PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 108
Współczesne tendencje rozwojowe badań operacyjnych 2010

Radosław Jadczak
Uniwersytet Łódzki

HEURYSTYKI I METAHEURYSTYKI W PROBLEMACH VRP

Streszczenie: Zagadnienia układania tras dla pojazdów (*vehicle routing problem* – VRP) interesują wielu badaczy ze względu na łatwość sformułowania konkretnego problemu – w przeciwieństwie do jego rozwiązania. Wśród znacznej liczba propozycji uzyskiwania najlepszych rozwiązań można wyróżnić metody dokładne, dające rozwiązania optymalne oraz metody przybliżone, które pozwalają na uzyskanie rozwiązań bliskich optymalnym. Metody heurystyczne stosowane są w tych problemach, w których czas potrzebny na uzyskanie rozwiązania optymalnego metodą dokładną jest zbyt długi i nie jest do zaakceptowania. W niniejszym artykule dokonano przeglądu wybranych algorytmów heurystycznych zaproponowanych dla problemów VRP, opartych na algorytmach ewolucyjnych, przeszukiwania tabu, algorytmach symulowanego wyżarzania oraz algorytmach mrówkowych.

Słowa kluczowe: metody heurystyczne, metody iteracyjne, metody planowania tras pojazdów

1. Wstęp

Wiele przedsiębiorstw stoi przed problemem właściwej organizacji transportu ludzi, dóbr lub informacji. Nie jest to jedynie problem przedsiębiorstw z samego sektora transportowego, lecz także np. zakładów produkcyjnych posiadających własne oddziały transportu. Ich zadaniem jest również organizacja przewozu bądź to surowców z jednego miejsca w inne, bądź to dystrybucji gotowych wyrobów do magazynów czy sieci detalicznej.

Do takich problemów można zaliczyć jedno z najbardziej popularnych zagadnień transportowych, z jakim spotykają się m.in. przedstawiciele handlowi. Jest nim problem komiwojażera, znany pod nazwą zagadnienia podróżującego sprzedawcy. Komiwojażer, startując z pewnego miasta, musi odwiedzić zadaną liczbę innych miast w taki sposób, aby każde miasto zostało zwizytowane tylko raz oraz droga, którą pokona, była jak najkrótsza (lub koszt podróży komiwojażera był jak najniższy). Rozszerzeniem problemu komiwojażera jest zagadnienie wielu komiwojażerów. Wszystkie miasta, które mają oni odwiedzić, muszą zostać jednoznacznie przyporządkowane danemu komiwojażerowi, a następnie należy znaleźć optymalną kolejność ich odwiedzania. Zarówno ustalenie przyporządkowania miast do komiwojażerów, jak i określenie kolejności ich wizyt muszą zostać wykonane w taki sposób, aby suma długości tras wszystkich komiwojażerów (lub łączny koszt podróży) była jak najniższa.

Najczęściej rola komiwojażerów nie ogranicza się tylko do samego odwiedzania miast. Często muszą oni np. dostarczyć pewną ilość dobra lub dóbr. Dodatkowo mogą być wprowadzone inne warunki dotyczące chociażby czasu pracy komiwojażerów czy czasu, w którym należy przybyć do danego miasta. Oznacza to, że wśród szerokiej klasy zagadnień wielu komiwojażerów można wyodrębnić wiele odmian, które określa się w literaturze ogólnie mianem zadań układania tras dla pojazdów (*vehicle routing problem* – VRP).

Cechą charakterystyczną zagadnień układania tras dla pojazdów jest łatwość sformułowania konkretnego problemu, w przeciwieństwie do jego rozwiązania. Dlatego też zagadnienia te interesują wielu badaczy, czego wynikiem jest znaczna liczba propozycji uzyskiwania najlepszych rozwiązań. Wśród zaproponowanych metod rozwiązań można wyróżnić metody dokładne, dające rozwiązania optymalne, oraz metody przybliżone, które pozwalają na uzyskanie rozwiązań bliskich optymalnym. Metody heurystyczne stosowane są w tych problemach, w których czas potrzebny na uzyskanie rozwiązania optymalnego metodą dokładną jest zbyt długi i nie jest do zaakceptowania.

