

UNIWERSYTET PRZYRODNICZY W LUBLINIE
WYDZIAŁ INŻYNIERII PRODUKCJI

Wojciech Misztal

**Optymalizacja transportu surowców dla przetwórstwa
rolno-spożywczego**

Autoreferat rozprawy doktorskiej

Promotor

Prof. dr hab. Andrzej Marczuk

Uniwersytet Przyrodniczy w Lublinie

Recenzenci:

Prof. dr hab. Józef Kowalski

Uniwersytet Rolniczy w Krakowie

Prof. dr hab. Wiesław Piekarski

Uniwersytet Przyrodniczy w Lublinie

LUBLIN 2018

Spis treści

1. Wstęp.....	3
2. Układ rozprawy.....	4
3. Cel pracy i koncepcja rozwiązania problemu.....	5
4. ILS-RVND.....	7
5. Analiza struktur sąsiedztwa.....	14
Literatura.....	20

1. Wstęp

Przetwórstwo rolno-spożywcze odgrywa kluczową rolę w zaspokajaniu potrzeb żywnościowych ludności, w znacznym stopniu wpływa również na rozwój gospodarczy kraju, bądź regionu.

Surowce rolnicze stanowią specyficzne i zarazem bardzo wymagające zasoby, odznaczające się z reguły niską trwałością, będącą przyczyną ograniczonej podatności transportowej i magazynowej. Duży wpływ na jakość tych zasobów wywierają wobec tego systemy ich odbioru i transportu, determinujące możliwość zapewnienia odpowiednich warunków, a także wpływające na wielkość ponoszonych kosztów [Baran 2012; Baryła-Paśnik i inni 2014; Klepacki i Rokicki 2011; Rudziński 2010].

Procesy transportowe ukierunkowane na przemieszczanie surowców rolniczych charakteryzują się: dużą liczbą z reguły znacznie oddalonych od siebie punktów odbioru, wysoką zmiennością parametrów, występującą często koniecznością stosowania pojazdów specjalistycznych i/lub odpowiednich pojemników transportowych, dużą częstotliwością odbioru ładunków, różnorodnością warunków pracy wykorzystywanych pojazdów (będącą następstwem zmienności stanu pogody oraz rodzajów nawierzchni), stosunkowo niskim wykorzystaniem środków transportu (związanym z jednokierunkowością przewozów oraz niewielkim udziałem czasu przewozu w ogólnym czasie transportu), a także stanowiącą niemal nieodłączny element wszystkich procesów ukierunkowanych na przemieszczanie wysoką kapitałochłonnością.

Ważną cechą jest również to, że przypadku pewnych surowców (jak np. mleko) produkcja w gospodarstwach ma charakter ciągły, a co za tym idzie istnieje potrzeba nieprzerwanego ich odbioru. Dodatkowo istotny czynnik stanowi zmienność wielkości podaży, która w znacznym stopniu determinuje złożoność organizacji tego rodzaju procesów transportowych [Baran 2012; Baryła-Paśnik i inni 2014; Gaworski, Makowska 2013].

W przypadku przetwórstwa rolno-spożywczego duże możliwości wydaje się nieść za sobą optymalizacja działań transportowych wykonywanych w ramach pozyskiwania surowców. Koszty z tym związane stanowią bowiem znaczny procent ich wartości [Álvarez, Munari 2016; Caceres-Cruz i inni 2014; Baran 2012;

Baryła-Paśnik i inni 2014; Jafari-Eskandari, Aliahmadi i Khaleghi 2010; Jakubowska, Pichocka 2015; Sadjadi, Jafari i Amini 2009; Takes, Kusters 2010].

Pomimo istotnych korzyści, jakie niesie za sobą optymalizacja prac transportowych znaczna część przedsiębiorstw nie stosuje żadnych tego rodzaju rozwiązań. Wiele spośród podmiotów, które podejmują pewne działania w zakresie usprawniania transportu wykorzystuje jedynie proste narzędzia geoinformacyjne. Kompleksowe systemy optymalizacji transportu stanowią rozwiązania, z których korzysta się znacznie rzadziej. Ponadto z reguły są to programy bazujące na starszych wersjach algorytmów.

Wobec powyższego można stwierdzić, że istnieje zapotrzebowanie na komputerowe systemy optymalizacji transportu surowców odbieranych z gospodarstw rolniczych. Odznaczających się wysoką jakością zwracanych rozwiązań oraz odpowiednią szybkością działania.

2. Układ rozprawy

Rozprawa składa się ze wstępu oraz zakończenia, a także czterech rozdziałów, do których zaliczają się:

Rozdział pt. „Przegląd literatury” zawierający wprowadzenie do tematyki zagadnień optymalizacyjnych. W pierwszej części tego rozdziału przedstawione zostały sformułowania, modele matematyczne (przepływu towarów, podziału zbioru oraz przepływu pojazdów i ich rozszerzenia) oraz charakterystyki *Problemów komiwojażera i wielu komiwojażerów*, a także *Problemu komiwojażera o ograniczeniach zasobowych*. Druga część rozdziału poświęcona została natomiast *Problemowi wyznaczania tras przejazdów (Vehicle Routing Problem)*. W części tej zawarte zostały sformułowania, modele matematyczne oraz charakterystyki komponentów zarówno wersji podstawowej, jak również i wariantów tego zagadnienia (*Capacitated Vehicle Routing Problem, Distance-Constrained Vehicle Routing Problem, Vehicle Routing Problem with Time Windows, Vehicle Routing Problem with Backhauls* oraz *Vehicle Routing Problem with Pickup and Delivery*). Podrozdział poświęcony został również dokładnym i heurystycznym metodom rozwiązywania, a także warunkom ich oceny.

Rozdział "Postępowanie *ILS-RVND*" zawiera dokładny opis rozpatrywanej heurystyki *ILS-RVND*, uwzględniający każdy z jej elementów składowych oraz modyfikacje wprowadzone przez autora rozprawy. W podrozdziale "Struktury sąsiedztwa" zamieszczone zostały opisy, schematy oraz pseudokody rozpatrywanych struktur sąsiedztwa *intra-* oraz *inter-route*. Należy dodać, że struktury te stanowią algorytmy opracowane przez autora wyłącznie na bazie idei wykorzystywanych przez nie operatorów, tj. sposobu wymiany, zamiany lub przeniesienia atrybutów tras (wierzchołków, bądź krawędzi). Działanie tych algorytmów oparte zostało na *Strategii Stromeego Zejścia*.

W rozdziale "Analiza struktur sąsiedztwa" zamieszczone zostały wyniki analizy porównawczej rozpatrywanych struktur sąsiedztwa *intra-* oraz *inter-route*.

Rozdział "Podsumowanie i wnioski" zawiera natomiast informacje opisujące problem, na przykładzie którego przeprowadzona została weryfikacja, a także otrzymane w wyniku tej weryfikacji wyniki oraz wyciągnięte na ich podstawie wnioski.

3. Cel pracy i koncepcja rozwiązania problemu

Niniejsze opracowanie skupia się na tematyce związanej z organizacją pracy środków transportu przemieszczających surowce rolnicze na trasach łączących gospodarstwa z zakładem przetwórczym. Celem rozprawy jest opracowanie i oprogramowanie komputerowego systemu optymalizacji procesów przemieszczania tego rodzaju przedmiotów transportu. Rozpatrując zagadnienie w ujęciu kompleksowym tj. rozwiązującego dwa wzajemnie uwarunkowane problemy decyzyjne ukierunkowane na wyznaczenie kompletnych tras, minimalizujących wartość założonej funkcji celu, w postaci sumy reprezentujących odległości wag krawędzi tworzących dane rozwiązanie, a co za tym idzie również i całkowitego kosztu związanego z realizacją rozpatrywanego procesu transportowego. Pierwszy z tych problemów dotyczy dokonania (w możliwie najbardziej odpowiedni sposób) podziału wszystkich gospodarstw na grupy, z których każda obsługiwana będzie przez inny pojazd, a drugi ustalenia optymalnej kolejności, w jakiej gospodarstwa przypisane do poszczególnych grup będą odwiedzane.

