 0$ .

Najpierw należy ustalić maksymalną ilość feromonu nałożoną na wierzchołki

$$\forall v \in V \text{ if } APT[v] > \text{max\_feromon} \text{ then } \text{max\_feromon} = APT[v] \text{ (wiersz algorytmu)}$$

a potem – jeśli istnieje taka potrzeba – spośród wszystkich wierzchołków  $v$  mających maksymalną ilość feromonu,  $APT[v] = \text{max\_feromon}$ , należy wybrać jeden wierzchołek w sposób losowy według wzoru

$$p(v_i) = \frac{APT[v_i]_{\max}}{\sum_{k \in V_{APT_{\max}}} APT[v_k]_{\max}}$$

Po włączeniu wierzchołka  $v$  do konstruowanej kliki na wierzchołku tym składany jest feromon, czyli:  $APT[v] = APT[v] + 1$  i w ten sposób dokonywane jest uaktualnienie aktualnej ścieżki feromonowej tworzonej przez pracującą mrówkę.

Feromon zakazu składany jest na wierzchołkach  $t$ , które nie sąsiadują z wierzchołkiem  $s$  włączonym do konstruowanej kliki przez mrówkę powracającą do mrowiska

$$\text{dla każdego } t: \text{if } (s, t) \notin E, FZ[t] = 1$$



Wynika to z faktu, że klikę tworzą wierzchołki, które są połączone ze wszystkimi pozostałymi wierzchołkami wchodzącymi w skład tej kliky, a to oznacza, że jeżeli wierzchołek wchodzący w skład konstruowanej kliky nie jest połączony z pewnymi wierzchołkami, to wierzchołki te nie mogą wejść w skład tej konstruowanej kliky i dlatego można zaznaczyć je feromonem zakazu  $FZ[t] = 1$ .

Feromon zakazu pozostawiany jest także na wierzchołkach włączonych do konstruowanej kliky i ma to na celu uniknięcie sytuacji, w której po raz wtóry ten sam wierzchołek byłby włączany do konstruowanej kliky.

Klika maksymalna jest wybrana spośród wszystkich klik utworzonych przez mrówki powracające do mrowiska podczas działania algorytmu.

```

for(okres=1; okres<=max_okres ; okres++)
{
  wybierz losowo v, gdy  $MP[v]=min\_feromon$ ;
   $s \leftarrow v$ ;
  for(cykl=1; cykl<=max_cykl; cykl++)
  {
    for(mrowka=1; mrowka<=max_mrowka; mrowka++)
    {
      dla każdego v:  $APT[v]=0$  oraz  $FZ[v]=0$ ;
       $C \leftarrow s$ ;
      dla każdego s:  $APT[s]=APT[s]+1$ ;
      for(krok=1; krok<=max_krok;
      {
        wybierz losowo v sąsiedni do wierzchołka s;
        dla każdego v:  $APT[v]=APT[v]+1$ ;
        wybierz losowo "między pracą a eksploracją"
         $s \leftarrow v$ ;
        if mrówka-eksplorująca do następnego krok
          else  $krok=max\_krok+1$ ;
      }
      while(idz-do_mrowiska)
      {
        dla każdego s:  $APT[s]=APT[s]+1$ ;
        wybierz losowo v,  $APT[v]=max\_feromon$  i  $FZ[v]=0$ ;
         $s \leftarrow v$ ;  $C \leftarrow s$ ;
        dla każdego t: if  $(s,t) \notin E$ ,  $FZ[t]=1$ ;
        if nie ma sąsiadów do s then  $idz\_do\_mrowiska=0$ 
      }
      zapamiętaj największą klikę C;
      dla każdego v:  $PT[v]=r*PT[v]+APT[v]$ ;
    } //(mrowka)
  } //(cykl)
  dla każdego v:  $MP[v]=MP[v]+PT[v]$ ;
} //(okres)

```

Rys. 2. Pseudokod dla algorytmu z mrówką-eksplorującą

Fig. 2. Pseudo code for algorithm with explorer ANT

Gdy każda mrówka powróci do mrowiska po utworzeniu kliku, nowo powstała aktualna ścieżka feromonowa dla każdej z tych mrówek jest dodawana do ścieżki feromonowej wszystkich mrówek według następującej zależności

$$PT[v] = r * PT[v] + APT[v]$$

gdzie:

- APT – aktualna ścieżka feromonowa,
- $v$  – wierzchołek,
- APT[ $v$ ] – liczba odwiedzin przez mrówkę wierzchołka  $v$ ,
- $r$  – stopień trwałości ścieżki feromonowej,
- PT – ścieżka feromonowa.

