AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/2

Lidia Dutkiewicz*, Edyta Kucharska*

Algorytm planowania tras dostaw dla wielu komiwojażerów

1. Wprowadzenie

Celem artykułu jest przedstawienie algorytmu optymalizującego problem planowania tras dostaw do firm wielooddziałowych. Problem planowania tras dostaw jest modyfikacją powszechnie znanego problemu wielu komiwojażerów. Zaproponowany algorytm należy do klasy heurystycznych algorytmów konstrukcyjnych i opracowany został na podstawie metody zadań zastępczych wykorzystującej ogólny schemat modelu algebraiczno-logicznego.

Ogólny schemat modelu algebraiczno-logicznego stanowi pewien sposób reprezentacji wiedzy o problemie. Model algebraiczno-logiczny jest modelem w przestrzeni stanów i przeznaczony jest do optymalizacji procesów decyzyjnych. Odpowiada on jednocześnie formalnej postaci wieloetapowego procesu decyzyjnego połączonego z symulacją procesu dyskretnego.

Ideę ogólnego schematu modelu algebraiczno-logicznego zaproponowano w [2]. Schemat ten powstał na podstawie metody logiczno-algebraicznej, wprowadzonej przez Z. Bubnickiego [1]. Istotą klasy modeli algebraiczno-logicznych jest fakt, że zarówno współrzędne stanu, jak i decyzje (sterowania) mogą być zmiennymi indywiduowymi lub zmiennymi wyższego rzędu. Funkcja przejścia i ograniczenia mogą natomiast być zdefiniowane zarówno za pomocą zależności algebraicznych jak i logicznych.

Jeśli przyjmiemy oznaczenia: X – zbiór stanów właściwych, $T \subset \boldsymbol{R}^{+}$ – podzbiór nieujemnych liczb rzeczywistych reprezentujących chwile czasowe, $S = X \times T$ – zbiór stanów uogólnionych, U – zbiór decyzji, to model można zdefiniować jako czwórkę:

$$P = (s_0, f, S_N, S_G) \tag{1}$$

gdzie:

$s_0 = (x_0, t_0)$, $s_0 \in S$ – uogólniony stan początkowy,

f: $U \times S \rightarrow S$ – funkcja częściowa (określona tylko dla pewnych par $(u, s) \in U \times S$), zwana funkcją przejścia,

$S_N \subset S$ – zbiór uogólnionych stanów niedopuszczalnych,

$S_G \subset S$ – niepusty zbiór uogólnionych stanów docelowych.

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

Funkcja przejścia jest zdefiniowana za pomocą dwóch funkcji $f = (f_x, f_t)$, gdzie:

f_x: $U \times X \times T \rightarrow X$ – określa następny stan właściwy,
f_t: $U \times X \times T \rightarrow T$ – określa następny moment czasu i spełnia następujący warunek: $\Delta t = f_t(u, x, t) - t$ ma wartość dodatnią i skończoną.

Zdefiniowanie funkcji przejścia jako funkcji częściowej pozwala na uwzględnienie wszystkich ograniczeń dotyczących decyzji sterujących za pomocą tzw. zbiorów sterowań możliwych w stanie s, oznaczonych $U_p(s)$. Jeśli decyzja u jest możliwa (sensowna) w stanie s, to funkcja przejścia jest określona dla tej pary (u, s). W przeciwnym wypadku nie jest określona.

Zadanie poszukiwania rozwiązania dopuszczalnego polega na znalezieniu ciągu decyzji wyznaczającego trajektorię dopuszczalną. Odpowiednie zadanie optymalizacji sterowania procesem polega na znalezieniu takiego ciągu sterowań dopuszczalnych, który ekstremalizuje pewien funkcjonał Q. Zadanie optymalizacji określone jest zatem przez model procesu P oraz funkcjonał Q i zapisywane jest jako para (P, Q).

Model algebraiczno-logiczny stanowi podstawę do tworzenia różnego rodzaju algorytmów optymalizacji opartych na symulacji procesu decyzyjnego [3]. Ogólna zasada działania algorytmów opartych na modelu algebraiczno-logicznym jest następująca. W każdym, nowo wyznaczonym stanie symulowanego procesu podejmowana jest decyzja dotycząca następnego kroku jego prowadzenia. Decyzja ta wybierana jest spośród możliwych (sensownych) dla danego stanu decyzji. Dla danego stanu i wybranej decyzji można wyznaczyć następny stan procesu oraz odpowiadający mu moment czasu. Wyznacza się je, korzystając z funkcji przejścia procesu. Jeśli wyznaczony stan należy do zbioru stanów docelowych procesu, przebieg symulacji jest zakończony pomyślnie i można dokonać jego oceny. Jeśli nowo wyznaczony stan i związany z nim czas nie spełniają warunków narzuconych ograniczeniami technologicznymi i czasowymi na proces, to stan taki znajduje się w zbiorze tzw. stanów niedopuszczalnych i nie ma sensu prowadzenie dalszej symulacji procesu.