Osiągnięcie założonego celu głównego wymagało zrealizowania celów pośrednich, w postaci: identyfikacji problemu decyzyjnego; doboru metody optymalizacyjnej, a także przeprowadzenia jej modyfikacji; opracowania struktury systemu komputerowego, pisania kodu wykonywalnego oraz weryfikacji poprawności funkcjonowania systemu.

Problem optymalizacyjny związany z przemieszczaniem surowców rolniczych wpisuje się w specyfikę zagadnienia znanego, jako *Heterogeneous Vehicle Routing Problem (HVRP)* stanowiącego istotną, w większym stopniu odpowiadającą specyfice rzeczywistych problemów, modyfikację zagadnienia *Capacitated Vehicle Routing Problem (CVRP)*, uwzględniającą zróżnicowaną flotę pojazdów.

Zagadnienie *Heterogeneous Vehicle Routing Problem* może zostać sformułowane w następujący sposób. Dany jest graf skierowany $G = (V, A)$, w którym $V = \{0, 1, \dots, n\}$ stanowi zbiór $n+1$ wierzchołków, natomiast $A = \{(i, j) : i, j \in V, i \neq j\}$ zbiór występujących pomiędzy nimi łuków. Wierzchołek oznaczony symbolem 0 opisuje tzw. bazę, czyli miejsce, w którym znajduje się flota dostępnych zróżnicowanych pojazdów. Pozostałe wierzchołki, tworzące zbiór $V' = V \setminus \{0\}$ reprezentują n klientów, z których każdy $i \in V'$ wyraża nieujemny popyt q_i . Flota stanowi zbiór m różnych typów środków transportu, $M = \{1, \dots, m\}$ i zawiera m_u pojazdów każdego typu $u \in M$, charakteryzujący się pojemnością Q_u , d_{ij} stanowi odległość (wagę gałęzi) występującą pomiędzy wierzchołkami (i, j) . Trasa opisywana jest jako (R, u) , gdzie $R = (i_1, i_2, \dots, i_{|R|})$, $i_1 = i_{|R|} = 0$ oraz $\{i_2, \dots, i_{|R|-1}\} \subseteq V'$, co oznacza, że każda trasa stanowi powiązany z pojazdem $u \in M$ cykl G , którego zarówno początek, jak również i koniec znajduje się w bazie. Trasę tego rodzaju uznaje się za wykonalną w przypadku, w którym suma popytów obsługiwanych klientów, nie jest wyższa niż pojemność przypisanego do niej pojazdu $u \in M$. Koszt pokonania tej trasy przez dany środek transportowy obliczany jest jako suma kosztów związanych z przemierzeniem przez niego łuków, z których ta trasa została utworzona. Wobec tego *HVRP* stanowi problem, którego rozwiązywanie przybiera postać wyszukiwania wykonalnych tras, takich że: każdy klient odwiedzany jest wyłącznie raz, liczba tras wyznaczonych dla pojazdów typu $u \in M$ nie jest większa od dostępnej

ilości pojazdów tego typu, natomiast całkowity koszt związany z pokonaniem wszystkich tras jest możliwie najmniejszy [Gendreau i inni 1999; Gragas i inni 2013; Kachitvichyanukul, Sombuntham i Kunnapapdeelert 2012; Penna, Subramanian, Ochi 2013; Sadouni 2006; Wassan, Osman 2002].

Problem ten wymaga znalezienia efektywnego sposobu rozwiązywania, zdolnego do zwracania dobrych jakościowo rozwiązań. Dokładność wpływa bowiem bezpośrednio na wielkość osiągniętych oszczędności, a co za tym idzie na zapewnianie konkurencyjności prowadzonej działalności. Zagadnienie to doczekało się wielu metod rozwiązywania, opartych na różnorodnych podejściach, odznaczających się zróżnicowaną jakością zwracanych rozwiązań i czasem działania, a także różnym stopniem przydatności przy rozwiązywaniu rzeczywistych problemów [Baldacci, Battarra i Vigo 2008; Caceres-Cruz, Gragas, Romalinho i Juan 2014; Gendreau i inni 1999; Sadouni 2006; Wassan, Osman 2002]. Wśród nich na szczególną uwagę zasługuje będące rozszerzeniem metody zaprojektowanej przez Subramaniana i innych [2010] dla *VRP with Simultaneous Pickup and Delivery* podejście *ILS-RVND*.

4. ILS-RVND

ILS-RVND stanowi hybrydową heurystykę, której podstawą jest odznaczająca się cechami takimi, jak prostota, krzepkość, efektywność oraz łatwość implementacji meta-heurystyka *Iterated Local Search (ILS)*, wykorzystująca podczas przeszukiwania lokalnego procedurę *Variable Neighborhood Descent (VND)*, systematyzującą sąsiedztwo w sposób losowy (*RVND*) [Gragas i inni 2013; Labadie, Prins i Prodhon 2016; Penna, Subramanian i Ochi 2013; Sadouni 2006; Subramanian i inni 2010; Subramanian i inni 2011].

Iterated Local Search (ILS) stanowi meta-heurystykę, odznaczającą się wysoką efektywnością przy rozwiązywaniu problemów z rodziny *VRP*, której ideą jest odejście od przeprowadzania analizy całkowitej przestrzeni rozwiązań, skupiając się przy tym jedynie na ograniczonym podzbiorze, określonym przez lokalne optimum wybranego algorytmu optymalizacyjnego [Lourenço, Martin i Stützle 2003]. Zasada działania tej meta-heurystyki jest następująca. Po znalezieniu rozwiązania

lokalnie optymalnego postępowanie nie powtarza tych samych instrukcji dla zupełnie nowego rozwiązania, tylko wywołuje wielokrotnie procedurę lokalnego przeszukiwania dla zbioru złożonego z rozwiązań uzyskiwanych w procesie zaburzenia uprzednio odwiedzonych lokalnych optimum [Álvarez, Munari 2016; Labadie, Prins i Prodhon 2016].

Struktura omawianej meta-heurystyki przedstawiona została w formie pseudokodu w *Algorytmie 1*.

Algorytm 1. Meta-heurystyka *Iterated Local Search*

1. **Procedure** *ILS*():
2. $s_0 \leftarrow \text{GenerateInitialSolution}$
3. $s^* \leftarrow \text{LocalSearch}(s_0)$
4. **while** nie zostanie napotkane kryterium zatrzymania:
5. $s' \leftarrow \text{Perturb}(s^*)$
6. $s^{*'} \leftarrow \text{LocalSearch}(s')$
7. $s^* \leftarrow \text{AcceptanceCriterion}(s^*, s^{*'})$
8. **return** s^*

