Uniwersytet Ekonomiczny w Katowicach

Studia Ekonomiczne. Zeszyty Naukowe
Uniwersytetu Ekonomicznego w Katowicach
ISSN 2083-8611 Nr 235 · 2015

Jakub Witkowski

Szkoła Główna Handlowa w Warszawie
Kolegium Analiz Ekonomicznych
Instytut Ekonometrii
Zakład Wspomagania i Analizy Decyzji
jzmwitkowski@gmail.com

Bogumił Kamiński

Szkoła Główna Handlowa w Warszawie
Kolegium Analiz Ekonomicznych
Instytut Ekonometrii
Zakład Wspomagania i Analizy Decyzji
bkamins@sgh.waw.pl

Wit Jakuczun

WLOG Solutions
wit.jakuczun@wlogsolutions.com

DOBÓR OPTYMALNEJ TARYFY TELEKOMUNIKACYJNEJ PRZY UŻYCIU PROGRAMOWANIA W LOGICE Z OGRANICZENIAMI

Streszczenie: Praca opisuje algorytm optymalizacyjny rozwiązujący w efektywny sposób problem wyboru optymalnej taryfy w telefonii komórkowej. Ze względu na bardzo dużą liczbę możliwości łączenia usług telekomunikacyjnych w taryfy rozważany problem optymalizacyjny jest złożonym nieliniowym zagadnieniem programowania kombinatorycznego. W niniejszej pracy pokazujemy, że tego typu zadanie może zostać efektywnie rozwiązane przy pomocy programowania w logice z ograniczeniami (*constraint logic programming*). Wykorzystanie takiego podejścia dodatkowo pozwala na stworzenie modelu, który może być łatwo modyfikowany. Zapewnia to możliwość jego łatwego wykorzystania w praktyce biznesowej, gdzie składowe taryf telekomunikacyjnych podlegają częstym zmianom.

Słowa kluczowe: programowanie w logice z ograniczeniami, optymalizacja doboru taryfy, optymalizacja kombinatoryczna.

Wprowadzenie

Zwiększanie się konkurencji na rynku telekomunikacyjnym doprowadziło do spadku cen usług, jak i rozszerzenia ofert operatorów o nowe taryfy i usługi. Jednym ze sposobów mającym przyciągnąć nowych klientów było oferowanie przez operatora gwarancji indywidualnego doboru optymalnej taryfy na podsta-

wie bilingów klienta. W takim przypadku, na koniec okresu rozliczeniowego, operator nalicza opłaty na podstawie taryfy, w której wykorzystane przez klienta usługi kosztowały najmniej. Ponieważ oferowanych taryf (wraz z różnymi usługami dodatkowymi) jest bardzo wiele, stanowi to skomplikowany kombinatoryczny problem optymalizacyjny. Do rozwiązania takiego problemu potrzebny jest algorytm *gwarantujący* dobór najlepszej, z punktu widzenia konsumenta, taryfy.

Poza dużą liczbą dostępnych taryf i usług, problem jest także komplikowany przez konieczność uwzględnienia kolejności w czasie wykonywanych przez klienta połączeń oraz wzajemne powiązania pomiędzy dostępnymi usługami. Z tego względu w modelu pojawia się bardzo wiele ograniczeń, a funkcja celu często staje się nieliniowa.

Dotychczas w literaturze pojawiło się wiele propozycji wykorzystania metod optymalizacji w problemach doboru taryfy. Część prac zajmuje się maksymalizacją zysku operatora – np. praca [Bouhtou, Erbs i Minoux, 2007] opisuje wykorzystanie w tym celu programowania dwupoziomowego (*bilevel programming*) oraz proponuje model programowania całkowitoliczbowego i metody jego rozwiązania (zwykle problem jest za duży dla zwykłych metod rozwiązywania tego typu problemów, dlatego też wykorzystywane są techniki, które skracają czas obliczeniowy, ale nie gwarantują osiągnięcia optimum globalnego). W pracy [Bouhtou, Hoesel, Kraaaij i Lutton, 2007] zaproponowano modelowanie problemu doboru taryf jako sieci w celu maksymalizacji zysku operatora. Do znalezienia optymalnego zestawu taryf wykorzystano metody programowania całkowitoliczbowego oraz metody podziału i ograniczeń (*branch and bound)*. Do rozwiązywania problemu doboru taryfy w telekomunikacji z perspektywy maksymalizacji zysku operatora wykorzystywano także metody heurystyczne [por. Schlereth, Stepanchuk i Skiera, 2010], takie jak np. symulacyjne wyżarzanie czy poszukiwanie stochastyczne. Problem doboru stawek taryfowych w zależności od stopnia wykorzystania zasobów sieciowych przez użytkowników został poruszony m.in. w [Bouhtou, Medori i Minoux, 2011]. W pracy [Pytlak i Stecz, 2007] rozwiązywany problem polega na minimalizacji kosztu dla klienta. Zagadnienie zostało sprowadzone do postaci, gdzie znajdują się tylko dwa typy opłat: stały miesięczny abonament zapewniający pewną liczbę darmowych minut oraz zmienna stawka, po której są wyceniane minuty przekraczające podstawowy limit. Takie podejście do problemu pozwala na rozwiązanie go za pomocą metod liniowego programowania całkowitoliczbowego. W pracy [Pytlak i Stecz, 2014] podobny problem jest także rozwiązywany (poza metodami opartymi na programowaniu całkowitoliczbowym) za pomocą metod programowania w logice z ograniczeniami.