Podczas testu wartość  $r$  przyjęto jako równą 0,998.

Pseudokod prezentowanego algorytmu mrówkowego opracowanego do wyznaczania kliku maksymalnej grafu zaprezentowano na rys. 1.

#### 4. Porównanie algorytmów mrówkowych

Oba algorytmy mrówkowe zostały porównane ze względu na wymiar maksymalnej kliku, jaką znajdują w grafie o  $n = 200$  wierzchołkach. Pomiary z tab. 2 wykonano przy gęstości grafu  $q = 0,999$  i liczbie kroków mrówki-eksplorującej wynoszącej 800 i zmieniającej się liczbie okresów i cykli oraz mrówek. Pomiary z tab. 3 wykonano przy zmieniającej się gęstości grafu i przy stałej liczbie okresów, cykli oraz mrówek wynoszących – odpowiednio – 50, 50 i 50, a także stałej liczbie kroków mrówki-eksploracyjnej wynoszącej 1000, poza jednym pomiarem dla gęstości grafu  $q = 0,985$ , gdzie liczba kroków wynosiła 2000. W obu zamieszczonych poniżej tab. 2 i 3 zawarto średnie wartości wyznaczonych klik o największych wymiarach z 10 pomiarów.

Tabela 2

Średni wymiar kliku w funkcji liczby ścieżek feromonowych dla algorytmów mrówkowych

Liczba okresów	1	5	5	5	10	10	10	10	20	20
Liczba cykli	1	5	5	10	10	10	10	20	20	30
Liczba mrówek	1	5	10	10	10	20	30	30	30	30
Ścieżki feromonowe	1	125	250	500	1000	2000	3000	6000	12 000	18 000
Algorytm mrówkowy tylko z fazą pracy	180,4	182,4	182,9	181,8	183,6	183,1	183,8	184,4	183,6	184,4
Algorytm mrówkowy z fazą eksploracji	177,2	182,2	182,9	181,8	183,6	183,1	183,8	184,4	183,6	184,4

Z tabeli 2 wynika, że największe wymiary kliku znajdowane przez oba algorytmy są takie same poza niewielkim obszarem niskich wartości liczby okresów i liczby cykli oraz liczby mrówek.

Z tabeli 3 wynika, że oba algorytmy nie różnią się również w średnim wymiarze wyznaczanych największych kliku w zależności od gęstości grafu. Jedyna różnica pojawiła

się przy gęstości  $q = 0,985$ , tj. w sytuacji, gdy w grafie istnieje największa liczba klik o wymiarach wynoszących ok. połowy liczby wierzchołków grafu, która to różnica w wyznaczonym średnim wymiarze klik dla obu algorytmów została zlikwidowana przez podniesienie liczby kroków jakie musi wykonać mrówka w fazie eksploracji. Generalnie można więc stwierdzić, że jakość obu porównywanych algorytmów jest prawie identyczna.

Tabela 3

## Średni wymiar klik w funkcji gęstości grafów

$q$	0,5	0,9	0,95	0,985	0,985	0,99	0,995	0,999
Liczba kroków	1000	1000	1000	1000	2000	1000	1000	1000
Algorytm mrówkowy tylko z fazą pracy	11,0	38,5	58,1	103,3	104,2	119,9	145,2	182,4
Algorytm mrówkowy z fazą eksploracji	11,0	38,5	58,1	103,0	104,2	119,9	145,2	182,4

## 5. Porównanie algorytmu mrówkowego z algorytmem 457

Opracowany algorytm mrówkowy oparty o fazę eksploracji i pracę mrówki w celu wyznaczenia klik maksymalnej grafu został poddany testom, podczas których rozmiary klik otrzymanych algorytmem mrówkowym i algorytmem 457 [8] zostały porównane.