Na podstawie modelu algebraiczno-logicznego możliwe jest stworzenie zarówno algorytmu dokładnego, jak i algorytmów przybliżonych. Rozwiązanie dokładne uzyskiwane jest metodą przeglądu zupełnego w łatwy sposób, gdyż model zapewnia przebadanie wszystkich możliwości (w każdym stanie s jest określony zbiór decyzji możliwych do podjęcia). Niestety, jako że algorytmy dokładne mogą efektywnie działać tylko dla instancji problemu o małych rozmiarach, mają one ograniczone zastosowanie.

Model algebraiczno-logiczny daje duże możliwości tworzenia algorytmów przybliżonych [6]. Głównym zadaniem takiego algorytmu przybliżonego jest odpowiedni wybór i podejmowanie decyzji w trakcie symulacji procesu. Jednym ze sposobów jest wygenerowanie wszystkich decyzji ze zbioru $U_p(s)$ decyzji możliwych do podjęcia w danym stanie s, ocena tych decyzji za pomocą lokalnych procedur, a następnie wybór jednej z nich. Innym podejściem do wyznaczania decyzji w danym stanie jest jej ustalenie na podstawie specjalnie skonstruowanych zasad. Zasady wyboru takiej decyzji muszą być zaprojektowane tak, aby zapewnić zarówno dopuszczalność decyzji, jak i odpowiednią jakość tworzonych na ich podstawie rozwiązań. Najczęściej korzysta się z różnego rodzaju reguł priorytetowych.

Niezwykle istotne jest to, iż algorytmy stworzone na bazie formalnego modelu jakim jest model algebraiczno-logiczny, pozostają niezależne od konkretnego języka programowania i struktury danych. Precyzyjne określenie składników systemu umożliwia łatwe wprowadzanie ewentualnych modyfikacji w algorytmie czy nawet całkowitą zmianę metody optymalizacji. Dla tego samego problemu można podać różne algorytmy oparte na modelu algebraiczno-logicznym, zatem istnieje możliwość obiektywnego porównania ich efektywności i jakości.

Proponowana metodologia ma szczególne zastosowanie w problemach należących do klasy problemów NP-trudnych, w których optymalizowane kryterium jest addytywnie separowalne i monotonicznie rosnące.

2. Opis problemu

Rozważany problem dotyczy planowania tras dostaw dla firmy dostawczej, która współpracuje z wieloma odbiorcami, w tym z firmami posiadającymi wiele oddziałów. Oddziały firm rozproszone są po całym obsługiwanym obszarze. W firmie dostawczej zatrudniona jest pewna liczba dostawców (komiwojażerów), którzy mają dostarczyć towar. W przypadku firm wielooddziałowych najpierw należy odwiedzić główną siedzibę (w celu negocjacji, ustalenia warunków sprzedaży dotyczących wszystkich oddziałów tej firmy, itp.). Tak postawiony problem można potraktować jako zmodyfikowany problem wielu komiwojażerów i w związku z tym sformułować następująco. Dany jest zespół komiwojażerów oraz zbiór miast do odwiedzenia, przy czym każde miasto reprezentuje siedzibę pewnej firmy lub jej oddział. Dane jest też miasto początkowe (czyli firma dostawcza). Komiwojażerowie wyruszają z miasta początkowego i ich zadaniem jest odwiedzenie wszystkich miast, a następnie powrót do miasta początkowego. Znane są odległości między poszczególnymi miastami. Każde miasto (oprócz początkowego) musi zostać odwiedzone dokładnie jeden raz, przy czym odwiedzającym może być dowolny komiwojażer z zespołu. Dodatkowo żaden komiwojażer nie może wyruszyć do miasta reprezentującego jeden z oddziałów danej firmy, zanim nie zostanie odwiedzona siedziba tej firmy. Natomiast jeżeli siedziba została już odwiedzona, to dowolny komiwojażer z zespołu może odwiedzić którykolwiek z jej oddziałów.

Celem optymalizacji jest takie zaplanowanie tras komiwojażerów, aby cały zbiór miast był odwiedzony w jak najkrótszym czasie.

W pracy przyjęto następujące oznaczenia: zbiór komiwojażerów oznaczony został poprzez $M = \{m_1, m_2, ..., m_{|M|}\}$, zbiór miast do odwiedzenia poprzez $N = \{n_1, n_2, ..., n_{|N|}\}$, miasto początkowe jako n_0.

Ponadto zdefiniowany został ciąg $E = (e_1, e_2, ..., e_E)$, którego poszczególne wyrazy to miasta reprezentujące siedziby firm i firmy jednooddziałowe. Każdemu wyrazowi tego ciągu e_j (dla $j = 1, ..., |E|$) odpowiada jeden zbiór F_j miast reprezentujących oddziały danej firmy, przy czym dla firm jednooddziałowych zbiór taki jest pusty. Firmy jednooddziałowe mogą być zatem traktowane jako miasta-siedziby bez oddziałów.

