

Problem komiwojażera dla symulowanego wyżarzania

Wojciech Wróblewski
250349

Opis problemu komiwojażera

Problem komiwojażera (Traveling Salesman Problem) to jeden z najbardziej rozpoznawalnych oraz studiowanych zagadnień zaliczonych do problemów NP-trudnych w dziedzinie zagadnień optymalizacji kombinatorycznej. Nazwa problemu sugerowana jest problemem znalezienia przez komiwojażera trasy przebiegającej przez n miast charakteryzującej się najmniejszym kosztem. Trasa przebiega dokładnie jeden raz przez każde z miast. Na koniec komiwojażer powraca do miasta początkowego. Koszt możemy utożsamiać z funkcją ostatecznej długości trasy. Możemy rozróżnić dwa przypadki. Kiedy odległość między parą miast A oraz B jest taka sama jak odległość między B oraz A, wówczas mamy do czynienia z symetrycznym TSP. Jednak nie zawsze tak musi być, a relacja odległości nie musi być symetryczna. W drugim przypadku mówimy o niesymetrycznym problemie komiwojażera. Przechodząc do terminologii grafowej, miasta są reprezentowane poprzez wierzchołki grafu ważonego, a szukaną trasą jest cykl hamiltona o najniższej wadze. Złożoność problemu sugeruje, że próżne jest szukanie algorytmu, który wyznaczy dokładne rozwiązania dla instancji problemu TSP w czasie wielomianowym, dlatego bardzo często stosowane są, mniej czasochłonne, heurystyczne algorytmy.

Heurystyka

Podejścia heurystyczne lub konkretne rozwiązania metaheurystyczne pozwalają na eksplorację przestrzeni rozwiązań danego problemu, proponują strategię uciekania z lokalnego minimum, jednak nie gwarantują odnalezienia rozwiązania oraz trudno przewidzieć czas w jakim wygenerują oczekiwane rozwiązanie. Ponadto niezwykle istotną częścią w działaniu danego algorytmu są jego parametry początkowe, które różnicują skuteczność jednych podejść algorytmicznych od innych dla instancji danego problemu. Różnorodność problemów nie pozwala na wskazanie uniwersalnych parametrów dla każdego problemu, dlatego podejście heurystyczne stosowane jest w szerokiej gamie problemów o klasie złożoności wykraczającej ponad złożoność wielomianową. W ogólności możemy wyróżnić charakterystyczne punkty w planowaniu algorytmów metaheurystycznych.

- rozwiązanie początkowe - rozwiązanie początkowe lub metoda wyznaczenia danych początkowych, na których opieramy strategię podejścia metaheurystycznego.
- wyznaczenie sąsiedztwa - sposób generowania kolejnych kandydatów na potencjalne lepsze rozwiązanie, gwarantujący różnorodność gamy rozwiązań jak i skuteczną eksplorację przestrzeni rozwiązań.
- strategia optymalizacji- schemat realizacji algorytmu wraz z zdefiniowaniem jego parametrów oraz procesu akceptacji kandydujących rozwiązań dający możliwość skutecznego przeszukiwania i ucieczek od minimów lokalnych.

Symulowane wyżarzanie

Jest to rodzaj algorytmu przeszukującego przestrzeń alternatywnych rozwiązań problemu w celu wyszukania rozwiązań najlepszych. Możemy wyróżnić podstawowe pojęcia występujące w algorytmie:

- Warunek zatrzymania - z reguły reprezentujący końcową temperaturę układu (dla której algorytm przestaje się wykonywać), czas działania t , maksymalną liczbę iteracji N lub sytuację, w której osiągnięte zostało wystarczająco dobre rozwiązanie lub od zadanej liczby iteracji nie nastąpiła żadna poprawa podczas jego wyznaczania.
- Temperatura układu T - parametr kluczowy w strategii symulowanego wyżarzania jako determinant prawdopodobieństwa akceptacji rozwiązań sąsiednich.
- Funkcja kosztu f - określana jako odwzorowanie różnicujące korzyść jednego rozwiązania nad innym.
- Schemat chłodzenia - metoda zmiany temperatury układu podczas działania algorytmu symulowanego wyżarzania.
- Wyznaczanie sąsiedztwa oraz jego rozmiar - metody kreowania nowych kandydatów na rozwiązania bazujące na aktualnym stanie układu.
- Prawdopodobieństwo przejścia/warunek akceptacji - określenie prawdopodobieństwa z jakim możliwe jest wybranie gorszego kandydata w danej iteracji.
- Długość epoki - interwał czasowy lub iteracyjny po którym następuje ochłodzenie układu.

