

Jarosław Koźlak*, Małgorzata Żabińska*

Rozwiązywanie problemów transportowych za pomocą agentów**

1. Wprowadzenie

Optymalizacja transportu jest kwestią kluczową dla sprawnego działania firmy. W niektórych gałęziach przemysłu koszty transportu mogą sięgać nawet 70% wartości przewożonego asortymentu. Dlatego uzasadnione jest dążenie do minimalizacji kosztów związanych z transportem, co bezpośrednio przekłada się na końcową cenę towarów oraz usług. Ze względu na dużą złożoność obliczeniową algorytmów (problem NP-trudny) rozwiązujących problemy transportowe, stosowane są różne podejścia heurystyczne. Szeroko badane klasyczne problemy transportowe takie jak VRP (*Vehicle Routing Problem*) oraz jego różnorodne warianty (np. *Vehicle Routing Problem with Time Windows* – VRPTW, *Pickup and Delivery* – PDPTW, *Pickup and Delivery Problem with Time Windows* – PDPTW) operują na uproszczonym modelu świata, który nie uwzględnia istotnych problemów z jakimi borykają się firmy transportowe w swej codziennej działalności. Chodzi tu w szczególności o uwzględnienie heterogeniczności zespołów transportowych (ciągniki, naczepy, kierowcy). Ponieważ tak złożony zespół może być traktowany jako pojedynczy element podczas większości działań, przy jednoczesnej możliwości jego reorganizacji, usuwania i wymiany elementów składowych, użyteczne wydaje się użycie podejść opartych na holonach

Zadaniem systemu będzie rozwiązanie szeregu zadań począwszy od zadań dla statycznego problemu po dynamiczne zadania problemu rozszerzonego z zastosowaniem agentów holonicznych. Zespoły transportowe są traktowane jako złożone struktury, budowane przez elementy składowe takie jak ciągniki, naczepy, przyczepy, kierowcy itp. Przedstawiciele poszczególnych klas elementów składowych mogą różnić się między sobą cechami i atrybutami, np. mocą i szybkością ciągników, ładownością naczep i przyczep, typami ładunków jakie mogą one przewozić, kwalifikacjami kierowców, ich znajomością poszcze-

* Katedra Informatyki, Akademia Górniczo-Hutnicza w Krakowie

** Niniejsza praca jest częściowo finansowana przez Katedrę Informatyki AGH, umowa numer 11.11.120.865, a częściowo z grantu Ministerstwa Nauki i Szkolnictwa Wyższego numer N N516 366236.

gólnych tras i języków, doświadczeniem czy też ilością godzin, którą aktualnie nieprzerwanie pracował.

W szczególności, system powinien umożliwiać rozwiązywanie PDPTW przy lepszym odwzorowaniu warunków rzeczywistych. PDPTW rozszerzony został m.in. o złożoną sieć transportową, dynamikę zleceń transportowych oraz sytuacje kryzysowe – nieprzewidziane zdarzenia, których zaistnienie wymusza reorganizację tras przejazdu zespołów transportowych. Obiekty występujące w firmie logistycznej, takie jak dystrybutor zleceń, pojazdy transportowe, czy byt odpowiedzialny za obsługiwane sytuacje kryzysowych, w prosty i intuicyjny sposób mogą zostać zaimplementowane przy użyciu agentów.

Do rozwiązywania problemów transportowych stosowane są różne podejścia heurystyczne jak algorytmy ewolucyjne, poszukiwanie z tabu [7], *Squeaky Wheel* [8] czy też podejście agentowe. Rozwiązania agentowe często są oparte na protokole Contract Net [10], a do optymalizacji jest używane *simulated trading* [3] lub DCSP (*Distributed Constraint Satisfaction Problem*) [9]. Pierwszym agentowym systemem transportowym opartym na podejściu holonowym był Teletruck [2]. W przeciwieństwie do modeli używanych do rozwiązywania klasycznych problemów transportowych, odznaczał się on znacznie większą elastycznością i bogactwem uwzględnionych elementów – brał pod uwagę różne rodzaje ciężarówek, przyczep, kontenerów, kierowców, a także charakteryzujące je atrybuty. Podstawowe idee były oparte na rozwiązaniach istniejących we wcześniejszym systemie MARS [3].