2. Sformułowanie problemu VRP

Zagadnienie układania tras pojazdów można sformułować w następujący sposób. Załóżmy, iż pewna baza transportowa ma za zadanie dostarczyć (lub odebrać) pewne jednorodne dobro do sieci $\boldsymbol{N}$ punktów obsługi wokół niej zlokalizowanych. Baza dysponuje pewną liczbą $\boldsymbol{K}$ środków transportu, z których każdy ma określoną ładowność $\boldsymbol{Q}$. Z każdym obsługiwanym punktem związana jest pewna wielkość $\boldsymbol{q_i}$ oznaczająca popyt (podaż) dobra. Znana jest także macierz kosztów $\boldsymbol{D}$ przedstawiająca koszt (długość) przejazdu pojazdu pomiędzy bazą a zlokalizowanymi wokół niej punktami oraz pomiędzy samymi punktami. Macierz ta ma wymiary $(N + 1) \times (N + 1)$.

Celem jest ułożenie tras dostaw (odbioru) dobra w taki sposób, aby spełnione były następujące warunki:

– wszystkie punkty muszą być obsłużone, czyli w ramach dostaw popyt każdego punktu musi być w całkowicie zaspokojony, natomiast w ramach zadań odbioru dobra podaż każdego punktu musi być całkowicie zrealizowana,
– każdy punkt może być odwiedzony tylko przez jeden pojazd oraz każdy punkt może być obsłużony tylko jeden raz,
– ładowność każdego użytego pojazdu nie może zostać przekroczona,

– suma kosztów (lub długości) tras, pokonanych przez wszystkie użyte w tym celu pojazdy, ma być jak najmniejsza.
– W tak sformułowanym zadaniu układania tras pojazdów występują dwa problemy optymalizacyjne:
– podział zbioru wszystkich punktów na rejony, z których każdy zostanie przypisany do jednego pojazdu, którym dysponuje baza,
– wyznaczenie kolejności obsługi punktów przez pojazd w ramach rejonu, do którego został on przydzielony.

Po raz pierwszy problem wielu komiwojażerów został zdefiniowany w końcu lat pięćdziesiątych ubiegłego wieku. Do dziś zaproponowano wiele metod przeznaczonych do jego rozwiązania. Podstawowy ich podział wyróżnia metody dokładne oraz metody przybliżone (heurystyczne). Wśród algorytmów pozwalających dokładnie ustalić rozwiązanie optymalne na uwagę zasługują przede wszystkim metody realizujące strategię podziału i ograniczeń. Należy do nich procedura J.D. Little'a [Little i in. 1963] dla zadania jednego komiwojażera oraz metoda EAAM L.J. Jasińskiego [1987] dla problemu wielu komiwojażerów. Ze względu na dużą pracochłonność, bardzo długi czas poszukiwania rozwiązania optymalnego oraz efektywność tych metod tylko dla zadań o niewielkich rozmiarach (małej liczbie obsługiwanych punktów) znacznie większą popularność zyskały sobie metody heurystyczne.

3. Klasyczne metody heurystyczne dla problemów VRP

Algorytmy heurystyczne nie dają pewności co do tego, że uzyskane rozwiązanie jest optymalne. Jest ono raczej rozwiązaniem suboptymalnym, czyli bliskim najlepszemu. Jednak czas, w jakim może ono zostać znalezione, jest znacznie krótszy niż w metodach dokładnych. W zależności od wcześniej przyjętego podejścia do ustalania rozwiązania problemu wielu komiwojażerów można wyróżnić trzy podstawowe klasy metod heurystycznych:
– metody konstrukcyjne,
– metody dekompozycyjne,
– metody wzrostu.

3.1. Metody konstrukcyjne

Ogólna idea metod należących do procedur konstrukcyjnych polega na stopniowej budowie rozwiązania dopuszczalnego z uwzględnieniem jego kosztu. Jednocześnie odbywa się tutaj proces przyporządkowywania obsługiwanych przez bazę punktów do poszczególnych pojazdów oraz ustalania optymalnej kolejności ich obsługi.

Reprezentantem tej klasy metod dla zadania wielu komiwojażerów jest algorytm *savings* opracowany przez G. Clarke'a i J.W. Wrighta [1964]. Nazwa pochodzi od obliczanych podczas wykonywania procedury pewnych wielkości, nazywanych *oszczędnościami*. Przebieg działania algorytmu można przedstawić można w następujący sposób:

I. Utworzyć N tras: $\boldsymbol{B} - Pi - \boldsymbol{B}$.

II. Obliczyć „oszczędności": $\boldsymbol{s_{ij}} = d_{Bi} + d_{jB} - d_{ij}$.

III. Odrzucić $s_{ij} < 0$ oraz posortować malejąco pozostałe s_{ij}.

IV. Dla wszystkich s_{ij} rozpatrzyć możliwość połączenia w jedną trasę punktów $\boldsymbol{P_i}$ oraz $\boldsymbol{P_j}$ wskazywanych przez s_{ij}.