W tej pracy rozważany problem polega na optymalizacji doboru taryfy i usług przez operatora, tak aby cena płacona przez klienta była jak najniższa (gwarancja najniższej ceny). Do rozwiązania tego problemu wykorzystywany jest algorytm oparty na programowaniu w logice z ograniczeniami (*constrained logic programming*, por. np. [Marriott i Stuckey, 1998]). Podejście to gwarantuje, że znalezione rozwiązanie będzie optymalne, a co za tym idzie, może być wykorzystane do rozwiązania rozpatrywanego problemu. Proponowany algorytm znajduje rozwiązanie poprzez efektywne poruszanie się po binarnym drzewie poszukiwania, rozpatrując jedynie dopuszczalne rozwiązania oraz zawężając zbiór przeszukiwanych rozwiązań dzięki technice podziału i ograniczeń [por. [Land i Doig, 1960; Apt, 2003]. Do przyśpieszenia działania algorytmu wykorzystywane jest przyrostowe obliczanie wartości funkcji celu oraz wykorzystanie leniwej ewaluacji przy narzucaniu ograniczeń (dokładne wyjaśnienie tych mechanizmów znajduje się w trzeciej części artykułu). Dodatkowo algorytm jest bardzo elastyczny i łatwo jest w nim dodać kolejne ograniczenia, co jest bardzo ważne w tak dynamicznie zmieniającej się branży jak telekomunikacja.

Celem tego artykułu jest przedstawienie skutecznego wykorzystania metod programowania w logice z ograniczeniami do doboru optymalnej taryfy do bilingów klienta. Pierwsza część zawiera opis rozwiązywanego problemu, druga objaśnia działanie algorytmu, a trzecia pokazuje jego implementacje. Czwarta część przedstawia porównanie algorytmu z innymi metodami zapewniającymi znalezienie optimum.

1. Opis problemu

Obliczenie kosztu obciążającego klienta za dany okres rozliczeniowy odbywa się na podstawie trzech czynników: danych bilingowych, dostępnych taryf oraz możliwych do wyboru usług opcjonalnych oferowanych przez operatora.

Dla uproszczenia, na potrzeby tej pracy, przez dane bilingowe będziemy rozumieć zbiór wykonanych przez klienta połączeń w okresie rozliczeniowym. Tak więc zbiór ten będzie zawierał informacje o wykonywanych połączeniach oznaczanych jako p_k, gdzie k należy do $1 \ldots K$. Zbiór wszystkich możliwych sekwencji połączeń będzie oznaczany jako $\boldsymbol{P}$, zaś $\boldsymbol{p}$ będzie wektorem takim, że $\boldsymbol{p} = (p_1, p_2, \ldots p_K)$. Każde p_k posiada dwie cechy: czas trwania połączenia (dodatnia liczba rzeczywista, oznaczana jako d_k) oraz atrybuty charakteryzujące połączenie (a_k). Atrybuty opisują szczegółowo każde połączenie, zawierając dokładne informacje o jego typie. Przykładowo połączenie może być opisane

jako wychodzące, międzynarodowe i wykonywane w godzinach nocnych. Na podstawie tych atrybutów możliwe jest dopasowanie odpowiednich usług wyceniających do połączeń. Ważną cechą każdego połączenia jest kolejność, w jakiej było wykonywane (co jest zawarte w subskrypcie k) ze względu na sposób wycenienia połączeń przez usługi.

Do wyceny połączeń operator wykorzystuje taryfy, czyli plany, na podstawie których wyceniane są połączenia, oznaczane jako t_i, gdzie i należy do $(1 \dots T)$. Zbiór wszystkich taryf jest oznaczany jako $\boldsymbol{T}$. Możliwe jest korzystanie z jednej taryfy naraz. W ramach taryfy można korzystać z pewnej gamy opcjonalnych usług.

