Piotr KISIELEWSKI
Politechnika Krakowska, Kraków

OPTYMALIZACJA PRZYDZIAŁU ZADAŃ TRANSPORTOWYCH

Słowa kluczowe

Zadania transportowe, plan przydziału zadań, grafik służb kierowców, optymalizacja przydziału zadań.

Streszczenie

W pracy przedstawiono problemu przydziału kierowców do zdefiniowanych zadań transportowych z celem minimalizacji kosztów operacyjnych dużej floty pojazdów. Zadanie w sensie matematycznym jest zagadnieniem optymalnego pokrycia zbiorów z uwzględnieniem ograniczeń wynikających z charakteru zadań, specyfiki przedsiębiorstwa i licznych przepisów prawa. W artykule omówiono etapy realizacji procesu przydziału ze szczególnym uwzględnieniem problemu optymalizacji. Zaproponowane dwie metody optymalizacyjne i przedstawiono wyniki przeprowadzonych dla nich testów na danych rzeczywistych przedsiębiorstw komunikacyjnych.

Wprowadzenie

Zagadnienie przydziału kierowców do realizowanych przez dużą flotę pojazdów zadań transportowych stanowi podstawę logistyki i efektywnej pracy przedsiębiorstwa. Powyższe zagadnienie stanowi istotny problem dla wszystkich przedsiębiorstw operujących liczebną flotą niezależnie od typu pojazdów, jak przedsiębiorstwa pasażerskiej komunikacji miejskiej i międzymiastowej, przedsiębiorstwa kolejowe, przedsiębiorstwa logistyczne i spedycyjne, linie lotnicze itp.

W sensie matematycznym problem definiowany jest jako zagadnienie optymalnego przydziału lub optymalnego pokrycia zbiorów (ang. *set covering problem*). Celem jest minimalizacja funkcji ogólnych kosztów planowanych przewozów. W teorii proste modele liniowe podobnych zagadnień rozwiązywane są za pomocą różnych odmian tzw. algorytmu węgierskiego.

W praktyce transportowej zagadnienie jest dużo bardziej skomplikowane i prowadzi do budowy znacznie bardziej złożonych modeli matematycznych, których rozwiązanie wymaga specjalistycznych, odmiennych algorytmów, przy czym z uwagi na dynamikę zmian planów transportowych wymagana jest bardzo duża szybkość algorytmów, umożliwiająca planistom pracę w czasie niemal rzeczywistym. Złożoność zagadnienia zwiększają liczne, trudne do zdefiniowania ograniczenia, problem różnej mocy pokrywanych zbiorów oraz wielowymiarowość relacji: zadanie – kierowca – pojazd.

W pracy przedstawiono system komputerowego wspomagania planowania przydziału zadań transportowych o umownej nazwie OptiGraf wg projektu autora, wdrożony w licznych przedsiębiorstwach transportowych.

1. Koncepcja systemu

W projekcie systemu o umownej nazwie OptiGraf przyjęto koncepcję interakcyjnego systemu planowania z zastosowaniem określonych procedur automatycznej generacji i optymalizacji. W tym sensie system jest programem projektowym, wyposażonym w odpowiednie narzędzia projektanta, zatem bardziej zbliżonym do programów klasy CAD niż klasycznych programów obsługi baz danych. Program umożliwia budowę harmonogramu pracy według dowolnych, definiowanych przez użytkownika, wielu różnych schematów stosowanych w przedsiębiorstwie.

Program umożliwia nie tylko budowę planu, ale także jego dynamiczną edycję w trakcie wykonania dzień po dniu, tj. bieżącą weryfikację wykonania.

Planowanie pracy drużyn trakcyjnych jest procesem skomplikowanym, wymagającym niezbędnej wiedzy w zakresie prawa pracy i doskonałej znajomości zasad funkcjonowania transportu danego typu.

Proces konstrukcji planów pracy składa się z etapów:

– generacja planu wg zdeterminowanych schematów dla różnych grup kierowców,
– przydział weekendów, dni wolnych i świąt,
– optymalizacja planu,
– dokładne zrównoważenie normy czasu pracy.

