

Algorytmy ewolucyjne i metaheurystyki

Sprawozdanie 2

1. Opis etapu projektu

Celem kolejnego etapu projektu było wykorzystanie przygotowanego podczas ćwiczenia 1 środowiska testowego. W projekcie należało wykorzystać naiwny algorytm przydziału punktów do grup oraz algorytm losowy, które stanowiły podstawę tworzenia rozwiązania startowego dla algorytmów lokalnego przeszukiwania przygotowanych podczas tego etapu projektu.

Jako naiwny algorytm wykorzystano algorytm przydziału punktu do grupy (bez żalu), natomiast algorytm losowy został oparty o całkowicie losowy przydział każdego punktu do jednej z 20 grup.

W ramach projektu przygotowano dwa algorytmy przeszukiwania lokalnego. Pierwszym z nich jest algorytm *Greedy Local Search*, natomiast drugim *Steepest Local Search*.

Podobnie jak w poprzednim etapie, funkcja celu została zdefiniowana jako *minimalizacja średniej odległości wszystkich par obiektów umieszczonych w ramach tej samej grupy*.

W rozdziale 2 zaprezentowano pseudokody przygotowanych algorytmów, natomiast w rozdziale 3 wyniki działania algorytmów dla 100 iteracji. Ostatni rozdział dotyczy wizualizacji najlepszych uzyskanych rozwiązań.

2. Pseudokody algorytmów przeszukiwania lokalnego

2.1. Generowanie listy ruchów

W algorytmach Greedy oraz Steepest Local Search wykorzystano metodę tworzącą listę możliwych ruchów dla aktualnego przydziału punktów do grup. Aby ułatwić zrozumienie algorytmów, wydzielono ją z pseudokodów i zaprezentowano osobno:

```
Zainicjalizuj listę ruchów jako pustą listę
Dla każdego punktu p z listy możliwych punktów {
  Dla każdej grupy g z listy grup {
    Jeżeli punkt p nie należy do grupy g {
      Do listy możliwych ruchów dodaj potencjalne przesunięcie punktu p
        z obecnej grupy do grupy g
    }
  }
}
Jako wynik metody zwróć listę ruchów
```

2.2. Greedy Local Search

```
Wykonuj dopóki przydział punktów do grup ulega zmianie {
  Przygotuj listę możliwych ruchów dla obecnego sąsiedztwa
  Dokonaj losowego posortowania listy ruchów

  Dla każdego ruchu r z listy możliwych ruchów {
    Oblicz deltę zmiany funkcji celu dla ruchu r
    Jeżeli delta poprawia wartość funkcji celu {
      Oznacz że dokonano zmiany przydziału punktów (kolejna iteracja możliwa)
      Zaaplikuj ruch r
      Dokonaj aktualizacji aktualnej wartości funkcji celu
      Przerwij pętlę sprawdzania ruchów
    }
  }
}
```

2.3. Steepest Local Search

```
Wykonuj dopóki przydział punktów do grup ulega zmianie {
  Przygotuj listę możliwych ruchów dla obecnego sąsiedztwa

  Dla każdego ruchu r z listy możliwych ruchów {
    Oblicz deltę zmiany funkcji celu dla ruchu r
    Jeżeli delta poprawia wartość funkcji celu {
      Zapamiętaj ruch jako dotychczas najlepszy
      Dokonaj aktualizacji aktualnej wartości funkcji celu
    }
  }

  Jeżeli dokonano zapamiętania ruchu (zmieniono funkcję celu) {
    Zaaplikuj najlepszy ruch (najbardziej poprawiający funkcję celu)
    Oznacz że dokonano zmiany przydziału punktów (aby wykonać kolejną iterację)
  }
}
```

