

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
<b>2</b>	<b>Przegląd literatury</b>	<b>5</b>
2.1	Przegląd - notatki . . . . .	5
2.2	O przestrzeni rozwiązań . . . . .	9
2.3	Sieć optimów lokalnych . . . . .	9
<b>3</b>	<b>Badania eksperymentalne</b>	<b>13</b>
3.1	Opis eksperymentów . . . . .	13
3.2	Zaimplementowane algorytmy . . . . .	13
3.3	Wyniki . . . . .	13
<b>4</b>	<b>Opis implementacji</b>	<b>15</b>
<b>5</b>	<b>Podsumowanie</b>	<b>17</b>
	<b>Dodatki</b>	<b>19</b>
<b>A</b>	<b>Jakiś dodatek</b>	<b>21</b>
	<b>Literatura</b>	<b>21</b>
	<b>Spis rysunków</b>	<b>23</b>
	<b>Spis tablic</b>	<b>25</b>



# Rozdział 1

## Wstęp



# Rozdział 2

## Przegląd literatury

### 2.1 Przegląd - notatki

#### Mapping the global structure of TSP fitness landscapes (2018) [3]

Celem pracy było zbadanie przestrzeni rozwiązań różnych instancji problemu komiwojażera (TSP), oraz opracowanie procedury próbkowania tych przestrzeni. W pracy przedstawiony został algorytm próbkujący oparty o algorytm *Chained-Lin-Kernighan*. Do próbkowania wykorzystano implementację tego algorytmu z biblioteki *Concorde*. Algorytm konstruuje sieć optimów lokalnych (LON) - odmianę z definicją krawędzi *escape edges*.

Wykonane zostało 1000 niezależnych przebiegów algorytmu. Połowa z nich korzystała z metody inicjalizacji Quicka-Borůvki, a druga połowa za punkt startowy przyjmowała losowe rozwiązanie. Algorytm opiera się na generowaniu nowych rozwiązań poprzez zastosowanie perturbacji typu double-bridge, a następnie uruchomieniu algorytmu Lin-Kernighan, by znaleźć lokalne minimum. Operacja jest powtarzana 10000 razy i budowana jest w ten sposób sieć optimów lokalnych.

Jednym z zadań, które postawili sobie autorzy pracy, była identyfikacja struktur lejowych w przestrzeni rozwiązań. Lej jest sekwencją lokalnych optimów, których wartość dopasowania się nie pogarsza, i która zmierza do pewnego optimum - globalnego (lej pierwszorzędny - ang. primary funnel) lub lokalnego (lej drugorzędny - ang. secondary funnel). To optimum nazywane jest spodem leja (ang. funnel floor/bottom).

Sieci optimów lokalnych wygenerowane przez kolejne uruchomienia algorytmu próbkującego zawyżały liczbę lejów w przestrzeni rozwiązań. Badacze scalili więc znalezione optima lokalne z wszystkich przebiegów algorytmu. Znalezione w ten sposób rozwiązania nazwali atraktorami i utworzyli graf sieci atraktorów, w której są one wierzchołkami. Następnie spody lejów zostały zidentyfikowane jako wierzchołki, z których nie ma krawędzi wychodzących. Przynależność optimów lokalnych do poszczególnych lejów odbywała się poprzez sprawdzenie, czy istnieje ścieżka w grafie pomiędzy danym optimum a spodem leja.

Do badań wykorzystano 20 instancji TSP o rozmiarach od 500 do 1500 miast. Wykorzystano instancje o miastach rozłożonych równomiernie i w klastrach. Część została wygenerowana losowo, a część pochodziła z biblioteki TSPLIB.

Zbadano właściwości powstałych sieci optimów lokalnych, m.in. liczbę unikalnych optimów lokalnych i globalnych, liczba wywołań funkcji celu, stosunek liczby wierzchołków i wywołań funkcji celu, stosunek wierzchołków jest połączonych ścieżką z optimum globalnym do całkowitej liczby wierzchołków, średnią siłę (ang. strength) wierzchołków. Zbadano również własności lejów, takie jak rozmiar największego leja czy liczba atraktorów.

