

ALHE Dokumentacja

Artificial Bee Colony na sieciach

SK.ALHE.10

Michał Iwanek i Wojciech Wolny
pod opieką Dr Inż. Stanisława Kozdrowskiego

1. Temat

Wstępnym tematem było zastosowanie dla sieci [zhttp://sndlib.zib.de/home.action](http://sndlib.zib.de/home.action) algorytmu sztucznej kolonii pszczół (Artificial Bee Colony) do znalezienia najlepszych (wg. Ustalonej metryki) ścieżek w danej sieci. Wziąć do testów co najmniej 3 przykładowe sieci. Po konsultacjach zostało postanowione, że tematem będzie stworzenie programu dla jednej sieci Polski z pliku podesłanego przez prowadzącego projekt na podstawie danych Polska z wyżej wymienionej strony internetowej. Wszelkie ograniczenia, funkcja kosztu oraz informacja o zapisie danych w pliku została podana w artykule "Ultra-Wideband WDM Optical Network Optimization" pod autorstwem S. Kozdrowskiego, M. Zotkiewicza oraz S. Sujeckiego.

2. Analiza Zadania:

Celem zadania jest znalezienie najtańszej konfiguracji transponderów realizujących wszystkie zapotrzebowania transmisyjne dla wybranej sieci telekomunikacyjnej.

Założenia:

1. Do wyboru mamy kilka rodzajów transponderów, każdy o podanej mocy i koszcie.
2. Transpondery zostają ustawione na daną ścieżkę w konkretnym zapotrzebowaniu i na każdej krawędzi w tej ścieżce posiadają ten sam pasek częstotliwości startowy.
3. Wszystkie zapotrzebowania w sieci telekomunikacyjnej muszą zostać spełnione.
4. Każdy typ transpondera posiada pewną potrzebną do prawidłowego funkcjonowania ilość pasków częstotliwości.
5. Jednego pasku częstotliwości na jednej krawędzi nie może wykorzystywać więcej niż jeden transponder.
6. Funkcja celu jest zadana i uwzględnia szereg dodatkowych czynników takich jak długość krawędzi, strata mocy na 1 km przy wykorzystaniu danej długości fali.

W celu znalezienia najlepszego rozwiązania należało zoptymalizować wybór zbioru odpowiednich transponderów dla zapotrzebowania, które realizowałyby całe zapotrzebowanie najmniejszym kosztem. Następnym ważnym elementem rozwiązania było maksymalne możliwe wykorzystanie tańszego pierwszego pasu częstotliwości. Dodatkowo każde zapotrzebowanie było reprezentowane przez trzy ścieżki. Ścieżki były ustawione od najkrótszej do najdłuższej. Ze względu na dodatkowy koszt pochodzący z faktu wykorzystywania kolejnych pasków należało stworzyć rozwiązanie, które korzystałoby z preferencji dla ścieżek.

3. Wykorzystywane biblioteki i programy

- biblioteki Python:
 - random – generacja liczb pseudolosowych
 - scipy.constants – stałe fizyczne
 - Matplotlib – rysowanie wykresów