Główną ideę algorytmu możemy przedstawić jako [Syg20]:

- Wybieramy temperaturę początkową układu $T = T_0$ oraz początkowe rozwiązanie i_0 (losowe).

- Do momentu znalezienia dość dobrego rozwiązania lub osiągnięcia warunku zatrzymania:
 - Wybieramy sąsiada x , metodami wyznaczania sąsiedztwa.
 - Decydujemy się, czy przejść do x .
 - Decydujemy o zmniejszeniu temperatury układu T .

Sposób działania symulowanego wyżarzania przypomina zjawisko wyżarzania w metalurgii. W algorytmie symulowanego wyżarzania dla każdego z testowych rozwiązań sąsiednich przechodzimy z pewnym prawdopodobieństwem, które jest warunkiem akceptacji danego kandydata. Jeśli funkcja kosztu określona na kandydacie j jest mniejsza od funkcji kosztu na aktualnym rozwiązaniu i , nowy kandydat jest zawsze akceptowany. W przeciwnym wypadku akceptacja następuje z pewnym prawdopodobieństwem zadany jako:

$$P_T(\text{akceptacja } j) = \begin{cases} 1 & \text{jeżeli } f(j) \leq f(i) \\ \exp\left(\frac{f(i)-f(j)}{T}\right) & \text{jeżeli } f(j) > f(i) \end{cases} \quad (1)$$

Akceptacja jest ściśle zależna od temperatury układu który zmienia się zgodnie z zaproponowanym schematem chłodzenia. Wykorzystywane są różne metody mające na celu modyfikacje temperatury układu oraz różna częstotliwość ich stosowania. W niektórych przypadkach temperatura może być zmieniana z każdą iteracją, a w niektórych określa się pewien interwał liczby iteracji po którym następuje zmiana (długość epoki).

Schematy chłodzenia w symulowanym wyżarzaniu

Możemy rozróżnić wiele schematów chłodzenia. [Kli16]. Niektóre z nich to:

liniowy schemat chłodzenia

$$T_{p+1} = T_p + \lambda \quad \lambda = \frac{T_k - T_0}{N}$$

Geometryczny schemat chłodzenia

$$T_{p+1} = T_p \cdot \lambda \quad \lambda = \left(\frac{T_k}{T_0}\right)^{\frac{1}{N}}$$

Logarytmiczny schemat chłodzenia

$$T_{p+1} = \frac{T_p}{1+\lambda \cdot T_p} \quad T_{p+1} = \frac{T_0 - T_k}{N \cdot T_0 \cdot T_k}$$

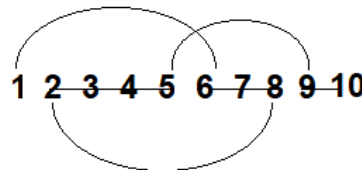
Gdzie p - indyktor p -tej iteracji algorytmu, T_0 - temperatura początkowa układu, T_k - temperatura końcowa układu, T_p - temperatura w p -tej iteracji algorytmu, λ - parametr schematu chłodzenia, N - maksymalna liczba iteracji.

Metody generowania sąsiedztwa

Sąsiedztwo rozwiązań możemy kreować wprowadzając pewne zmiany w cyklu, który reprezentuje naszą drogę w problemie TSP. Przykłady generowania sąsiedztwa na podstawie mutacji

- Zamiana pozycji sąsiadnich: M_1 [LL09]- zamiana pozycji dwóch sąsiadujących wierzchołków w cyklu.
- Dodanie wierzchołka: VI [Gen+11] - wydzielenie jednego wierzchołka w cyklu i dodanie go w inne miejsce tworząc w ten sposób inny cykl.
- Dodanie bloku: BI [Gen+11] - wydzielenie ścieżek prostych z cyklu i ich zamiana.

