

Praca magisterska, wersja robocza

Hubert Guzera

2015-09-02

Wprowadzenie

Wal-Mart, amerykański gigant handlowy, co godzinę umieszcza w swoich bazach danych 2.5 petabajtów danych, pochodzących z blisko miliona transakcji ([?]. I nie jest wyjątkiem - przeciętna ilość danych przechowywanych przez przedsiębiorstwa w Stanach Zjednoczonych jest większa niż zbiory Biblioteki Kongresu (szacowane na 235 terabajtów)([?]. W erze informacji większość z danych działań trafia na serwery tej bądź innej firmy, w formie historii transakcji, koordynatu GPS czy zdjęcia.

Często informacje te zbierane są przypadkiem - ze względu na prowadzenie rachunkowości, specyfikę świadczonych usług, lub też względy archiwizacyjne. Jednak wydobyte z nich *wiedzy* może stanowić źródło znaczącej przewagi konkurencyjnej. Jak wskazują Brynjolfsson, Hitt i Kim [?], przedsiębiorstwa podejmujące decyzje na podstawie analizy dużych zbiorów danych (*data driven decision making*) osiągają efektywność o 5-6 proc. większą niż grupa porównawcza. Mają także większy zwrot z kapitału i wycenę rynkową - jednym słowem, radzą sobie lepiej. Nic więc dziwnego, że coraz częściej data analytics staje się priorytetem wśród dużych spółek. Skalę popularności business intelligence unaczynia badanie PwC [?], według którego 44 proc. CEO planuje oparcie rozwoju firmy o inwestycje w tej dziedzinie.

Ale dzisiejsze zastosowania *big data* to tylko preludium do tego, co czeka nas w przyszłości. Trwający równolegle trend robotyzacyjny spowoduje, że w ciągu 20 lat w przedsiębiorstwie zamiast kierowców możemy zarządzać flotą autonomicznych pojazdów, a magazynierów zastąpią roboty. Fakt, że Google i Daimler już testują takie auta nie pozwala na nazwanie tego science-fiction. Według Carla Freya i Michela Osborne'a z Uniwersytetu w Oxfordzie [?], blisko 47 proc. miejsc pracy jest zagrożonych komputeryzacją. Większość z nich to zawody wykonujące rutynowe, mechaniczne czynności, ale postęp technologiczny powoduje, że na tej liście znajdują się też prace wymagające umiejętności kognitywnych i wnioskowania - jak pracownicy biurowi, analitycy czy operatorzy.

Jeśli więc w jednej strony mamy do czynienia z flotą autonomicznych pojazdów, z a drugiej z petabajtami informacji o tym gdzie i co kupują nasi klienci, możemy znaleźć się w sytuacji, gdzie koordynacja łańcucha dostaw będzie wykraczać poza możliwości człowieka. Dla komputera, wyprognozowanie popytu na podstawie danych i zaplanowanie dostaw nie będzie żadnym problemem.

Potwierdza to The McKinsey Global Institute [?], który wskazuje, że coraz częściej maszyny będą zastępować ludzi w podejmowaniu decyzji i brać udział w sterowaniu przedsiębiorstwem.

W teorii, ze względu na możliwość przeprowadzania złożonych obliczeń i analizy gigabajtów danych, decyzje te będą trafniejsze i poprawią efektywność przedsiębiorstwa.

Niniejsza praca ma na celu skonfrontowanie tej hipotezy. Po pierwsze, poprzez zaproponowanie jednego z wielu możliwych algorytmów optymalizacji działania przedsiębiorstwa poprzez wykorzystanie istniejących technik *modelowania predykcyjnego*. Po drugie przez sprawdzenie, jak tak podejmowane decyzje będą wpływać na funkcjonowanie przedsiębiorstwa i czy będzie ono funkcjonować efektywniej, niż gdyby zastosować w nim dotychczasowe praktyki biznesowe.