Opcjonalnie usługi będą oznaczane jako u_j $(j = 1, \dots, U)$. Ponieważ każda z usług może być w jednym z dwóch stanów (aktywna/nieaktywna), dlatego też $\boldsymbol{u}$ będzie wektorem binarnym oznaczającym, które usługi są aktywne. Każda usługa jest opisana przez koszt aktywacji i parametry (np. priorytet, relacje z innymi usługami). Obie cechy są zależne od taryfy, w jakiej usługa jest oferowana, i innych usług oferowanych w taryfie. Na przykład w różnych taryfach różne usługi mogą znajdować się w grupach typu „dokładnie jedna usługa z wielu", ponadto w związku z wymaganiami biznesowymi oraz polityką operatora nie każda usługa jest dostępna w każdej taryfie, jak również różne są relacje pomiędzy poszczególnymi usługami.

Relacje pomiędzy taryfami a usługami, z których wynikają podstawowe ograniczenia w modelu optymalizacyjnym, są następujące:

- usługi wymagane w danej taryfie,
- usługi wykluczane przez daną taryfę,
- usługi wymagane przez daną usługę,
- usługi wykluczane przez daną usługę,
- zbiór usług, z których dokładnie jedna może być wybrana,
- zbiór usług, z których dokładnie jedna musi być wybrana.

Cena C, którą musi zapłacić klient, jest funkcją wykonanych połączeń ($\boldsymbol{p}$), taryfy (t) oraz wybranych usług ($\boldsymbol{u}$). Warto zauważyć, że w funkcji $C(\boldsymbol{p}, t, \boldsymbol{u})$ nie każda usługa u_j może być zastosowana do wyceniania każdego połączenia p_k. Dzieje się tak z dwóch względów: po pierwsze, dana usługa może nie dotyczyć danego połączenia (np. za pomocą usługi „darmowe połączenia w weekend" nie można wyceniać połączenia wykonywanego w dzień powszedni), o czym decydują cechy połączenia p_k, zarówno jego długość, jak i atrybuty. Drugi powód dotyczy usług limitowanych (np. 100 darmowych minut) – są one używane do wyceny tych połączeń, które zostały wykonane wcześniej (w kolejności wykonywania) i nie przekraczają limitu wartościowego usługi.

Proces wyceny połączeń (informacje o wycenionych połączeniach przechowuje wektor $\boldsymbol{b}$) za pomocą usług jest następujący: aktywne usługi są wykorzystywane do wyceny połączeń (tylko tych, które spełniają ich wymagania, zależnie od zgodności atrybutów usługi i połączenia; połączenia możliwe do wyceny przez usługę są zapisane w wektorze $\boldsymbol{b'}$) sekwencyjnie, wg priorytetu usługi (pozycji usługi u_j w wektorze usług $\boldsymbol{u}$). Przykładowo najpierw aktywna usługa u_1 jest używana do wyceny wszystkich połączeń, które spełniają jej kryteria, a następnie kolejne usługi z wektora $\boldsymbol{u}$ są kolejno używane do dalszej wyceny. Jeżeli połączenie p_k zostało wycenione przez usługę u_i, to zostaje ono oznaczone jako wycenione i nie podlega wycenie przez inne usługi[1]. Wyliczona cena jest przechowywana w wektorze $\boldsymbol{c}$. Ceny są obliczane przyrostowo, tzn. wzrost ceny wynikający z użycia następnej usługi jest dodawany do wcześniej wyliczonych wartości; nie są one obliczane od początku dla każdej konfiguracji usług. Jeżeli po wykorzystaniu wszystkich usług pozostaną jeszcze niewycenione połączenia, zostają one wycenione na podstawie obowiązujących w taryfie zasad.

Algorytm 1: wycenianie połączeń

dla sekwencji $\boldsymbol{p}$ *zawierającej K połączeń, dla każdej* taryfy $i = 1 \ldots T$ {

$\boldsymbol{b_i} = [0]_{1xK}$ $C(\boldsymbol{p}, t, \boldsymbol{u}) = c_i^s$

  dla każdej usługi u_j $j = 1 \ldots U$ {

    jeżeli $u_j = 1$ {

niech $\boldsymbol{b'} \in \{0,1\}^K$ *zawiera połączenia, które może wyceniać* u_j

      dla każdego połączenia takiego, że $\boldsymbol{b'} + \boldsymbol{b_i} < 2$ {

        $C(\boldsymbol{p}, t, \boldsymbol{u}) = C(\boldsymbol{p}, t, \boldsymbol{u}) + c_{ij}$

        $\boldsymbol{b_i} = \boldsymbol{b_i} + \boldsymbol{b'}$

      }}}

  dla każdego niewycenionego połączenia $\boldsymbol{b'} = [1]_{1xK} - \boldsymbol{b_i}$ {

    $C(\boldsymbol{p}, t, \boldsymbol{u}) = C(\boldsymbol{p}, t, \boldsymbol{u}) + c_i^s$ }

  zwróć $C(\boldsymbol{p}, t, \boldsymbol{u})$ }

Połączenie w i-tej taryfie przy wykorzystaniu usługi s_j oznaczane jest jako c_{ij}, a cena połączenia wg zasad taryfy i jako c_i^z. Stała opłata związana z taryfą jest oznaczana jako c_i^s. Podsumowując, proces wyceniania bilingu pokazuje algorytm 1.