Każdy z powyższych etapów w systemie OptiGraf jest wspomagany przez całkowicie zautomatyzowane procedury matematyczne. Wszystkie automaty są parametryczne – tj. pracują na grupach służb, kierowców i przedziale dni zadanym przez operatora, wykorzystując dodatkowo definiowane przez użytkownika parametry. Parametryzacja automatów umożliwia ich szerokie stosowanie. Automaty zbudowane są w oparciu o funkcje deterministyczne i losowe.

2. Generacja harmonogramu wg schematów

Przykładem automatu deterministycznego jest możliwość generacji planu wg dowolnie długiego zdeterminowanego przez użytkownika schematu następstwa przedzielanych zadań/służb, w którym określone są zadania alternatywne, w kolejności ich preferencji obsady dla danego dnia i kierowcy.

ZR	ZR	ZR	ZR	ZR	W	ZP	ZP	ZP	ZP	ZP	W

Rys. 1. Przykład schematu następstwa zadań/służb

W przykładzie na rys.1 pokazano pracę wg schematu 5 dni pracy, 6 dzień wolny, gdzie: W oznacza dzień wolny, ZR oznacza pracę ranną, a ZP – popołudniową. Komórki/dni definiowanego schematu mogą być wypełniane zadaniami (identyfikatory służb), tzw. rezerwami lub wszelkimi typami dni wolnych (wolne zwykłe, wolne za sobotę, za niedzielę itp.).

3. Przydział dni wolnych

Przykładem algorytmu losowego jest algorytm przydziału dni wolnych w którym kolejność kierowców jest generowana losowo. Algorytm przebiega wg poniższych punktów:

P1. Pobieramy kierowców i układamy ich w losowej kolejności.

P2. Przechodzimy po wszystkich dniach i zapamiętujemy:
- a) ilość świąt, niedziel, sobót,
- b) wszystkie służby kierowcy – będzie nam to potrzebne przy przesuwaniu w lewo wszystkich służb,
- c) ilość kierowców i służb dostępnych każdego dnia.

P3. Dokładamy dla każdego kierowcy brakujące wolne w dni kalendarzowe:
- a) święta,
- b) niedziele,
- c) soboty.

P4. Usuwamy dla każdego kierowcy nadmiarowe wolne:
- a) święta,
- b) niedziele,
- c) soboty.

Dla każdego rodzaju automat przeszukuje kolejne dni i szuka wolnego danego rodzaju. Gdy znajdzie, to usuwa wolne, gdy jest spełniony warunek:

liczba kierowców < liczba służb + liczba przydzielonych wolnych (*)

Celem powyższego warunku jest, aby automat nie usunął na początku miesiąca nadmiarowych wolnych wszystkim kierowcom, ponieważ wtedy niepotrzebnie będzie szukał dni z wolnym i psuł wcześniej przypisane zdeterminowane schematy.

P5. Przechodzimy po dniach i dla każdego z nich:

a) przechodzimy po wszystkich kierowcach i sprawdzamy:

- czy kierowca jest dostępny w danym dniu (czy jest zatrudniony i czy nie ma zaplanowanego wolnego); jeżeli nie jest, to przechodzimy do następnego kierowcy,
- czy kierowca pracuje siódmy dzień pod rząd; jeżeli tak, to (**)
- czy kierowca ma co x (wartość z parametru) niedzielę wolną oraz spełniony jest warunek (*); jeżeli nie, to wykonujemy działanie (**)
- czy kierowca ma założoną liczbę prawdziwych weekendów wolnych oraz spełniony jest warunek (*); jeżeli nie, to wykonujemy działanie (**)
- czy kierowca ma założoną liczbę prawdziwych niedziel wolnych oraz spełniony jest warunek (*); jeżeli nie, to wykonujemy działanie (**)
- czy kierowca ma założoną liczbę prawdziwych sobót wolnych oraz spełniony jest warunek (*); jeżeli nie, to wykonujemy działanie (**).