Spis treści

1	Cel, założenia i podstawy teoretyczne pracy	4
1.1	Koncepcja pracy	4
1.2	Podstawy teoretyczne	4
1.2.1	Przedsiębiorstwo jako system wieloagentowy	4
1.2.2	Zadanie optymalizacyjne	6
1.2.3	Modelowanie predykcyjne	7
1.3	Proponowany algorytm optymalizacyjny	10
2	Model	12
2.1	Koncepcja modelu	12
2.2	Zastosowane narzędzia	13
2.3	Struktura programu	14
2.4	Generowanie mapy	14
2.4.1	Algorytm wyszukiwania drogi	16
2.5	Agenci, ich rodzaje i właściwości	18
2.5.1	Konsumenci	18
2.5.2	Przedsiębiorstwo	19
2.5.3	Produkt	20
2.5.4	Konkurencja	21
2.5.5	Trasy	21
2.6	Symulowanie decyzji konsumenckich	21
3	Efekty działania algorytmu optymalizacyjnego	23
3.1	Charakterystyka badanego środowiska	23
3.2	Przewidywanie decyzji konsumentów	23
3.3	Wyniki przedsiębiorstwa przy braku optymalizacji	23
3.4	Optymalizacja przy stałych cenach i braku efektu skali	23
3.5	Optymalizacja przy stałych cenach i istnieniu efektu skali	23
3.6	Optymalizacja przy zmiennych cenach i istnieniu efektu skali	23

1 Cel, założenia i podstawy teoretyczne pracy

1.1 Koncepcja pracy

Praca ma na celu zaproponowanie algorytmu optymalizacji podejmowania decyzji w przedsiębiorstwie na podstawie modelowania predyktywnego oraz sprawdzenie, jak zaimplementowanie takiego algorytmu wpływa na wyniki firmy.

Przyjmując, że optymalizowane przedsiębiorstwo z sektora FMCG zajmuje zarówno produkcją, jak i dystrybucją towarów do sklepów detalicznych, będziemy starali się w prognozować w krótkiej perspektywie wolumen sprzedaży w każdym z prowadzonych sklepów. Wykorzystaną wiedzę wykorzystamy do optymalizacji procesów logistycznych, tj. zbudowanie takich tras dostaw i alokację wśród nich wolumenów produktów, żeby zysk firmy był jak największy.

W celu zaprezentowania wyniku działania powstałego w ten sposób algorytmu, zostanie zbudowany model wieloagentowy symulujący rynek i zachowania klientów. Z jego pomocą sprawdzimy funkcjonowanie algorytmu w trzech przypadkach założeń:

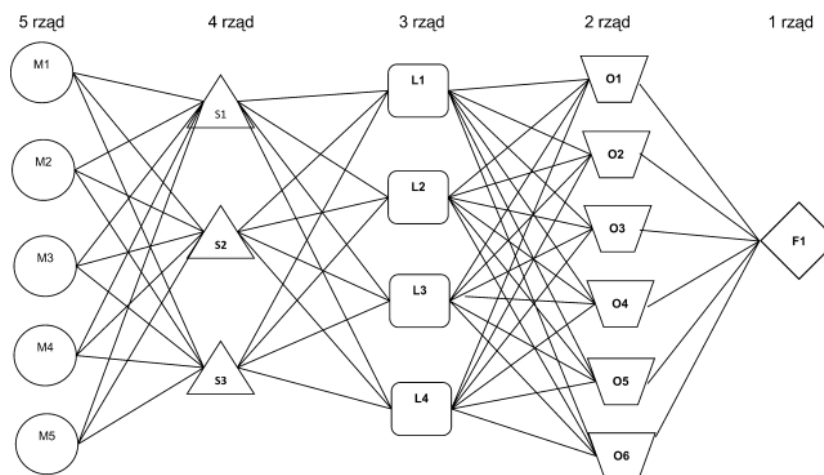
1. Cena produktu jest stała, a w łańcuchu produkcyjnym nie występują efekty skali
2. Cena produktu jest stała, a w łańcuchu produkcyjnym występują efekty skali
3. Cena produktu jest decyzją przedsiębiorstwa, a w łańcuchu produkcyjnym występują efekty skali

1.2 Podstawy teoretyczne

1.2.1 Przedsiębiorstwo jako system wieloagentowy

Na możliwość wykorzystania modeli wieloagentowych do badania i zarządzania systemami logistycznymi wskazują m.in. Moyaux et al, 2006 [?] czy Kawa, 2010 [?]. W swoich pracach zauważyli oni, że *producenci*, *dostawcy* i *odbiorcy* i inni uczestnicy łańcucha logistycznego mogą być opisani jako sieć autonomicznych, współpracujących ze sobą agentów. Takie podejście, i wynikająca z niego możliwość wykorzystania modeli wieloagentowych pomaga w rozwiązaniu problemów operacyjnych, na jakie wskazuje Moyaux [?]. Należy bowiem zwrócić uwagę, że w zakresie wyboru tras i zarządzania flotą wieloetapowe łańcuchy dostaw wielu produktów są problemami NP-trudnymi, szczególnie, że decyzje podejmowane lokalnie są współzależne.¹ Ponadto, jak zauważa Kawa, w sieci przedsiębiorstw pomiędzy dostawcami kolejnych rzędów (tj. fabryki, magazyny, sklepy) może istnieć wiele połączeń które są wobec siebie konkurencyjne, ponieważ jeden magazyn może zaopatrywać się w wielu fabrykach. Zastosowanie w tej dziedzinie modeli wieloagentowych pozwala więc na zbadanie, jak decyzje podejmowane na jednym z etapów łańcucha dostaw wpłyną na cały system i innych uczestników.