---

[1] W rzeczywistości możliwa jest sytuacja, gdy dwie usługi powinny wyceniać jedno połączenie (np. w sytuacji, gdy połączenie przekracza limit wartościowy danej usługi). Takie połączenie powinno zostać w takim wypadku podzielone na części. W tej pracy założono, że taki proces odbywa się automatycznie, a jego implementacja została opisana w części trzeciej.

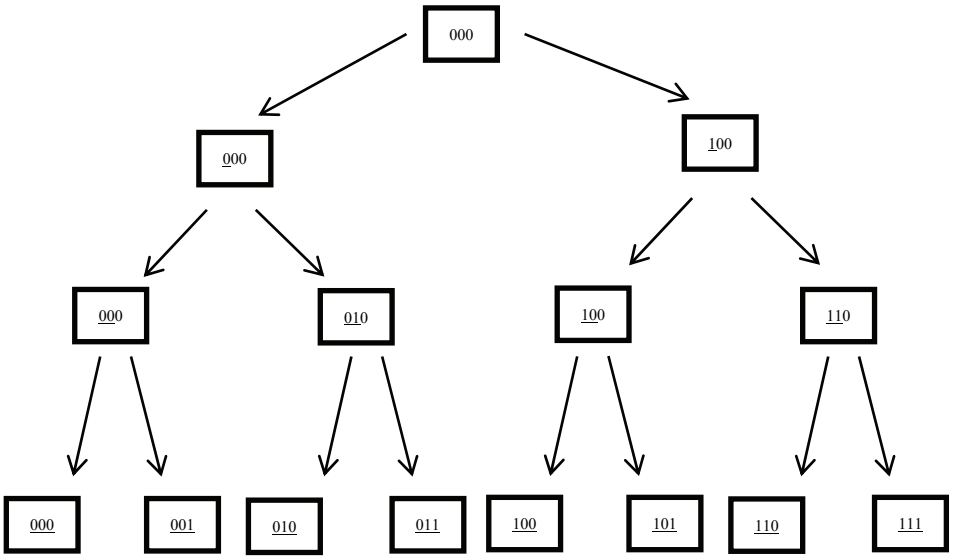
2. Opis algorytmu

Celem działania algorytmu jest znalezienie takiej kombinacji taryfy i usług przy zadanych połączeniach, aby zminimalizować cenę płaconą przez klienta. Liczba wszystkich możliwych rozwiązań wyraża się więc wzorem $2^U T$, gdzie T jest liczbą taryf, a U liczbą dostępnych usług. Jak łatwo zauważyć, wraz ze wzrostem liczby taryf i usług liczba wszystkich rozwiązań rośnie bardzo szybko (dla 2 taryf i 5 usług mamy 64 rozwiązania, a dla 4 taryf i 10 usług już 4096). W rzeczywistości liczba oferowanych taryf i usług jest bardzo duża, a co za tym idzie, prosta metoda obliczania ceny dla wszystkich możliwych kombinacji jest bardzo czasochłonna i nieefektywna.