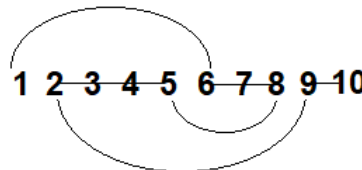
1 2 3 4 5 6 7 8 9 10



1 6 7 8 2 3 4 5 9 10

- Dodanie bloku z odwróceniem: M_6 [LL09] - modyfikacja BI z innym łączeniem ciągu w wynikowy cykl.

1 2 3 4 5 6 7 8 9 10



1 6 7 8 5 4 3 2 9 10

- Odwrócenie bloku: BR [Gen+11] - wydzielenie ścieżki prostej w cyklu i odwrócenie jej kolejności wierzchołków.

Generowanie nowych kandydatów na potencjalne lepsze rozwiązania to kluczowy element podejścia heurystycznego, jednak nie zawsze pewne wyrobione schematy modyfikacji sprawdzają się w każdej instancji TSP.

Według badań [Gen+11], strategię w dobieraniu oraz łączeniu sposobów na generowanie sąsiedztwa, mogą usprawnić lub pogorszyć metodę przeszukiwań dla danego problemu. Dlatego podczas stosowania wielu sposobów na generowanie sąsiedztwa, wyboru mutacji można dokonywać z pewnym prawdopodobieństwem

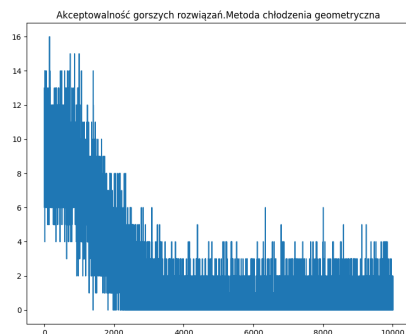
lub rozplanować strategię doboru sąsiedztwa podczas wyszukiwania. Sposoby generowania sąsiedztwa oznaczone jako VI,BI,BR, M_6 wywierają większy wpływ na generowanie różnorodnych sąsiedztw, które mogą znacząco różnić się o rozwiązanie bazowego. Mutacja M_1 wprowadza relatywnie mniejsze zmiany, jednak w momencie przeszukiwania przestrzeni rozwiązań, gdy rozwiązania należą już do grupy lokalnych minimów, może wprowadzić subtelne zmiany kierowane ku znalezieniu minimum globalnego. Zastosowanie strategii z czasowym lub zależnym od temperatury doбором sąsiedztwa może znacząco wpłynąć na rezultaty przeszukiwań.

Rozwiązywania TSP wykorzystując symulowane wyżarzanie

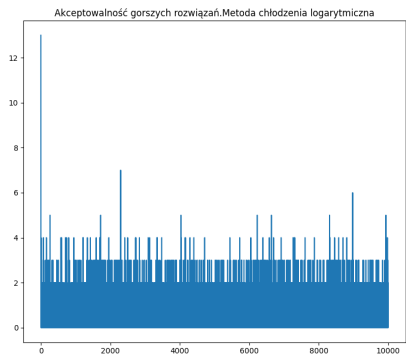
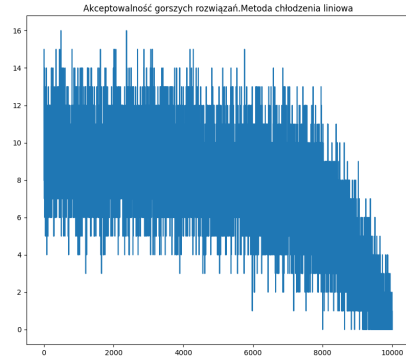
Standardowe symulowane wyżarzanie SA zakłada istnienie strategii zmiany temperatury układu. Poprawę w znalezieniu dostatecznie dobrego rozwiązania dla zadanej instancji problemu, może wnieść wykorzystanie innej metody chłodzenia układu oraz dobranie odpowiedniej temperatury początkowej T_0