3. Wyniki eksperymentów obliczeniowych

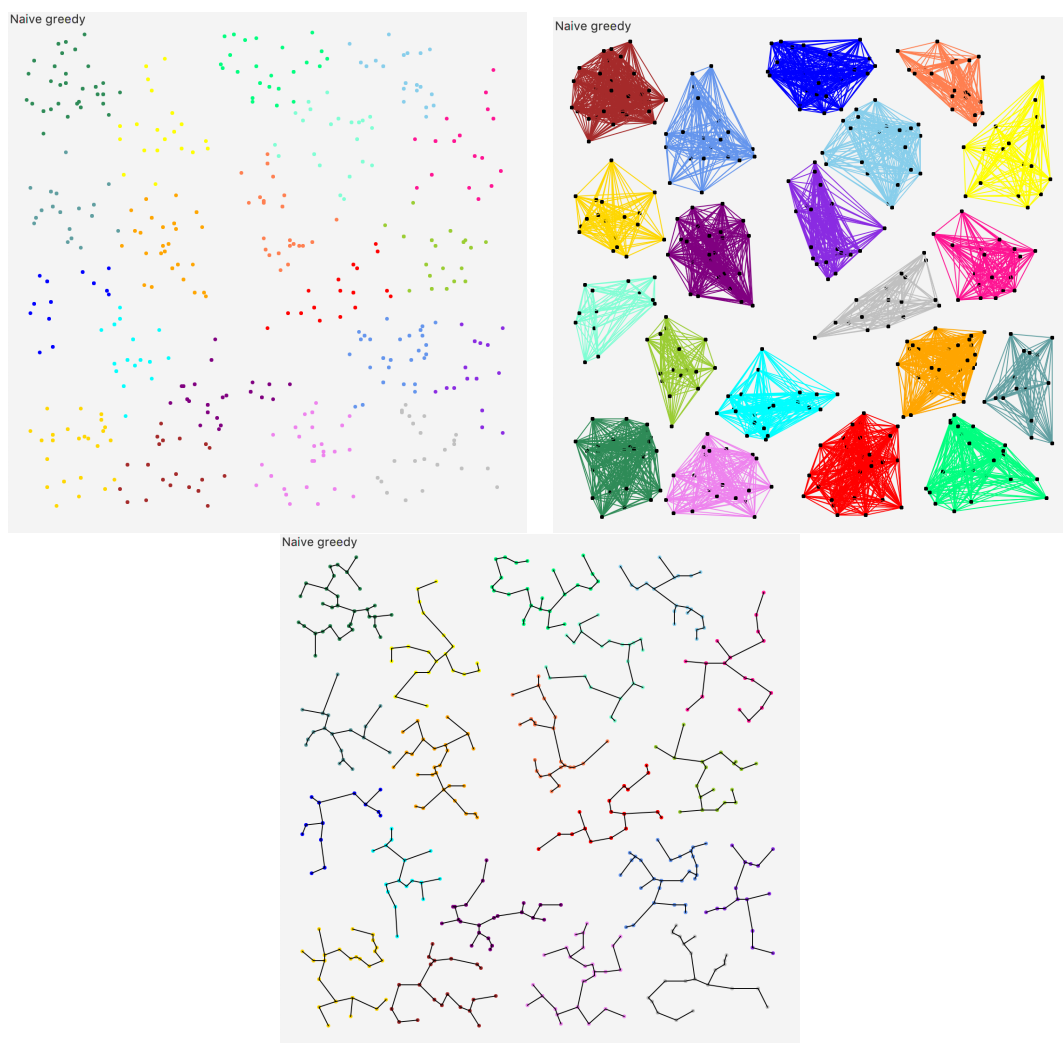
W tabeli 1 zaprezentowano wyniki eksperymentów obliczeniowych. Dokonano 100 powtórzeń obliczeń. Za każdym razem algorytmy zostały uruchomione dla dwóch rozwiązań startowych. Pierwszy z nich to wynik działania algorytmu naiwnego przygotowanego w poprzednim ćwiczeniu, a drugi to przydział przygotowany przez algorytm losowego przydziału punktów do grup.