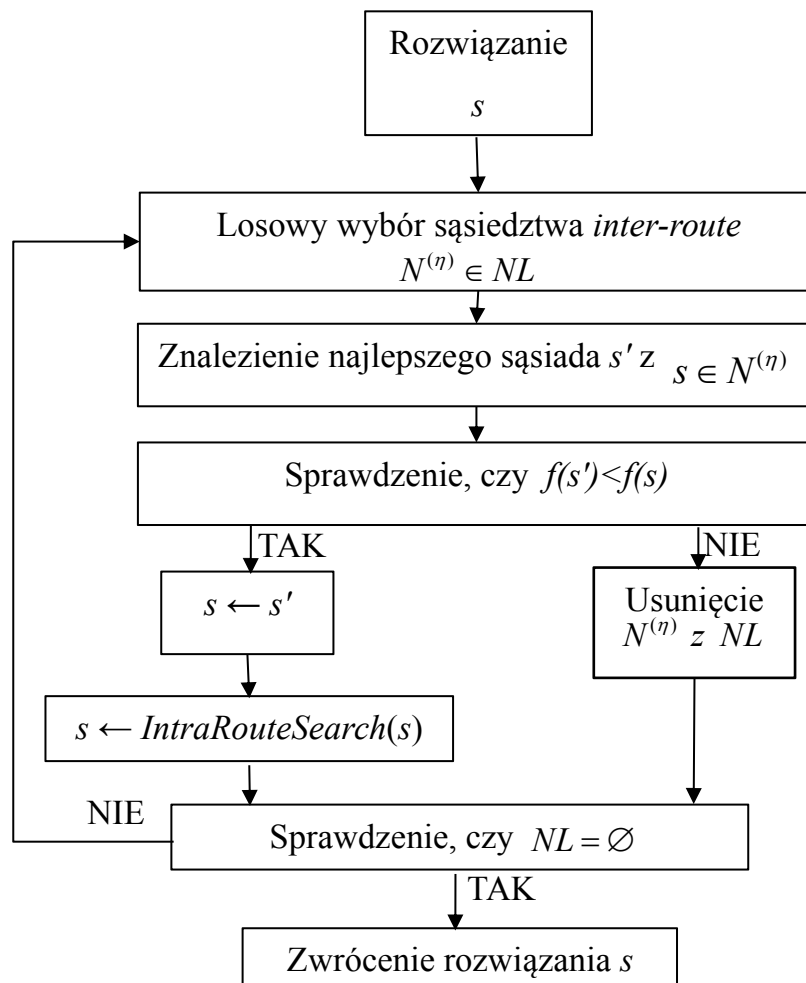
Podejście *Variable Neighborhood Descent (VND)* stanowi składnik opracowanej przez Mladenovicia i Hansena [1997] meta-heurystyki *Variable Neighborhood Search (VNS)*, której idea opiera się na dokonywaniu zmian sąsiedztwa w celu osiągnięcia lokalnych minimum oraz ucieczki z dolin, w których się one znajdują. *VNS* zostało opracowane na bazie pewnych wniosków sformułowanych na podstawie wyników badań doświadczalnych, przyjmujących następującą postać:

1. globalne minimum stanowi lokalne minimum wobec wszystkich możliwych struktur sąsiedztwa;
2. w przypadku wielu problemów lokalne minima jednego lub większej ilości sąsiedztw nie są zbyt oddalone od siebie nawzajem.

Na podstawie punktu drugiego można wnioskować, że pewne informacje odnoszące się do optimum globalnego zawarte są w optimum lokalnym. Przykładowo zarówno jedno, jaki i drugie z nich może zawierać pewne zmienne o takich samych wartościach. W rzeczywistości jednak nie istnieje sposób, który pozwoliłby na ich wskazanie przez co za właściwe uznaje się rozpatrywanie sąsiedztw zidentyfikowanego lokalnego

optimum w celu znalezienia rozwiązania bardziej korzystnego [Mladenović, Hansen 1997].

Procedura *VND* zaimplementowana w heurystyce *ILS-RVND* przybiera następującą postać. Wykorzystywany jest zbiór $N = \{N^{(1)}, \dots, N^{(r)}\}$ sąsiedztw *inter-route*, z których w pierwszej kolejności wybierane jest na drodze losowej jedno sąsiedztwo $N^{(n)}$, które następnie wykorzystywane jest w celu podjęcia próby poprawy bieżącego rozwiązania. W przypadku, w którym sąsiedztwo to nie pozwoli na uzyskanie rozwiązania lepszego od rozwiązania bieżącego jest usuwane, po czym algorytm ponownie dokonuje losowego wyboru kolejnego sąsiedztwa $N^{(n)} \in N$, dla którego powtarza te same instrukcje [Mladenović, Hansen 1997; Penna, Subramanian, Ochi 2013; Subramanian i inni 2011]. Procedura *RVND* przedstawiona została w postaci pseudokodu w *Algorytmie 2* oraz schematu na *Rys.1*.



Rys. 1. Schemat działania algorytmu *RVND*

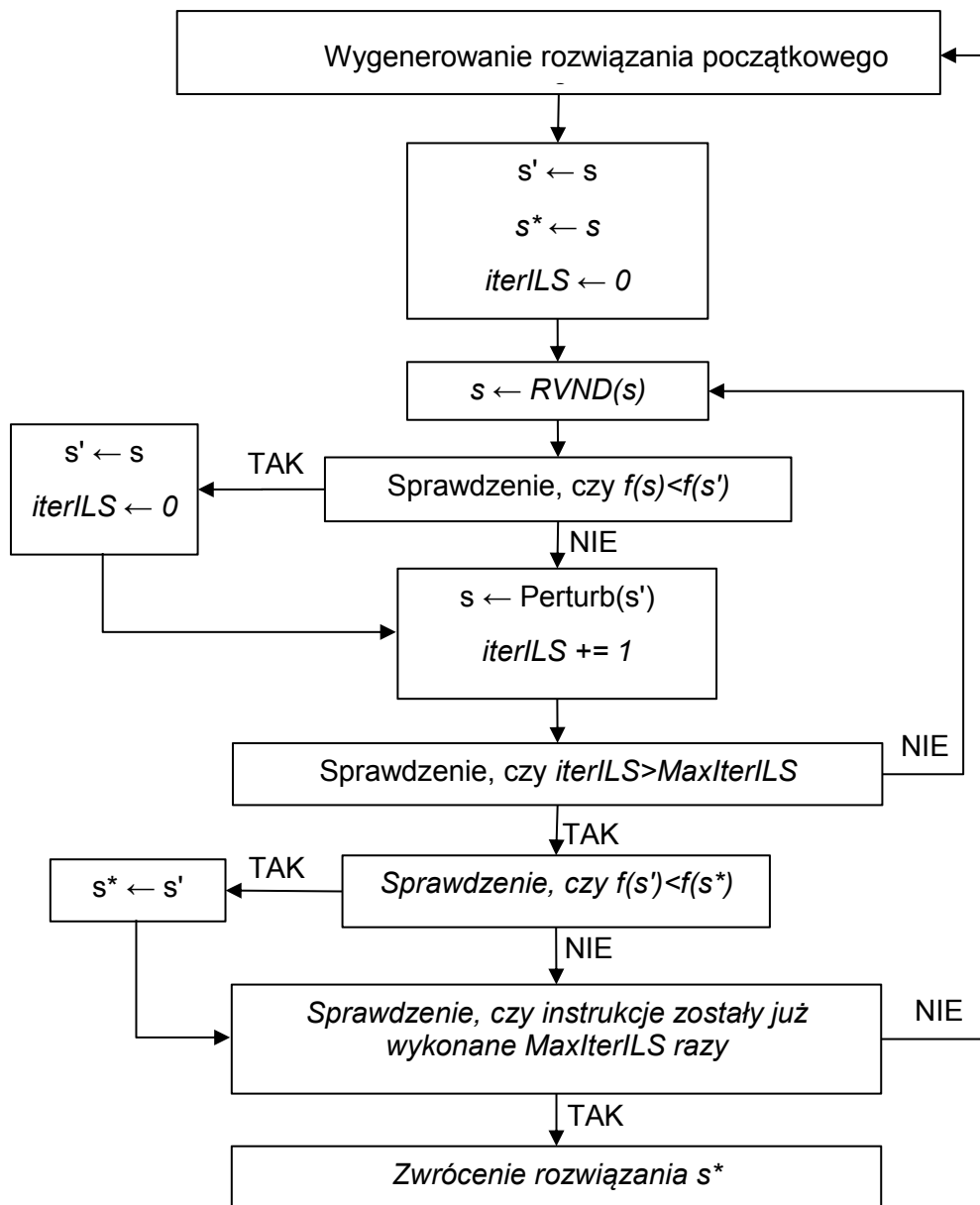
Algorytm 2. RVND

1. **Procedure** RVND (s):
2. **while** $NL \neq \emptyset$:
3. losowy wybór sąsiedztwa $N^{(n)} \in NL$
4. znalezienie najlepszego sąsiada s' z $s \in N^{(n)}$
5. **if** $f(s') < f(s)$:
6. $s \leftarrow s'$
7. $s \leftarrow \text{IntraRouteSearch}(s)$
8. **else**:
9. usunięcie $N^{(n)}$ z NL
10. **return** s

Postępowanie *ILS-RVND* przedstawione zostało natomiast w formie pseudokodu w *Algorytmie 3* oraz za pomocą schematu na *Rys. 2*.