Wyniki porównań przeprowadzono dla grafów o gęstości wynoszącej 0,999, gdzie wartość gęstości jest prawdopodobieństwem, z którym była generowana krawędź pomiędzy dwoma wierzchołkami grafu, tzn. z takim prawdopodobieństwem dana krawędź istniała bądź też nie, i zestawiono je w tab. 4–6 dla różnych testów.

Tabela 4

## Średni wymiar klik w funkcji liczby ścieżek feromonowych

Liczba okresów	10	10	10	10	20	20	30	30	40	40	50
Liczba cykli	10	10	10	20	20	30	30	40	40	50	50
Liczba mrówek	10	20	30	30	30	30	30	30	30	30	30
Ścieżki feromonowe	1000	2000	3000	6000	12 000	18 000	27 000	36 000	48 000	60 000	75 000
Algorytm mrówkowy	183,7	183,1	183,9	184,1	183,8	183,9	182,8	182,8	185,3	180,7	183,9
Algorytm 457	184,4	183,6	184,5	184,6	184,2	184,3	183,2	183,1	185,4	180,8	183,9

Wszystkie parametry algorytmu mrówkowego są następujące: liczba wierzchołków grafu wynosi 200, liczba kroków mrówek-eksplorujących wynosi 800, liczba okresów jest

równa 50, liczba cykli wynosi 50, a liczba mrówek wynosi 30, prawdopodobieństwo konwersji z mrówki-eksplorującej w mrówkę-pracującą wynosi 0, więc mrówka-eksplorująca wykonuje wszystkie przewidziane dla niej kroki.

Liczbę ścieżek feromonowych otrzymuje się przez pomnożenie liczby okresów przez liczbę cykli i liczbę mrówek. Jeśli jakiś z parametrów algorytmu mrówkowego ulega zmianie podczas przeprowadzanego testu zostanie to zaznaczone, podczas gdy pozostałe wartości parametrów zostaną pominięte w opisie testu. Wszystkie wyniki są średnimi z 10 pomiarów.

Algorytm 457 potrafi wyznaczyć klikę maksymalną w grafie, więc wyniki przedstawione w tab. 2 wskazują, że algorytm mrówkowy wyznacza klikę maksymalną, gdy mrówki utworzą 75 000 ścieżek feromonowych. Liczba ścieżek feromonowych ulega zmianie podczas przeprowadzanych testów, których wyniki przedstawiono w tab. 4.

Przeprowadzone testy dotyczące doboru liczby kroków wykonywanych przez mrówki-eksplorujące do liczby wierzchołków w grafie wykazały, że odpowiednia ich liczba uzyskiwana jest wtedy, gdy jest ona ok. 4 razy większa niż liczba wierzchołków w grafie (zob. tab. 5) i przy takim stosunku największe wymiary klik grafu zostały znalezione przez algorytm mrówkowy.

Tabela 5

**Wymiar klik i stosunek liczby kroków mrówek-eksploratorów do liczby wierzchołków**

Liczba kroków/liczba wierzchołków	1,0	1,5	2,0	2,5	3,0	3,5	4,0
Algorytm mrówkowy	138,1	161,6	173,7	179,2	180,4	183,4	183,9
Algorytm 457	182,5	182,3	182,8	183,5	181,4	183,6	183,9

Należałoby z tego wywnioskować, że wymiar największej wyznaczonej klik przez opracowany algorytm zależy od sumarycznego prawdopodobieństwa wyboru wierzchołków wchodzących w skład tej klik  $\sum p_i$  z sąsiedztwa wierzchołka-mrowiska i wierzchołka  $s$ , w którym aktualnie znajduje się mrówka, w ciągu decyzyjnym od  $i = 1$  do  $i = \text{max-krok}$ . Im większa wartość  $\text{max-krok}$ , tym większe sumaryczne prawdopodobieństwo wyboru wierzchołka z największej klik w otoczeniu wierzchołka-mrowiska i tym mniejsza wartość sumarycznego prawdopodobieństwa niewybrania wierzchołka wchodzącego w skład klik o największym wymiarze z sąsiedztwa wierzchołka-mrowiska w ciągu decyzyjnym od  $i = 1$  do  $i = \text{max-krok}$ .