Działanie (**): przesuwamy wszystkie służby w lewo do momentu aż w sprawdzanym dniu kierowca będzie miał wolne.

b) wyliczamy warunek:

delta = dostępni kierowcy w dniu – (ilość służb w dniu +
+ ilość wolnych w dniu)

Jeżeli *delta < 0* (za dużo wolnych), to przechodzimy po kierowcach i sprawdzamy:

- czy nie został zmodyfikowany w pkt. a),
- czy jest zatrudniony i nie ma planowanego wolnego (urlop, chorobowe itd.)

Jeżeli te warunki są spełnione, to przesuwamy wszystkie służby w lewo aż do momentu znalezienia dnia pracy.
Jeżeli *delta > 0* (za mało wolnych), to przechodzimy po kierowcach, sprawdzamy:
– czy kierowca ma w poprzednich 4 dniach pracę.
Jeżeli ten warunek jest spełniony, to przesuwamy wszystkie służby w lewo do momentu znalezienia dnia wolnego, jeżeli nie jest spełniony, to przechodzimy do kolejnego kierowcy.

3. Optymalizacja przydziału zadań

Najtrudniejszym, a zarazem najciekawszym z punktu widzenia metodyki etapem planowania jest matematyczna optymalizacja przydziału zadań dla operatorów pojazdów.

Optymalizacja opiera się na takim przypisaniu zadań, by ich sumaryczny koszt, wyliczany z użyciem funkcji kosztu (1) był jak najniższy.

$$K = \sum_{i=1}^{10} w_i p_i \tag{1}$$

gdzie: K to całkowity koszt przypisania, p_i to uwzględniane składniki kosztów, a w_i — wagi (koszty jednostkowe) określające wpływ określonego składnika na ostateczny wynik. Składniki kosztów określono w tabeli 1.

Tabela 1. Składniki kosztów funkcji celu optymalizacji

Lp.	Składnik	Wartość
1	Nakładanie się służb	Czas nakładania się służby na przypisane wcześniej (w minutach)
2	Nakładanie się służb u zmienników	Czas nakładania się służby na służby przypisane wcześniej zmiennikom danego kierowcy
3	Nadgodziny w dniu	Przekroczenie dziennego czasu pracy (wyliczane według zasad obowiązujących w danym przedsiębiorstwie), w minutach
4	Nad- i podgodziny w okresie rozliczeniowym	Przypisanie powodujące zwiększenie ilości nadgodzin lub podgodzin danego kierowcy w okresie rozliczeniowym (różnica bezwzględna pomiędzy normą i czasem pracy, w minutach)
5	Przekroczenie doby pracowniczej	Nadgodziny wynikające z przekroczenia doby pracowniczej względem dnia poprzedniego (w minutach)
6	Odwołanie z wolnego	480 (liczba minut w ośmiu godzinach), jeśli dany kierowca ma już zaplanowane wolne w tym dniu; zero w przeciwnym razie
7	Ponad sześć dni pracy	480, jeśli wskazany dzień jest co najmniej siódmym z kolei dniem pracy danego kierowcy; zero w przeciwnym razie

8	Brak preferencji służba/kierowca	48 min x (10 – preferencja); preferencja najniższa wynosi 0, najwyższa 10.
9	Nieprawidłowa zmiana	480 (liczba minut w ośmiu godzinach), jeżeli zarazem: (1) służba ma zdefiniowaną zmianę (2) kierowca ma w tym dniu zaplanowaną zmianę i (3) są to różne zmiany.
10	Niezachowanie 11 godzin odpoczynku	Czas (w minutach), którego zabrakło do pełnych 11 godzin

Poszczególne wagi definiowane są przez użytkownika i przechowywane w bazie danych. Wagi przed wyliczeniem funkcji kosztu są normalizowane do przedziału [0,1000]. Przedział ten został dobrany eksperymentalnie, tak by wartości nie były ani zbyt małe (co mogłoby wypaczyć macierz kosztu składającą się z liczb całkowitych), ani zbyt duże (co mogłoby spowodować przekroczenia zakresu przy sumowaniu liczb całkowitych).

Dla zminimalizowania sumarycznego kosztu opracowano i testowano następujące metody:

- GA algorytm zachłanny z losowaną kolejnością przedziału zadań,
- HLN algorytm węgierski programowania liniowego.