Odległości między miastami przedstawione są w postaci macierzy odległości A, której poszczególne elementy a_{ij} określają odległość pomiędzy miastem n_i oraz miastem n_j, gdzie $i, j = 0, 1, 2, ..., |N|$ oraz $i \neq j$. Przyjmuje się, że elementy a_{ii} wynoszą nieskończoność.

Ponadto zakłada się, że wszyscy komiwojażerowie podróżują z tą samą, jednostkową prędkością, zatem wartości czasu przejazdu i pokonywanej w tym czasie drogi są takie same.

Rozważany problem należy do klasy problemów NP-trudnych, gdyż nawet relaksując problem przez usunięcie ograniczenia związanego z koniecznością udostępniania miast, otrzymujemy nadal problem NP-trudny (problem *m*-komiwojażerów).

3. Model algebraiczno-logiczny problemu

W niniejszym rozdziale przedstawiony został model algebraiczno-logiczny rozważanego problemu: postać stanu systemu, zbiór stanów docelowych oraz stanu niedopuszczalnego, postać decyzji, zbiór decyzji możliwych do podjęcia w poszczególnych stanach oraz zbiór decyzji dopuszczalnych, a także elementy składające się na funkcję przejścia.

Dla rozważanego problemu stan systemu $s = (x, t)$ w danej chwili czasu t można opisać przez aktualny stan wszystkich komiwojażerów oraz zbiór odwiedzonych miast. Stan właściwy systemu x określany jest następująco:

$$x = (x^0, x^1, x^2, ..., x^{|M|}) \tag{2}$$

gdzie:

x^0 – zbiór miast, które zostały odwiedzone do chwili t,

x^k – stan k-tego komiwojażera, dla $k = 1, 2, ..., |M|$.

W danej chwili komiwojażer może albo jechać do następnego miasta, albo stać w ostatnio odwiedzonym mieście. Stan pojedynczego komiwojażera w danej chwili t jest następujący:

$$x^k = (n, r) \tag{3}$$

gdzie poszczególne zmienne przyjmują wartości:

$n \in N \cup n_0$ – wyznaczone miasto, do którego komiwojażer jedzie, lub miasto w którym stoi,

$r \in R^+$ – długość odcinka drogi jaka pozostała komiwojażerowi do dotarcia do wyznaczonego miasta; w przypadku postoju komiwojażera przyjmowana jest wartość 0.

Stan początkowy $s_0 = (x_0, t_0)$ systemu jest następujący:

$$x_0 = (x_0^0, x_0^1, x_0^2, \ldots, x_0^{|M|}), \quad t_0 = 0 \tag{4}$$

W stanie początkowym (gdy $t_0 = 0$) wszyscy komiwojażerowie stoją w mieście początkowym n_0, zatem $x_0^k = (n_0, 0)$. Ponadto w chwili t_0 żadne miasto nie zostało odwiedzone, a więc zbiór x^0 jest zbiorem pustym: $x_0^0 = \varnothing$.

W zbiorze stanów niedopuszczalnych S_N znajdują się stany związane z przekroczeniem warunków ograniczających dla danego zadania. W rozważanym problemie stanem niedopuszczalnym jest stan, w którym wszyscy komiwojażerowie powrócili do miasta początkowego, a nie wszystkie miasta zostały odwiedzone.

$$S_N = \left\{ s = (x, t) :\ x^0(s) \neq N \wedge \forall_{k=1,\ldots,|M|} \left(x^k(s) = (n_0, 0) \right) \wedge t > 0 \right\} \tag{5}$$

Celem komiwojażerów jest odwiedzenie wszystkich miast i powrót do miasta początkowego n_0. Zbiór stanów docelowych S_G jest zatem następujący:

$$S_G = \left\{ s = (x, t) : s \notin S_N \wedge x^0(s) = N \wedge \forall_{k=1,\ldots,|M|} \left(x^k(s) = (n_0, 0) \right) \right\} \tag{6}$$

W danym stanie s można wyróżnić następujące dwa zbiory: komiwojażerów wolnych $M_W(s)$ oraz komiwojażerów zajętych $M_Z(s)$.