Aby zaprojektować algorytm znajdujący optymalną taryfę (w czasie umożliwiającym jego zastosowanie biznesowe), należy w efektywny sposób przeszukiwać rozwiązania. Można to zrobić, opierając się na drzewie przeszukiwania. Drzewo to posiada U poziomów (poziom 0 to korzeń). Na każdym poziomie algorytm podejmuje decyzję o aktywacji danej usługi (tzn. przy poruszaniu się z poziomu j na poziom $j+1$ zostaje podjęta decyzja o aktywacji j-tej usługi). Na każdym poziomie drzewa znajdują się wierzchołki N, w których przechowywane są informacje potrzebne do obliczania ceny. Przede wszystkim są to dane o usługach – informacje o aktualnym stanie aktywacji każdej z usług (binarny wektor $\boldsymbol{u}(N)$) oraz o tym, o aktywacji której usługi zapadała decyzja w tym wierzchołku (poziom drzewa $h(N)$). Binarny wektor $\boldsymbol{r}(N)$ przechowuje zaś informacje o taryfach. Wektor ten ma wymiary $1xT$, gdzie T jest liczbą rozpatrywanych taryf. Wartości w wektorze oznaczają dostępność taryf w danym wierzchołku. Początkowo dostępne są wszystkie taryfy, jednak wraz z poruszaniem się po drzewie część taryf może stać się niedostępna ze względu na powiązanie taryf z usługami (pewne usługi mogą być niedopuszczalne/wymagane w pewnych taryfach). Do każdego wierzchołka przyporządkowane są także informacje używane do obliczania ceny. W wektorze $\boldsymbol{c}(N)$ przechowywane są wartości cen obliczane wg algorytmu 1. Bazując na tym algorytmie w korzeniu, $\boldsymbol{c}(N)$ przechowuje wartości stałej opłaty abonamentowej, a wraz z poruszaniem się po drzewie jest modyfikowany o koszt wynikający z aktywowania i użycia danej usługi do wyceny połączeń. Informacje o wycenionych przez usługi połączeniach są zawarte w wektorze $\boldsymbol{b}(N)$ składającym się z wektorów $\boldsymbol{b}_i$ z algorytmu 1.

Wektory opisujące cechy drzewa aktualizują się wraz z przejściami między wierzchołkami. Wektor $\boldsymbol{u}(N)$ jest aktualizowany wg następującej konwencji: każdy wierzchołek dzieli się na dwoje potomków, w prawym z nich dodatkowa usługa jest aktywowana, zaś w lewym dana usługa nie zostanie aktywowana (tak

więc na poziomie $h(N)$ i lewym potomku na poziomie $h(N)+1$ aktywne będą te same usługi). Przykładowo z pierwszego wierzchołka drzewa ($h(N)=0$) odchodzić będą dwa wierzchołki. W lewym żadna z usług nie będzie aktywna (wektor $\boldsymbol{u}(N)$ nie zmienia się), zaś w prawym usługa u_1 będzie aktywna, a pozostałe nieaktywne (pierwsza pozycja w wektorze $\boldsymbol{u}(N)$ ulegnie zmianie). Proces aktywowania usług przedstawia rys. 1.



Rys. 1. Drzewo przeszukiwania dla trzech usług. Liczby w liściach oznaczają stan aktywacji poszczególnych usług. Początkowo wszystkie trzy usługi są nieaktywne, ale wraz ze schodzeniem na kolejne poziomy drzewa część z nich jest aktywowana

Wektor $\boldsymbol{r}(N)$ zmienia się w trakcie przechodzenia między wierzchołkami w następujący sposób: przechodząc z j-tego poziomu na poziom $j+1$ do lewego wierzchołka wszystkie taryfy, które wymagają usługi $u_{(j+1)}$, zostają uznane za niedopuszczalne. W przypadku prawego wierzchołka za niedopuszczalne zostają uznane te taryfy, które wykluczają aktywację $u_{(j+1)}$. Ponadto poruszanie się pomiędzy wierzchołkiem N a jego potomstwem jest tylko możliwe, jeżeli taki ruch nie powoduje powstania niedopuszczalnych kombinacji usług. Wartości wektora $\boldsymbol{c}(N)$ także są przeliczane wraz z poruszaniem się po wierzchołkach drzewa, w każdym obliczana jest cena wynikająca z aktywacji usług wyceniających połączenia (połączenia niewycenione przez usługi są wyceniane w liściach drzewa przy użyciu zasad taryfy). W momencie, kiedy osiągnięty zostanie liść drzewa, najniższa cena w nim zostaje aktualnym minimum. Po osiągnięciu kolejnego liścia w przypadku, gdy cena jest mniejsza od aktualnego minimum,

zostaje ono zaktualizowane. Taki sposób przechowywania osiągniętego minimum pozwala na znaczne zmniejszenie liczby koniecznych obliczeń – jeżeli w pośrednim wierzchołku N cena dla taryfy t jest większa lub równa aktualnemu minimum, to taryfa ta zostaje usunięta z dalszych obliczeń we wszystkich wierzchołkach wywodzących się z wierzchołka N (taki zabieg jest możliwy, ponieważ wszystkie ceny są nieujemne).

Warto podkreślić, że wraz z przejściem od jednego wierzchołka do drugiego obliczany jest jedynie przyrost ceny wynikający z aktywacji (lub jej braku) danej usługi, co pozwala na skrócenie czasu obliczeń, ponieważ nie jest konieczne obliczanie ceny dla każdego wierzchołka od nowa, z uwzględnieniem wszystkich usług z poprzednich wierzchołków. Zwykle oszczędność czasu obliczeniowego jest znaczna, ponieważ w sytuacji, gdy w wektorze $\boldsymbol{p}$ znajduje się wiele połączeń, obliczenia mogą być czasochłonne.