W algorytmie maksymalna liczba kroków wykonywanych przez mrówki-eksplorujące reprezentowana jest przez zmienną  $\text{max\_krok}$ .

Rezultaty testu wykazały, że mrówka-eksplorująca nie powinna powracać do mrowiska i przemienić się w mrówkę-pracującą zbyt szybko. Większy wymiar klik jest wyznaczony, gdy mrówka-eksplorująca rozpoczyna poszukiwanie klik po wykonaniu wszystkich przewidzianych kroków na eksplorację sąsiedztwa wierzchołka-mrowiska i zachodzi to przy  $r = 0,0$ , gdzie  $r$  jest prawdopodobieństwem konwersji z mrówki-eksplorującej w mrówkę-pracującą.

Wyniki testów przedstawiono w tab. 6, gdzie prawdopodobieństwo konwersji  $r$  z mrówki-eksplorującej w mrówkę-pracującą ulega zmianie od  $r = 0,0$  do  $r = 0,8$ .

Tabela 6

**Wymiar klikli i prawdopodobieństwo konwersji  $r$  z mrówki-eksplorującej w mrówkę-pracującą**

$r$	0,8	0,2	0,1	0,05	0,01	0,002	0,0
Algorytm mrówkowy	8,0	48,9	84,4	127,5	182,6	184,9	183,9
Algorytm 457	184,1	183,7	186,3	184,6	185,1	185,0	183,9

## 6. Algorytmy z przeszukiwaniem lokalnym

Prezentowane algorytmy wyznaczają klikli, które niekoniecznie są klikami o największych rozmiarach. Dla poprawy otrzymywanych wyników wprowadzone zostało przeszukiwanie lokalne dla już wyznaczonych klikli, które ma na celu poprawę otrzymanego rezultatu przez zwiększenie wymiaru klikli w wyniku przeszukiwania lokalnego. Przeszukiwanie lokalne dokonuje się dla największej klikli wyznaczonej przez mrówki z całej ich populacji. Istnieje wiele procedur lokalnego przeszukiwania wykorzystywanych do poprawy otrzymanego rozwiązania. W tym artykule prezentowane są dwie takie procedury.

Jedną z nich jest znana procedura (2,1)-wymiany [1], w której jeden wierzchołek z klikli jest zastępowany dwoma wierzchołkami spoza klikli, gdy po zastąpieniu uzyskuje się klikli o większym wymiarze. W skrócie procedurę tę można przedstawić następująco: Dla danej klikli  $C$  procedura ta szuka wierzchołków  $v_i$ ,  $v_j$  i  $v_k$  takich, że wierzchołek  $v_i$  należy do klikli  $C$ , wierzchołki  $v_j$  i  $v_k$  nie należą do klikli  $C$ , wierzchołki  $v_j$  i  $v_k$  są połączone krawędzią oraz wierzchołki  $v_j$  i  $v_k$  są połączone krawędziami ze wszystkimi wierzchołkami struktury grafu  $C - \{v_i\}$  i gdy takowe wierzchołki znajduje, zastępuje wierzchołek  $v_i$  wierzchołkami  $v_j$  i  $v_k$ , powiększając tym samym wymiar wyznaczonej klikli  $C$  o 1. Procedura ta powtarzana jest iteracyjnie do momentu osiągnięcia stanu lokalnego optimum, które nie może być poprawione tą procedurą (2,1)-wymiany.

Druga procedura prezentowana w tym artykule wykorzystuje metaheurystykę opartą na kolonii mrówek i będzie oznaczana ANT – *local procedura*. Procedura ta zastosowana jest do wierzchołków niewchodzących w skład klikli  $C$ . Wśród tych wierzchołków konstruowana jest klikli  $K$  o jak największym wymiarze przez kolonię mrówek. Następnie tworzona jest klikli będąca połączeniem klikli  $C$  i klikli  $K$  w ten sposób, że wszystkie wierzchołki klikli  $K$  wchodzi w jej skład, gdy tymczasem w jej skład wchodzi tylko te wierzchołki klikli  $C$ , które są połączone krawędziami ze wszystkimi wierzchołkami klikli  $K$ . Wymiar wyznaczonej w ten sposób klikli określa ilość feromonu pozostawianą na wszystkich wierzchołkach klikli  $K$  zgodnie ze wzorem

$$dt = \frac{1}{(1 + |C_{\max} - C_{KC}|)}$$

gdzie:

$C_{\max}$  – maksymalny wymiar klikli od początku działania algorytmu,

$C_{KC}$  – wymiar klikli otrzymanej w wyniku połączenia klikli  $K$  i  $C$ .