Komiwojażer wolny w stanie s to komiwojażer, który stoi w mieście (ostatnio odwiedzonym lub początkowym) i może mu zostać przydzielone kolejne miasto do odwiedzenia.

$$M_W(\mathrm{s}) = \left\{ m_k \in M : x^k(s) = (n, 0) \right\} \tag{7}$$

Komiwojażer zajęty w stanie s to komiwojażer, który jedzie do wyznaczonego miasta lub zakończył swoją podróż i powrócił do miasta początkowego 0.

$$M_Z(\mathrm{s}) = \left\{ m_k \in M : x^k(s) = (n, r > 0) \vee \left(x^k(s) = (n_0, r = 0) \wedge t(s) > 0 \right) \right\} \tag{8}$$

W danym stanie $s = (x, t)$ podejmowana jest decyzja, na podstawie której system przeprowadzany jest do następnego stanu. Wyznaczona decyzja $u(s)$ musi należeć do zbioru $U_p(s)$ decyzji możliwych (sensownych) w danym stanie.

W rozważanym problemie przyjęto, iż decyzja o przydzieleniu komiwojażerowi następnego miasta do odwiedzenia może być podjęta dopiero po dotarciu komiwojażera do miasta poprzednio wyznaczonego, ponadto podjęta decyzja nie ulega zmianie. W konsekwencji powyższych założeń, dla komiwojażera, który jeszcze nie dotarł do wskazanego mu wcześniej miasta, dozwolona jest tylko decyzja o kontynuacji przemieszczania się w kierunku tego miasta. Natomiast dla komiwojażera, który w danej chwili stoi, może zostać podjęta decyzja o odwiedzeniu kolejnego miasta, o powrocie do miasta początkowego lub o pozostaniu w danym mieście. Kolejnym miastem wskazanym do odwiedzenia może być tylko miasto nieodwiedzone, do którego nie zmierza aktualnie żaden z komiwojażerów. Jednocześnie miasto takie musi reprezentować siedzibę firmy albo oddział takiej firmy, której siedziba została już odwiedzona.

Uwzględniając powyższe rozważania strukturę decyzji u można zdefiniować jako m-elementowy wektor $u = (u^1, u^2, \ldots, u^{|M|})$, gdzie kolejne współrzędne oznaczają odrębne decyzje dla poszczególnych komiwojażerów.

Wartością każdej współrzędnej u^k jest miasto, do którego komiwojażer ma jechać lub w którym ma nadal stać, czyli wartość $u^k \in N \cup n_0$.

Cały zbiór decyzji możliwych w stanie $s = (x, t)$ jest następujący:

$$U_p(s) = U_p^1(s) \times U_p^2(s) \times \ldots \times U_p^m(s) \setminus H \tag{9}$$

gdzie H – zbiór decyzji przydzielających jednocześnie to samo miasto do odwiedzenia więcej niż jednemu komiwojażerowi oraz niewyznaczających żadnego nowego miasta do odwiedzenia w sytuacji, kiedy wszyscy komiwojażerowie w danym stanie są wolni.

Jeżeli natomiast decyzja możliwa przeprowadza stan $s = (x, t)$ do zbioru S_N, to jest to decyzja niedopuszczalna. Zbiór decyzji dopuszczalnych $U_d(s)$ w stanie s definiowany jest zatem następująco:

$$U_d(s) = \left\{ u \in U_p(s): \quad s' = f(u, s) \notin S_N \right\} \tag{10}$$

Kolejno podjęte decyzje tworzą ciąg decyzyjny $\tilde{u} = (u_1, u_2, \ldots, u_c)$ gdzie c jest liczbą podjętych decyzji. Ciąg decyzyjny wyznacza jednoznacznie jedną z trajektorii systemu.

Na podstawie aktualnego stanu $s = (x, t)$ i decyzji u podjętej w tym stanie, generowany jest za pomocą funkcji przejścia f następny stan $s' = f(u, x, t)$. Funkcja przejścia jest zdefiniowana za pomocą dwóch funkcji $f = (f_x, f_t)$, gdzie f_x określa następny stan właściwy, a f_t określa następny moment czasu.

Wyznaczanie funkcji przejścia jest zadaniem kombinatorycznym. Należy rozważyć różne możliwe podzbiory stanów i przejścia pomiędzy nimi. Konieczne jest określenie, jakie stany możemy otrzymać w wyniku realizacji konkretnej decyzji, oraz w jaki sposób należy obliczać wartości poszczególnych składników następnego stanu.

W celu określenia momentu t' wystąpienia następnego stanu (czyli najbliższego momentu, w którym co najmniej jeden komiwojażer dotrze do wyznaczonego miasta), dla każdego komiwojażera obliczany jest czas t_k niezbędny na zrealizowanie podjętej dla niego decyzji. Najmniejszy z tych czasów wyznacza czas t':

$$t' = t + \Delta t, \quad \text{gdzie } \Delta t = \min_{k=1,2,\ldots,|M|} t_k \tag{11}$$

Po wyznaczeniu chwili $t' = t+\Delta t$ wystąpienia następnego stanu, obliczane są nowe wartości współrzędnych stanu właściwego.