Etap IV powtarzany jest do momentu, gdy nie uda się już połączyć ze sobą istniejących tras. Trasy można łączyć tylko w punktach krańcowych, tzn. takich, które w swoich trasach obsługiwane są jako pierwsze lub ostatnie, a jednocześnie nie będzie przekroczona ładowność pojazdu.

Powyższy algorytm rozpoczyna działanie od niedopuszczalnej liczby tras, równej liczbie obsługiwanych punktów, które stopniowo potem łączy ze sobą z zachowaniem dopuszczalnej ładowności pojazdów oraz wcześniej ustalonej w tych trasach kolejności obsługi punktów.

Innym przykładem metod konstrukcyjnych są algorytmy zaproponowane przez R.H. Mole'a i S.R. Jamesona [1976] oraz N. Christofidesa, A. Mignozziego i P. Totha [1979], nazywane algorytmami sekwencyjnego wstawiania.

I. Utworzyć pierwszą trasę: $\boldsymbol{B} - Pi - \boldsymbol{B}$.

II. Dla każdego z pozostałych do obsłużenia punktów obliczyć koszt jego wstawienia do już istniejącej trasy i wybrać punkt o koszcie najniższym.

Dla powstałej częściowej trasy rozwiązać problem jednego komiwojażera.

Kroki II i III powtarzane są do momentu, gdy nie można już rozbudować trasy pojazdu (np. przekroczona ładowność pojazdu). Wtedy należy zainicjować nową trasę (krok I).

3.2. Metody dekompozycyjne

Odrębną klasę algorytmów heurystycznych stanowią metody opierające się na strategii rozdzielającej w rozpatrywanym problemie zagadnienie przydziału punktów obsługi do pojazdów oraz zagadnienie ustalania kolejności odwiedzania tych punktów przez dany pojazd.

Strategia ta może być realizowana w dwóch kierunkach: najpierw rejon obsługi, potem kolejność lub odwrotnie – najpierw kolejność, potem rejon. W drugim przypadku celem jest budowa jednej niedopuszczalnej trasy dla wszystkich obsługiwanych przez bazę punktów, by następnie podzielić ją na mniejsze podtrasy obsługiwane przez poszczególne pojazdy. Najczęściej rozpatrywanym podejściem jest najpierw ustalenie rejonów, które mają być obsłużone przez środki transportu, a potem układanie kolejności, w jakiej mają być obsłużone punkty przez przydzielone im pojazdy.

Reprezentantem tej klasy metod jest algorytm zaproponowany przez B.E. Gilletta i L.R. Millera [1974], nazywany popularnie metodą *sweep*[1]. Algorytm ten przeznaczony jest do modeli planarnych, w których punkty obsługi P_i reprezentowane są w przestrzeni Euklidesowej przez współrzędne (x_i, y_i).

I. Poprowadzić promień R^* przez dowolny punkt P^*.

II. Poprowadzić promienie R_i przez pozostałe punkty P_i i obliczyć kąty nachylenia α_i do R^*.

III. Posortować punkty P_i rosnąco względem α_i.

IV. Przyporządkować kolejno punkty P_i do pojazdów (warunek ładowności).

V. Dla każdego pojazdu rozwiązać problem komiwojażera.

Inną propozycją jest algorytm M. Fishera i R. Jaikumara [1981], w którym wykorzystuje się zagadnienie uogólnionego przydziału.

I. Dla każdego rejonu obsługi (pojazdu) należy wybrać punkty początkowe P^*_k.

II. Dla każdego z pozostałych do obsłużenia punktów obliczyć koszt jego wstawienia do każdej trasy.

III. Rozwiązać uogólnione zadanie przydziału z obliczonymi w kroku II kosztami, wagami odpowiadającymi zapotrzebowaniu poszczególnych punktów oraz ograniczeniami w postaci ładowności pojazdów.

IV. Dla uzyskanych w kroku III rejonów obsługi rozwiązać problem komiwojażera. Interesujący przykład dwuetapowego podejścia do rozwiązania problemu układania tras dla pojazdów zaprezentował A. Całczyński [1992] w algorytmie o nazwie BF-WOT (baz fikcyjnych – wymiany odcinków tras), w której na szczególną uwagę zasługuje pierwszy etap. Chcąc ustalić rejon obsługi dla pojazdu, należy rozwiązać klasyczne zagadnienie transportowe składające się z dwóch nadawców oraz odbiorców, którymi są punkty obsługi nie będące jeszcze przyporządkowane do żadnego z pojazdów. Jednym nadawcą jest baza, a drugim wybrany arbitralnie punkt, który nosi nazwę bazy fikcyjnej. Ilość dostarczonego przez bazę fikcyjną towaru równa jest ładowności pojazdu pomniejszonej o popyt punktu obsługiwanego przez pojazd. W drugim etapie, podobnie jak w poprzednich metodach, optymalizowane jest zagadnienie komiwojażera.