Tabela 1. Wyniki eksperymentów obliczeniowych dla 100 iteracji

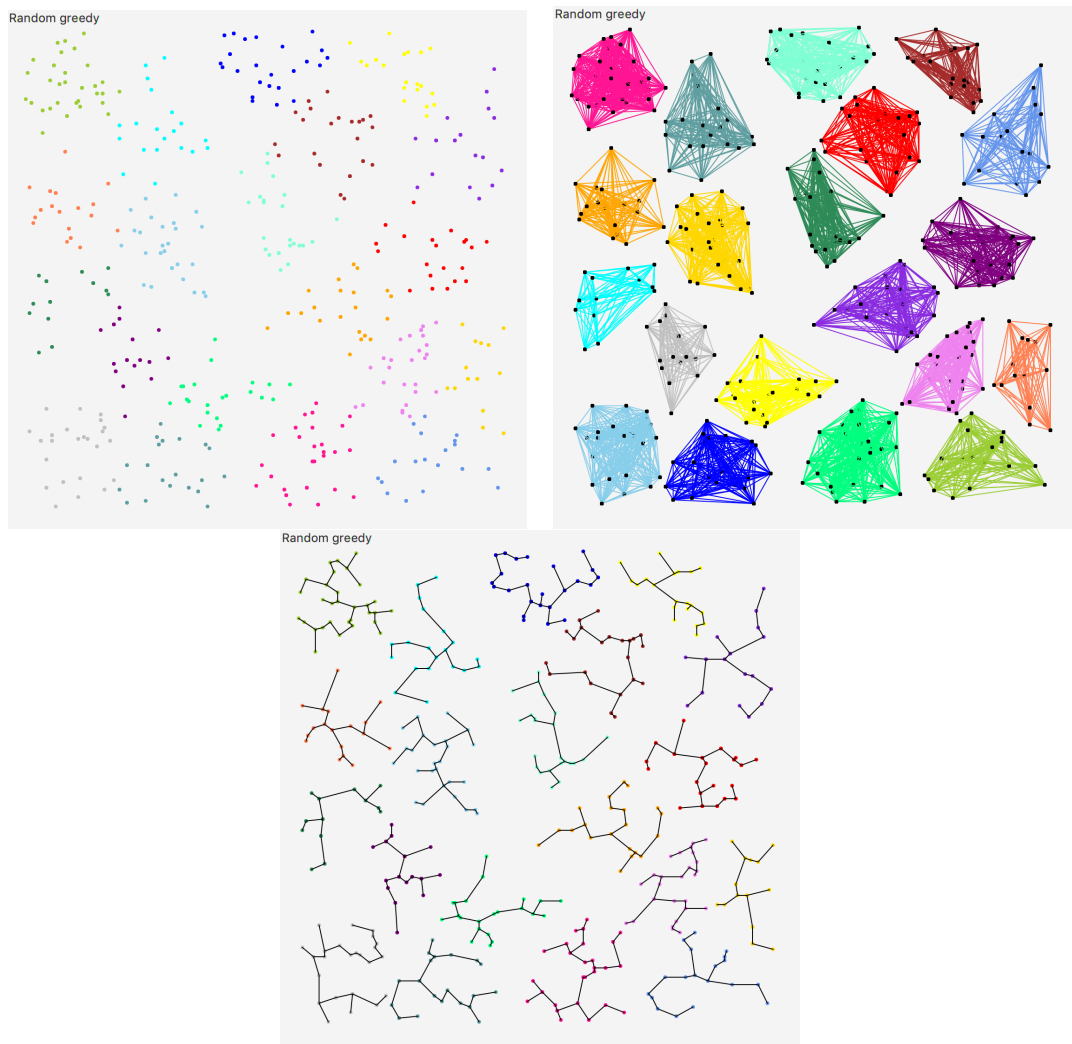
Cecha	Naive Greedy LS	Random Greedy LS	Naive Steepest LS	Random Steepest LS
Wartość minimalna funkcji celu	26.39	26.37	26.39	26.39
Wartość maksymalna funkcji celu	29.07	27.99	28.82	28.86
Wartość średnia funkcji celu	27.00	26.95	27.27	27.12
Wartość minimalna czasu obliczeń [sec]	0.11	0.34	0.83	4.50
Wartość maksymalna czasu obliczeń [sec]	0.60	0.71	3.28	7.51
Wartość średnia czasu obliczeń [sec]	0.19	0.41	1.48	5.52