Zbiór x^0, czyli pierwsza współrzędna stanu s', która reprezentuje zbiór odwiedzonych miast, zawiera dodatkowo te miasta, do których dojechał jeden z komiwojażerów w momencie t':

$$x^{0\prime} = x^0 \cup \left\{ n \in N : \exists_{k=1,2,\ldots,|M|} \left(u^k(s) = n \wedge t_k = \Delta t \right) \right\} \tag{12}$$

Stany poszczególnych komiwojażerów również są przeprowadzane do nowego stanu:

$$x^{k\prime} = f_x\left(u, x^k, t\right) \quad \text{dla } k = 1, 2, \ldots, |M| \tag{13}$$

Jeśli przyjmie się, że $x^k=(n, r)$ przechodzi w $x^{k\prime} = (n', r')$, to w zależności od realizowanej decyzji, wartości poszczególnych elementów stanu komiwojażera $x^{k\prime}$ w chwili $t' = t + \Delta t$ wynoszą:

1) Dla k-tego komiwojażera wolnego w danym stanie s, stojącego w mieście n_i, dla którego została podjęta decyzja $u^k(s) = n_j$ o odwiedzeniu miasta n_j (w tym o powrocie do miasta początkowego):

$$\begin{aligned} n' &= n_j \\ r' &= \begin{cases} 0 & \text{gdy } t_k = \Delta t \\ a_{ij} - \lambda & \text{gdy } t_k > \Delta t \end{cases} \end{aligned} \tag{14}$$

2) Dla k-tego komiwojażera wolnego w danym stanie s, stojącego w mieście n_i, dla którego została podjęta decyzja $u^k(s) = n_i$ o pozostaniu w tym mieście:

$$\begin{aligned} n' &= n_i \\ r' &= 0 \end{aligned} \tag{15}$$

3) Dla k-tego komiwojażera zajętego w danym stanie s, zmierzającego do wskazanego wcześniej miasta n_i, w tym do miasta początkowego:

$$\begin{aligned} n' &= n_i \\ r' &= \begin{cases} 0 & \text{gdy } t_k = \Delta t \\ r - \lambda & \text{gdy } t_k > \Delta t \end{cases} \end{aligned} \tag{16}$$

gdzie λ – odległość jaką pokonuje komiwojażer w czasie Δt.

4. Zaproponowany algorytm

NP-trudność problemu zaplanowania tras dostaw dla wielu komiwojażerów uzasadnia zastosowanie algorytmów przybliżonych do jego rozwiązania. Zaproponowany dla rozważanego problemu algorytm należy do klasy heurystycznych algorytmów konstrukcyjnych i opracowany został na podstawie metody zadań zastępczych [4].

W metodzie zadań zastępczych konstruowana jest w przestrzeni stanów pojedyncza trajektoria. W każdym rozważanym stanie s procesu decyzja wyznaczana jest na podstawie specjalnie skonstruowanego zadania optymalizacji, zwanego dalej zadaniem zastępczym $ZZ(s)$. W różnych stanach procesu zadanie to może mieć inną postać. Celem tworzenia takich zadań jest ułatwienie wyznaczenia decyzji w danym stanie poprzez zastąpienie optymalizacji zadania globalnego prostszym zadaniem lokalnym. Po wyznaczeniu decyzji generowany jest następny stan trajektorii s', dla którego ponownie przeprowadzana jest „automatyczna analiza" procesu, mająca na celu wygenerowanie najlepszego odcinka trajektorii ze stanu s' do zbioru stanów docelowych S_G. W wyniku tej analizy określane jest nowe, zmodyfikowane zadanie zastępcze. Tak więc w każdej iteracji obliczenia prowadzone są na dwóch poziomach:

1) poziom automatycznej analizy procesu i skonstruowania zadania zastępczego,
2) poziom wyznaczania możliwie dobrej decyzji dla zrealizowania zadania zastępczego i obliczania następnego stanu.

Aby można było określić zadanie zastępcze, stanowiące podstawę wyboru decyzji, konieczna jest analiza procesu w danym stanie. Na jej podstawie określane są, w sposób heurystyczny, tzw. cele pośrednie, które są następnie wykorzystywane do zdefiniowania zbioru stanów docelowych zadania zastępczego S_{Gz}. Jako *cel pośredni d* rozumiane jest jak najszybsze osiągnięcie przez proces stanu należącego do pewnego wyróżnionego zbioru stanów S_d.

Zaproponowany algorytm opiera się na metodzie zadań zastępczych i na jego potrzeby w rozważanym problemie określone zostały cele pośrednie dwojakiego rodzaju:

– osiągnięcie jak najszybciej takiego stanu, w którym odwiedzone jest miasto reprezentujące siedzibę jednej z firm lub firmę jednooddziałową,
– osiągnięcie jak najszybciej takiego stanu, w którym odwiedzone są wszystkie miasta reprezentujące oddziały jednej z firm.