¹W praktyce, decyzje podjęte na wczesnym etapie łańcucha rezonują na dalsze etapy, co Moyaux opisuje jako "bullwhip effect"



Rysunek 1: Przykładowa sieć relacji w łańcuchu logistycznym

W niniejszej pracy stosowane jest rozszerzenie tego podejścia, poprzez zaprogramowanie jako agentów jednostek organizacyjnych przedsiębiorstwa (*fabryka, magazyn, sklep, zarząd*), które razem tworzą system (*przedsiębiorstwo*).

To podejście opiera się na obserwacji, że relacje pomiędzy jednostkami w przedsiębiorstwie są analogiczne do relacji uczestników łańcucha dostaw. Michael Porter zauważył, że działalność przedsiębiorstwa to de facto sekwencja działań, która na każdym ogniwie zwiększa wartość dla odbiorcy. Zasady funkcjonowania opisywanego przez Portera *łańcucha wartości* są identyczne co do opisywanego przez Moyeux i Kawę łańcucha dostaw - szczególnie, że przedsiębiorstwa poprzez strategię *integracji wertykalnej* mogą swym zasięgiem mogą w rzadkich przypadkach objąć całość łańcucha dostaw.

Ponieważ przedsiębiorstwa często dysponują wieloma duplikującymi swoje działania jednostkami ², łańcuch ten jest nieliniowy i w jego przypadku mamy do czynienia z podobnymi wyzwaniem co w łańcuchu logistycznym.

Zdefiniowanie jako agentów poszczególnych jednostek przedsiębiorstwa jest przy tym spójne z określoną przez Wooldridge i Jennings charakterystyką agenta, który według ich postulatów posiada :

- autonomię - poszczególne jednostki przedsiębiorstwa podążają za strategią i celami narzuconymi przez zarząd, ale mają zazwyczaj swobodę w podejmowaniu decyzji mających na celu ich realizację
- zdolności do komunikacji - jednostki przedsiębiorstwa komunikują się z otoczeniem (relacje z klientami) oraz między sobą (raportowanie do zarządu, spotkania), a w ramach pomiędzy jednostkami przedsiębiorstwa istnieje asymetria informacji

²Dobrym przykładem są tutaj zakłady samochodowe, które mogą produkować dany model w różnych krajach. Zmiana fabryki powoduje przy tym radykalną zmianę łańcucha dostaw

- reaktywność - jednostki przedsiębiorstwa reagują na zmiany rynkowe oraz zmiany wewnątrz przedsiębiorstwa
- proaktywność - jednostki przedsiębiorstwa podejmują inicjatywy mające na celu zwiększyć wartość przedsiębiorstwa, jak działalność innowacyjna bądź ekspansja.

Dlatego w niniejszej pracy będziemy rozważać model wieloagentowy, w którym według założeń na przedsiębiorstwo składać się będzie szereg autonomicznych agentów,

- $fabryk \in FA = \{fa_1, fa_2, fa_3..fa_m\}$
- $magazynów \in MA = \{ma_1, ma_2, ma_3..ma_k\}$
- $sklepów \in SK = \{sk_1, sk_2, sk_3..sk_i\}$
- $zarząd$, pełniący rolę centralnego koordynatora i przechowujący wszystkie powyższe $\in ZA = \{FA, MA, SK\}$

przez które kolejno będzie musiał przejść produkt zanim będzie mógł być zakupiony przez klienta. Rozpatrując ten system w proponowanym przez Kawa, 2013 kontekście teorii grafów oznacza to, że łańcuch produkcyjny może składać się z $m = \binom{n}{1} \times \binom{k}{1} \times \binom{i}{1}$ kombinacji d_m połączeń pomiędzy jednostkami przedsiębiorstwa. Jak zauważa Kawa, każde z tych połączeń będzie miało swoją maksymalną przepustowość oraz koszt $k_d \in f[d]$, będący sumą kosztów ponoszonych na każdym z ogniw łańcucha wartości. Na krańcach grafu może dojść do sprzedaży towaru i przychodu dla całego systemu - jednak zaopatrzenie sklepu w zbyt dużą ilość towaru doprowadzi do jego zmarnowania i strat.