Algorytm SA

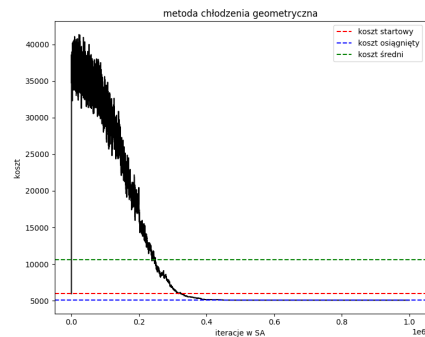
1. Wylosuj rozwiązanie początkowe X_0 , Ustaw temperaturę początkową T_0 , temperaturę końcową T_k , końcową liczbę iteracji N , numer aktualnej iteracji $i = 0$
2. Wygeneruj nowego kandydata $X_j = N(X_i)$ na podstawie mutacji (przykładowo BR). Oblicz różnicę $\delta = f(X_j) - f(X_i)$, gdy f jest funkcją kosztu.
3. Zaktualizuj aktualną temperaturę układu T_i dla iteracji i zgodnie z zadaną metodą chłodzenia .
4. Jeżeli $\delta < 0$ idź do kroku 6, w przeciwnym wypadku idź do kroku 5.
5. Jeżeli $\exp\left(\frac{-\delta}{T_i}\right) > \text{Random}(0, 1)$ idź do kroku 6, w przeciwnym wypadku idź do kroku 7.
6. Zaakceptuj rozwiązanie $X_i = X_j$
7. Jeżeli temperatura układu $T_i > T_k \wedge i < N$ idź do kroku 2, w przeciwnym wypadku zakończ działanie.



Wpływ schematów chłodzenia zamieszczonych w sekcji 'Schematy



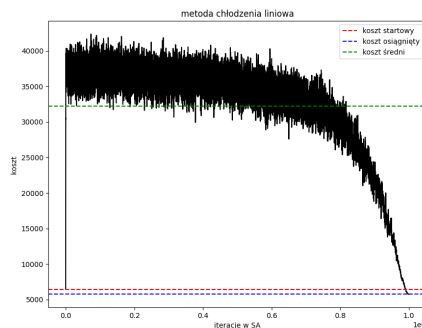
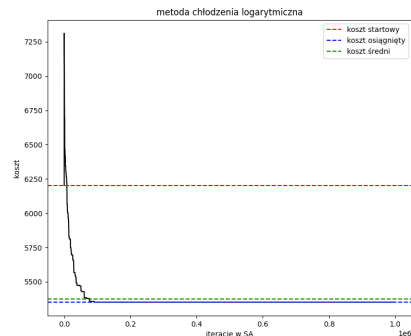
Poniżej przedstawione jest zestawienie jak zmienia się koszt akceptowalnych rozwiązań w zależności od rodzaju chłodzenia układu. Doświadczenie przeprowadzone na małej instancji problemu TSP dla 150 miast. Temperatura początkowa $T_0 = 1000$, temperatura końcowa $T_k = 0.001$. Warunkiem stopu było osiągnięcie temperatury końcowej lub maksymalnej liczby iteracji $N = 1000000$. Sąsiedztwo generowane poprzez mutację BR.



chłodzenia w symulowanym wyrażaniu' na akceptowalność gorszego rozwiązania dla małego symetrycznego problemu TSP, przy dodatkowym założeniu, że z jednego miasta możemy dotrzeć do każdego innego. Temperatura początkowa $T_0 = 1000$, temperatura końcowa $T_k = 0.01$. Doświadczenie przeprowadzone dla 20 miast rozmieszczonych na płaszczyźnie 500×500 jednostek. Warunkiem stopu było osiągnięcie temperatury końcowej lub maksymalnej liczby iteracji $N = 1000000$. BR [Gen+11].