2. Koncepcja systemu wieloagentowego

Tworzony system jest funkcjonalnością zbliżony do systemu TeleTruck. Porównano rozwiązania uzyskane jako wynik działania istniejących, klasycznych algorytmów z wynikami uzyskiwanymi dla problemu uwzględniającego stosowanie holonów, co umożliwia elastyczną konfigurację możliwości zespołów transportowych. Następnie przeprowadzono porównanie wyników uzyskanych przez taki system z klasycznymi rozwiązaniami problemów. Do analiz wybrano problem PDPTW oraz jego wprowadzone przez nas rozszerzenia, związane z holonami [5]. System umożliwia rozwiązywanie problemów statycznych i dynamicznych, może operować w środowisku opartym na odległościach euklidesowych i grafie reprezentującym połączenia drogowe między lokacjami. Ponadto, uwzględnia nieprzewidziane warunki, które występują w rzeczywistym środowisku działania firm transportowych i powodują różnego rodzaju sytuacje nietypowe i kryzysowe [6].

Opis agentów. W realizowanym systemie istnieją następujące główne typy agentów: Dispatcher Agent – DA (Dyspozytor), Driver Agent – DRA (Kierowca), Truck Agent – TKA (Pojazd), Trailer Agent – TRA (Przyczepa), Transportation Unit Agent – TUA (Jednostka Transportowa – Holon) oraz Crisis Manager Agent – CMA (Zarządca Kryzysów).

Dispatcher Agent DA pełni rolę dyspozytora zarządzającego całym zespołem pojazdów. Steruje pracą i działaniami TUA, jak również rozdziela zlecenia transportowe pomię-

dzy jednostkami transportowymi, jest reprezentowany przez n -kę:

$DA = (G_D, S_D, K_D, A_D)$, gdzie:

G_D – cel, którym jest minimalizacja funkcji

$$G_D = \sum_{i=1}^n C_v^i + \sum_{i=1}^n C_d^i TD^i,$$

gdzie n – liczba TUA,

C_v^i – koszt eksploatacji (głównie koszt amortyzacji i płaca kierowcy),

TUA^i, TD^i – dystans przemierzony przez TUA^i ,

C_d^i – średni koszt przemierzenia jednostki odległości przez TUA^i ,

S_D – stan,

$$S_D = \{LocAgent_j\},$$

$j = 1, \dots, n$, przybliżone położenie wszystkich agentów j ,

K_D – wiedza,

$$K_D = \{Reqs, StatReqs, Env\},$$

gdzie *Reqs* – informacje o zleceniach,

StatReqs – informacja o statusie zleceń, która przypisuje do każdego zlecenia wartości ze zbioru $\{przyjęte, odrzucone, przydzielone, załadowane, dostarczone\}$,

Env – informacja na temat sieci transportowej,

A_D – akcje do wykonania,

$$A_D = \{AllocReq, GetAgPos, SendAgPos\},$$

AllocReq – aukcja zleceń i ich przydział do Jednostki Transportowej,

GetAgPos – pobranie informacji o położeniu agenta,

SendAgPos – wysłanie informacji o położeniu agenta.

Dokładniejszy opis akcji, uwzględniający te elementy agenta, które mają na nie bezpośredni wpływ (lewa część wyrażenia, przed strzałką) i będące modyfikowane w wyniku akcji (prawa część wyrażenia, po strzałce) są następujące:

AllocReq: $Reqs \times StatReqs \rightarrow StatReqs$,

GetAgPos: $LocAgent_j \rightarrow LocAgent_j$,

SendAgPos: $LocAgent_j \rightarrow another_agent$ (akcja komunikacyjna, przesłanie informacji do innego agenta).