1.2.2 Zadanie optymalizacyjne

Dlatego optymalizując działanie przedsiębiorstwa będziemy dążyć do tego, żeby dla danego poziomu produkcji x i zbioru *produktów* $\in PR = \{pr_1, pr_2, pr_3..pr_x\}$ ze zbioru możliwych sterowań (możliwych tras dostawy) $< d_1, d_2, d_3..d_m >$ dla każdego z produktów wybrać trasy maksymalizujący wyrażenie

$$\max \sum_{pr=1}^x \text{zysk}(cena, d) = \begin{cases} cena - k_d & \text{jeśli dokonano sprzedaży} \\ k_d & \text{jeśli nie dokonano sprzedaży} \end{cases} \quad (1)$$

gdzie $k_d \in f[d] \wedge k_d < cena$

To zadanie optymalizacyjne byłoby trywialne, gdybyśmy mieli doskonałą informację na temat poziomu sprzedaży w każdym ze sklepów w $t + 1$. Na taką wiedzę nie możemy liczyć ani w tej pracy, ani w rzeczywistości, dlatego rozwiązaniem proponowanym w niniejszej pracy jest zastosowanie modelowania

predykcyjnego (*predictive analytics*), w celu prognozowania ilości klientów i ich wyborów w każdym ze sklepów w najbliższych okresach czasu.³

Ponadto, jak warto zauważyć, nie możemy liczyć na to, że funkcja kosztu k_d będzie liniowa. Wspominana przez Kawę maksymalna przepustowość każdego z łańcuchów, która w przedsiębiorstwie będzie spowodowana ograniczonymi mocami produkcyjnymi, może spowodować, że funkcja kosztu będzie wykładnicza⁴. W niektórych przypadkach mogą istnieć także korzyści skali, które spowodują, że funkcja będzie logarytmiczna, a całkiem realne jest istnienie obu tych efektów⁵, przez co musimy zakładać, że $f(d)$ może być dowolną funkcją nieliniową.

Trzecim aspektem który trzeba wziąć pod uwagę jest złożoność obliczeniowa. Nawet dla prostego układu, lecz wolumenu produkcji ponad 1000 sztuk sprawdzenie zysku w przypadku wszystkich kombinacji alokacji wymaga olbrzymiej ilości iteracji. Mimo znaczącego wzrostu mocy komputerów w ostatnich latach, wolumeny produkcji w dużych przedsiębiorstwach oraz złożoność tras logistycznych sprawia, że to podejście jest kompletnie niepraktyczne i należy szukać alternatywnych podejść, upraszczających problem.

Proponowany algorytm będzie brał wszystkie te aspekty pod uwagę, a pierwszy z wymienionych problemów rozwiążemy wykorzystując *modelowanie predykcyjne*.

1.2.3 Modelowanie predykcyjne

Jak wskazuje James, 2013, zakładając, że dysponujemy zbiorem n obserwacji p zmiennych, możemy zbadać ich relację ze zmienną wyjaśnianą y i otrzymać *model*, który dla nowych - nieanalizowanych wcześniej - obserwacji x_1, x_2, \dots, x_n zwraca przewidywaną wartość zmiennej objaśnianej \hat{y} . Różnorodne metody dzięki którym możemy otrzymać \hat{y} James nazywa *statistical learning* i może być wykorzystany do modelowania predykcyjnego, tj. przewidywania przyszłych wydarzeń na podstawie przeszłej historii danych.

Zastosowanie metod *statistical learning* w przedsiębiorstwach potwierdza Buckinx, 2007, który wskazywał na możliwość prognozowania lojalności klienta na podstawie wewnętrznych danych o transakcjach, oraz Davenport, który w Harvard Business Review 12/2011 opisuje szereg *case studies* firm, w których wykorzystuje się istniejące dane o transakcjach do przewidywania przyszłych zakupów klientów. Jednym z podanych przez niego przykładów jest Tesco, które na podstawie zebranych danych przewiduje, jak będzie wyglądał koszyk zakupów klienta podczas następnych zakupów, i odpowiednio wcześniej wysyła mu bony zniżkowe. Podczas panelu *Strategia B2C w erze Big Data - jak wykorzystać potencjał danych* na XXV Forum Ekonomicznego w Krynicy przedstawi-

³Obecnie w przedsiębiorstwach rzadko stosuje się zaawansowane sposoby prognozowania sprzedaży (*predictive analytics*), a zarządzanie dostawami odbywa się raczej metodą manualnego uzupełniania zapasów.