Wybór pierwszego wierzchołka wchodzącego w skład klikli  $K$  dokonuje się ze zbioru wierzchołków niewchodzących w skład wierzchołków klikli  $C$  w sposób losowy z jedna-

kowym prawdopodobieństwem, natomiast pozostałe wierzchołki wchodzące w skład klikli  $K$  wybiera się losowo spośród zbioru wierzchołków połączonych krawędziami z wierzchołkami tworzącymi klikę  $K$ , zbioru oznaczonego jako  $V_{SK}$ , z prawdopodobieństwem określonym wzorem

$$p(v_i) = \frac{t[v_i]}{\sum_{k \in V_{SK}} t[v_k]}$$

gdzie  $t[v_i]$  to ilość feromonu na wierzchołku  $v_i$ .

### 7. Porównanie algorytmów z przeszukiwaniem lokalnym

Porównaniu podlegały algorytmy wzbogacone o procedury lokalnego przeszukiwania, a mianowicie o procedurę (2,1)-wymiany i procedurę lokalnego przeszukiwania opartą na zachowaniu się kolonii mrówek. Porównania dokonano przy następujących parametrach: liczbie wierzchołków grafu równej  $n = 200$ , liczbie kroków mrówek-eksplorujących równej 1000, liczbie okresów równej 30, liczbie cykli równej 30 oraz liczbie mrówek równej 30 przy różnych gęstościach grafu  $q = \{0,5, 0,9, 0,95, 0,985, 0,99, 0,995 \text{ i } 0,999\}$ .

Z przeprowadzonych pomiarów zawartych w tab. 7 wynika, że najlepszą lokalną procedurą wspomagającą wyznaczanie największej klikli w grafie jest procedura lokalna (2,1)-wymiany wierzchołków.

Tabela 7

Porównanie wymiarów klik wyznaczonych przy różnych przeszukiwaniach lokalnych

$q$	0,999	0,995	0,99	0,985	0,95	0,9	0,5
Mrówka-robotnik	184,4	144,2	121,1	101,1	56,0	36,9	10,5
Mrówka-eksplorator	183,7	142,5	119,8	101,5	56,3	37,2	10,1
Mrówka-robotnik – (2,1)-wymiana	183,9	144,9	123,7	105,6	60,3	40,0	10,6
Mrówka-eksplorator – (2,1)-wymiana	183,9	144,9	123,5	105,2	59,7	39,8	10,6
Mrówka-robotnik procedura ANT	183,9	144,0	120,8	101,5	55,7	36,4	10,5
Mrówka-eksplorator procedura ANT	183,3	143,0	119,6	100,1	56,0	37,1	10,3

### 8. Podsumowanie

Prezentowane w niniejszym artykule algorytmy wykorzystują dwie główne aktywności w życiu mrówek: ich pracę i ich eksplorację otoczenia mrowiska. Z uzyskanych rezultatów wynika, że im lepsza eksploracja otoczenia mrowiska, tym lepsze uzyskuje się rozwiązanie problemu wyznaczania klikli maksymalnej grafu. Z jednej więc strony, nie należy się spie-

szyc z wprowadzaniem fazy pracy mrówek, z drugiej zaś – należy pamiętać, że faza eksploatacji ograniczona jest nasyceniem wierzchołków ilością złożonego na nich feromonu. Porównanie z algorytmem 457 wskazuje, że prezentowane w tym artykule algorytmy mrówkowe wyznaczają przy pewnych określonych i wyznaczonych parametrach klikę maksymalną grafu, jak np. stosunek liczby kroków mrówek do liczby wierzchołków przeszukiwanego grafu czy też odpowiednio dobrana liczba mrówek, bądź też odpowiednio dobrana liczba cykli i okresów. Badanie zastosowanych procedur lokalnego przeszukiwania wykazało, że procedura (2,1)-wymiany wierzchołków daje lepsze rezultaty niż procedura lokalnego przeszukiwania oparta na metaheurystyce zachowania się kolonii mrówek. Z porównania procedur lokalnego przeszukiwania wynika także, że procedura (2,1) jest zdecydowanie szybszą procedurą. Ogólnie można stwierdzić, że przy algorytmach mrówkowych należy zawsze rozważyć kompromis pomiędzy dokładnością uzyskiwanego rozwiązania a czasem potrzebnym na uzyskanie większej dokładności tego rozwiązania.