Truck Agent. Truck Agent i TKA^i reprezentuje główny element (pojazd) należący do Jednostki Transportowej, który może samodzielnie przemieszczać się razem z kierowcą ale bez możliwości przewożenia ładunku i jest reprezentowany przez n -kę:

$TKA^i = (S_{TKA}^i, K_{TKA}^i, A_{TKA}^i)$, gdzie

S_{TKA}^i – stan,

$$S_{TKA}^i = (TKA^i, Loc^i, Bel_{TUA}^i, Res_{TUA}^i),$$

gdzie TKA^i – cechy TKA (między innymi moc i możliwe rodzaje połączeń z przyczepą),
 Loc^i – położenie,
 Bel_{TUA}^i – przynależność do TUA,
 Res_{TUA}^i – rezerwacja na rzecz TUA i okres rezerwacji.

K_{TKA}^i – wiedza,
 $K_{TKA}^i = (Info_{DRA}^i, Info_{TRA}^i)$,
 zawierająca informacje o przebywających w pobliżu (w tej samej bazie) kierowcach ($Info_{DRA}^i$) i przyczepach ($Info_{TRA}^i$),

A_{TKA}^i – akcje,
 $A_{TKA}^i = (Join_{TUA}^i, Disj_{TUA}^i)$,
 gdzie $Join_{TUA}^i$ – przyłączenie się do TUA,
 $Disj_{TUA}^i$ – odłączenie się od TUA.

Trailer Agent. Trailer Agent i (TRA^i) reprezentuje wszelkiego rodzaju przyczepy i naczepy, które mogą być używane do przewozu ładunku i jest opisywany przez n-kę:

$TRA^i = (S_{TRA}^i, K_{TRA}^i, A_{TRA}^i)$, gdzie:

S_{TRA}^i – stan,
 $S_{TRA}^i = (TRA^i, Loc^i, Bel_{TUA}^i, Res_{TUA}^i)$,
 gdzie TRA^i – cechy TRA (między innymi rodzaj ładunku, ładunek bieżący i maksymalny, akceptowalne rodzaje połączenia z pojazdem),
 Loc^i – położenie,
 Bel_{TUA}^i – przynależność do TUA,
 Res_{TUA}^i – rezerwacja na rzecz TUA i okres rezerwacji.

K_{TRA}^i – wiedza,
 $K_{TRA}^i = (Info_{DRA}^i, Info_{TKA}^i)$, zawierająca informacje o przebywających w pobliżu (w tej samej bazie) kierowcach ($Info_{DRA}^i$) i pojazdach ($Info_{TKA}^i$).

A_{TRA}^i – akcje: $A_{TRA}^i = (Join_{TUA}^i, Disj_{TUA}^i)$,
 gdzie $Join_{TUA}^i$ – przyłączenie się do TUA,
 $Disj_{TUA}^i$ – odłączenie się od TUA.

Driver Agent. Driver Agent i (DRA^i) reprezentuje kierowcę pojazdu i jest opisywany przez następującą n-kę: $DRA^i = (S_{DRA}^i, K_{DRA}^i, A_{DRA}^i)$, gdzie:

S_{DRA}^i – stan,
 $S_{DRA}^i = (DRA^i, Loc^i, Bel_{TUA}^i, Res_{TUA}^i)$,
 gdzie DRA^i – cechy DRA (między innymi przejechane dotąd godziny pracy, licencje dotyczące prowadzonych różnych typów pojazdów oraz przewozu różnych ładunków, wiedza na temat trasy, znajomość obcych języków, preferencje dotyczące punktów pośrednich),
 Loc^i – położenie,

- Bel_{TUA}^i – przynależność do TUA,
 Res_{TUA}^i – rezerwacja na rzecz TUA i okres rezerwacji.
 K_{DRA}^i – wiedza,
 $K_{DRA}^i = (Info_{TKA}^i, Info_{TRA}^i)$, zawierająca informacje o przebywających w pobliżu (w tej samej bazie) pojazdach ($Info_{TKA}^i$) i przyczepach ($Info_{TRA}^i$),
 A_{DRA}^i – akcje,
 $A_{DRA}^i = (Join_{TUA}^i, Disj_{TUA}^i)$,
gdzie $Join_{TUA}^i$ – przyłączenie się do to TUA,
 $Disj_{TUA}^i$ – odłączenie się od TUA.