Algorytm 3. Heurystyka ILS-RVND

1. **Procedure** ILS-RVND (*CandidateList*, *MaxIter*, β):
2. $v \leftarrow$ całkowita ilość pojazdów
3. $\text{MaxIterILS} \leftarrow n + v * \beta$
4. **for** i in range(1, *MaxIter*):
5. $s \leftarrow \text{GenerateInitialSolution}(v, \text{CandidateList})$
6. $s' \leftarrow s$
7. $s^* \leftarrow s'$
8. $\text{iterILS} \leftarrow 0$
9. **while** $\text{iterILS} \leq \text{MaxIterILS}$:
10. $s \leftarrow \text{RVND}(s)$
11. **if** $f(s) < f(s')$:
12. $s' \leftarrow s$
13. $\text{iterILS} \leftarrow 0$
14. $s \leftarrow \text{Perturb}(s')$
15. $\text{iterILS} += 1$
16. **if** $f(s') < f(s^*)$:
17. $s^* \leftarrow s'$
18. **return** s^*



Rys. 2. Schemat działania heurystyki ILS-RVND

Generowanie rozwiązań początkowych dokonywane jest przy zastosowaniu algorytmu *GenerateInitialSolution*, stanowiącego procedurę konstrukcyjną, wykorzystującą dwie strategie wstawiania, tj. *Strategię sekwencyjnego wstawiania (Sequential Insertion Strategy - SIS)* oraz *Strategię równoległego wstawiania (Parallel Insertion Strategy - PIS)*, z których każda stanowi odrębny, rozbudowany algorytm. *GenerateInitialSolution* wykorzystuje również dwa rodzaje kryteriów wstawiania tj. *Zmodyfikowane kryterium najtańszego wykonalnego wstawienia (Modified Cheapest Feasible Insertion Criterion - MCFIC)* oraz *Kryterium*

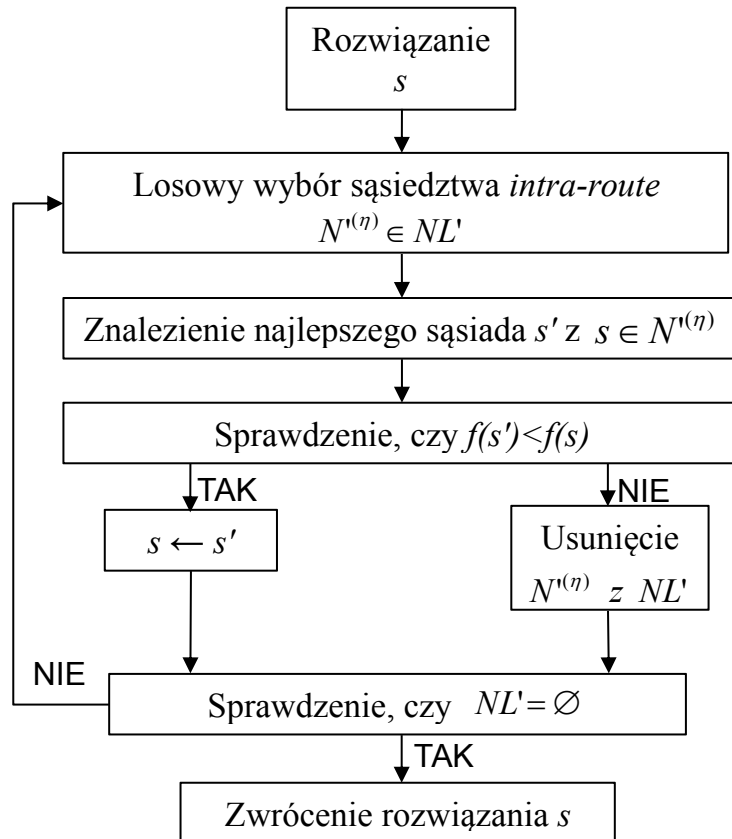
najbliższego wykonalnego wstawienia (*Nearest Feasible Insertion Criterion - NFIC*) [Labadie, Prins i Prodhon 2016, Penna, Subramanian i Ochi [2013]. Należy podkreślić, że algorytm *GenerateInitialSolution* zaproponowany został przez Pennę, Subramaniana i Ochiego [2013] w wersji, w przypadku której wielokrotnie może dochodzić do generowania rozwiązań niedopuszczalnych. W celu poprawy efektywności autor rozprawy wprowadza wobec tego pewną modyfikację, polegającą na włączeniu do procedury *GenerateInitialSolution*, a także współpracujących z nią procedur *SequentialInsertion* oraz *ParallelInsertion* instrukcji odpowiedzialnych za zapewnienie generowania wyłącznie rozwiązań wykonalnych. Pseudokod postępowania *GenerateInitialSolution* zamieszczony został w *Algorytmie 4*.

Algorytm 4. *GenerateInitialSolution*

1. *Procedure GenerateInitialSolution* (v , *CandidateList*):
2. $s \leftarrow$ zbiór w postaci $\{s^1, \dots, s^v\}$ tworzony przez v pustych tras
3. **for** v' in range($1, v$):
4. $s^{v'} \leftarrow k \in \text{CandidateList}$ wybrane losowo, z uwzględnieniem warunku wykonalności
5. $\text{CandidateList} \leftarrow \text{CandidateList} \setminus \{k\}$
6. $\text{InsertionCriterion} \leftarrow$ wybrane losowo spośród MCFIC lub NFIC
7. $\text{InsertionStrategy} \leftarrow$ wybrane losowo spośród SIS lub PIS
8. **if** $\text{InsertionStrategy} == \text{SIS}$:
9. $s \leftarrow \text{SequentialInsertion}(s, v, \text{CandidateList}, \text{InsertionCriterion})$
10. **else**:
11. $s \leftarrow \text{ParallelInsertion}(s, v, \text{CandidateList}, \text{InsertionCriterion})$
12. **return** s

Budowa algorytmu *IntraRouteSearch* zawierającego instrukcje przeszukiwania *intra-route* w znacznym stopniu zbliżona jest do budowy uprzednio omawianego algorytmu *RVND* i podobnie, jak on korzysta ze zbioru struktur sąsiedztwa. Zbiór ten, określony mianem *Listy sąsiedztwa intra-route* (oznaczany jako NL') tworzony jest przez r' tego rodzaju struktur, tzn. $NL' = \{N^{(1)}, \dots, N^{(r')}\}$.

Schemat postępowania zamieszczony został na *Rys. 3.*, natomiast jego pseudokod w *Algorytmie 5*.



Rys. 3. Schemat działania algorytmu *IntraRouteSearch*

Algorytm 5. *IntraRouteSearch*

1. **Procedure** *IntraRouteSearch* (s):
2. zainicjowanie listy sąsiedztwa *Intra-Route* (NL')
3. **while** $NL' \neq \emptyset$:
4. losowy wybór sąsiedztwa $N^{(n)} \in NL'$
5. znalezienie najlepszego sąsiada s' z $s \in N^{(n)}$
6. **if** $f(s') < f(s)$:
7. $s \leftarrow s'$
8. **else**:
9. usunięcie $N^{(n)}$ z NL'
10. **return** s

W rozprawie rozpatrywano różnorodne struktury sąsiedztwa *intra-* oraz *inter-route* bazujące zarówno na znanych operatorach, jak również i proponowanych przez autora, których wykaz zawarty został w *Tab. 1*.