Temperatura układu zmniejszana co 20 iteracji, w celu obserwacji ile gorszych rozwiązań zostanie zaakceptowanych dla danej długości epoki. Generowanie nowych kandydatów na podstawie mutacji

Na podstawie wykresów akceptowalności gorszych rozwiązań, można zauważyć, że różne techniki chłodzenia oferują różne możliwości. Zastosowanie logarytmicznej metody chłodzenia [Kli16] sprawia że akceptowalność gorszych rozwiązań bardzo szybko maleje.



Przy metodzie geometrycznej otrzymujemy mniej drastyczny spadek w początkowej fazie niż w metodzie logarytmicznej. Przy dużych temperaturach prawdopodobieństwo akceptacji gorszego rozwiązania jest duża jednak z czasem ona maleje. W liniowej metodzie chłodzenia otrzymujemy najwolniejszy spadek akceptacji gorszych rozwiązań, który początkowo jest duży i przez dłuższy czas działania algorytmu pozostaje dużo

wiekszy niż w innych metodach, jednak gdy układ zbliża się do końcowej temperatury zbieżność prawdopodobieństwa zaakceptowania gorszego rozwiązania bardzo szybko maleje.

Logarytmiczna metoda chłodzenia sprawia, że ekspansja przestrzeni rozwiązań jest mniejsza niż w przypadku liniowym oraz geometrycznym. Bardzo szybko koszt aktualnych rozwiązań zbiega do puli rozwiązań o potencjalnie niskim koszcie. Następnie eksploracja w poszukiwaniu minimum globalnego odbywa się już w wąskim zakresie. Metoda geometryczna dla zadanego przykładu spowodowała większą ekspansję przestrzeni rozwiązań, a zbieżność kosztu rozwiązań jest wolniejsza od tej realizowanej w metodzie logarytmicznej.

Problem TSP mimo swojej prostoty sprawia dużo problemów podczas optymalizacji. Dlatego też powstało wiele badań i algorytmów mających na celu optymalizację tego problemu. Z badań wynika, że mimo iż dla pewnych instancji problemu TSP podstawowy algorytm symulowanego wyżarzania SA potrafi znajdować optymalne trasy, jednak kolejnym krokiem, który może usprawnić przeszukiwania jest łączenie heurystyk. W artykule [LL09] przedstawione zostaje podejście hybrydowe łączące SA z implementacją listy Tabu, znanej z algorytmu Tabu-search. W rezultacie otrzymujemy algorytm :

1. Wylosuj rozwiązanie początkowe x_0 , Ustaw temperaturę początkową T_0 , zainicjalizuj listę $TL = None$, zainicjalizuj parametr $k = 0.75$, zainicjalizuj maksymalny czas działania programu $Time_{max}$, Ustaw $R_0 = 0$.
2. Jeśli czas działania programu jest mniejszy od $Time_{max} \cdot k$, wygeneruj nowego kandydata $X_j = N(X_i)$ na podstawie mutacji M_6 . W przeciwnym wypadku wygeneruj nowego kandydata na podstawie mutacji M_1 . Oblicz różnicę $\delta = f(X_j) - f(X_i)$, gdy f jest funkcją kosztu.
3. Zaktualizuj R_i, T_i oraz TL
4. Jeżeli $\delta < 0$ idź do kroku 6, w przeciwnym przypadku idź do kroku 5.
5. Jeżeli $\exp\left(\frac{-\delta}{T_i}\right) > \text{Random}(0, 1)$ idź do kroku 6, w przeciwnym przypadku idź do kroku 8.
6. Jeżeli $X_i \in TL$ idź do kroku 8, w przeciwnym przypadku idź do kroku 7.
7. Zaakceptuj rozwiązanie $X_i = X_j$
8. Jeśli czas działania programy jest mniejszy od $Time_{max}$ idź do kroku 2, w przeciwnym przypadku zakończ działanie.