3. Implementacja modelu w programowaniu logicznym z ograniczeniami

Do implementacji opisanego algorytmu zostało wybrane środowisko ECL^{i}PS^e w wersji 6.10 [por. Schimpf i Kish, 2011]. Środowisko to jest oparte na języku programowania Prolog oraz posiada rozbudowany moduł programowania w logice z ograniczeniami (w pracy wykorzystana została biblioteka IC). Implementacja algorytmu została podzielona na trzy etapy: zdefiniowanie zmiennych, nałożenie ograniczeń oraz szukanie rozwiązań.

Poza algorytmem opisanym w artykule zaimplementowany został także algorytm przeszukujący wszystkie dopuszczalne rozwiązania (jako punkt odniesienia dla skuteczności algorytmu i jego technik przeszukiwania).

Do kontrolowania wykorzystania usług do wyceny połączeń wykorzystane zostały globalne ograniczenia (nakładane na zmienną opisującą zużycie czasu połączeń dla każdej kombinacji usług i połączeń). Aby efektywnie nakładać takie ograniczenia, wykorzystana została technika leniwej ewaluacji (po każdej aktywacji/dezaktywacji usługi zmiana w wykorzystaniu połączeń była obliczana przyrostowo), co znacznie skróciło czas potrzebny na obliczenia [por. Apt i Wallace, 2007].

4. Wyniki eksperymentów

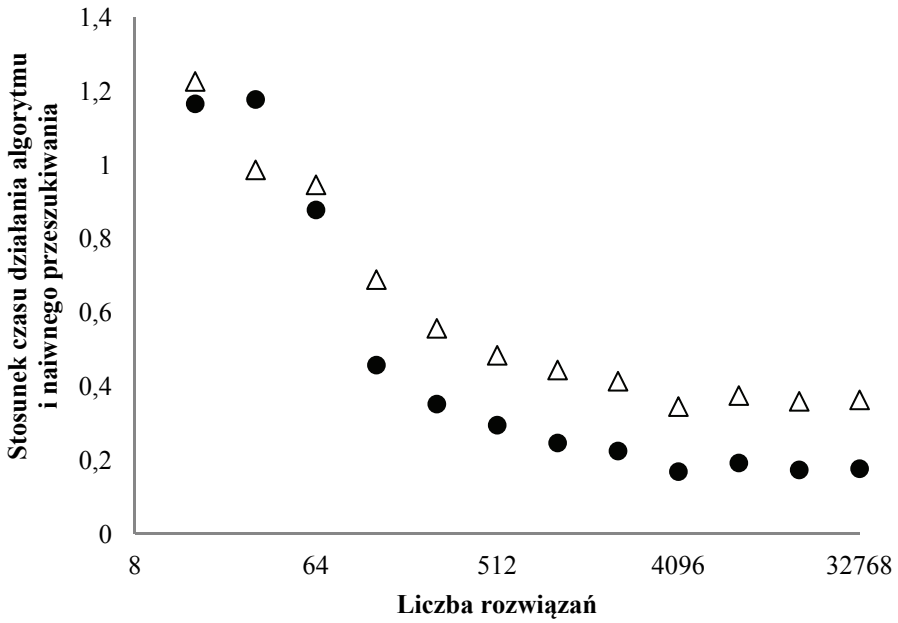
Wykorzystanie przedstawionego algorytmu do rozwiązania rzeczywistych problemów (kombinacje 10 taryf i 30 usług dla ponad miliona klientów) pokazało, że jest on znacznie bardziej skuteczny od prostych metod przeszukiwania wszyst-

kich rozwiązań. W tym artykule zaprezentowane zostaną wyniki[2] działania algorytmu dla zanimizowanych danych dotyczących jednego bilingu składającego się z 4055 połączeń.

Aby w pełni ocenić skuteczność algorytmu, powinien on zostać porównany z innymi metodami. Grupa metod, które mogą służyć za punkt odniesienia, wydaje się dość duża, ale muszą być to metody, które gwarantują odnalezienie minimum globalnego, a nie lokalnego. Dlatego też nie mogą to być algorytmy heurystyczne (opisane np. w [Schlereth, Stepanchuk i Skiera, 2010]). Znalezienie optimum gwarantują metody programowania całkowitoliczbowego, ale opisywany problem jest zbyt skomplikowany, aby można było bezpośrednio używać tego typu algorytmów, możliwe jest jednak ich zastosowanie, jak np. w [Martin, 1999] albo [Pytlak i Stecz, 2007]. Jednakże metody z tej rodziny są wysoce nieelastyczne, co znacznie obniża ich efektywność wraz ze zwiększaniem się liczby ograniczeń. Ze względu na to zdecydowaliśmy się porównywać przedstawiony algorytm do prostej metody przeszukiwania wszystkich rozwiązań, opartej na dostępnej w ECL^{i}PS^e funkcji *findall*, która przeszukuje wszystkie możliwe rozwiązania i co za tym idzie, pozwala na znalezienie optymalnego rozwiązania.