Badania wykazały, że przestrzeń rozwiązań badanych instancji TSP zawiera wiele lejów a nie, jak wcześniej sądzono, jeden lej kończący się w globalnym minimum. W przestrzeni rozwiązań instancji o regularnej strukturze wykryto istnienie rozległych płaskowyzów(ang. plateaus). Zauważono różnicę między instancjami wygenerowanymi losowo a rzeczywistymi. Te pierwsze miały zwykle jedno globalne optimum i małą neutralność, podczas gdy instancje rzeczywiste zwykle miały więcej globalnych optimów, czasem formujących płaskowyz globalnych optimów. Zauważono zależność między rozmiarem instancji a liczbą lejów - ich liczba zwiększa się wraz ze wzrostem rozmiaru instancji. Najwięcej lejów pojawiło się w instancjach z równomiernym losowym rozmieszczeniem miast. Zbadano korelację między różnymi właściwościami przestrzeni rozwiązań. Wykazano, że w instancjach, w których miasta rozmieszczone są w klastrach, występuje większa korelacja właściwości, niż w instancjach o rozłożeniu równomiernym.

## Local Optima Networks in Solving Algorithm Selection Problem for TSP (2018) [1]

W pracy badano możliwość wykorzystania analizy sieci optimów lokalnych w problemie wyboru algorytmu heurystycznego dla danej instancji problemu komiwojagera (Algorithm selection problem).

Wykorzystano algorytm dwufazowy (ang. Two-Phase Sampling). Algorytm tego typu podzielony jest na dwa etapy - poszukiwanie wierzchołków i poszukiwanie krawędzi. Wierzchołki wyszukiwane są poprzez generowanie losowych rozwiązań i uruchamianie na nich algorytmu 2-opt typu best-improvement. Algorytm generuje LON opartą o escape edges. Próbkowanie krawędzi polega na poddaniu każdego ze znalezionych wcześniej lokalnych optimów perturbacji typu 2-change i optymalizacji algorytmem 2-opt typu first-improvement. Sprawdzany jest następnie warunek istnienia krawędzi i, jeśli jest spełniony, do zbioru dodawana jest odpowiednia krawędź bądź zwiększana jest jej waga, jeśli już istnieje ona w zbiorze. Wśród badanych parametrów wymienić można m.in stosunek liczby krawędzi do liczby wierzchołków, średnia odległość optimum lokalnego od optimum o najniższej wartości funkcji celu, stosunek wierzchołków połączonych z tym optimum do pozostałych oraz różnorodność(ang. assortativity) i współczynnik klasteryzacji grafu. Do eksperymentów wykorzystano instancje problemu z biblioteki TSPLIB, oraz instancje generowane losowo o równomiernym rozkładzie miast. Obliczono korelację różnych miar, aby wyeliminować redundantne dane.

Na każdej instancji problemu uruchomiono algorytmy heurystyczne i porównano parami jakość ich rozwiązań. Obliczono również średnią względną jakość rozwiązań dla każdego algorytmu. Porównane zostały algorytmy *Greedy Descent*, *Guided Local Search*, *Simulated Annealing*, *Taboo Search*, *Objective Taboo Search*. Wykorzystano implementacje zawarte w pakiecie *Google Optimization Tools*. Sprawdzono również obecny w pakiecie tryb automatyczny, który sam dobiera algorytm dla danej instancji.

Na podstawie zebranych danych nauczono klasyfikatora dla każdej pary algorytmów. Klasyfikatory przypisywały wyniki do jednej z trzech klas: 1 - algorytm A daje lepszy wynik od algorytmu B, 2 - algorytm B daje lepszy wynik od algorytmu A, 3 - algorytmy dają ten sam wynik. Utworzone klasyfikatory porównano z klasyfikatorem ZeroR. Przygotowane klasyfikatory dokonały poprawnej predykcji częściej niż ZeroR w przypadku instancji ze zbioru TSPLib. W przypadku instancji generowanych losowo, tylko w jednym przypadku klasyfikator okazał się lepszy od ZeroR.

Eksperymenty wykazały, że analiza sieci optimów lokalnych może zostać z powodzeniem wykorzystana do selekcji algorytmu heurystycznego, jednak czas próbkowania, dłuż-

szy od czasu samego rozwiązywania problemu, ogranicza jej użyteczność.