Tab. 1. Rozpatrywane struktury sąsiedztwa

Sąsiedztwa <i>inter-route</i>	<i>Shift(1,0), Shift(2,0), Swap(1,1), Swap(2,1), Swap(2,2), Cross(1,1), Cross(2,1), Cross(2,2), Cross(3,1), Cross(3,2), Cross(3,3), Cross(4,1), Cross(4,2), Cross(4,3), Cross(4,4), iCross(2,1), iCross(2,2), iCross(3,1), iCross(3,2), iCross(3,3), iCross(4,1), iCross(4,2), iCross(4,3), iCross(4,4), Mixture-Cross(2,1), Mixture-Cross(2,2), Mixture-Cross(3,1), Mixture-Cross(3,2), Mixture-Cross(3,3), Mixture-Cross(4,1), Mixture-Cross(4,2), Mixture-Cross(4,3), Mixture-Cross(4,4), K-Shift(1), K-Shift(2), K-Shift(2)C, K-Shift(3), K-Shift(3)C, K-Shift(3,2,1), K-Shift(3,2,1)C, K-Shift(3,2,1)D, K-Shift(3,2,1)DC</i>
Sąsiedztwa <i>intra-route</i>	<i>2-opt, 3-opt, Or-opt (1), Or-opt (2), Or-opt (3), Or-opt (3,2,1), Or-opt (3,2,1)R, Exchange, Displacement, Reinsertion</i>

W pracy rozpatrywano dodatkowo trzy mechanizmy zaburzające. Dwa z nich, w postaci *Multiple-Swap (1-1)* oraz *Multiple-Shift (1-1)*, są algorytmami opracowanymi przez autora z uwzględnieniem operatorów przedstawionych przez Pennę, Subramaniana i Ochiego [2013]. Trzeci natomiast, nazwany *Modified-Multiple-Shift (1-1)* zaproponowany został przez autora z myślą o zwiększeniu siły wprowadzanych zaburzeń.

5. Analiza struktur sąsiedztwa

Ocena poszczególnych struktur dokonywana była przy wykorzystaniu problemów testowych obejmujących sieci o 10, 20, 50, 75, 100, 125, 150, 175 oraz 200 wierzchołkach. Dla każdego z tych wymiarów rozpatrywano pięć odrębnych problemów (różniących się strukturą dostępnej floty oraz macierzami odległości). Pomiary obejmowały dziesięciokrotne wywołania algorytmów, podczas których mierzony był czas ich działania (w sekundach CPU).

System oprogramowany został w języku *Python* w wersji 2.7.3., pomiary przeprowadzone były natomiast przy wykorzystaniu komputera stacjonarnego wyposażonego w procesor i5-4590 3.30 GHz, 8GB pamięci RAM oraz kartę graficzną GeForce 960GTX z systemem *Windows 7 Professional* w wersji 64-bitowej.

Wyniki analiz struktur sąsiedztwa zamieszczone zostały w *Tab. 2 - 5*. Tabele 2 oraz 3 zawierają dane dotyczące jakości rozwiązań generowanych przez rozpatrywane

algorytmy. Tabele te zaprojektowane zostały w taki sposób, aby dostarczały informacji donośnie tego średnio w jakim procencie rozwiązania, które zostały wygenerowane za pomocą danej struktury dobiegają (są gorsze - "+", bądź lepsze - "-") od rozwiązań zwróconych przez strukturę zidentyfikowaną, jako dającą rozwiązania najwyższej jakości (w przypadku *intra-route* jest to postępowanie *3-opt*, a w przypadku *inter-route* *Swap(2,1)*). Należy dodać, że *Tab. 3.* zawiera wyłącznie najkorzystniejsze spośród struktur należących do poszczególnych grup.

Tab. 2. Wyniki analiz struktur sąsiedztwa *intra-route*

	Najlepsze uzyskane rozwiązanie	Średnia wartość uzyskanych rozwiązań	Najgorsze uzyskane rozwiązanie
<i>2-opt</i>	+1,22%	+3,17%	+10,28%
<i>Displacement</i>	+4,62%	+13,26%	+46,43%
<i>Or-opt 1</i>	+6,60%	+16,58%	+60,71%
<i>Or-opt 2</i>	+6,60%	+13,95%	+45,24%
<i>Or-opt 3</i>	+6,26%	+14,08%	+45,24%
<i>Or-opt 321</i>	+6,78%	+18,42%	+67,86%
<i>Or-opt 321R</i>	+5,56%	+14,95%	+45,24%
<i>Reinsertion</i>	+6,32%	+16,58%	+60,71%
<i>Exchange</i>	+6,12%	+15,78%	+52,03%

Tab. 3. Wyniki analiz struktur sąsiedztwa *inter-route*

	Najlepsze uzyskane rozwiązanie	Średnia wartość uzyskanych rozwiązań	Najgorsze uzyskane rozwiązanie
<i>iCross (3,1)</i>	-0,64%	+31,20%	+62,14%
<i>K-Shift (1)</i>	+23,40%	+142,76%	+195,71%
<i>Shift (1,0)</i>	-29,81%	+15,85%	+80,30%
<i>Shift (2,0)</i>	+20,74%	+134,75%	+282,31%
<i>Swap (2,2)</i>	-26,60%	+7,34%	+57,58%
<i>Swap (1,1)</i>	-42,63%	+14,96%	+92,42%
<i>Cross (2,1)</i>	+21,47%	+8,18%	+54,55%
<i>Mixture-Cross(2,2)</i>	-38,46%	+28,72%	+103,03%

Tab. 4. Średni czas [sekundy CPU] zwracania rozwiązań przez struktur sąsiedztwa *intra-route*

Rozmiar problemu	<i>iCross</i> (3,1)	<i>K-Shift</i> (1)	<i>Shift</i> (1,0)	<i>Shift</i> (2,0)	<i>Swap</i> (2,2)	<i>Swap</i> (2,1)	<i>Swap</i> (1,1)	<i>Cross</i> (2,1)	<i>MixtureCross</i> (2,2)
10	0,0095	0,0016	0,0013	0,0005	0,0005	0,0046	0,0017	0,0025	0,001
20	0,0224	0,0119	0,0138	0,0004	0,0004	0,0079	0,0282	0,0229	0,0273
50	1,2507	0,1838	0,9519	0,0008	0,0008	2,0181	2,2343	3,1119	2,2099
75	8,7755	1,1596	7,1006	4,2915	4,2915	11,0135	10,2097	20,13	13,1437
100	44,0911	3,0131	30,011	13,3527	13,3527	62,9098	65,6051	83,7027	69,142
125	58,369	4,0839	32,1582	9,6102	9,6102	67,4691	79,7277	109,8399	103,6338
150	117,2526	9,237	46,7843	0,0036	0,0036	108,937	136,3634	204,3572	233,5933
175	217,858	14,2017	108,8364	43,625	43,625	286,5273	262,2534	287,6927	305,0709
200	394,402	25,3341	236,0808	156,6968	156,6968	474,9717	490,5828	468,0362	437,5308

Tab. 5. Średni czas [sekundy CPU] zwracania rozwiązań przez struktur sąsiedztwa *inter-route*