Aby można było zbadać, w jaki sposób złożoność problemu wpływa na skuteczność algorytmu, rozważone zostały przypadki z różnymi liczbami możliwych rozwiązań. Liczba rozwiązań zmieniała przez modyfikację zestaw danych z jedną taryfą i pewną liczbą usług. Początkowo dostępne były trzy usługi, a w kolejnych zestawach danych ta liczba zwiększała się. Wyniki testów przedstawia rys. 2. Metoda naiwnego przeszukiwania jest tu zaimplementowana w dwóch wariantach: w pierwszym (nazwanym dalej metodą pierwszą), prostszym, obliczane są wszystkie możliwe rozwiązania, a następnie sprawdzana jest ich dopuszczalność, zaś w drugim (nazwanym dalej metodą drugą) najpierw stosowane są ograniczenia i wykorzystana jest metoda przeszukiwania drzewa z proponowanego algorytmu, jednak nie jest wykorzystywane inkrementalne obliczanie funkcji celu.

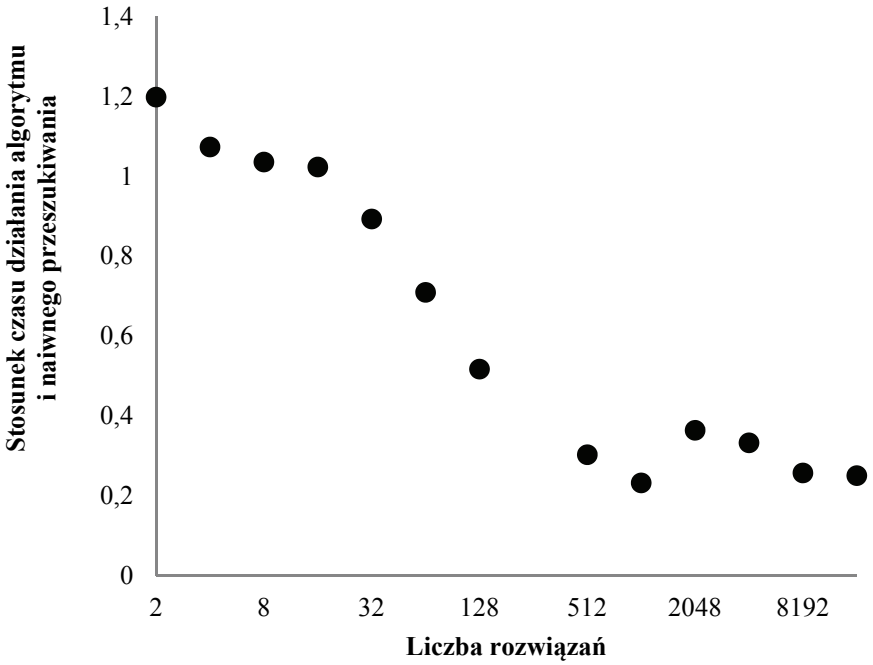
[2] Wszystkie obliczenia zostały przeprowadzone na komputerze o następujących parametrach: procesor Intel i7, 8 GB pamięci RAM.



Rys. 2. Porównanie czasu działania algorytmu z metodą pierwszą (trójkąty) i z metodą drugą (koła)

Dla problemów z małą liczbą możliwych rozwiązań wszystkie testowane metody zachowują się podobnie, jednakże wraz ze złożonością problemu rośnie efektywność algorytmu w stosunku do metod naiwnego przeszukiwania wszystkich rozwiązań. Jak widać, zastosowanie technik inkrementalnego obliczania funkcji celu, leniwej ewaluacji oraz przycinania drzewa przeszukiwania pozwala na znaczne oszczędności w czasie obliczeniowym potrzebnym do rozwiązania problemu.

Ciekawa wydaje się obserwacja niskiej efektywności algorytmu w przypadku małych problemów, jednak jej przyczyna leży nie tyle w konstrukcji algorytmu, lecz po stronie oprogramowania, w którym został on zaimplementowany.