Zaprezentowany wyżej algorytm jest rozszerzeniem klasycznego symulowanego wyżarzania o zapamiętywanie rozwiązań w liście tabu (o określonej w badniu wielkości), przez co unikamy powracania do rozwiązań wygenerowanych wcześniej. Taki zabieg redukuje czas przeszukiwań podczas szukania minimum globalnego [LL09]. Algorytm dodatkowo przewiduje zmienienie metody chłodzenia układu poprzez rozszerzenie go o czynnik R_i związany z aktualnym stanem przeszukiwań, rozszerzając możliwość dynamicznej adaptacji temperatury układu. Temperaturę obliczamy jako $T_i = T_0 + \lambda \cdot \ln(1 + R_i)$, gdzie λ jest zdefiniowanym parametrem spadku temperatury. Aktualizacja parametru R_i polega na zwiększeniu go o 1 gdy w danej iteracji i wartość funkcji kosztu dla nowego kandydata jest większa od wartości funkcji kosztu na obecnie rozpatrywanym rozwiązaniu. Jeżeli wartość funkcji kosztu jest mniejsza w iteracji i, to parametr R_i ustawiamy na 0, a gdy wartość funkcji kosztu na nowym kandydacie pozostaje taka sama to parametr pozostaje niezmieniony.

Dane numeryczne z badania wykazują, że zastosowanie takiej modyfikacji zapewnia szybszą zbieżność rozwiązań do puli rozwiązań optymalnych. Co jest wynikiem zastosowania nowej metody chłodzenia układu, która dynamicznie dostosowuje zdolność układu do akceptacji rozwiązań. Jednak wciąż istnieje ryzyko złej eksploracji przestrzeni rozwiązań w końcowej fazie działania algorytmu gdy zmieniamy sąsiedztwo (mutacja BI) na to mniej ekspansywne a bardziej lokalnie wyszukujące lepszych rozwiązań (mutacja M_1).

Według badań korzystnym jest zastosowanie modyfikacji algorytmu symulowanego wyżarzania o algorytm zachłanny .

Algorytm ASA-GS [Gen+11]

1. Wylosuj rozwiązanie początkowe x_0 , Ustaw temperaturę początkową T_0 , temperaturę końcową T_k , parametr chłodzenia t_{cool} , parametr określający liczbę kandydatów podczas etapu zachłannego t_{greedy} , bieżącą temperaturę $T_{current} = T_0$, oraz licznik $G = 0$.
2. Wygeneruj nowe rozwiązanie x' na podstawie metod generowania sąsiedztwa(mutacje VI, BI ,BR - wybierane z pewnym prawdopodobieństwem)
3. Oblicz $\delta = f(x') - f(x_0)$, gdzie f jest funkcją kosztu. Jeśli $\delta \leq 0$ idź do kroku 7, w przeciwnym przypadku idź do kroku 4.
4. Zwiększ licznik $G = G + 1$ i zachowaj $f(x')$. Jeżeli $G \geq t_{greedy}$ idź do kroku 5, w przeciwnym wypadku idź do kroku 2.
5. Wybierz najlepszego z zachowanych t_{greedy} elementów, czyli weź taki x'_{best} , że $f(x'_{best}) = \min \{f(x'_1), \dots, f(x'_{t_{greedy}})\}$. Przypisz $x' = x'_{best}$. Przejdź do kroku 6.
6. Zaakceptuj x' , czyli wykonaj $x_0 = x'$ z prawdopodobieństwem p . Idź do kroku 8.
7. Zaakceptuj x' , czyli wykonaj $x_0 = x'$. Idź do kroku 8.
8. Zaktualizuj temperaturę $T_{current} = T_{current} \cdot t_{cool}$ i ustaw $G = 0$. Jeżeli $T_{current} < T_k$ idź do kroku 2, w przeciwnym przypadku zakończ działanie.