Rozmiar problemu	<i>3-opt</i>	<i>2-opt</i>	<i>Displacement</i>	<i>Or-opt 1</i>	<i>Or-opt 2</i>	<i>Or-opt 3</i>	<i>Or-opt 321</i>	<i>Or-opt 321R</i>	<i>Reinsertion</i>	<i>Exchange</i>
10	0,0113	0,0022	0,0005	0,0004	0,0004	0,0005	0,0005	0,0006	0,0004	0,0005
20	0,3359	0,0162	0,0013	0,0006	0,0007	0,0006	0,0015	0,0013	0,0006	0,0008
50	7,545	0,2805	0,0153	0,0021	0,0028	0,0028	0,0057	0,0063	0,0015	0,0028
75	13,2567	0,945	0,0487	0,0041	0,0055	0,0045	0,0111	0,0155	0,0028	0,0062
100	23,3727	2,2471	0,1159	0,0086	0,0079	0,0071	0,0209	0,0265	0,0059	0,0169
125	32,4404	4,3414	0,2226	0,0112	0,0133	0,011	0,0309	0,0415	0,0074	0,0427
150	52,7689	7,5016	0,3949	0,0193	0,019	0,0163	0,048	0,0698	0,0123	0,0351
175	121,0709	11,8221	0,6119	0,0213	0,021	0,0224	0,0551	0,046	0,0139	0,0487
200	356,7254	17,3999	0,8964	0,0281	0,0295	0,0267	0,0953	0,0981	0,0183	0,0422

Zgodnie z wcześniej podaną informacją na podstawie wyników przeprowadzonych pomiarów można stwierdzić, że najwyższą skutecznością w swoich grupach odznaczają się struktury $3-opt$ oraz $Swap(2,1)$. Rozwiązania, które generują charakteryzują się bowiem jakością zdecydowanie przewyższającą jakość wyników zwracanych przez pozostałe struktury.

Wśród struktur *intra-route* na uwagę zasługują również zwracający wyniki nieznacznie gorsze od $3-opt$, algorytm $2-opt$ (charakteryzujący się jednocześnie znacznie krótszym czasem działania) oraz odznaczający się niedługim czasem generowania stosunkowo dobrych rozwiązań algorytm *Displacement*. Pozostałe struktury tego typu reprezentują natomiast podobny poziom, zarówno w kwestii skuteczności, jak i efektywności. Warto jednak zauważyć, że postępowanie $Or-opt3,2,1R$ zwraca rozwiązania korzystniejsze od $Or-opt3,2,1$.

Pomiędzy strukturami sąsiedztwa *inter-route* dobrą jakością zwracanych rozwiązań cechują się również algorytmy $Swap(2,2)$ oraz $Cross(2,1)$. Istotnym jest jednak, że pierwszy z nich odznacza się zdecydowanie korzystniejszym czasem działania. W grupie tej wyróżnia się również postępowanie $K-Shift(1)$, które w prawdzie wykazuje zdolność do generowania rozwiązań gorszych jakościowo, jednak odznacza się znacznie krótszym czasem zwracania wyników, w porównaniu z pozostałymi strukturami tego typu. Dodatkowo specyfika wykonywanego przez niego ruchu znacznie odbiega od operatorów pozostałych struktur, w związku z czym może on pełnić korzystną funkcję w poszerzaniu obszaru przeszukiwań.

Istotnym jest, że największe różnice w wartościach funkcji celu uzyskiwanych rozwiązań dotyczą przypadków sieci o niewielkich rozmiarach (zawierających małą liczbę wierzchołków), związane jest to z faktem, iż specyfika niektórych struktur obejmuje przemieszczanie dłuższych fragmentów tras, a co za tym idzie w sieciach tego rodzaju są one w stanie wykonywać stosunkowo niewiele iteracji.

Sugerując się wynikami przeprowadzonych pomiarów, pamiętając również o konieczności przeciwdziałania uwięzieniu w lokalnym optimum przyjęto, że w finalnej wersji systemu powinny zostać uwzględnione struktury sąsiedztwa w postaci: $Swap(2,1)$, $Swap(2,2)$, $Shift(2,0)$, $K-Shift(1)$, $iCross-exchange(3,1)$, *Displacement*, $2-opt$, $Or-opt2$ oraz $Or-opt3,2,1R$.

6. Podsumowanie i wnioski

Weryfikacja poprawności działania systemu przeprowadzona została na przykładzie transportu mleka odbieranego z gospodarstw rolniczych. Stanowiącego niezwykle wymagający surowiec, odznaczający się niską podatnością transportową i magazynową, wykazujący zdolność do szybkiego pogarszania swoich parametrów jakościowych, w przypadku którego wymagana jest konieczności zapewnienia łańcucha chłodniczego.

Podczas weryfikacji uwzględniano rzeczywisty problem transportowy związany z przemieszczaniem mleka surowego na trasach łączących gospodarstwa rolne z zakładem przetwórczym. Rozpatrywany przypadek obejmował obsługę 114 punktów odbioru surowca o łącznej jego podaży wynoszącej 42 139 litrów. Obsługa gospodarstw dokonywana była przy wykorzystaniu czterech pojazdów o pojemnościach wynoszących kolejno: 10 800, 10 800, 11 400 oraz 11 700 litrów. Stosowane w zakładzie przetwórczym rozwiązanie stanowiło natomiast zbiór tras o całkowitej długości wynoszącej 356 km.

Wyniki pomiarów stanowiących rezultat wywołań systemu zamieszczone zostały w *Tab. 6* oraz *7*.

Tab. 6. Weryfikacja poprawności działania systemu

Lp.	Wartość funkcji celu rozwiązania początkowego [km]	Ilość iteracji algorytmu ILS-RVND	Wartość funkcji celu rozwiązania końcowego [km]	Czas działania [s]
1	1159	35	283	138,1868
2	556	36	286	145,4927
3	559	73	274	219,8506
4	547	39	280	142,4653
5	531	66	272	153,2134
6	526	77	293	356,376
7	522	33	287	167,7279
8	483	43	274	197,8181
9	1043	38	276	143,1186
10	1099	50	268	206,3598

Tab. 7. Wyniki weryfikacji poprawności działania systemu

	Rozwiązanie najlepsze [km]	Średnia wartość funkcji celu uzyskanych rozwiązań [km]	Rozwiązanie najgorsze [km]	Minimalny czas działania algorytmu [s]	Średni czas działania algorytmu [s]	Maksymalny czas działania algorytmu [s]
	268	279,3	293	138,1868	187,06092	356,376
Wartość funkcji celu względem stosowanego rozwiązania	-25%	-22%	-18%			

Na podstawie danych zawartych w Tab. 6-7, a także wyników analiz struktur sąsiedztwa zamieszczonych w Tab. 2-5 zostały wysunięte następujące wnioski:

- Metoda heurystyczna *ILS-RVND* stanowi skuteczne narzędzie optymalizacji problemów wpisujących się w specyfikę zagadnienia *Heterogeneous Vehicle Routing Problem*. Odznacza się bowiem dobrą jakością zwracanych rozwiązań, a także korzystnym stosunkiem tej jakości do czasu działania.
- Heurystyka *ILS-RVND* nie może wykorzystywać wyłącznie struktur sąsiedztwa zwracających rozwiązania najlepszej jakości. Zróżnicowanie wpływa bowiem korzystnie na przeciwdziałanie uwięzieniu w lokalnym optimum. Istotnym jest również, żeby wykorzystywane ruchy opierały się na różnych strategiach, co przynosi korzyść w postaci poszerzenia obszaru przeszukiwań, a tym samym zwiększa szansę na znalezienie rozwiązania korzystniejszego.
- Ogólnie struktury sąsiedztwa *inter-route* odznaczają się lepszą jakością zwracanych rozwiązań w stosunku do struktur typu *intra-route*, ich czas działania jest jednak w większości przypadków zdecydowanie dłuższy.
- Odejście od deterministycznego charakteru wykonywanych modyfikacji w sposób korzystny wpływa na jakość zwracanych rozwiązań. Należy jednak pamiętać, że w celu zwiększania szansy na osiągnięcie lepszych wyników konieczne jest wielokrotne wywoływanie rozpatrywanych algorytmów. Wzrost ilości wywołań wpływa jednak w znacznym stopniu na wydłużenie czasu generowania rozwiązań. Ich ilość powinna być wobec tego ustalana na podstawie maksymalnego dopuszczalnego czasu działania systemu.