Rys. 3. Porównanie czasu działania algorytmu i metody drugiej (w modelu z dodatkowym ograniczeniem)

Dodatkowo, w celu poszerzenia zakresu testów do standardowych relacji między usługami, został dodany jeszcze jeden typ relacji: „wybór dokładnie dwóch usług z zadanego zbioru". Wyniki porównania między proponowanym algorytmem a metodą drugą (nie zostaje uwzględniona w tym przypadku metoda pierwsza, ponieważ wcześniejsze obliczenia pokazały, że czas działania obu metod zmienia się według tej samej tendencji) przedstawia rys. 3. Wprowadzenie dodatkowego ograniczenia nie zmienia skuteczności algorytmu, w dalszym ciągu pozwala on na znaczne zmniejszenie ilości czasu potrzebnego do obliczeń.

Podsumowanie

Artykuł prezentuje ideę algorytmu rozwiązywania problemu doboru optymalnej taryfy dla klienta, wdrożonego przez jednego z operatorów telekomunikacyjnych.

W pracy zaproponowano sposób zapisu problemu i opracowano metodę pozwalającą na efektywne znalezienie rozwiązania. Dodatkowo zaprezentowana

została implementacja algorytmu w programowaniu w logice z ograniczeniami, co pozwala na wykorzystanie go do rozwiązywania realnych i złożonych problemów. Wysoka wydajność algorytmu wynika z przyrostowego obliczania funkcji celu, efektywnego przycinania drzewa rozwiązań oraz wykorzystania leniwej ewaluacji do nakładania ograniczeń. Ponadto implementacja w programowaniu w logice z ograniczeniami daje możliwość elastycznego wprowadzania zmian w algorytmie, co pozwala na łatwe dostosowanie algorytmu do zmieniającej się oferty operatorów telekomunikacyjnych.

Warto także podkreślić, że wiele mechanizmów wykorzystanych w algorytmie (leniwa ewaluacja, przyrostowe obliczanie funkcji celu, przycinanie drzewa poszukiwań) można zastosować w większości problemów zapisanych w programowaniu w logice z ograniczeniami. Mechanizmy te mogą być zastosowane w łatwy sposób do rozwiązywania wielu skomplikowanych problemów kombinatorycznych.

Literatura

Apt K.R. (2003), *Principles of Constraint Programming*, Cambridge University Press.

Apt K.R, Wallace M.G. (2007), *Constraint Logic Programming using ECLiPSe*, Cambridge University Press.

Bouhtou M., Erbs G., Minoux M**.** (2007), *Joint Optimization of Pricing and Resource Allocation in Competitive Telecommunications Networks*, „Networks", Vol. 50.

Bouhtou M., Hoesel S., Kraaaij A., Lutton J. (2007), *Tariff optimization in networks*, „INFORMS Journal on Computing", Vol. 19.

Bouhtou, M., Medori, J.R., Minoux, M. (2011), *Mixed Integer Programming model for pricing in telecommunication* [w:] J. Pahl, T. Reiners, S. Voß (eds.), *Network Optimization*, Springer, Berlin, Heidelberg.

Land A.H., Doig A.G. (1960), *An automatic method of solving discrete programming problems*, „Econometrica", Vol. 28(3).

Marriott K., Stuckey P.J. (1998), *Programming with Constraints*, The MIT Press.

Pytlak R., Stecz W. (2007), *Tariff optimization problem – formulation and algorithms* [w:] Korytkowski A., Mitkowski W., Szymkal M. (eds.) , 23rd IFIP TC 7 *Conference on System Modelling and Optimization Conference Materials*.

Pytlak R., Stecz W. (2014), *Models for solving the tariff optimization problem*, „Research in Logistics and Production", No. 2.

Martin R.K. (1999), *Large scale linear and integer optimization: a unified approach*, Kluwer Academic Publishers.

Schimpf J., Kish S. (2011), *ECLiPSe – from LP to CLP*, „Theory and Practice of Logic Programming", No. 12 (Special Issue on Prolog System).

Schlereth Ch., Stepanchuk T., Skiera B. (2010), *Optimization and Analysis of the Profitability of Tariff Structures with Two-Part Tariffs*, „European Journal of Operational Research", Vol. 206(3).

SOLVING THE OPTIMAL TELECOMMUNICATION RATE PLAN CONFIGURATION PROBLEM WITH CONSTRAINED LOGIC PROGRAMMING

Summary: We present an efficient algorithm that solves the telecommunication rate plan optimization problem. It is a complex and non-linear combinatorial programming task if we take into account realistic structures of offers available for mobile telephony subscribers. In the paper we show that constrained logic programming is an efficient approach to finding an optimal solution of this problem. Additionally, application of constrained logic programming allows us to formulate the problem in a simple way that provides a low-cost maintenance of the solution in practical applications when the rate plan structure often changes.

Keywords: rate plan configuration, constrained logic programming, combinatorial optimization.