Zatem na bazie każdego miasta n_i będącego elementem ciągu E określany jest cel pośredni $d_i^{[1]}$ pierwszego typu ($i \in \{1, 2, ... |E|\}$). Odpowiednim celem pośrednim jest wtedy osiągnięcie jak najszybciej stanu należącego do zbioru $S_{d_i^{[1]}}$:

$$S_{d_i^{[1]}} = \left\{ s = (x, t) : n_i \in x^0(s) \right\} \tag{17}$$

Natomiast na bazie każdego z podzbiorów F_i miast reprezentujących oddziały poszczególnych firm n_i, określany jest cel pośredni $d_i^{[2]}$ drugiego typu ($i \in \{1, 2, ... |E|\}$).

Odpowiednim celem pośrednim jest wtedy osiągnięcie jak najszybciej stanu należącego do zbioru zdefiniowanego następująco:

$$S_{d_i^{[2]}} = \left\{ s = (x, t) : \forall_{n \in F_i} \; n \in x^0(s) \right\} \tag{18}$$

W każdym stanie aktualizowany jest zbiór celów D. Po pierwsze, usuwane są z niego cele zrealizowane, tzn. jeżeli odwiedzone jest jakieś miasto reprezentujące siedzibę firmy, to usuwany jest cel z nim związany oraz jeżeli odwiedzone są już wszystkie oddziały jednej firmy, to usuwany jest cel z nimi związany. Po drugie, dodawane są do zbioru D cele $d_i^{[2]}$ drugiego typu, związane z oddziałami firm, których siedziby właśnie zostały odwiedzone.

Istotne jest, żeby do realizacji wybierane były cele pośrednie tak, aby jak najszybciej udostępnić wiele oddziałów, ale jednocześnie komiwojażerowie kierowali się do miast znajdujących się najbliżej. Priorytet p celu pośredniego d zależy więc wprost proporcjonalnie od liczby nieodwiedzonych jeszcze oddziałów danej firmy, a odwrotnie proporcjonalnie od odległości siedziby (dla celów pierwszego typu) lub najbliższego oddziału (dla celów drugiego typu) od aktualnego położenia komiwojażerów wolnych w danym stanie s. Wyznaczany jest na podstawie następującego wzoru:

$$p(d_i) = \frac{|F_i \setminus x^0(s)|}{a_{\min}} \tag{19}$$

gdzie $a_{\min}$ to minimalna odległość ze wszystkich odległości między miastami, w których znajdują się aktualnie komiwojażerowie $m \in M_W(s)$, a miastami związanymi bezpośrednio z celem $d \in D$.

W każdym kroku wybierana jest określona liczba celów z najwyższymi wartościami priorytetów, która tworzy zbiór celów pośrednich wybranych do realizacji $D_W(s)$. Cele w każdym stanie wybierane są na nowo na podstawie aktualnych priorytetów.

Aby wyznaczyć decyzję w danym stanie s, należy wyznaczyć wszystkie współrzędne tej decyzji $u = (u^1, u^2, \ldots, u^{|M|})$. Współrzędne $u^k(s)$ wyznaczane są po kolei dla poszczególnych komiwojażerów uwzględniając następujące przypadki:

– jeżeli k-ty komiwojażer jest zajęty (jedzie do i-tego miasta), to dla niego można podjąć jedynie decyzję o kontynuacji działania:

$$u^k = n_i \tag{20}$$

– jeżeli k-ty komiwojażer jest wolny (stoi w i-tym mieście), to przydzielamy mu do odwiedzenia miasto n_j – najbliższe ze wszystkich miast związanych z celami wybranymi aktualnie do realizacji $d \in D_W(s)$. Oczywiście musi być to miasto do tej pory nieodwiedzone ani nieprzydzielone wcześniej innemu komiwojażerowi.

$$u^k = n_j \tag{21}$$

– jeżeli nie ma już żadnych miast do odwiedzenia, to k-temu komiwojażerowi przydzielamy powrót do miasta początkowego (lub dalszy postój w mieście początkowym):

$$u^k = n_0 \tag{22}$$

Poniżej przedstawione zostaną dwie wersje tego algorytmu.

Algorytm 1

Krok 1. Ustalenie początkowego stanu procesu $s := s_0$

Krok 2. Ustalenie zbioru celów pośrednich D

Na bazie każdego z miast $n_i \in E$ reprezentujących siedziby firm i firmy jednooddziałowe wyznaczany jest jeden cel pośredni $d_i^{[1]}$. Liczba celów pośrednich jest zatem równa $|E|$.

Krok 3. Wyznaczenie priorytetów celów pośrednich

Dla każdego celu pośredniego d obliczany jest priorytet $p(d)$ według wzoru (19).

Krok 4. Wyznaczenie zbioru $D_W(s)$

W zbiorze $D_W(s)$ umieszczanych jest $|M_W(s)|$ celów pośrednich o najwyższych priorytetach. Jeżeli zbiór D jest mniej liczny niż zbiór $M_W(s)$, to w zbiorze $D_W(s)$ znajdzie się $|D|$ celów pośrednich.