⁴Wyprodukowanie każdego kolejnego produktu ponad moc produkcyjną będzie wymagało nadgodzin, dodatkowej przestrzeni magazynowej, pomocy poddostawców - a każde z tych wydarzeń będzie generowało dodatkowe koszty

⁵Korzyści skali odczuwalne do pewnego optymalnego poziomu produkcji, powyżej którego brak mocy produkcyjnych powoduje powiększanie się kosztów krańcowych

cieli polskiego biznesu zwracali uwagę na szerokie wykorzystanie modelowania predykcyjnego również w nasz gospodarce.⁶

To daje nam podstawy, żeby w przypadku optymalizowanego przedsiębiorstwa zakładać, że dane o każdej transakcji są zapisywane i poza podstawowymi informacjami (produkt, cena, data) zawierają one także dane o kliencie, a cały zbiór danych może być wykorzystany do przewidywania sprzedaży w $t + 1$.

Dla każdej transakcji w sklepie dysponujemy więc zbiorem informacji $X = n \times x_i = \langle tr_1..tr_k, pr_1..pr_k, kl_1..kl_k \rangle$ gdzie tr to identyfikatory transakcji (miejscę, data, rodzaj płatności), pr to identyfikatory produktu (nazwa, cena, ilość) oraz kl to identyfikatory klienta (płeć, wiek, zarobki, wykształcenie etc.).

Na podstawie takiego zbioru danych chcemy przewidzieć popyt na produkty w każdym ze sklepów, co w praktyce oznacza konieczność stworzenia modelu estymującego -

- liczbę poszczególnych grup klientów⁷ odwiedzających sklep w $t + 1$, gdzie $\hat{y} \in R$

oraz w zależności od zastosowanego podejścia

- prawdopodobieństwo p z jakim klient o danej charakterystyce kupi produkt pr , gdzie $\hat{y} \in (0, 1)$
- produkt wybrany przez danego klienta, gdzie $\hat{y} \in \{pr_1, pr_2..pr_n\}$, a pr_i to dostępny produkt.

Jak wskazuje James, zmienna objaśniana \hat{y} może przyjąć formę liczby rzeczywistej, zmiennej binarnej (1/0), prawdopodobieństwa, *log odds* lub klasy, jednak ze względu na różne dziedziny \hat{y} , każdy z tych przypadków różni się metodami które możemy zastosować. Według sugestii James i Hastie, zastosujemy następujące metody

- do liczby klientów - regresję metodą OLS (*ordinary least squares, metoda najmniejszych kwadratów*)
- do prawdopodobieństwa zakupu - regresję logistyczną (*logistic regression*)
- do wyboru produktu- metody klasyfikacyjne - *k-means* oraz *drzewa klasyfikacyjne*

oraz metody nie służące bezpośrednio do modelowania \hat{y} , jednak wspierające proces predykcyjny

- *Distance scaling*, poprzez liczenie odległości euklidesowej (*euclidean distance*) pomiędzy dwoma zbiorami danych

⁶Jednak jak zwracano uwagę, *big data* i *modelowanie predykcyjne* służą głównie do zyskiwania wiedzy o kliencie i rynku do manualnego przetworzenia *insight*, a nie automatyzacji podejmowania decyzji, co rozważamy w tej pracy

⁷Przez "poszczególne grupy klientów" rozumiemy klientów o wspólnej charakterystyce, czyli takich samych zestawach zmiennych identyfikujących $\langle kl_1..kl_n \rangle$

- *Subset selection*, do wybrania zmiennych wyjaśniających które wspólnie tworzą najlepszy model
- Prawdopodobieństwo warunkowe i naive bayes do przewidywania, jacy konsumenci odwiedzą sklep w $t + 1$, a co za tym idzie, jakie są wartości zmiennych wyjaśniających do prognozowania \hat{y}

1.3 Proponowany algorytm optymalizacyjny

Znając przewidywaną sprzedaż na krańcach grafu, proponowany w niniejszej pracy algorytm rozwiązania powyższego problemu opiera się na obserwacji, że równanie 1 będzie równoważne zapisowi sumującemu po możliwych kombinacjach łańcucha (zamiast dla każdego produktu z osobna).