W obu przypadkach dokonywana jest generacja kolumnowa przydziału w zadanym okresie, tj. wg kolejności dni. Dla obydwu metod wyliczana macierz kosztów przydziału dla każdego dnia uwzględnia przydział zadań w dniach poprzedzających i następnych całego okresu rozliczeniowego, tj. projektowanego planu.

Algorytm zachłanny GA zaproponowano wg następującego schematu:

A. Lista służb jest układana w losowej kolejności. Do układania wykorzystuje się standardowy algorytm, który wybiera losowo jedno z $n!$ możliwych uporządkowań (gdzie n jest liczbą służb/zadań na liście).
B. Dla każdego dnia, od pierwszego dnia ujętego w projekcie kolejno do ostatniego, wykonywane są następujące działania:
   a) Służby przypisywane są w kolejności określonej w punkcie A. Dla każdej służby obliczany jest koszt przypisania jej do każdego z kierowców.
   b) Jeżeli w dowolnym momencie wyliczony koszt przypisania wynosi zero, pomija się obliczenia dla pozostałych kierowców (żaden inny koszt nie może być niższy).
   c) Dokonywane jest przypisanie, które jest obciążone najniższym kosztem. W przypadku gdy więcej niż jedno przypisanie ma ten sam koszt, wybierane jest to, w którym kierowca ma wyższą pozycję na liście kierowców.
C. Całkowity koszt znalezionego w ten sposób rozwiązania wyliczany jest jako suma kosztów poszczególnych przypisań.

D. Kroki A–C powtarzane są zadaną liczbę iteracji, po czym jako ostateczne wybierane jest rozwiązanie, które było obciążone najmniejszym kosztem całkowitym.

Algorytm HLN oparto na tzw. metodzie węgierskiej programowania liniowego [1, 2]. Dla każdego dnia wyznaczana jest macierz kosztów Aij przydziału zadań. Algorytm HLN pozwala bardzo szyko wyznaczyć macierz permutacji Xij przydziału zadań w dniu. Klasyczny algorytm węgierski bazuje jednak na przypisaniu jeden-do-jeden, tzn. jeden kierowca-jedno zadanie, co uniemożliwia przydział wielu zadań dla kierowcy w dniu.

Ponadto każda optymalizacja opierająca się na wyliczaniu macierzy kosztów ma problem z takim obsadzaniem służb, by służby przypisane do danego kierowcy nie kolidowały ze służbami przypisanymi do jego zmienników. Wynika to z faktu, że kolidowanie może zostać uwzględnione w macierzy kosztów dopiero po przypisaniu przynajmniej jednej służby, a optymalizacja liniowa zakłada, że cała macierz znana jest przed dokonaniem jakiegokolwiek przypisania.

Rozwiązanie problemu zmienników i możliwości przypisania wielu zadań jednemu kierowcy zostało rozwiązane przez modyfikację alogorytmu HLN w kolejnych pracach autora.

W tabelach 2 i 3 przedstawiono wybrane wyniki eksperymentów optymalizacyjnych. Wynik optymalizacji w istotny sposób zależy od zbioru uprzednio przygotowanych, zdefiniowanych zadań transportowych. Nie rozpatrywano tutaj zagadnienia optymalizacji budowy samych zadań. Wstępne testy algorytmów optymalizacyjnych przeprowadzono dla rzeczywistych baz danych z przedsiębiorstw transportowych.

Tabela 2. Test 1 – Baza Białystok

Optymalizacja	Koszt całkowity	Czas obliczeń (min:s)
GA, 10 iteracji (średnia z 3 prób)	347 542 934	1:56
GA, 20 iteracji (średnia z 3 prób)	347 548 210	3:44
HLN	358 071 138	0:11

Kierowców: *229*
Dni: *30 (kwiecień)*
Służb: *168–170 (dni powszednie), 49 (soboty), 37 (niedziele i święta)*