Transportation Unit Agent. Transportation Unit Agent i (TUA_i) reprezentuje zespół transportowy realizujący zlecenia transportowe. Razem z innymi jednostkami, uczestniczy w aukcjach zleceń transportowych a następnie wykonuje przydzielone zlecenia. Opisany jest on następująco: $TUA_i = (G_{TUA}^i, P_{TUA}^i, Loc_{TUA}^i, S_{TUA}^i, K_{TUA}^i, A_{TUA}^i)$, gdzie

- G_{TUA}^i – cel agenta, polegający na uzyskaniu maksymalnego możliwego przychodu
 $G_{TUA}^i = C_v^i + C_d^i TD_i^i$,
gdzie C_v^i – koszt eksploatacji (głównie koszt amortyzacji i wynagrodzenia kierowcy) TUA_i ,
 TD_i^i – dystans pokonany przez TUA_i ,
 C_d^i – średni koszt pokonania jednostki odległości przez TUA_i ,
 P_{TUA}^i – planowana trasa, obejmuje listę miejsc do odwiedzenia, wykonane akcje załadunku i rozładunku oraz reorganizację holona,
 Loc_{TUA}^i – bieżące położenie TUA,
 S_{TUA}^i – stan agenta,
 $S_{TUA}^i = (AIReqs^i, HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i)$,
gdzie $AIReqs^i$ – przydzielone zlecenia,
 $HolStruc^i$ – informacja o strukturze holonu i agentach składowych.,

oraz

- Res_{DRA}^i, Res_{TRA}^i i Res_{TKA}^i – kolejno zarezerwowani kierowcy, pojazdy i przyczep,
 K_{TUA}^i – wiedza agenta
 $K_{TUA}^i = (Reqs^i, Env^i)$ obejmująca $Reqs$ – zestaw zleceń,
 Env – informacje o otoczeniu (sieci transportowej),
 A_{TUA}^i – akcje TUA:
 $A_{TUA}^i = (ReqPart^i, Move^i, PickUp^i, Delv^i, HolCrt^i, HolDis^i, HolRorg^i)$,
gdzie:
 $ReqPart^i$ – uczestnictwo w aukcjach zleceń transportowych,
 $Move^i$ – przemieszczenie pomiędzy położeniami,
 $PickUp^i$ – załadunek,

$Delv^i$ – rozładunek,
 $HolCrt^i$ – tworzenie holonu, $HolDis^i$ – rozpad holonu,
 $HolRorg^i$ – reorganizacja holonu.

Dokładniejszy opis akcji jest następujący:

ReqPart: $(AIReqs^i, Env^i) \rightarrow (AIReqs^i, P_{TUA}^i)$,
 Move: $Loc_{TUA}^i \rightarrow Loc_{TUA}^i$,
 PickUp: $(AIReqs^i, Loc_{TUA}^i) \rightarrow AIReqs^i$,
 Delv: $(AIReqs^i, Loc_{TUA}^i) \rightarrow AIReqs^i$,
 HolCrt: $(HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i) \rightarrow (HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i)$,
 HolDis: $(HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i) \rightarrow (HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i)$,
 HolRorg: $(HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i) \rightarrow (HolStruc^i, Res_{DRA}^i, Res_{TRA}^i, Res_{TKA}^i)$.

Crisis Mager Agent (CMA) dedykowany jest do zarządzania powstającymi w systemie sytuacjami kryzysowymi [6]. Odpowiedzialny jest za detekcję sytuacji kryzysowych oraz ich obsługę. Detekcja sytuacji kryzysowej może odbyć się bezpośrednio przez CMA, lub pośrednio poprzez informację od innego agenta. Obsługiwanie sytuacji kryzysowych, polega na takiej współpracy z pozostałymi agentami w systemie, aby zminimalizować wszelkie negatywne skutki zaistnienia sytuacji.