Krok 5. Wyznaczanie decyzji dla danego stanu $u(s) = (u^1, u^2, ..., u^{|M|})$

Dla każdego komiwojażera wyznaczana jest decyzja $u^k(s)$ odpowiednio o kontynuacji działania lub realizująca cele pośrednie $d \in D_W(s)$.

Krok 6. Obliczenie następnego stanu s'

Na podstawie aktualnego stanu $s = (x, t)$ i wyznaczonej decyzji $u(s)$ obliczany jest następny stan procesu s' za pomocą funkcji przejścia: $s' = f(u, s)$.

Krok 7. Sprawdzenie warunków stopu

Jeżeli stan s' należy do stanów docelowych lub niedopuszczalnych, to generowanie trajektorii zostaje zakończone i wypisywany jest uzyskany wynik.
W przeciwnym przypadku należy przejść do kroku 8.

Krok 8. Aktualizacja zbioru celów pośrednich D

Po wyznaczeniu następnego stanu s' aktualizowany jest zbiór D celów pośrednich: usuwane są z niego cele zrealizowane, a dodawane są cele $d_i^{[2]}$ związane z oddziałami firm, których siedziby właśnie zostały odwiedzone.

Krok 9. Inicjowanie następnej iteracji algorytmu

Podstawiane jest: $s: = s'$
Następnie należy przejść do kroku 3, czyli do wyznaczania priorytetów.

Algorytm 2

Algorytm 2 jest modyfikacją *algorytmu 1* i różni się od niego tylko liczbą celów pośrednich wybieranych do zbioru $D_W(s)$, czyli krokiem 4. Krok ten jest następujący.

Krok 4. Wyznaczenie zbioru $D_W(s)$

W zbiorze $D_W(s)$ umieszczanych jest $|M|$ celów pośrednich o najwyższych priorytetach. Jeżeli zbiór D jest mniej liczny niż zbiór M, to w zbiorze $D_W(s)$ znajdzie się $|D|$ celów pośrednich.

5. Badania symulacyjne

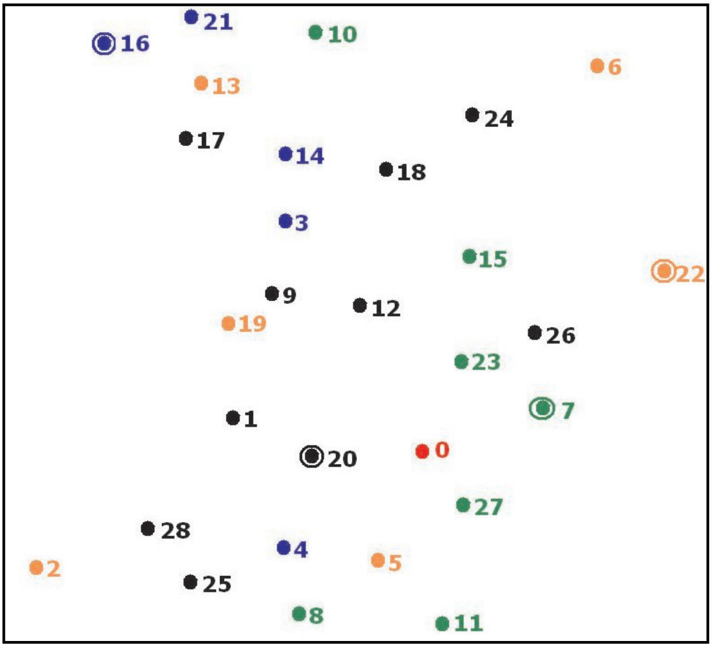
W ramach eksperymentów sprawdzono działanie dwóch wersji zaproponowanego algorytmu dla problemu planowania tras dostaw. Testy przeprowadzono dla przykładu zaczerpniętego z biblioteki TSPLIB95 [7]. Oryginalny przykład (*bayg29*) dotyczył instancji problemu komiwojażera dla 29 miast i został zmodyfikowany przez wyznaczenie miasta początkowego, a także ustalenie, które z miast reprezentują siedziby firm oraz oddziały poszczególnych firm. Przyjęto przy tym, że firmy posiadają różną liczbę oddziałów. Rozważano przypadek, w którym firma dostawcza zatrudnia dwóch komiwojażerów. Na rysunku 1 przedstawiono „mapę" miast. Miasto początkowe oznaczono kolorem czerwonym. Każdej firmie (tzn. siedzibie oraz ewentualnym oddziałom) przyporządkowano odrębny kolor. Dodatkowo kółkiem oznaczono miasta reprezentujące siedziby.