Badanie przewiduje dobranie prawdopodobieństwa wybranej mutacji tak, aby wartości takie jak poprawa rozwiązania oraz czas działania były najbardziej korzystne. Autorzy przedstawiają doświadczalne próby zachowania poszczególnych mutacji oraz ich efektywności na podstawie których, określają prawdopodobieństwo mutacji VI,BI,BR na odpowiednio 10%,1% oraz 89% szans wylosowania. Prawdopodobieństwo p oraz parametry początkowe badań jak t_{cool}, t_{greedy} (zależne od wielkości problemu TSP- liczby miast) można odnaleźć w badaniu. [Gen+11]. Zaprezentowany wyżej algorytm jest modyfikacją symulowanego wyżarzania o rozszerzenie metody generowania sąsiedztwa oraz o zmianę warunku przejścia (akceptacji gorszych rozwiązań). Zauważamy, że gorszy kandydat nie jest poddawany warunkowi akceptacji zaraz po określeniu jego wartości przez funkcję kosztu. W momencie gdy algorytm uważa gorszego kandydata umiesza go w niezależnym sąsiedztwie o długości t_{greedy} . W chwili, gdy sąsiedztwo osiąga swój maksymalny rozmiar, algorytm zachłanny wybiera najkorzystniejszą opcję z sąsiedztwa i akceptuje go na aktualne rozwiązanie z zadanym prawdopodobieństwem p . Można zauważyć, że taka sytuacja ma miejsce

jeżeli od t_{greedy} iteracji nie osiągnięta została poprawa, ponieważ w momencie wygenerowania lepszego rozwiązania licznik rozmiaru sąsiedztwa greedy jest zerowany.

Korzystając z danych zawartych w badaniach wspomnianych wyżej można spróbować porównać algorytmy TS-SA oraz ASA-GS dla pewnych instancji problemu TSP. Dane w tabelach poniżej zostały zaczerpnięte z badań. Porównanie algorytmów znajduje się w sekcji "Wnioski".

ASA-GS [Gen+11]

problem	opt	best	avg	time
eil 76	538	544.369	544.369	5.5
berlin52	7542	7544.37	7544.37	3.83
eil51	426	428.872	428.872	3.91

TS-SA [LL09]

problem	opt	best	avg	time
eil 76	538	542	564	10
berlin52	7542	7648	7718.5	10
eil51	426	430	432.5	10

SA (schemat chłodzenia geometryczny, ze stałym parametrem schematu chłodzenia $\lambda = 0.95$) [LL09]

problem	opt	best	avg	time
eil 76	538	557	573	10
berlin52	7542	7658	7809.2	10
eil51	426	426	445.3	10

Wnioski

Wspomniane wyżej algorytmy różnią się strukturą implementacyjną. Można zauważyć, że algorytm ASA-GS jest algorytmem zachowującym się bardziej probabilistycznie niż algorytm TS-SA. Zostaje w nim uwzględnione pewne prawdopodobieństwo doboru generowanego sąsiedztwa, bazujące na statystycznych danych w badaniu. Dodatkowo parametryzacja prawdopodobieństwa p w warunkach akceptacji jest również bardziej złożona. Algorytm TA-SA posiada stały parametr k , który decyduje o zmianie sposobu modyfikacji sąsiedztwa. Algorytm w pierwszej fazie czasowej generuje bardziej zróżnicowane sąsiedztwa tym samym rozszerzając przestrzeń rozwiązań, a następnie w drugiej fazie korzystając z mniej inwazyjnej mutacji M_1 stara się znaleźć globalne minimum z obszaru wartości, który jest lokalnie akceptowalny. Ponadto algorytm TS-SA implementuje nowe podejście do metody chłodzenia układu, które dynamicznie się zmienia, co jest korzystnym rozwiązaniem problemu ucieczki z minimów lokalnych. Sam warunek stopu w obu przypadkach jest różny. Algorytm ASA-GS kończy działanie w momencie uzyskania temperatury końcowej, natomiast algorytm TS-SA jest algorytmem działającym czasowo. Warunki stopu w obu algorytmach wywierają kluczowy wpływ na ich działanie.