Rozróżnia się trzy rodzaje sytuacji kryzysowych: dotyczące zleceń transportowych (np. wycofanie zlecenia), floty transportowej (np. awarie zespołu transportowego) oraz infrastruktury drogowej (np. korki lub wyłączenia dróg z ruchu). CMA jest opisywany jako następująca n-ka: $CMA = (K_{ACM}, R_{ACM}, A_{ACM})$, gdzie:

K_{ACM} – wiedza o zachodzących sytuacjach kryzysowych,
 R_{ACM} – reguły decyzyjne opisujące akcje do podjęcia, dla danej sytuacji kryzysowej i sytuacji zachodzących podczas jej obsługi,
 A_{ACM} – akcje wykonywane przez agenta,
 $A_{ACM} = (SendInfo, RecStat, DecReallocate)$,
 SendInfo – rozsyłanie informacji o zajściu sytuacji kryzysowej,
 ResStat – odbiór informacji o statusie sytuacji kryzysowych,
 DecReallocate – propozycja realokacji zleceń transportowych.

Algorytm optymalizacyjny. Przydzielenie zadań agentom jest dokonywane używając *Contract Net Protocol* [10]. Jednostki transportowe (TUA) (skompletowane lub nie) ogłaszają przybliżone koszty realizacji zleceń. Te oszacowania mogą się opierać na właściwościach jednostek transportowych (jej parametrach, położeniu oraz zbiorze już zaakceptowanych zleceń) lub szacowanych możliwościach tworzenia przez nie holonów. W taki sposób

przeprowadzane są aukcje oferowanych komponentów, z których można złożyć holon. Zlecenia są przydzielane tym agentom, którzy mogą je zrealizować z najniższym możliwym przyrostem kosztów. Następnie przeprowadzana jest optymalizacja rozwiązania, co zadany przedział czasu, każda jednostka transportowa TUA próbuje pozbyć się zleceń, które są najmniej korzystne, biorąc pod uwagę kryterium kosztowe, i wysłać je na aukcje. W ten sposób zlecenia mogą być przekazane innym jednostkom transportowym. Holon może także samodzielnie dokonywać swojej reorganizacji, w celu dobrania lepszej konfiguracji, aby spełnić przydzielone do realizacji zlecenia przy niższych kosztach.

3. Zastosowanie holonów

Współpraca pomiędzy agentami polega na zestawianiu zespołów transportowych reprezentowanych przez holony potrzebnych do wykonania zleceń, a także minimalizacji kosztów wykonania danego zestawu zleceń.

Zastosowany przez nas model zakłada, że agenci porzucają jedynie część swojej niezależności. Cały holon jest reprezentowany przez agenta TUA, który komunikuje się z całą społecznością. Zarządca ma prawo przydzielania zasobów, planowania, a także negocjowania z innymi agentami na podstawie planów i celów agentów składowych. Jednocześnie zadaniem TUA jest wymiana elementów holonu lub też ich odłączanie i przyłączanie.

Holony są tworzone w momencie nadejścia nowego zlecenia, o ile umożliwia to mniej kosztowne jego obsłużenie. Ponadto, można przyjąć, że pewne holony istnieją już na początku symulacji. Stworzona jednostka TUA nie posiada na początku żadnych elementów holonu, przez co holon musi być stworzony od podstaw. Do wszystkich typów elementów rozsyłane jest zadanie, po czym jednostka oczekuje na odpowiedzi. Po ich otrzymaniu wybierane są najlepsze oferty, następuje obliczenie kosztu wykonania zadania i przesłanie oferty do *Dispatchera*. W przypadku otrzymania od niego pozytywnej odpowiedzi, powiadamiane są elementy i tworzony jest holon. Pozostałe elementy otrzymują odpowiedź odmowną. Taką odpowiedź otrzymują wszystkie elementy, jeśli dana jednostka wykonawcza nie zostanie wybrana do danego zadania. W tym przypadku jednostka ta zakończy swoje istnienie. Natomiast jeśli holon zostanie utworzony, jego elementy tracą część swojej autonomii. Nie będą mogły odbierać informacji o nowych zleceniach od jednostek wykonawczych innych niż jednostka będąca zarządcą ich holonu, nie będą mogły również wysyłać odpowiedzi na takie zlecenia.