3.3. Metody wzrostu

Algorytmy wzrostu[2], zwane inaczej metodami przeszukiwania lokalnego, opierają się na strategii poprawy istniejącego już pewnego rozwiązania (rys. 4). Ogólnie podejście to można przedstawić w kilku etapach:

I. Ustalić jakiekolwiek początkowe rozwiązanie dopuszczalne X i przyjąć jako najlepsze dotychczas znalezione rozwiązanie X^*.

---

[1] *Sweep* – ang. zagarniać, zamiatać.

[2] W literaturze obcej metody te określane są mianem *improving heuristics*,

II. Wygenerować zbiór rozwiązań sąsiednich $N(X)$.
III. Znaleźć najlepsze rozwiązanie X' ze zbioru $N(X)$.
IV. Jeżeli X' jest lepsze od X^*, przyjąć je jako X^*.

Generowanie rozwiązań sąsiednich może polegać na dokonywaniu prostych przekształceń obecnie rozpatrywanego zbioru tras pojazdów, np. poprzez:
- wymianę podciągów punktów obsługi pomiędzy trasami poprzez skrzyżowanie krawędzi dwóch tras,
- bezpośrednią wymianę punktów obsługi między trasami,
- przeniesienie punktu obsługi z jednej trasy do drugiej.

4. Metaheurystyki dla problemów VRP

Jeżeli pojęciem heurystyka określa się zbiór reguł służących do przeszukiwania przestrzeni rozwiązań, to przez pojęcie metaheurystyka można rozumieć ramy i zasady generowania tych reguł. Metaheurystyka jest zatem pojęciem znacznie szerszym, co oznacza, że dla jednej metaheurystyki można zaproponować wiele algorytmów heurystycznych, tzw. instancji metaheurystyk.

Wśród algorytmów realizujących omawianą wcześniej strategię przeszukiwania przestrzeni rozwiązań – metod wzrostu można zaliczyć heurystyki zaliczane do kilku klas metaheusytyk:
- algorytmy przeszukiwania *tabu* (*tabu search*),
- algorytmy symulowanego wyżarzania (*simulated annealing*),
- algorytmy ewolucyjne (*evolutionary algorithms*),
- algorytmy (systemy) mrówkowe (*ant systems*).

4.1. Algorytmy przeszukiwania tabu

Algorytmy przeszukiwania tabu są procesami iteracyjnymi, realizującymi strategię przeszukiwania lokalnego (podpunkt 3.3), przeznaczonymi głównie do optymalizacji problemów kombinatorycznych.

I. Ustalić jakiekolwiek początkowe rozwiązanie dopuszczalne X i przyjąć jako najlepsze dotychczas znalezione rozwiązanie X^*.

II. Wygenerować zbiór rozwiązań sąsiednich $N(X)$ z uwzględnieniem reguł tabu w pamięci.

III. Znaleźć najlepsze rozwiązanie X' ze zbioru $N(X)$.
IV. Jeżeli X' jest lepsze od X^*, przyjąć je jako X^*.
V. Zaktualizować struktury pamięci.

Istotą tych algorytmów jest określenie sposobu generowania sąsiedztwa aktualnie rozpatrywanego rozwiązania, a szczególnie określenie mechanizmu konstruującego ograniczenia dotyczące generowania pewnych rozwiązań sąsiednich. Celem nakładanych ograniczeń jest zminimalizowanie ryzyka zapętlenia się algorytmu, tzn. ponownego rozpatrywania rozwiązań już wcześniej egzaminowanych, a tym samym skierowanie procesu poszukiwania rozwiązania optymalnego w inne rejony przestrzeni rozwiązań. Ograniczenia nakładane na zbiór rozwiązań sąsiednich mogą mieć charakter zakazu lub restrykcji. W przypadku rozwiązania sąsiedniego objętego zakazem nie jest w ogóle możliwe przejście do niego od rozwiązania aktualnie rozpatrywanego. Natomiast w przypadku restrykcji jest to możliwe pod warunkiem, że w całym procesie działania algorytmu może to przynieść określoną korzyść.