4. Wytworzone pliki:

- **/network** – zawiera pliki, które tworzą reprezentację wczytanej sieci
 - demand.py
 - klasa zapotrzebowania w sieci
 - path.py
 - klasa ścieżki dla zapotrzebowania
 - transponder.py
 - klasa informująca o budowie transpondera
 - network.py
 - klasa reprezentująca sieć z jej informacjami
- **/solution** – zawiera pliki dotyczące reprezentacji pojedynczego rozwiązania w tym:
 - solution_network.py
 - klasa reprezentująca rozwiązanie pojedynczej sieci
 - solution_demand.py
 - klasa reprezentująca i informująca o pokryciu konkretnego zapotrzebowania
 - solution_transponder.py
 - klasa reprezentująca transponder wykorzystany w rozwiązaniu dla zapotrzebowania
- **/ABC** – zawiera pliki dotyczące algorytmu sztucznej kolonii pszczół dla podanej wcześniej reprezentacji rozwiązania
 - bee_colony.py
 - główny plik kontrolujący przydział rozwiązań do odpowiednich pszczół
 - onlooker.py
 - plik posiadający implementację obserwatorów posiadające dostęp do rozwiązania i funkcję opracowujące je
 - scout.py
 - plik posiadający implementację zwiadowczyń, które resetują aktualne rozwiązanie i przygotowujące losowe inne
 - worker.py
 - plik posiadający implementację robotnic, które przeszukują potencjalne sąsiednie rozwiązania obecnego
- environment.py
 - plik posiadający reprezentację środowiska rozwiązań, na których zapuszczamy algorytm sztucznej kolonii pszczół
- main.py
 - główny plik uruchamiający program

5. Metoda rozwiązania

W rozwiązaniu zastosowaliśmy Algorytm Sztucznej Kolonii Pszczół. Algorytm Sztucznej Kolonii Pszczół (Artificial Bee Colony, ABC) jest algorytmem optymalizacyjnym, który ma na celu odzwierciedlenie zachowania roju pszczoł. W algorytmie mamy 3 typy osobników. Pierwszą stanowią pszczoły robotnice, każda osobniczka w tej grupie przeszukuje najbliższą okolicę zapamiętanego miejsca i wybiera najlepsze rozwiązanie. Robotnice informują o efektach swojej pracy pszczoły zwiadowcze i obserwatorce. Zwiadowcy wybierają pseudolosowe miejsca w środowisku i informują o nich kolonię. Trzecim rodzajem osobniczek są obserwatorce. Obserwatorce wybierają te miejsca, które dają najlepsze wyniki wśród zwiadowczyń i robotnic i przeszukują te obszary. Zaletą ABC jest fakt, że łączy w sobie przeszukiwanie lokalne jak i globalne. Lokalnym zajmują się robotnice i obserwatorce, a globalnym zwiadowcy i obserwatorce. Dzięki takiemu rozwiązaniu zostaje zachowany balans między eksploracją i eksploatacją środowiska.

Nasze podejście:

W naszym rozwiązaniu każda pszczoła ma dokładnie zdefiniowaną rolę i ściśle współpracuje z innymi pszczołami:

- a) **Zwiadowczynie** - resetują podane im rozwiązanie w **każdej** iteracji, po czym po kolei idąc od pierwszego paska częstotliwości losowo dobierają zbiory ścieżek, które możliwie maksymalnie zapełniają pasek na wszystkich krawędziach sieci. Dzięki takiemu podejściu minimalizujemy liczbę wolnych przestrzeni w potencjalnych rozwiązaniach. Mają one także określoną, maksymalną liczbę operacji, które mogą wykonać w danej iteracji. Rozwiązanie wygenerowane przez Zwiadowczynie nie musi być konieczne poprawne (niektóre zapotrzebowania mogą nie być wcale spełnione). Pozwala to jednak Robotnicom na dalszą eksplorację tego potencjalnego rozwiązania. Po każdej iteracji rozwiązanie, które jest obiecującym materiałem dla robotnic może trafić do specjalnej puli "obiecujących rozwiązań", pod warunkiem, że jest ono lepsze od któregośkolwiek z obecnie dostępnego w puli.
- b) **Robotnice** - mają dwa scenariusze przeszukiwania. Jeśli wymagania zapotrzebowania nie zostały spełnione dla danego rozwiązania, to robotnice wybierają niespełnione zapotrzebowanie i starają się je uzupełnić szukając miejsca w sieci, w które mogłyby wstawić brakujący transponder. Jeśli wszystkie zapotrzebowania są spełnione to robotnice starają się przenieść transpondery z drugiego pasa na pierwszy oraz zmienić ścieżkę dla niektórych zapotrzebowania na krótszą. Każde poprawne rozwiązanie może trafić do puli N "najlepszych rozwiązań", pod warunkiem, że jest lepsze od któregośkolwiek z obecnych w puli. W przypadku, gdy przez X iteracji rozwiązanie wygenerowane przez robotnice nie poprawia się, wtedy robotnica bierze nowe rozwiązanie z puli "obiecujących" rozwiązań i rozpoczyna na nowo pracę.