Literatura

1. Aarts E., Lenstra J.K. 2003: Local search in combinatorial optimization. Princeton University Press. ISBN: 978-0-691115-22-1.
2. Álvarez A., Munari P. 2016: Metaheuristic approaches for the vehicle routing problem with time windows and multiple deliverymen. *Gest. Prod. São Carlos*. V.32.n.2.279-293. DOI: 101590/0104-530X2359-15.
3. Baldacci R., Battarra M., Vigo D. 2008: Routing a Heterogeneous Fleet of Vehicles. In: Golden B., Raghaven S., Wasil E. (eds) *The Vehicle Routing Problem: Latest Advances and New Challenges*. Operations Research/Computer Science Interfaces. Vol. 43. Springer, Boston. MA. ISBN: 978-0-387-77777-1. DOI: 10.1007/978-0-387-77778-8_1.
4. Baldacci R., Mingozzi A. 2006: Lower Bounds and an Exact Methods for the Capacitated Vehicle Routing Problem. *Service Systems and Service Management*. 2 pp. 1536-1546. DOI: 10.1109/ICSSSM.2006.320764.
5. Baran J. 2012: Skala działania a rozwiązania w zakresie magazynowania i transportu w przedsiębiorstwach przetwórstwa mleka. *Logistyka* 2/210.
6. Baryła-Paśnik M., Piekarski W., Dudziak A., Piecak A. 2014: Zarządzanie systemem logistycznym transportu w dystrybucji mleka. *Innowacje w zarządzaniu i inżynierii produkcji*. Tom I/pod red. Ryszarda Knosali. Polskie Towarzystwo Zarządzania Produkcją. Opole. 937-944. ISBN: 978-83-930399-6-8.
7. Bendkowski J., Kramarz M., Kramarz W. 2010: *Metody i techniki ilościowe w logistyce stosowanej. Wybrane zagadnienia*. Wydawnictwo Politechniki Śląskiej. Gliwice. ISBN: 978-83-7335-597-2.
8. Bladi M.M., Crainic T.G., Perboli G., Tadei R. 2012: The generalized bin packing problem. *Transportation Research Part E: Logistics and Transportation Review*. Volume 48, Issue 6. 1205-1220.
9. Brar G.S., Saini G. 2011: Milk Run Logistics: Literature Review and Directions. *Lecture Notes in Engineering and Computer Science*. 1. 797-801.

10. Bräysy O., Gendreau M. 2005: Vehicle Routing Problem with Time Windows. Part I: Route Construction and Local Search Algorithms. *Transportation Science*. Vol. 39. No. 1: 104-118. ISSN: 0041-1655.05.3901.0119.
11. Bräysy O., Gendreau M. 2005: Vehicle Routing Problem with Time Windows. Part II: Metaheuristics. *Transportation Science*. Vol. 39. No. 1: 119-139. ISSN: 0041-1655.05.3901.0119.
12. Caceres-Cruz J., Grasas A., Ramalhinho H., Juan A.A. 2014: A savings-based randomized heuristic for the heterogeneous fixed fleet vehicle routing problem with multi-trips. *Journal of Applied Operational Research* (2014).6(2), 69-81. ISSN 1735-8523.
13. Całczyński A., Kędzierska-Stróż D., Orzechowska D., Śleszyński Z. 2000: Elementy badań operacyjnych w zarządzaniu. Tom II. Wydawnictwo Politechniki Radomskiej. Radom. ISBN:
14. Chang Y., Chen L. 2007: Solve the Vehicle Routing Problem with Time Windows via a Genetic Algorithm. *Discrete and Continuous Dynamical Systems Supplement*. 240-249.
15. Choi E., Tcha D. 2006: A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*.
16. Clarke G., Wright J.W. 1964: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*. 12:568–581.
17. Coffman E.G., Garey M.R., Johnson D.S. 1997: Approximation algorithms for Bin Packing: A survey. *Approximation algorithms for Np-hard problems*. PWS Publishing C. Boston. 46-93. ISBN: 0-534-94968-1.
18. Coello C. A. 2011: Learning and Intelligent Optimization. 5th international Conference. Lion 5 Rome. Italy. ISBN: 978-3-642255-65-6.
19. Dantzig G.B., Ramser J.H. 1959: The truck dispatching problem. *Management Science*. 6:80–91.
20. Dell'Amico M., Monaci M. Pagani C., Vigo D. 2006: Heuristic approaches for the fleet size and mix vehicle routing problem with time widnows. Technical report, DISMI, University of Modena and Reggio Emilia. Italy.

21. Dutkiewicz L., Kucharska E. 2010: Algorytm lanowania tras dostaw dla wielu komiwojazerów. *Automatyka*. T.14.z.3/2.
22. El-Sherbeny N.A. 2010: Vehicle routing problem with time windows. An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University (Science)* (2010) 22.123-131. DOI: 10.1016/j.jksvs.2010.03.002.
23. Gaworski M., Makowska K. 2013: Analiza wykorzystania cystern samochodowych do odbioru mleka w małym zakładzie mleczarskim. *Inżynieria Rolnicza*. Z.2(143) T.1 s.61-68.
24. Gendreau M., Laporte G., Musaraganyi Ch., Taillard É.D. 1999: A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*. Elsevier Science. 26 (1999) 1153-1173.
25. Grasas A., Caceres-Cruz J., Loureço H.R., Juan A.A., Roca M. 2013: Vehicle routing in a Spanish distribution company: Saving using a savings-based heuristic. *Operational Research Society*. Vol. 26.3.191-202.
26. Hanczar P. 2010: Wspomaganie decyzji w obszarze wyznaczania tras pojazdów. *Decyzje*.13.55-80.
27. Harmanani H., Azar D., Helal N. 2011: A Simulated Annealing Algorithm for the Capacitated Vehicle Routing Problem. *Proceedings of the IJCA 26th International Conference on Computers and Their Applications, CATA 2011*. 96-101.
28. Jachimowski R., Ambroziak T. 2012: Klasteryzacja punktów obsługi i wyznaczanie tras pojazdów w dwuszczeblowym systemie dystrybucji. *Logistyka* 4/2012.
29. Jadczyk R. 2005: Rozwiązywanie zagadnień układania tras pojazdów z wykorzystaniem algorytmów ewolucyjnych. *Badania operacyjne i decyzyjne*. Nr 3-4.
30. Jafari-Eskandari M., Aliahmadi A.R., Khaleghi G.H.H. 2010: A robust optimisation approach for the milk run problem with time windows with inventory uncertainty: an auto industry supply chain case study. *Int. J. Rapid Manufacturing*. Vol. 1.No.3.334-347.