Reorganizacja holonu może nastąpić w przypadku nadejścia nowego zlecenia, którego dany zespół nie jest w stanie zrealizować. Gdy jednostka wykonawcza otrzymuje zlecenie, sprawdzana jest konfiguracja holonu. Jeśli konfiguracja jest nieodpowiednia do danego zadania, jednostka wykonawcza podejmie próbę jej rekonfiguracji. Rozsyłane są zgłoszenia zlecenia do podelementów, które mają być wymienione. Następnie odpowiadają one swoimi ofertami zawierającymi koszt przyłączenia ich do danego holonu. Jeden element może

odpowiedzieć kilku jednostkom oferujących mu to samo zadanie, gdyż tylko jedna z nich je otrzyma. Nie może natomiast odpowiadać na inne zlecenia, które pojawiły się w tym samym czasie, ponieważ mogłoby to prowadzić do konfliktów. Z nadesłanych ofert jednostka wykonawcza wybiera tę najlepszą, oblicza koszt wykonania zadania i wysyła swoją ofertę na jego wykonanie. Po otrzymaniu odpowiedzi powiadamia kandydatów na elementy holonu, czy zostali do niego wybrani.

Rozpad holonu następuje w momencie, gdy zespół transportowy wykona wszystkie powierzone mu zadania. Zasoby zostają uwolnione i mogą przyłączyć się do innych holonów, natomiast agent TUA oczekuje na kolejne zlecenia. Jeśli się pojawiają, będzie próbowała zebrać kolejny zespół transportowy i rozpocząć ich realizację.

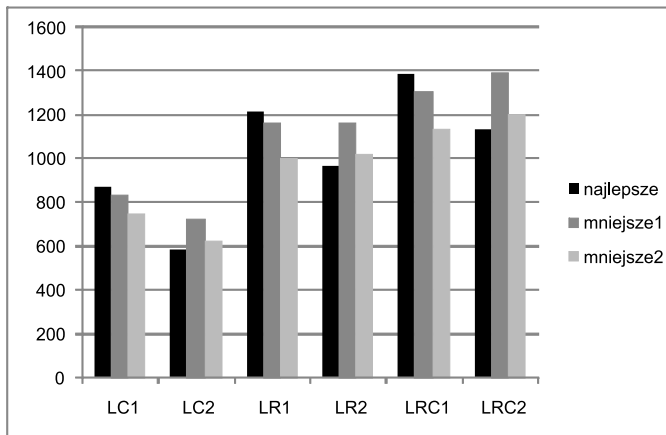
4. Realizacja

System podzielono na kilka modułów. Są to: moduł obliczeniowy, warstwa sterująca symulatorem z interfejsem użytkownika, wizualizatorem i generator testów. W szczególności, moduł obliczeniowy obejmuje agentów w systemie odpowiedzialnych za znalezienie rozwiązania. Generator testów pozwala na zamianę testów statycznych na dynamiczne, zwiększanie liczby dostępnych baz transportowych, zmianę sieci transportowej oraz wprowadzanie sytuacji kryzysowych. Agenci wykorzystują platformę JADE (*Java Agent Development Framework*, wersja 3.3) [4] i komunikują się ze sobą podejmując akcje w celu rozwiązania zadanego problemu. Realizacja zadań agentów w systemie związana jest z komunikacją pomiędzy nimi. Można wymienić następujące główne zadania agentów:

- przydzielanie nowego zlecenia, gdy istnieje już zespół (holon) ze zleceniami;
- przydzielanie nowego zlecenia, gdy istnieje zespół (holon), ale bez zleceń;
- przydzielanie nowego zlecenia, gdy nie istnieje jeszcze zespół (holon) i trzeba go utworzyć – dokonane jest stworzenie holonu;
- brak nowego zlecenia dla zespołu (po zakończeniu zadań), zwolnienie zespołu – w rezultacie następuje rozpad holonu.

5. Badania

Badania zostały przeprowadzone na bazie zbiorów testowych przygotowanych przez [7, 1], przy czym zestawy modyfikowano w celu uwzględnienia podejścia holonicznego oraz sytuacji kryzysowych. Testowano problemy LC (punkty zleceń rozmieszczone w klustrach), LR (punkty zleceń rozrzucone równomiernie) oraz LRC (punkty zleceń rozmieszczone w klustrach i równomiernie), z wąskimi (1) oraz szerokimi oknami czasowymi (2) (rys. 1).