Podstawowymi elementami algorytmów tabu są: struktura pamięci oraz zbiór reguł ograniczających wybór rozwiązania sąsiedniego. W strukturze pamięci gromadzone są informacje o przeszukiwanym obszarze przestrzeni rozwiązań (o zrealizowanych dotychczas przejściach). Może ona także przechowywać informacje o częstotliwości przejść lub czasie (iteracjach), jaki upłynął od wykonania każdego z przejść. Struktura pamięci przyjmuje często formę listy.

Zbiór reguł ograniczających budowany jest na podstawie:

– aktualności zapisanych w pamięci danych (o dotychczasowych przejściach),
– częstotliwości zapisu w pamięci określonych danych,
– wpływu danych zgromadzonych w pamięci na jakość uzyskiwanych wyników.

Przykładem algorytmu tabu jest algorytm TABUROUTE zaproponowany przez M. Gendreau, M. Hertza i G. Laporte'a [1994], w którym sąsiednie rozwiązanie w stosunku do aktualnie rozpatrywanego zbioru tras generowane jest poprzez usunięcie wybranego punktu obsługi z jednej trasy i wstawienie go do innej. Algorytm ten może rozpatrywać także rozwiązania niedopuszczalne (przekroczenie ładowności pojazdu). Jednak proponowana konstrukcja funkcji oceniającej rozwiązanie uwzględnia taką możliwość. Oprócz elementu określającego długość wszystkich tras w rozwiązaniu (funkcja celu problemu) zawiera także element kary, powiększający jej wartość, gdy rozwiązanie nie spełnia nałożonych ograniczeń zasobowych. Elementem ograniczającym wybór rozwiązania sąsiedniego (tabu) jest przyjęty zakaz powrotnego umieszczania wybranego punktu obsługi w trasie, z której został on wcześniej usunięty. Zakaz ten utrzymywany jest przez pewien czas θ (liczbę iteracji), który jest parametrem tego algorytmu. Przyjęty parametr ustalany jest losowo z przedziału (5; 10). Oznacza to, że punkt P_i usunięty z trasy pojazdu k nie może być ponownie do niej włączony wcześniej niż po upływie θ iteracji.

Wśród innych algorytmów heurystycznych należących do metod przeszukiwania tabu można wyróżnić algorytm z podziałem punktów na sektory i regiony koncentracji (z bazą jako środkiem koła), zaproponowany przez E.D. Taillarda [1993], czy algorytm Granular Tabu Serach P. Totha i D. Vigo [2001]. W tym ostatnim sąsiedztwo generowane jest poprzez wymianę określonej liczby połączeń (krawędzi) punktów w tej samej trasie lub pomiędzy dwiema trasami. Natomiast rozpatrywane są tylko te krawędzie, których długość jest mniejsza od pewnej średniej długości krawędzi w zbiorze tras otrzymanym jako rozwiązanie przy wykorzystaniu jednej z klasycznych metod heurystycznych.

4.2. Algorytmy symulowanego wyżarzania

Podobnie jak algorytmy przeszukiwania tabu, algorytmy symulowanego wyżarzania także są procesami iteracyjnymi, realizującymi strategię przeszukiwania lokalnego (podpunkt 3.3). Algorytm ten po raz pierwszy zaprezentowali: S. Kirkpatrick, C.D. Gelatt i M.P. Vecchi [1983], w którym zastosowane zostało prawo termodynamiki mówiące, że przy ustalonej temperaturze t prawdopodobieństwo wzrostu energii cząsteczki o ΔE określonej jest wzorem: $e^{-\Delta E/t}$. Algorytm symulowanego wyżarzania można przedstawić następująco:

I. Ustalić jakiekolwiek początkowe rozwiązanie dopuszczalne X i przyjąć jako najlepsze dotychczas znalezione rozwiązanie X^*.

II. Określić temperaturę początkową $T_{\max}$, końcową $T_{\min}$ oraz określić funkcję jej spadku.

III. Dla określonego limitu prób k przy ustalonej temperaturze T:
- IIIa) wygenerować rozwiązanie sąsiednie X',
- IIIb) obliczyć $\Delta E = f(X) - f(X')$,
- IIIc) jeżeli $\Delta E < 0$, przyjąć $X^* = X'$ oraz $X = X'$,
- IIId) jeżeli $\Delta E \geq 0$, sprawdzić $e^{-\Delta E/t} \geq \theta$ i przyjąć $X = X'$.

IV. Obniżyć temperaturę $T = g(T)$.

Powyższy schemat jest powtarzany do momentu, aż proces ulegnie zamrożeniu, tzn. temperatura T osiągnie ustaloną wartość minimalną $T_{\min}$.