#### Literatura

- [1] Abello J., Pardalos P.M., Resende M.G.C., *On Maximum Clique problem in very large graph*, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 50, American Mathematical Society, 2000, 119-130.
- [2] Auguston G., Minker J., *Analysis of Some Graph theoretic Cluster Techniques*, J.ACM 17 (1970), 571-588.
- [3] Karp R.M., *Reducibility among combinatorial problems*, in *Complexity of Computer Computation*, [in:] R.E. Miller, J.W. Thatcher (eds.), Plenum Press, New York 1972, 85-103.
- [4] Johnson D.S., *Approximation algorithms for the combinatorial problems*, Journal of Computer Science 9, 1974, 256-278.
- [5] Battiti R., Protasi M., *Reactive local search for the maximum clique problem*, Algorithmica 29, 2001, 610-637, praca dostępna w GOOGLE.
- [6] Grosso A., Locatelli M., Della Croce F., *Combining swaps and node weights in an adaptive greedy approach for the maximum clique problem*, Journal of the Heuristics 10, 2004, 135-152.
- [7] Jagota A., Sanchis L.A., *Adaptive, restart, randomized greedy heuristics for the maximum clique*, Journal of the Heuristics 7, 2001, 565-585.
- [8] Bron C., Kerbosch J., *Algorithm 457: Finding all cliques of an undirected graph*, Comm. ACM 16, 1973, 575-577.
- [9] Glover F., Laguna M., *Tabu search*, [in:] *Modern Heuristics Technics for Combinatorial Problems*, Blackwell Scientific Publishing, Oxford, UK 1993, 70-141.
- [10] Aarts E.H.L., Korst J.H.M., *In simulated annealing and the Boltzmann machines*, John Wiley & Sons, Chichester, UK 1989.
- [11] Marchiori E., *Genetic, iterated and multistart local search for the maximum clique problem*, [in:] *Application of Evolutionary Computing, Proc. of Eco Workshops 2002: EvoCOP, EvoIASP, EvoSTim, 2279 of LNCS*, Springer-Verlag, 2002, 112-121, praca dostępna w GOOGLE.



- [12] Dorigo M., Di Caro G., *The Ant Colony Optimization meta-heuristics*, [in:] D. Corne, M. Dorigo, F. Glover (eds.) *New Ideas in Optimization*, McGraw-Hill, UK 1999, 11-32.
- [13] Dorigo M., Gambardella L.M., *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Transaction on Evolutionary Computation 1, 1997, 53-66.
- [14] Elliman D.G., Youssef S.M., *Reactive Prohibition-based Ant Colony Optimisation (RPACO): A New Parallel Architecture for Constrained Clique Sub-Graphs*, Computer Science Technical Report, No. NOTTCS-TR-2004-7, School of Computer Science and Information Technology, University of Nottingham, UK 2004, praca dostępna w GOOGLE.
- [15] Maniezzo V., Colorni A., *The Ant system applied to the quadratic assignment problem*, IEEE Transaction on Data and Knowledge Engineering 11, 1999, 769-778.
- [16] Solnon Ch., Fenet S., *Investigating ACO capabilities for solving the Maximum clique Problem (raport)*, University Lyon I, May 24, 2004, praca dostępna w GOOGLE.
- [17] Costa D., Herz A., *Ants can colour graphs*, Journal of the Operational Research Societz 48, 295-305.
- [18] Corne D., Dorigo M., Glover F., *New ideas in Optimization*, The McGraw-Hill Companies, Londyn, UK 1999.
- [19] Leguizamon G., Michalewicz Z., *A new version of ant system for subsets problems*, [in:] *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*, Washigton, USA, praca dostępna w GOOGLE.