- c) **Obserwatorki** - biorą podane zapotrzebowanie po czym usuwają wszystkie transpondery powyżej podanego poziomu i uzupełniają zapotrzebowania na nowo. Każde poprawne rozwiązanie może trafić do puli N “najlepszych rozwiązań”, pod warunkiem, że jest lepsze od któregośkolwiek z obecnych w puli. W przypadku, gdy przez X iteracji rozwiązanie wygenerowane przez obserwatorkę nie poprawia się, wtedy pszczoła bierze nowe rozwiązanie z puli “najlepszych rozwiązań” i rozpoczyna na nowo pracę.

Kolonia pszczół krok po kroku:

Znajdź_rozwiazania_startowe_dla_pszczol

FOR i = 0; i < MAKSYMALNA_LICZBA_ITERACJI; i++ {

 FOR robotnica : robotnice {
 znajdź_nowe_rozwiazanie

 IF limit_prób_robotnicy_przekroczony {
 wylosuj_nowe_rozwiazanie_z_puli_obiecujacych_rozwiazan
 }
 zaktualizuj_pule_rozwiazan_akceptowalnych

 IF nowe_rozwiazanie_lepsze_od_obecnie_najlepszego
 AND nowe_rozwiazanie_spełnia_ograniczenia {
 najlepsze_rozwiazanie_kolonii = robotnica.nowe_rozwiazanie
 }
 }

 FOR obserwatorka : obserwatorki {
 znajdź_nowe_rozwiazanie

 IF limit_prób_obserwatorki_przekroczony {
 wylosuj_nowe_rozwiazanie_z_puli_rozwiazan_akceptowalnych
 }
 zaktualizuj_pule_rozwiazan_akceptowalnych

 IF nowe_rozwiazanie_lepsze_od_obecnie_najlepszego
 AND nowe_rozwiazanie_spełnia_ograniczenia {
 najlepsze_rozwiazanie_kolonii = obserwatorka.nowe_rozwiazanie
 }
 }

 FOR zwiadowczyni : zwiadowczynie {
 zresetuj_obecne_rozwiazanie
 znajdź_nowe_rozwiazanie

 zaktualizuj_pule_rozwiazan_akceptowalnych
 zaktualizuj_pule_rozwiazan_obiecujacych

 IF nowe_rozwiazanie_lepsze_od_obecnie_najlepszego
 AND nowe_rozwiazanie_spełnia_ograniczenia {
 najlepsze_rozwiazanie_kolonii = zwiadowczyni.nowe_rozwiazanie
 }}}

Wyniki:

Problem optymalizacji sieci telekomunikacyjnej sprowadza się do kwestii, którą ścieżkę powinniśmy wybrać dla danego zapotrzebowania i jakich transponderów powinniśmy użyć. Chcemy przy tym wykorzystać jak najwięcej slice'ów w pierwszym paśmie, które jest po prostu tańsze. W zależności od zapotrzebowania dla danej sieci zadanie może okazać się trywialne lub wręcz niemożliwe do wykonania. Dla potrzeb badawczych bazowe zapotrzebowania z pliku res.dat zostały przemnożone przez odpowiedni współczynnik.

1) Ogromne zapotrzebowania (rzędu xxxxx GB)

Znalezienie rozwiązania w tym przypadku jest po prostu niemożliwe. Albo zabraknie nam slice'ów w drugim paśmie i nie wszystkie zapotrzebowania zostaną spełnione (ograniczenie 1) albo ograniczenie 2 zostanie przekroczone.