Istotnymi parametrami tych algorytmów są:
– temperatura początkowa $T_{\max}$ oraz końcowa $T_{\min}$,
– funkcja spadku temperatury $g(T)$,
– liczba prób k przy ustalonej temperaturze T,
– prawdopodobieństwo przejścia do rozwiązania sąsiedniego $\theta \in (0; 1)$.

Pierwsze cztery parametry: temperatura początkowa i końcowa, funkcja spadku, liczba prób, są elementami wpływającymi w sposób ogólny na przebieg procesu przeszukiwania przestrzeni rozwiązań przez cały algorytm, a zwłaszcza na jego tempo. Ostatni parametr, a mianowicie prawdopodobieństwo θ, umożliwia przejście do rozwiązania gorszego od aktualnie rozpatrywanego.

Przykład algorytmu symulowanego wyżarzania dla problemu VRP podał I.H. Osman [1993], w którym rozwiązanie początkowe generowane jest z wykorzystaniem algorytmu *savings*. Rozwiązania sąsiednie generowane są poprzez realokację punktu obsługi z jednej trasy do drugiej lub poprzez wymianę dwóch punktów pomiędzy trasami.

4.3. Algorytmy ewolucyjne

W przeciwieństwie do dwóch pierwszych metaheurystyk, algorytmy ewolucyjne należą do algorytmów wieloagentowego przeszukiwania przestrzeni rozwiązań.

Oznacza to, że w każdej iteracji procesu przeszukiwania rozpatrują jednocześnie nie jedno, a wiele rozwiązań optymalizowanego problemu. Są one metodami rozwiązywania zadań optymalizacyjnych opartymi na symulacji procesów ewolucyjnych zachodzących wśród organizmów żywych. Podstawowymi cechami każdego algorytmu ewolucyjnego są:

– rozwiązania zadania zakodowane są w postaci osobników (chromosomów), które podlegają procesom selekcji i reprodukcji;
– algorytm musi mieć funkcję oceny (kryterium), na podstawie której ocenia się jakość potencjalnych rozwiązań;
– zawiera procedury kształtujące kolejne lepsze rozwiązania (pokolenia);
– algorytmy ewolucyjne nie wymagają dogłębnej znajomości optymalizowanego problemu, lecz jedynie podstawowej wiedzy na jego temat.

Proces działania algorytmu ewolucyjnego zawiera poniższy schemat:

I. Ustalić pewną liczbę (populację) rozwiązań początkowych X i określić ich wartość $F(X)$.

II. Ustalić najlepsze dotychczas znalezione rozwiązanie $\boldsymbol{x}^*$ w X.

III. Dla określonej liczby pokoleń (iteracji algorytmu):

IIIa) przeprowadzić proces selekcji,
IIIb) zastosować operatory genetyczne,
IIIc) przeprowadzić proces sukcesji – utworzyć nową populację X',
IIId) znaleźć najlepsze rozwiązanie x^{pop} w X',
IIIe) jeżeli $f(x^{pop}) < f(x^*)$, przyjąć $x^* = x^{pop}$.

Propozycje algorytmów ewolucyjnych dla problemów VRP przedstawili Ch. Prins [2004] i R. Jadczak [2005].

Wykorzystywane metody kodowania rozwiązań w postaci chromosomów były oparte na kodowaniu całkowitoliczbowym i permutacyjnym. Oznacza to, że chromosomy były reprezentowane przez wektor liczb całkowitych odpowiadających poszczególnym punktom obsługi.

Procesy selekcji służyły do wygenerowania subpopulacji dobrych rozwiązań, których przekształcenie w wyniku zastosowania operatorów genetycznych dawało największe szanse na uzyskanie lepszych zbiorów tras w kolejnym pokoleniu (iteracji) algorytmu.

Operatory genetyczne to procesy krzyżowania, mutacji oraz lokalnej optymalizacji. Krzyżowanie polega na wymianie informacji pomiędzy dwoma chromosomami. Natomiast mutacja polega na przekształceniu jednego chromosomu w inny (np. poprzez zmianę położenia wybranego elementu wektora). Lokalna optymalizacja także może uchodzić za pewien rodzaj mutacji, z tą różnicą, iż często stosowany jest tutaj pewien prosty algorytm poprawy istniejącego rozwiązania, ale na jego zdekodowanej postaci.