Przebadane zostały obie wersje zaproponowanego algorytmu, czyli wersja, w której do zbioru celów pośrednich wybranych do realizacji $D_W(s)$ wybierane tyle jest celów, ile jest wolnych komiwojażerów oraz wersja, w której do zbioru $D_W(s)$ wybierane są zawsze dwa cele (gdyż taka jest ilość wszystkich komiwojażerów w rozważanym przykładzie).

Ponadto dla porównania rozważana instancja problemu planowania tras dostaw została rozwiązana za pomocą zmodyfikowanego algorytmu „najbliższego sąsiada". Algorytm ten działał jak klasyczny algorytm najbliższego sąsiada, z tą różnicą, że nie wybierano ze wszystkich miast a jedynie z aktualnie dostępnych (czyli miast reprezentujących siedziby lub oddziały firm, których siedziby zostały już odwiedzone).

Dla obu wersji zaproponowanego algorytmu oraz dla algorytmu „najbliższego sąsiada" wyznaczono rozwiązanie, czyli czas w którym komiwojażerowie odwiedzili wszystkie miasta i powrócili do miasta początkowego. Dla algorytmu 1 otrzymano wynik 1183, a dla algorytmu 2 wynik wyniósł 1194. Rezultat uzyskany za pomocą algorytmu najbliższego sąsiada to 1430.

Należy też nadmienić, że rozwiązanie dokładne klasycznego problemu komiwojażera dla przyjętych danych wynosi 1610 [7]. Na podstawie tego wyniku można oszacować dolne ograniczenie dla rozważanego problemu, biorąc pod uwagę fakt, że jest dwóch komiwojażerów oraz że każdy z nich musi powrócić do miasta początkowego. Oszacowana w ten sposób wartość wynosi 841, przy czym nie uwzględniono istotnego ograniczenia, iż nie wszystkie miasta są od razu dostępne do odwiedzenia.



Rys. 1. Rozmieszczenie miast w analizowanej instancji problemu planowania tras dostaw

Na podstawie uzyskanych wyników można stwierdzić, że zaproponowane algorytmy dostarczyły dość dobre przybliżenie rozwiązania problemu. Otrzymane rezultaty są lepsze od wyniku uzyskanego zmodyfikowanym algorytmem najbliższego sąsiada (o 17,3% i o 16,5%), natomiast od dolnego ograniczenia problemu zrelaksowanego są gorsze o ok. 40%.

Zaprezentowane wyniki stanowią zaledwie początek prac nad zastosowaniem metody zadań zastępczych dla problemów planowania tras dostaw i innych problemów tej samej klasy. W dalszych pracach planowane jest przebadanie między innymi różnych sposobów wyznaczania priorytetów dla celów pośrednich.

6. Podsumowanie

W artykule przedstawiono problem planowania tras dostaw dla wielu komiwojażerów. Zaprezentowany został model tego problemu, a więc postać stanu systemu, postać decyzji, zbiór decyzji możliwych do podjęcia w poszczególnych stanach oraz zbiór decyzji dopuszczalnych. Przedstawione zostały elementy składające się na funkcję przejścia.

Zaproponowany został algorytm planowania tras dostaw oparty na metodzie zadań zastępczych. Metoda ta bazuje na ogólnym schemacie modelu algebraiczno-logicznego. Przedstawione zostały wyniki wstępnych badań dotyczących zastosowania algorytmów tego typu do rozwiązywania problem planowania tras dostaw.

Należy podkreślić, że przedstawiona metodologia jest bardzo uniwersalna i może znaleźć szerokie zastosowanie, przede wszystkim w optymalizacji trudnych problemów decyzyjnych.

Literatura

[1] Bubnicki Z., *Wstęp do systemów ekspertowych*. PWN, Warszawa, 1990.
[2] Dudek-Dyduch E., *Formalizacja i analiza problematyki dyskretnych procesów produkcyjnych*. Zesz. Nauk. AGH, s. Automatyka, z. 54, Kraków, 1990.
[3] Dudek-Dyduch E., *Learning based algorithm in scheduling.* Journal of Intelligent Manufacturing (JIM), vol. 11, No. 2, Kluwer Academic Publishers, 2000, 135–143.
[4] Dudek-Dyduch E., Dutkiewicz L., *Metoda zadań zastępczych do rozwiązywania NP-trudnych problemów szeregowania*. Zeszyty Naukowe/Politechnika Śląska; nr 1726, Automatyka, Wydawnictwo PŚ, Gliwice 2006, 57–66.
[5] Dudek-Dyduch E., Dyduch T., *Learning algorithms for scheduling using knowledge based model*. Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence; 4029, Springer-Verlag, Berlin, 2006, 1091–1100.
[6] Dutkiewicz L., Kucharska E., Kraszewska M., *Szeregowanie prac przygotowawczych w kopalni – algorytmy symulacyjne.* Gospodarka Surowcami Mineralnymi. t. 24, z. 3/3, Kraków, 2008, 79–93.
[7] www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.