2) Niewielkie zapotrzebowania (rzędu xx GB)

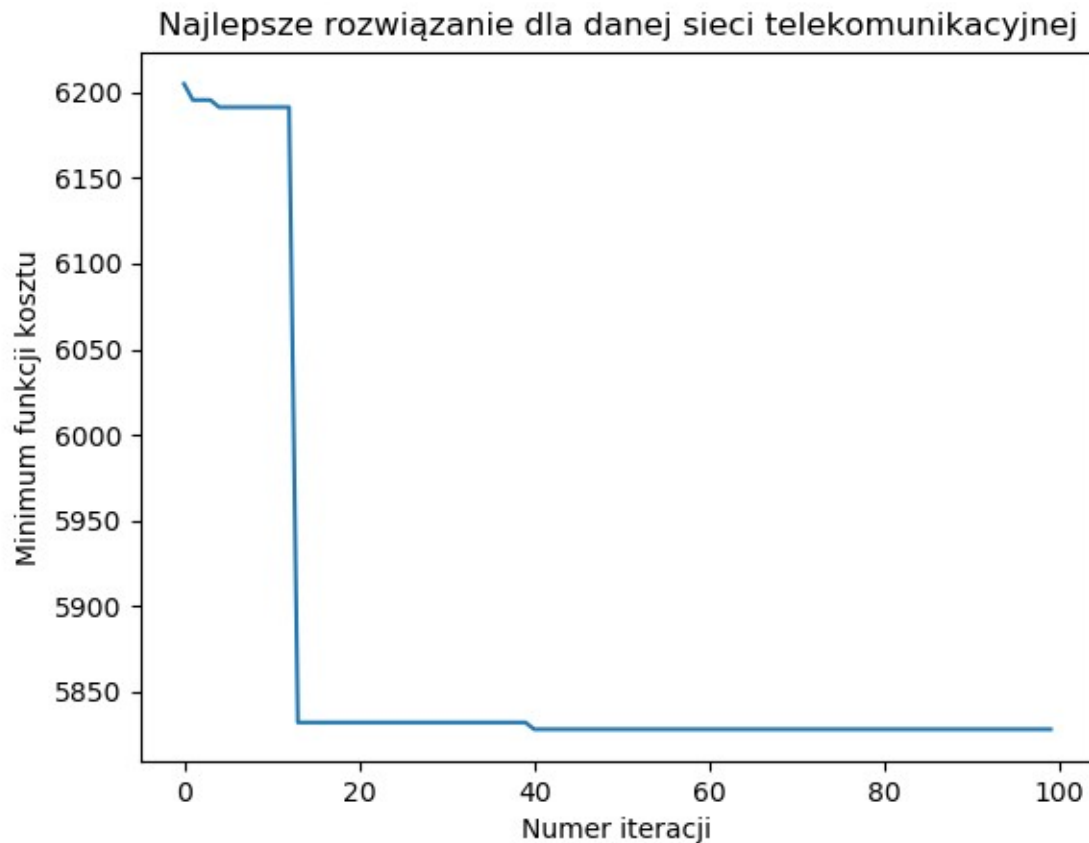
Znalezienie rozwiązania w tym przypadku staje się trywialne i nie ma potrzeby w tym przypadku stosować skomplikowanych algorytmów. Jak widać na rysunku optymalne rozwiązanie zostaje znalezione natychmiast i nie ma już co optymalizować. Jest to najlepsze matematycznie rozwiązanie jakie w ogóle moglibyśmy uzyskać.