## Clarifying the Difference in Local Optima Network Sampling Algorithms (2019) [6]

W pracy wykonane zostało statystyczne porównanie dwóch algorytmów próbkowania sieci lokalnych optimów dla kwadratowego problemu przydziału(QAP). Do badań wykorzystano instancje problemu z biblioteki QAPLIB. Wybrano 30 instancji o różnych klasach (rozłożone równomiernie, na siatce, instancje rzeczywiste, instancje losowe symulujące rzeczywiste). Porównano algorytmy *Snowball* oraz *Iterated Local Search*. Spróbkowane dane wykorzystano do stworzenia modeli regresji do przewidywania jakości rozwiązań dwóch algorytmów heurystycznych: *Robust Taboo Search* Taillarda oraz *Improved ILS* Stützla.

Wykonano próbkowanie przestrzeni rozwiązań i sprawdzono, czy wartości poszczególnych miar pochodzące od dwóch algorytmów są ze sobą skorelowane. Zbadanymi miarami były m.in: liczba optimów, liczba krawędzi, średnia liczba krawędzi wychodzących z wierzchołka grafu(ang. outdegree), średnica grafu, średnie dopasowanie lokalnych optimów(ang. meanfitness), sinkfitness (dopasowanie spływów? TODO: translate). Zauważono korelację metryk meanfitness i sinkfitness na poziomie 0.99. Dla wszystkich innych metryk korelacja było mniejsza od 0.5, dla średnicy grafu była ona ujemna. Na podstawie zebranych danych nauczono modele typu liniowego oraz *RandomForest*. Zbadano wartości metryki  $R^2$  dla różnych parametrów algorytmów próbkowania, algorytmów heurystycznych i modeli regresji. Obliczono również współczynniki korelacji pomiędzy różnymi właściwościami sieci lokalnych optimów a performance gap metric(TODO: translate) algorytmów optymalizujących.

Badania wykazały, że sieci wygenerowane przez dwa różne algorytmy próbkujące mają pewne wspólne cechy. *Snowball* okazał się bardziej przewidywalny i łatwiejszy w dobieraniu parametrów. ITS natomiast lepiej znajdował struktury skupiające (ang. hub-and-spoke structures, hubs) w przestrzeni rozwiązań. Wartości metryk LON uzyskane na podstawie algorytmu ITS były bardziej skorelowane z jakością rozwiązania heurystyki(ang. heuristic performance metric). Modele regresji utworzone na podstawie danych uzyskanych z ITS dokonywały lepszej predykcji jakości rozwiązań algorytmów heurystycznych, niż modele utworzone na podstawie algorytmu *Snowball*. Zauważono, że wartości funkcji celu w poszczególnych lokalnych optimach sieci LON były lepszymi predyktorami, niż informacje o krawędziach tej sieci.

## Inferring Future Landscapes: Sampling the Local Optima Level (2020) [7]

W pracy zostały porównane algorytmy próbkowania sieci optimów lokalnych(LON) dla kwadratowego problemu przydziału (QAP). Porównane zostały algorytmy *Markov-Chain* oraz *Snowball*. Dla małych instancji problemu wykonano kompletne przeszukanie przestrzeni rozwiązań, dla większych - przeszukiwanie z limitowanym budżetem obliczeniowym.

Jako instancje testowe wykorzystano przykłady ze zbioru QAPLIB. Instancje należały do pięciu kategorii: instancje z wartościami losowymi z rozkładu równomiernego, instancje z wartościami odległości ułożonymi "na siatce" i losowymi wartościami przepływu, rzeczywiste instancje problemu, instancje generowane symulujące rzeczywiste, oraz instancje nie pasujące do żadnej z poprzednich kategorii.

Badane właściwości sieci LON różniły się w zależności od algorytmu próbkowania. Dla Algorytmu *Markov-chain* były to: liczba wierzchołków i krawędzi grafu, średnie dopasowanie(ang. meanfitness), średnia liczba krawędzi wychodzących z wierzchołka (ang. mean outdegree), oraz najdłuższa ścieżka między wierzchołkami. Dodatkowo zbadano tzw. metryki lejowe(ang. Funnel metrics) - liczba krawędzi do globalnego optimum (ang. incoming global) oraz średnie dopasowanie spodu leja(ang. mean funnel-floor fitness).

Dla algorytmu *Snowball* metryki lejowe, oraz te oparte o gęstość i wzory połączeń krawędzi nie dają użytecznych informacji, zamiast tych metryk wybrano: średnią wagę pętli w grafie, średnią różnicę wag krawędzi wychodzących (ang. mean weight disparity), korelację dopasowania między sąsiadami (ang. fitness-fitness correlation), średnia długość "wspinaczki" do optimum lokalnego, maksymalna długość "wspinaczki" do lokalnego optimum oraz maksymalna liczba ścieżek prowadzących do lokalnego optimum.