Tabela 3. Test 2 – Baza Tarnobrzeg

Optymalizacja	Koszt całkowity	Czas obliczeń (min:s)
GA, 10 iteracji (średnia z 5 prób)	11 417 244	1:27
GA, 20 iteracji (średnia z 5 prób)	11 417 722	2:53
HLN	11 422 110	0:07

Kierowców: *208*
Dni: *28 (luty)*
Służb: *143–144 (dni powszednie), 80–83 (soboty), 39–42 (niedziele)*

W tabeli 3 dla algorytmu łapczywego GA podano wyniki (koszt i czas) jako średnią z 5 eksperymentów, przy czym dla każdego z eksperymentów uzyskano bardzo niewielki rozrzut wyników poniżej 0,1%. Niska liczba iteracji dla algorytmu łapczywego pozostawała bez praktycznego wpływu na wynik kosztów, znacząco zwiększając jednak czas obliczeń.

Podsumowanie

Wyniki testów przeprowadzonych na rzeczywistych danych z przedsiębiorstw transportowych wykazują dużą efektywność zaproponowanych algorytmów optymalizacji przydziału zadań zarówno pod względem czasu automatycznej realizacji planu, jak i uzyskiwanych efektów minimalizacji kosztów.

Zastosowany algorytm programowania liniowego wykazuje wielokrotnie krótsze czasy realizacji w porównaniu z tzw. algorytmem łapczywym, przy porównywalnych wartościach funkcji kosztów, co kwalifikuje go do dalszych prac badawczych celem rozwiązania problemu zmienników oraz możliwości przydziału wg relacji jeden-do-wielu.

Liczne warunki dodatkowe dotyczące czasu pracy, a wynikające z norm prawnych powinny zostać uwzględnione i rozszerzone w badaniach symulacyjnych optymalizacji z ograniczeniami [3].

Bibliografia

1. Caprara A, Fischetti M., Toth P., Vigo D., Guida P.L.: Algorithms for railway crew management. Technical report, DEIS, University of Bologna, Italy, DMI, Univerity of Udine, Italy, Ferrovie dello Stato SpA, Italy, June 1997.

2. Filipowicz B.: Badania Operacyjne. Wybrane metody obliczeniowe i algorytmy. FHU Poldex, Kraków 1997.
3. Ftulis S.G., Giordano M., Pluss J.J., Vota R.J.: Rule-based constrains programming: application to crew assignment. Expert Systems with Applications 15 (1998) 77–85.
4. Freling R., Lentink R., Odijk M.: Scheduling train crews: a case study for the Dutch Railways. Econometric Institute, Erasmus University Rotterdam, Econometric Institute Report EI2000-17/A.
5. Kisielewski P.: Komputerowy system planowania i dyspozycji dużą flotą pojazdów. IX Warsztaty Naukowe PTSK, Koszalin-Osieki 2002.
6. Kisielewski P., Komputerowe wspomaganie planowania pracy kierowców dużej floty pojazdów. Konferencja TRANSCOMP 2002, Politechnika Radomska, Radom – Zakopane 2002.
7. Kisielewski P.: Kolejowy system komputerowego wspomagania dyspozycji zadań trakcyjnych. Zeszyty Naukowe Politechniki Śląskiej; Seria Transport, Katowice 2004.
8. Optycom Ltd.: Rail crew scheduling, rostering and management. Optycom Ltd. 2001.
9. Ross C., Wren A.: Greedy genetic algorithms, optimizing mutations and bus driver scheduling. Computer-Aided Transit Scheduling, number 430 in Lecture Notes in Economics and Mathematical Systems Springer, 1993, 213–235.

Recenzent:
Wojciech BĄKOWSKI

Optimization of transportation task assignment

Key words

Transportation tasks, plan of task assignment, diver/crew rostering, optimization of task assignment.

Summary

The problem of assigning transportation tasks to drivers, while minimizing the operational costs of a large vehicle fleet, is presented in the paper. In a mathematical sense the issue presents a so called 'optimal set covering problem' taking into consideration constraints resulting from task characteristics, company profile and numerous law regulations. The stages of the assignment process have been presented with particular emphasis on the optimization problem. Two optimization methods are proposed and the results of their tests are presented on real data from existing large urban transportation companies.