Rys. 1. Zbiorczy oszacowany koszt rozwiązań – najlepsze – koszt najlepszych znanych rozwiązań przy przyjętych w testach pojemnościach, mniejsze1 i mniejsze2 – koszty rozwiązań przy pojazdach o coraz mniejszych pojazdach do dyspozycji

Uzyskane wyniki wykazały, że koszty przejazdu dla wersji benchmarks (czyli pojemności zgodne z tymi, które użyto do uzyskania najlepszych znanych rozwiązań podanych w *najlepsze*) zazwyczaj odbiegają jedynie w akceptowalnym naszym zdaniem stopniu od najlepszych znanych rozwiązań. Natomiast używając mniejszych pojazdów, co ogranicza ilość jednocześnie obsługiwanych zleceń, ale jednocześnie obniża koszt pokonania jednostki dystansu, w problemach *mniejsze1* (pojazdy o pojemnościach 100%, 75% i 50% pojemności domyślnych) oraz *mniejsze2* (pojazdy o pojemnościach 50%, 37,5% i 24% pojemności domyślnych) udało się znacznie obniżyć koszty, co jest dowodem na to, że w rozwiązaniach problemów testowych zazwyczaj nie wykorzystywano pełnych pojemności pojazdów. Pokazuje to także opłacalność elastycznego planowania, gdy mamy do dyspozycji przyczepy o różnych pojemnościach, gdyż może to w znacznym stopniu ograniczyć koszty.

6. Podsumowanie

Przedstawiono koncepcję wieloagentowego systemu bazującego na wykorzystaniu holonów. Pokazano także wyniki uzyskane przy użyciu pilotowej wersji naszego holonowego systemu do modelowania i optymalizacji przewozów transportowych. Dalsze prace będą zmierzać w kierunku wprowadzenia nowych algorytmów konstrukcji tras oraz zwiększenia elastyczności algorytmów zbudowanych z holonów. Kolejną dziedziną prac będzie mocniejsze uwzględnienie różnego rodzaju sytuacji kryzysowych, a w szczególności powstawania korków, zamykania tras oraz awarii pojazdów lub innych elementów holonu. Aspekty te były badane w innych naszych pracach [6], teraz zostaną one zintegrowane z podejściem holonistycznym [5].

Literatura

- [1] *Benchmarks-Vehicle Routing and Travelling Salesperson Problems*. <http://www.sintef.no/static/am/opti/projects/top/>.
- [2] Burckert H.-J., Fischer K., Vierke G., *Transportation scheduling with holonic MAS – the TELETRUCK approach*. [w:] Third International Conference on Practical Applications of Intelligent Agents and Multiagents, PAAM 98, 1998.
- [3] Fischer K., Muller J., Pischel M., *Cooperative Transportation Scheduling: an Application Domain for DAI*. Applied Artificial Intelligence, 1996, 1–33.
- [4] *Java Agent DEvelopment Framework*. <http://jade.tilab.com/>.
- [5] Gołacki M., Koźlak J., Żabińska M., *Holonic-Based Environment for Solving Transportation Problem*. [w:] Proceedings of HOLOMAS'2009, accepted for publication. LNCS, Springer-Verlag, 2009.
- [6] Konieczny M., Koźlak J., Żabińska M., *Multi-agent crisis management in transport domain*. [w:] Proceedings of ICCS'2009, LNCS, Springer-Verlag, 2009.
- [7] Li H., Lim A., *A Metaheuristic for the Pickup and Delivery Problem with Time Windows*. [w:] Proceedings of 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), Dallas, USA, 2001.
- [8] Lim H., Lim A., Rodrigues B., *Solving the Pick up and Delivery Problem using “Squeaky Wheel” Optimization with Local Search*. [w:] Proceedings of American Conference on Information Systems, AMCIS 2002, USA, 2002.
- [9] Neagu N., Dorer K., Calisti M., *Solving Distributed Delivery Problems with Agent-Based Technologies and Constraint Satisfaction Techniques*. [w:] Dist. Plan and Schedule Management, 2006 AAAI Spring Symp., USA, 2006. The AAAI Press.
- [10] Smith R., *The contract net protocol: high-level communication and control in a distributed problem solver*. IEEE Transactions on Computer, December 1980, 1104–1113.