Na podstawie danych zbudowano modele regresji, dokonujące predykcji jakości rozwiązań uzyskanych przy wykorzystaniu heurystyk *Improved Iterated Local Search* i *Robust Taboo Search*. Sprawdzono modele liniowe i *RandomForest*. Modele oparte o *RandomForest* okazały się lepsze.

Badania nie wykazały wyższości jednego algorytmu próbkowania nad drugim. Autorzy zwracają uwagę na fakt, że niektóre z cech sieci optimów lokalnych są lepiej odwzorowywane przez algorytm *Markov-chain*, a inne lepiej przez *Snowball*. Najlepsze wśród zbudowanych modeli regresji okazały się zaś te, które korzystały z kombinacji danych pozyskanych z obu algorytmów.

## Understanding AutoML Search Spaces with Local Optima Networks (2022) [5]

AutoML jest procesem automatyzacji konfiguracji procesów(pipelines) uczenia maszynowego. Może obejmować on takie zadania jak wstępne przetwarzanie danych, ekstrakcja cech, dobór odpowiedniego algorytmu i jego hiperparametrów. Mnogość możliwych konfiguracji procesu sprawia, że przestrzeń przeszukiwania AutoML jest duża. W opisaney pracy autorzy podejmują próbę analizy przestrzeni AutoML dla zadania klasyfikacji przy wykorzystaniu sieci optimów lokalnych.

Przestrzeń przeszukiwania AutoML cechuje się dużą liczbą parametrów warunkowych - parametrów, których istnienie zależy od wartości innych parametrów. Jest to spowodowane tym, że algorytmy uczenia maszynowego przyjmują inny zestaw hiperparametrów. Z tego powodu do reprezentacji przestrzeni rozwiązań wykorzystano odpowiednią gramatykę BNF z rozwiązaniami generowanymi przy użyciu drzewa wyprowadzania gramatyki (grammar derivation tree). Powstała gramatyka posiada 38 reguł decydujących o zastosowanych algorytmach wstępnego przetwarzania danych, zastosowanego algorytmu klasyfikacji oraz jego hiperparametrów. Spośród algorytmów klasyfikacji wykorzystano: Regresję logistyczną, MLP, KNN, Random Forest i Ada Boost.

Wierzchołki LON zdefiniowano jako lokalne maksima przestrzeni rozwiązań. Jako wartość dopasowania wybrano uśrednioną wartość metryki F-score dla danej konfiguracji procesu. Sąsiedztwo rozwiązania zdefiniowano jako zbiór rozwiązań możliwych do uzyskanie poprzez zastosowanie operatora mutacji na danym rozwiązaniu. Za operator mutacji przyjęto losowe wybranie jednego wierzchołka z drzewa rozwiązania i zastąpienie go losowym drzewem (przy zachowaniu zasad gramatyki procesu). Szansa na wylosowanie wierzchołka została uzależniona od głębokości jego pozycji w drzewie. W ten sposób operacje zmieniające proces w znaczny sposób (np. zmiana algorytmu) mają mniejsze prawdopodobieństwo wystąpienia niż operacje zmieniające jedynie wartość hiperparametrów. Zbudowano sieci



oparte o 3 różne definicje krawędzi - basin-transition edges, escape edges i perturbation edges. Do próbkowania przestrzeni rozwiązań w celu utworzenia LON wykorzystano algorytm wstępujący(hill-climb).

Do badań wykorzystano 7 zbiorów danych, zbadano 30 różnych przestrzeni konfiguracji. Ze względu na złożoność obliczeniową badania sąsiedztwa, próbkowano ograniczoną liczbę sąsiadów. Parametr ten nazwano wielkością sąsiedztwa i zbadano 4 warianty - 20, 30, 50 i 100 sąsiadów. W każdym przypadku wartość dopasowania została obliczona dla wszystkich rozwiązań w przestrzeni. Nadano budżet 120 sekund na wykonanie obliczeń określających wartość dopasowania danego rozwiązania. W przypadku przekroczenia otrzymywał on wartość 0.

Zbadano korelację między wartością dopasowania optimum lokalnego a rozmiarem jego basenu przyciągania w zależności od rozmiaru sąsiedztwa. Wykazano, że istnieje silna korelacja i zwiększanie sąsiedztwa w większości przypadków zmniejsza liczbę basenów przyciągania w przestrzeni i zwiększa rozmiar istniejących. Ta zależność nie zachodziła jednak w przypadku jednego z badanych zbiorów danych.