4. Wizualizacja najlepszych rozwiązań

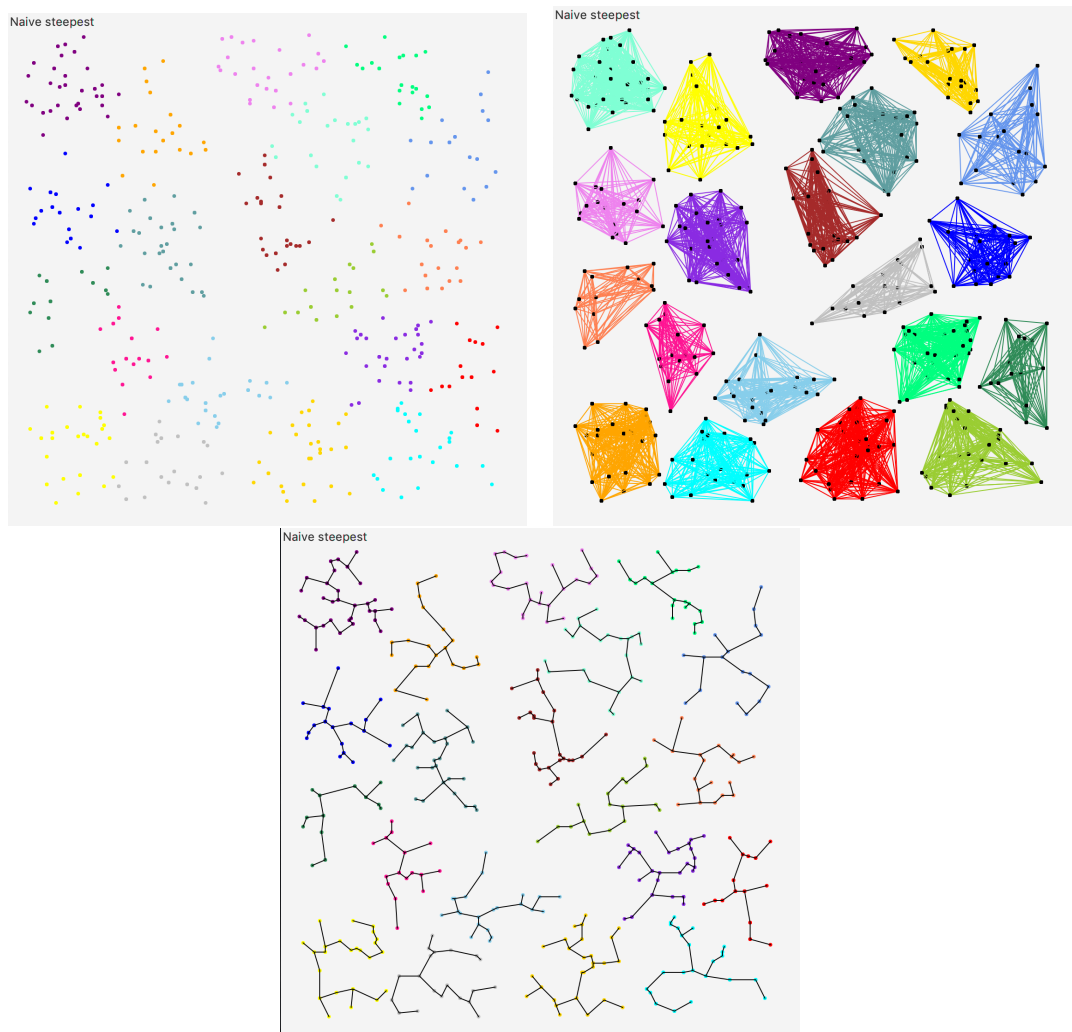
W wizualizacji najlepszych rozwiązań wykorzystano trzy sposoby prezentacji rozwiązań. Pierwszy z nich to zaprezentowanie samych grup punktów, bez jakichkolwiek powiązań. Drugim sposobem jest prezentacja zgodna z funkcją celu, czyli zaprezentowanie powiązań pomiędzy punktami w ramach grupy. Ostatni sposób wykorzystuje minimalne drzewo rozpinające, które w przejrzysty sposób prezentuje przydział punktów do grup.