Wreszcie ostatni z elementów algorytmu ewolucyjnego to sukcesja, czyli właściwy etap generowania rozwiązań sąsiednich. Sukcesja to proces utworzenia nowej populacji chromosomów, czyli nowego zbioru rozwiązań, które zostaną poddane ponownej ocenie. Nowa populacja może być tworzona na podstawie zbioru chromosomów z populacji wyjściowej (elitarność) oraz zbioru chromosomów z populacji utworzonej po zastosowaniu operatorów genetycznych.

4.4. Algorytmy mrówkowe

Innym przykładem systemów wieloagentowych zaproponowanych do wykorzystania w rozwiązywaniu problemów VRP są algorytmy symulujące zachowanie mrówek. Każdy z agentów (mrówek) realizuje taką samą strategię przeszukiwania przestrzeni rozwiązań. Są to algorytmy przeznaczone niemal wyłącznie do rozwiązywania zadań o charakterze kombinatorycznym. Jednym z pierwszych przykładów, do których zostały one wykorzystane, jest problem komiwojażera. Schemat działania algorytmu mrówkowego przedstawić można w następujący sposób:

I. Ustalić początkową ilość feromonu τ_{ij} na każdym połączeniu punktów P_i, P_j.

II. Odparować część feromonu ze wszystkich połączeń par punktów.

III. Wygenerować ścieżki dla wszystkich mrówek począwszy od punktu startowego (bazy).

IV. Nanieść nowy poziom feromonu na każdym połączeniu punktów P_i, P_j w zależności od jakości uzyskanych rozwiązań.

Wybór kolejnego punktu P_j w trasie odbywa się w sposób losowy, tzn. kolejne punkty dobierane są z pewnym prawdopodobieństwem p, którego wielkość zależy od aktualnego poziomu feromonu na połączeniu punktu, z którego mrówka wyrusza dalej do kolejnego potencjalnego punktu, a także od długości połączenia pomiędzy nimi. Im wyższy poziom feromonu na połączeniu lub im mniejsza długość połączenia, tym większe prawdopodobieństwo włączenia nowego punktu do trasy.

Poziom feromonu na połączeniu par punktów jest pewnego rodzaju przekaźnikiem informacji o jakości (długości) trasy pomiędzy mrówkami. Jednocześnie sposób (funkcja) nanoszenia nowego poziomu feromonu oraz funkcja jego zmniejszania (odparowywania) są elementami algorytmu warunkującymi zakres przeszukiwania przestrzeni rozwiązań rozwiązywanego problemu optymalizacyjnego.

Przykład wykorzystania przedstawionego powyżej algorytmu mrówkowego do problemu VRP zaprezentowali B. Bullnheimer, R.F. Hartl i Ch. Strauss [1999], w którym trasa pojedynczej mrówki jest przerywana w razie przekroczenia ładowności pojazdu i rozpoczynana ponownie z bazy B w celu obsłużenia pozostałych, nieprzydzielonych jeszcze do tras punktów.

5. Zadania testowe

W tabeli 1 przedstawiono grupę zadań testowych, które były rozwiązywane przy wykorzystaniu algorytmów heurystycznych należących do jednej z klas przedstawionych wcześniej metod: CH – algorytmy konstrukcyjne, DH – algorytmy dekompozycyjne, TS – algorytmy przeszukiwania tabu, SA – algorytmy symulowanego wyżarzania, EA – algorytmy ewolucyjne, AS – algorytmy mrówkowe.

Obok nazwy zadania w następnej kolumnie podana jest liczba punktów obsługiwanych przez bazę. Natomiast w kolumnie ostatniej podane są najlepsze dotychczas znalezione rozwiązania (długości zbioru tras).

Jak wskazują wyniki, realizowanie strategii wzrostu w rozwiązywaniu zagadnień VRP przynosiło lepsze rezultaty niż strategie konstrukcyjne lub dekompozycyjne. Z kolei, rozpatrując efektywność poszczególnych metaheurystyk, można stwierdzić, że najlepiej radziły sobie algorytmy heurystyczne oparte na przeszukiwaniu tabu lub realizujące podejście ewolucyjne.

Należy jednocześnie zaznaczyć, iż wiele proponowanych przez autorów algorytmów, opartych na przedstawionych metaheurystykach, zawiera elementy tzw. hybrydyzacji, czyli włączenia prostych i szybkich klasycznych heurystyk w celu poprawy wygenerowanego rozwiązania. Przykładem takiego procesu jest zastosowanie prostego algorytmu 2-opt w celu poprawy trasy (problemu komiwojażera) w zbiorze wygenerowanych tras jako uzyskanego rozwiązania.