Zauważono wpływ zastosowanej definicji krawędzi na powstałą sieć LON - w sieciach z krawędziami typu basin-transition nie zauważono obecności ścieków (wierzchołków bez krawędzi wychodzących, ang. sinks), natomiast istniało wiele źródeł (wierzchołków bez krawędzi wchodzących, ang. sources). Ilość źródeł zmniejszała się wraz ze wzrostem sąsiedztwa. Sieci oparte o escape edges miały natomiast dużo ścieków i mało źródeł. Zwiększanie rozmiaru sąsiedztwa prowadziło do zmniejszenia liczby ścieków. Podobny efekt dawało zwiększenie parametru D. W przypadku perturbation edges w sieci nie było ścieków ani źródeł niezależnie od rozmiaru sąsiedztwa.

Obliczono odległości między parami lokalnych i globalnych optimów. Stwierdzono, że odległości między optimami globalnymi były mniejsze od odległości między optimami lokalnymi i odległości pomiędzy optimami globalnymi i lokalnymi. Zwiększanie sąsiedztwa zmniejszało odległość między globalnymi optimami. Zauważono, że optima są skupione w pewnym rejonie przestrzeni rozwiązań, a nie rozłożone równomiernie.

## 2.2 O przestrzeni rozwiązań

Fitness landscape(Przestrzeń rozwiązań? Krajobraz dopasowania?) jest pojęciem wywodzącym się z biologii ewolucyjnej. W kontekście biologicznym jest to model opisujący relację między genotypem i fenotypem organizmów, a ich przystosowaniem (ang. fitness), które jest miarą opisującą sukces reprodukcyjny[2].

W kontekście optymalizacji Fitness landscape to trójka  $(S, V, f)$ , gdzie:

- $S$  jest zbiorem wszystkich możliwych rozwiązań - przestrzenią przeszukiwania,
- $V$  jest funkcją przypisującą każdemu rozwiązaniu  $s \in S$  zbiór sąsiadów  $V(s)$ ,
- $f$  jest funkcją  $f : S \rightarrow \mathbb{R}$  przypisującą danemu rozwiązaniu wartość przystosowania - zwykle jest to wartość funkcji celu dla danego rozwiązania.

## 2.3 Sieć optimów lokalnych

Sieć optimów lokalnych(ang. Local Optima Network, LON) jest konstruktem zaprezentowanym po raz pierwszy w artykule[8], i rozwiniętym w pracy[4].

Jest to graf  $G = (N, E)$  przedstawiający występujące w przestrzeni rozwiązań optima lokalne (zbiór wierzchołków  $N$ ) i relacje między nimi (zbiór krawędzi  $E$ ).

## Wierzchołki

Wierzchołki w sieci optimów lokalnych reprezentują optima lokalne w przestrzeni rozwiązań. Do optimów zaliczamy minima i maksima; w problemach optymalizacyjnych zazwyczaj poszukujemy tych pierwszych. Minimum lokalne to takie rozwiązanie  $s$ , w którego sąsiedztwie  $V(s)$  nie znajduje się żadne rozwiązanie  $x$ , dla którego  $f(x) < f(s)$ . Każde rozwiązanie w przestrzeni rozwiązań można przyporządkować do pewnego lokalnego minimum. Aby znaleźć lokalne minimum  $n \in N$ , do którego "prowadzi" dane rozwiązanie  $s \in S$ , wykonuje się lokalną optymalizację z tym rozwiązaniem przyjętym jako punkt startowy. W dalszej części pracy takie przyporządkowanie będzie oznaczane jako  $h(s) \rightarrow n \in N$ . Wierzchołkom w sieci LON można przypisać wagę równą wartości funkcji celu w danym optimum lokalnym.