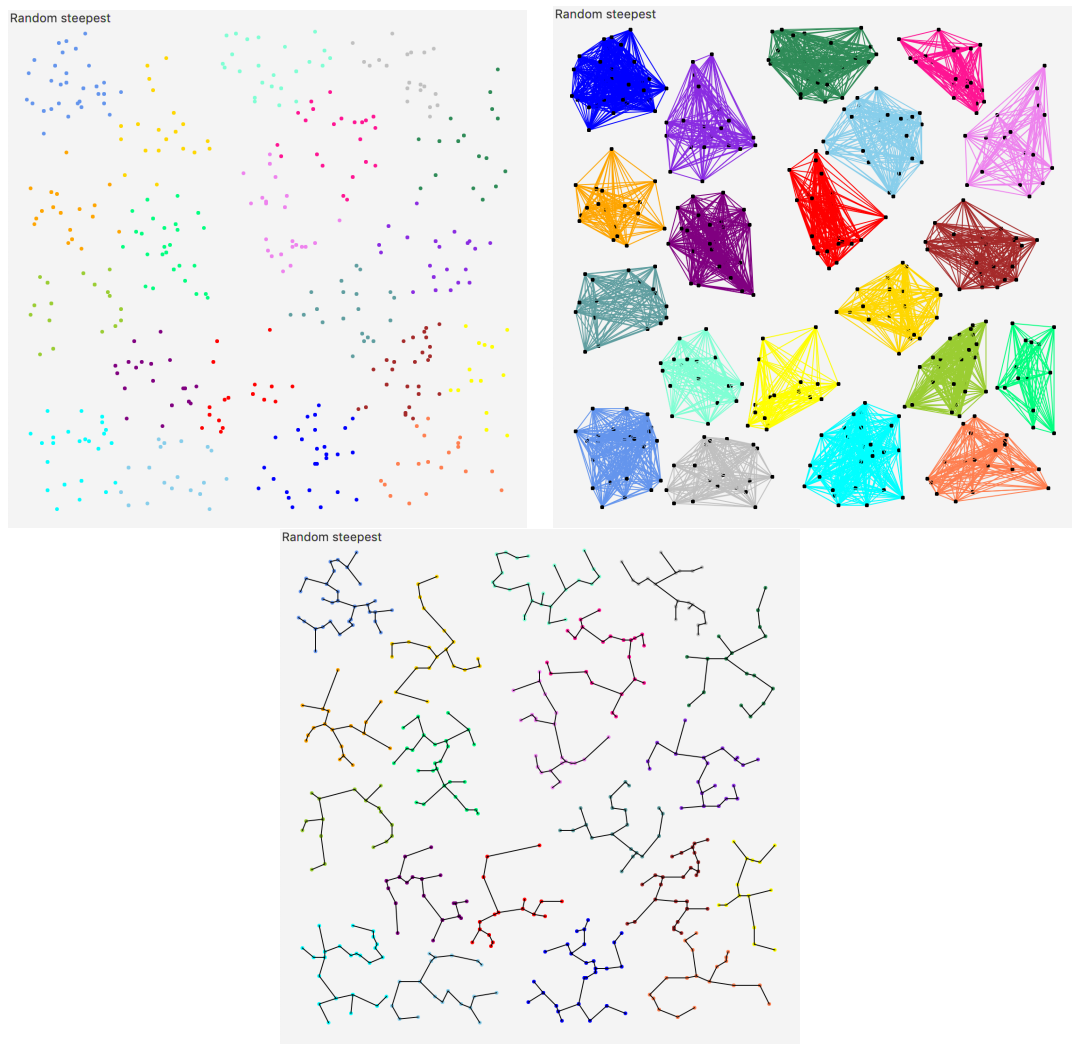
Rysunek 1. Naive Greedy Local Search



Rysunek 2. Random Greedy Local Search



Rysunek 3. Naive Steepest Local Search



Rysunek 4. Random Steepest Local Search