Można zauważyć, że w instancjach TSP dla których możemy porównać algorytmy, rozwiązania generowane przez ASA-GS cechuje większa skuteczność (odległość od optimum) w czasie działania algorytmu. Jednak warto zauważyć, że mimo końcowych lepszych wyników algorytm nie znajdował poprawy i nie udało mu się uciec z lokalnego minimum. Świadczy o tym wartość średnia oraz w źródle [Gen+11] wartość najgorsza. Z drugiej strony algorytm TS-SA cechował się większą ekspansją rozwiązań mniej optymalnych, o czym sugerują wartość średnia, jednak w każdym przypadku znajdował poprawę.

Na podstawie danych dla standardowej implementacji SA, można zauważyć, że algorytm TS-SA niekoniecznie wypada lepiej (dla wymienionych instancji TSP) pod względem skuteczności (odległości od optimum) w porównaniu do standardowej implementacji SA. W zasadzie czynni to rzadko dla testów przedstawionych w badaniach. Jednak zawsze cechuje się lepszą zbieżnością do rozwiązań lepszych, co możemy zauważyć porównując średnie wartości rozwiązań. Prawdopodobnie tą można zauważyć dla wszystkich instancji problemu TSP w doświadczeniu, które realizuje wspomniany algorytm.

Rezultaty podczas szukania rozwiązań dla problemu komiwojażera zależą niewątpliwie od parametrów algorytmu, jednak nie zawsze otrzymujemy jasny sposób ich inicjowania. W zależności czy oczekujemy większej ekspansji, czy eksploracji możemy manipulować temperaturą początkową oraz kluczowym w symulowanym wyżarzaniu schematem chłodzenia. Istotną kwestią jest dobre określenie strategii podczas określania sąsiedztwa. Czy chcemy, aby nasze sąsiedztwo było wywoływane mutacjami przez pewien czas działania algorytmu, przez pewien interwał temperaturowy, czy też z jakimś prawdopodobieństwem podczas działania algorytmu. Po objęciu strategii należy uwzględnić rodzaje mutacji generujących sąsiedztwo. Żeby pogodzić czas działania algorytmu oraz skuteczność jego rozwiązań (bliskość wartościom optymalnym) w badaniach dąży się do uzyskania właściwej zbieżności kosztu uzyskiwanych rozwiązań do rozwiązań optymalnych. Żeby algorytm szybko trafiał na rozwiązania akceptowalne, czyli nieoptymalne, ale bliskie rozwiązaniom optymalnym, podejście musi cechować się dość szybką zbieżnością. Jednak nie można zapominać o metodach aktywnej ucieczki z minimów, gdyż bez takich mechanizmów nasza metoda przeszukiwania może utknąć w minimum lokalnym i nie gwarantować nam poprawy do minimum globalnego. Na podstawie rezultatów z badań, można zauważyć, że nie ma idealnego sposobu dla wielu instancji problemu TSP. Może okazać się, że pewnie podejścia gwarantują optymalne wyniki dla pewnych instancji problemu TSP, a dla niektórych reagują bardzo niekorzystnie.

Bibliografia

- [LL09] Yi Liu i Hongbing Liu. “Hybrid simulated annealing algorithm based on adaptive cooling schedule for TSP”. W: *Odwiedzono: 14-06-2020*. Sty. 2009, s. 895–898. DOI: 10.1145/1543834.1543969.
- [Gen+11] Xiutang Geng i in. “Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search”.

W: *Applied Soft Computing* 11.4 (2011). Odwiedzono: 14-06-2020, s. 3680–3689. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2011.01.039>. URL: <http://www.sciencedirect.com/science/article/pii/S1568494611000573>.

- [Kli16] Marcin Klimek. “Symulowane wyżarzanie dla problemu harmonogramowania projektu z ograniczonymi zasobami”. W: *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki* 10 (grud. 2016). Odwiedzono: 14-06-2020, s. 53–65. DOI: 10.26348/znwwsi.15.53. URL: <http://books.google.com/books?id=W-xMPgAACAAJ>.
- [Syg20] Piotr Syga. *Algorytmy metaheurystyczne, Wykład 6-symulowanie wyżarzanie, kwantowe wyżarzanie*. <https://ki.pwr.edu.pl/syga/courses/amh/w06.pdf>. Odwiedzono: 14-06-2020. 2020.