3) Duże zapotrzebowania (rzędu xxxx GB)

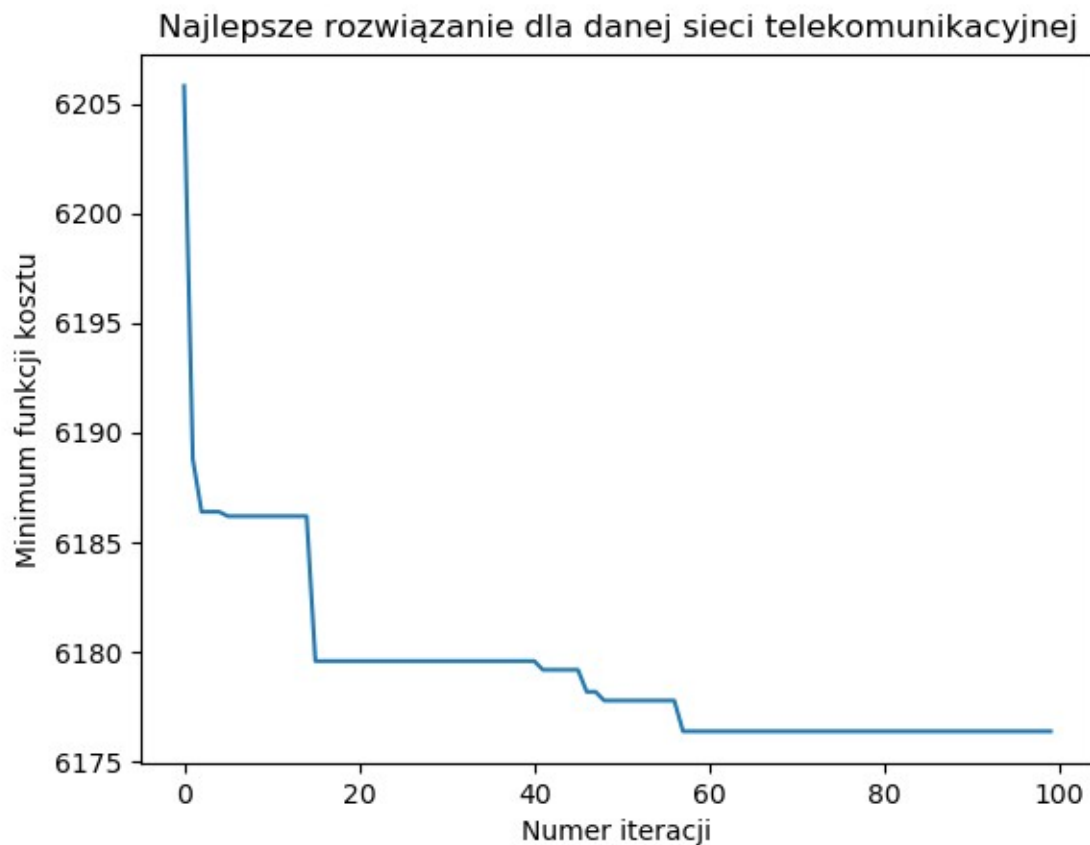
Jest to dla nas najciekawszy przypadek. Aby zrealizować zapotrzebowania konieczne zaczyna być wykorzystanie slice'ów z drugiego pasma. Co więcej nie wszystkie wygenerowane losowania muszą być akceptowalne. Problem staje się zatem nietrywialny. Możemy tu wreszcie zauważyć efekty współpracy różnych rodzajów pszczoł.

a) 3 robotnice, 5 obserwatorek, 2 zwiadowczynie



W okolicach 16 iteracji zakończyła się eksploracja optimum lokalnego dla robotnic, przez co na wykresie możemy znaczący spadek kosztu najlepszego rozwiązania. Pomimo to udało się jeszcze nieco poprawić rezultat w okolicach 40 iteracji.

b) 0 robotnic, 9 obserwatorek, 1 zwiadowczyni



W tym przypadku nie ma robotnic, które eksplorowałyby optima lokalne. Poprawa wyniku rozwiązania następuje zdecydowanie dłużej, lecz ostateczne rozwiązanie jest dużo gorsze. Brakuje tu synergii zwiadowczyń i robotnic, gdzie pierwsze pseudolosowo znajdują potencjalnie opłacalne rozwiązania częściowe, a robotnice dalej je eksplorują uzyskując często bardzo dobry rezultat.