## Krawędzie

Powstały dwa modele krawędzi: basin-transition i escape edges (TODO: translate?) TODO: można coś napisać jeszcze o perturbation edges

### Basin-transition

Basen przyciągania optimum lokalnego  $n$  jest zdefiniowany jako zbiór:

$$b_i = \{s \in S \mid h(s) = n\}$$

Rozmiarem basenu jest liczność tego zbioru oznaczana jako  $\#b_i$ . Dla każdej pary rozwiązań w przestrzeni można obliczyć prawdopodobieństwo przejścia z jednego rozwiązania do drugiego  $p(s \rightarrow s')$ . Dla rozwiązań reprezentowanych permutacją o długości  $M$ , prawdopodobieństwo takie wynosi:

$$p(s \rightarrow s') = \frac{1}{M(M-1)/2}, \quad \text{jeżeli } s' \in V(s),$$

$$p(s \rightarrow s') = 0, \quad \text{jeżeli } s' \notin V(s),$$

Mając informacje o prawdopodobieństwach przejścia między poszczególnymi rozwiązaniami można obliczyć prawdopodobieństwo przejścia od rozwiązania  $s$  do dowolnego rozwiązania należącego do basenu  $b_j$ :

$$p(s \rightarrow b_j) = \sum_{s' \in b_j} p(s \rightarrow s')$$

Całkowite prawdopodobieństwo przejścia z basenu jednego optimum lokalnego do drugiego wynosi więc:

$$p(b_i \rightarrow b_j) = \frac{1}{\#b_i} \cdot \sum_{s \in b_i} p(s \rightarrow b_j)$$

To całkowite prawdopodobieństwo stanowi wagę krawędzi w grafie.

### Escape Edges

Escape Edges zdefiniowane są przy pomocy funkcji dystansu  $d$  zwracającej najmniejszą odległość między dwoma rozwiązaniami i liczby całkowitej  $N$ . Krawędź  $e_{ij}$  między lokalnymi optimami  $n_i$  i  $n_j$  istnieje, jeśli istnieje rozwiązanie  $s$  takie, że:

$$d(s, n_i) \leq D \wedge h(s) = n_j$$

Wagę takiej krawędzi jest ilość rozwiązań spełniających powyższy warunek.

TODO: napisać o różnicach sieci tworzonych przez basin-transition i escape edges



# Rozdział 3

## Badania eksperymentalne

3.1 Opis eksperymentów

3.2 Zaimplementowane algorytmy

3.3 Wyniki



## Rozdział 4

### Opis implementacji





# Rozdział 5

## Podsumowanie



# Dodatki



Dodatek A

Jakiś dodatek



# Literatura

- [1] W. BOZEJKO, A. GNATOWSKI, T. NIZYNSKI, M. AFFENZELLER, AND A. BEHAM, *Local optima networks in solving algorithm selection problem for TSP*, in Contemporary Complex Systems and Their Dependability - Proceedings of the 13th International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2-6, 2018, Brunów, Poland, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, eds., vol. 761 of Advances in Intelligent Systems and Computing, Springer, 2018, pp. 83–93.
- [2] I. FRAGATA, A. BLANCKAERT, M. A. DIAS LOURO, D. A. LIBERLES, AND C. BANK, *Evolution in the light of fitness landscape theory*, Trends in Ecology & Evolution, 34 (2019), pp. 69–82.
- [3] G. OCHOA AND N. VEERAPEN, *Mapping the global structure of TSP fitness landscapes*, J. Heuristics, 24 (2018), pp. 265–294.
- [4] G. OCHOA, S. VÉREL, F. DAOLIO, AND M. TOMASSINI, *Local optima networks: A new model of combinatorial fitness landscapes*, CoRR, abs/1402.2959 (2014).
- [5] M. C. TEIXEIRA AND G. L. PAPPA, *Understanding automl search spaces with local optima networks*, in GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, USA, July 9 - 13, 2022, J. E. Fieldsend and M. Wagner, eds., ACM, 2022, pp. 449–457.
- [6] S. L. THOMSON, G. OCHOA, AND S. VÉREL, *Clarifying the difference in local optima network sampling algorithms*, in Evolutionary Computation in Combinatorial Optimization - 19th European Conference, EvoCOP 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24-26, 2019, Proceedings, A. Liefooghe and L. Paquete, eds., vol. 11452 of Lecture Notes in Computer Science, Springer, 2019, pp. 163–178.
- [7] S. L. THOMSON, G. OCHOA, S. VÉREL, AND N. VEERAPEN, *Inferring future landscapes: Sampling the local optima level*, Evol. Comput., 28 (2020), pp. 621–641.
- [8] M. TOMASSINI, S. VÉREL, AND G. OCHOA, *Complex-network analysis of combinatorial spaces: The nk landscape case*, Phys. Rev. E, 78 (2008), p. 066114.





# Spis rysunków



Spis tabel