Tabela 1. Porównanie algorytmów heurystycznych i metaheurystycznych

Zadanie	*n*	*CH*	*DH*	*TS*	*S.A.*	*EA*	*AS*	*R**
E051-05e	50	547	524	524,61	528	524,61	524,61	524,61
E076-10e	75	883	857	835,26	838	835,26	870,58	835,26
E101-08e	100	851	833	826,14	829	826,14	879,43	826,14
E101-10c	100	827	824	819,56	826	819,56	819,96	819,56
E121-07c	120	1066	1266	1042,11	1176	1042,11	1072,45	1042,11
E200-17c	199	1395	1389	1291,45	1378	1296,39	1473,40	1291,45
D051-06c	50	565	560	555,43	555	555,43	562,93	555,43
D076-11c	75	969	916	909,68	909	909,68	948,16	909,68
D101-09c	100	915	885	865,94	866	865,94	886,17	865,94
D101-11c	100	875	876	866.37	890	866,37	869,86	866,37
D121-11c	120	1590	1776	1541,14	1545	1542,86	1590,52	1541,14
D151-14c	150	1245	1230	1028,42	1058	1030,46	1147,41	1028,42
D200-18c	200	1508	1518	1397,94	1417	1402,75	1504,79	1395,85

Źródło: [Bullnheimer, Hartl, Strauss, 1999; Laporte, Semet 2000; Osman 1993; Mole, Jameson 1976; Prins 2004; Taillard 1993].

Literatura

Bullnheimer B., Hartl R.F., Strauss Ch., *Applying the ant system for the vehicle routing problem*, "Annals of Operation Research" 1999 vol. 89, s. 319-328.

Całczyński A., *Metody optymalizacji w obsłudze transportowej rynku*, PWE, Warszawa 1992.

Christofides N., Mignozzi A., Toth P., *The vehicle routing problem*, [w:] *Combinatorial Optimization*, eds. N. Christofides, A. Mignozzi, P. Toth, C. Sand, Wiley, New York 1979, s. 315-338.

Clarke G., Wright J.W., *Scheduling of vehicles from a central depot to a number of delivery points*, "Operations Reseach" 1964 vol. 12, s. 568-581.
Fisher M., Jaikumar R., *A generalized assignment heuristic for the vehicle routing*, "Networks" 1981 vol. 11, s. 109-124.
Gendreau M., Hertz M., Laporte G., *A tabu search heuristic for the vehicle routing problem*, "Management Science" 1994 vol. 40, s. 1276-1290.
Gillett B.E., Miller L.R., *A heuristic algorithm for the vehicle dispatch problem*, "Operations Research" 1974 vol. 22, s. 340-349.
Jadczak R., *Rozwiązywanie zagadnień układania tras pojazdów z wykorzystaniem algorytmów ewolucyjnych*, „Badania Operacyjne i Decyzje" 2005 nr 3-4, s. 7-22.
Jasiński L.J., *Optymalizacja dostawy towarów na zaopatrzenie rynku w warunkach niepewności*, Instytut Rynku Wewnętrznego i Konsumpcji, Warszawa 1987.
Kirkpatrick S., Gelatt J.C.D., Vecchi M.P., *Optimization by simulated annealing*, "Science" 1983 vol. 220, s. 671-680.
Laporte G., Semet F., *Classical heuristic for the vehicle routing problem*, [w:] *The Vehicle Routing Problem*, monograph on discrete mathematics and Applications, eds. P. Toth, D. Vigo, SIAM, Austin 2000.
Little J.D.C., Murty K.G., Sweeney D.W., Karel C., *An algorithm for the traveling salesman problem*, "Operations Research" 1963 vol. 11.
Osman I.H., *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, "Annals of Operations Research" 1993 vol 41, s. 421-451.
Mole R.H., Jameson S.R., *A sequential route-building algorithm employing a generalized savings criterion*, "Operations Research Quarterly" 1976 vol. 27, s. 503-511.
Prins Ch., *A simple effective evolutionary algorithm for the vehicle routing problem*, "Computers and Operations Research" 2004 vol. 31, s. 1985-2002.
Taillard E.D., *Parallel iterative search methods for the vehicle routing problem*, "Networks" 1993 vol. 23, s. 661-673.
Toth P., Vigo D., *Granular tabu search heuristic for vehicle routing problems with time windows*, "Journal of the Operational Research Society" 2001 vol. 52, s. 928-936.

HEURISTICS AND METAHEURISTICS FOR THE VEHICLE ROUTING PROBLEM (VRP)

Summary: The paper presents the vehicle routing problem (VRP), its modifications and main strategies used to solving this combinatorial problem. The review of proposed heuristics and the comparison of results are presented.