31. Jagiełło S., Żelazny D. 2013: Algorytm równoległy dla problemu marszrutyzacji. *Innowacje w zarządzaniu i inżynierii produkcji / pod red. Ryszarda Knosali. Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją. S.748-757.*
32. Jakubowska A., Piechocka K. 2015: W poszukiwaniu optymalnej trasy – Wybrane algorytmy w zastosowaniu do problemu komiwojażera. *Journal of Translogistics.*
33. Józefowska J. 2011: Modele i narzędzia optymalizacji w systemach informatycznych zarządzania. Wyd. Politechniki Poznańskiej. Poznań.
34. Kachitvichyanukul V., Sombuntham P., Kunnapapdeelert S. 2012: Benchmark Problem Instances for Generalized Multi-Depot Vehicle Routing Problems with Pickup and Delivery Requests. *Research Gate.*
35. Kaczmarek K., Nowa A. 2007: Sieci. Analiza i optymalizacja. Wyd. Politechniki Śląskiej. Gliwice. ISBN: 978-83-7335-393-0.
36. Kara B., Sabuncuoglu I., Bidanda B. 2014: *Global Logistics Management.* CRC Press. ISBN: 978-1-482226-94-2.
37. Kara I., Laporte G., Bektas T. 2003: A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operation Research.* DOI: 10.1016/50377-2217(03)00377-1.
38. Klepacki B., Rokicki T. 2011: Logistyka w przedsiębiorstwach przetwórstwa owoców i warzyw. *Koncepcje i strategie logistyczne. Logistyka 3/2011.*
39. Kobus J., Kmiecik D. 2006: Jakość mikrobiologiczna i skład chemiczny mleka surowego pochodzącego z wielkich i małych gospodarstw rolnych Wielkopolski w 2004 roku. *Żywność. Nauka. Technologia. Jakość. 2(47) Supl. 108-115.*
40. Korf R.E. 2002: A New Algorithm for Optimal Bin Packing. *Eighteenth national conference on Artificial Intelligence.* American Association for Artificial Intelligence. 731-736. ISBN: 0-262-51129-0.
41. Labadie N., Prins Ch., Proghon C. 2016: *Metaheuristics for Vehicle Routing Problems.* Wiley. ISBN: 978-1-848218-11-6. DOI: 101002/9781119136767.

42. Laporte G. 1992: The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*. 59 (1992) 345-358.
43. Laporte G., Nobert Y. 1983: A Branch and Bound Algorithm for the Capacitated Vehicle Routing Problem. *OR Spektrum*. Springer-Verlag.
44. Lau H.Ch., Sim M., Teo K.M. 2003: Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*. 148(2003) 559-569.
45. Letchford A.N., Lodi A. 2011: Mathematical programming approaches to the travelling salesman problem. *Wiley Encyclopedia of Operations Research and Management Science*. Ed./James J. Cachron, Louis A. Cox, Pinar Keshinocak, Jeffrey P. Kharoufreh, J. Cole Smith. Chichester. John Wiley and Sons Ltd.
46. Little J.D.C., Murty K.G., Sweeney D.W., Karel C. 1963: An algorithm for the travelling salesman problem. Massachusetts Institute of Technology. Massachusetts.
47. Lourenço H. R., Martin O. C., Stützle T. 2003: Iterated local search. *Handbook of Metaheuristics*. Kluwer Academic. Dordrecht.
48. Michłowicz E. 2009: Problem komiwojżera dla kilku centrów dystrybucji. *Prace naukowe Politechniki Warszawskiej. Transport*. Z.70.
49. Mladenović N., Hansen P. 1997: Variable neighborhood search. *Comput. Oper. Res.* 24, 1097-1100.
50. Osman I.H. 1993: Meta-strategy simulated annealing and Tabu search algorithms for the vehicle routine problem. *Annals of Operations Research*. 41(1993) 421-451. DOI: 10.1007/BF02023004.
51. Penna H. V. P., Subramanian A., Ochi L. S. 2013: An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *J. Heuristics* (2013) 19: 201-232.
52. Peterson B., von Hove W.J., Kekre S., Debo L. 2010: Flexible Milk-Runs for Stochastic Vehicle Routing.

53. Pichpibul T., Kawtummachai R. 2012: An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem. *Science Asia*. 38: 307-318. DOI: 10.2306/1513-1874.2012.38.307.
54. Pichpibul T., Kawtummachai R. 2013: A heuristic approach based on Clarke-Wright algorithm for Open Vehicle Routing Problem. *Scientific World Journal*. 2013: 874349. DOI: 10.1155/2013/874349.
55. Pop P.C., Fuksz L., Marc A.H. 2014: A Variable Neighborhood Search Approach for Solving the Generalized Vehicle Routing Problem. *Hybrid Artificial Intelligence Systems*. 9th International Conference. Salamanca. Springer. ISBN: 978-3-319-07616-4.
56. Rostami A.S., Mohanna F., Keshavarz H., Hosseinabadi A.A.R. 2015: Solving Multiple Traveling Salesman Problem using the Gravitational Emulation Local Search Algorithm. *App. Math. Inf. Sci.* 9, No. 2, 699-709.
57. Rudziński R. 2010: Organizacja logistyki w zakładach przetwórstwa mleka. *Zeszyty Naukowe Uniwersytetu Przyrodniczo-Humanistycznego w Siedlcach. Administracja i Zarządzanie*. Nr 87.
58. Sadjadi S.J., Jafari M., Amini T. 2009: A new mathematical modeling and a genetic algorithm search for milk run problem (an auto industry supply chain case study). *Int.J. Adv. Manuf. Technol.* Springer-Verlog London. 44. 194-200. DOI: 10.1007/s00170-008-1648-5.
59. Sadouni K. 2006: Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Nonlinearly Penalized Delays. *Jorunal of Applied Sciences* 6(9): 1969-1973.
60. Sedighpoura M., Yousefikhoshbakhtb M., Daranic N.M. 2011: An Effective Genetic Algorithm for Solving Multiple Traveling Salesman Problem. *Journal of Optimization in Industrial Engineering* 8 (2011) 73-79.
61. Simchi-Levi D., Chen X., Bramel J. 2005: Solving the VRP Using a Column Generation Approach. In: *The Logic of Logistic*. Springer Series in Operations Research. Springer, New York, NY. ISBN: 978-0-387-22199-1. DOI: 10.1007/0-387-22619-2_16.

62. Subramanian A., Drummond L. M. A., Bentes C., Ochi L. S., Farias R. 2010: A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Journal Computers and Operations Research*. 37. Issue 11. 1899-1911. DOI: 10.1016/j.car.2009.10.011.
63. Subramanian A., Penna P.H.V., Uchoa E., Ochi L.S. 2011: A Hybrid Algorithm for the Fleet Size and Mix Vehicle Routing Problem. *Industrial Engineering and Systems Management*. Metz-France.
64. Sysło M.M., Deo N., Kowalik J.S. 1993: *Algorytmy optymalizacji dyskretnej z programami w języku Pascal*. Wydawnictwo Naukowe PWN. Warszawa. ISBN:
65. Taillard É. D., Badeau P., Gendreau M., Guertin F., Patvin J.Y. 1997: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*. DOI: 10.1287/tresc.31.2.170.
66. Takes F.W., Kusters W.A. 2010: Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem. 22nd Benelux Conference on Artificial Intelligence (BNAIC 2010). Luxembourg 25-26 October 2010.
67. Tan K.C., Lee L.H., Zhu Q.L., Ou K. 2000: Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering* 15(2001) 281-295.
68. Toth P., Vigo D. 2001: *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematic. ISBN: 0-898-71851-1.
69. Toth P., Vigo D. 2014: *Vehicle Routing Problems, Methods and Applications*. Second Edition. Society of Industrial and Applied Mathematics Philadelphia. ISBN: 978-1-611973-58-7.
70. Vig M.G., Bolia N., Sundararaman G. 2013: Optimization of 'Milk Run' of JCB India Ltd. (Ballabgarh Plant). *Journal of Traffic and Logistics Engineering, Engineering and Technology Publishing*. Vol. 1. No. 2. DOI:10.12720/jtle.1.2.137-140.
71. Wassman N.A., Osman I.H. 2002: Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*. 53.768-782. DOI: 10.1057/palgrave.jors.2601344.

72. Yaman H.D. 2006: Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming*. 106(2): 365-390.
73. You Z., Jiao Y. 2014: Development and Application of Milk-Run Distribution Systems in the Express Industry Based on Saving Algorithm. *Mathematical Problems in Engineering*. Vol. 2014. DOI: 10.1155/2014/536459.