$$\max \sum_{d=1}^m zysk(cena) = cena \times pr_d - k_d \times pr_d \quad (2)$$

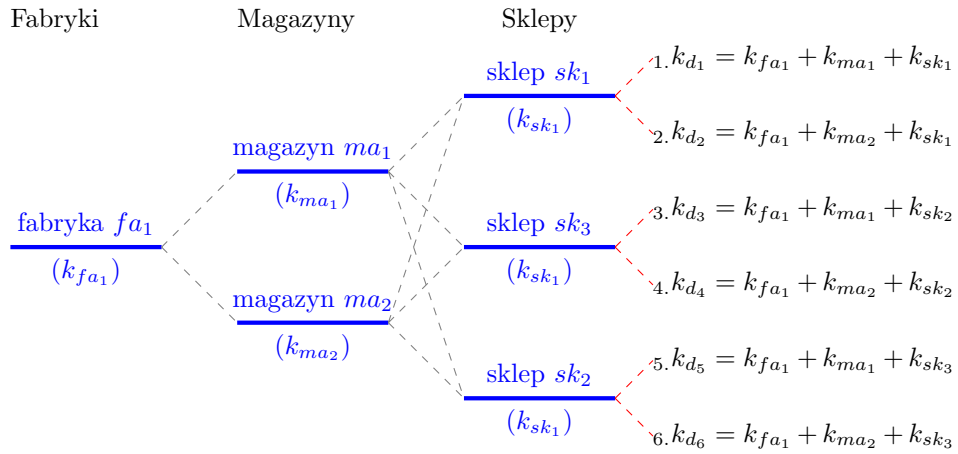
gdzie pr_d to ilość produktów przechodzącym w danym łańcuchu

Ponieważ ścieżka d_m składa się z $\langle fa_m, ma_m, sk_m \rangle$, a koszt ponoszony na całej trasie będzie równie sumie kosztów ponoszonym na każdym z ogniw łańcucha, równanie 2 możemy zapisać jako

$$\max \sum_{d=1}^m zysk(cena) = cena \times pr_d - \left(\sum_{j=1}^m k_j \right) \times pr_d \quad (3)$$

gdzie k_j to koszty ponoszone na każdym z elementów łańcucha wartości (koszty produkcji, magazynowania, transportu, etc.). Te co do zasady są nam znane, więc zostaje nam znalezienie takich wartości pr_d i $ceny$ (jeśli nie przyjmujemy założenia danej, stałej ceny) dla których firma będzie osiągać maksymalny zysk. Ze względu na występujące w przedsiębiorstwie współzależności oraz nieliniowość kosztów (korzyści skali) musimy cały układ rozważać łącznie.

Na przykładzie przedsiębiorstwa $= \langle fa_1, ma_1, ma_2, sk_1, sk_2, sk_3 \rangle$ mamy do czynienia z $m = \binom{1}{1} \times \binom{2}{1} \times \binom{3}{1} = 6$ kombinacji połączeń pomiędzy jednostkami przedsiębiorstwa, przy czym każdy do każdego ze sklepów można przeprowadzić dostawę jedną z dwóch tras.



Ponieważ przewidywany wolumen sprzedaży produktu w każdym z krańców grafu jest nam znana⁸, żeby umożliwić obliczenie optymalnego obłożenia każdej z tras prowadzącej do krańca grafu⁹ możemy dla każdej z tras ilość przechodzących nią produktów zapisać jako

$$pr_1 = \alpha pr_{sk_1}, \quad pr_2 = \beta pr_{sk_1} \quad \dots \quad pr_{d-1} = \gamma pr_{sk_i}, \quad pr_d = \delta pr_{sk_i}$$

gdzie pr_{sk_d} to ilość produktów, jakie mają trafić do sk_d krańca grafu leżącego na trasie d , a $\alpha, \beta, \dots, \delta$ to udział tej trasy w przepływie wszystkich towarów do danego krańca grafu. W ten sposób, równanie 3 możemy zapisać jako

$$\max \sum_{d=1}^m zysk(cena) = cena \times \alpha pr_{sk_d} - \left(\sum_{j=1} k_j \right) \times \alpha pr_{sk_d} \quad (4)$$

Co po rozwinięciu pozwoli nam na sprowadzeniu równania 4 do postaci

$$\begin{aligned} zysk(cena) = & \alpha(cena \times pr_{sk_1} - \left(\sum_{j=1} k_j \right) \times pr_{sk_1}) + \\ & \dots + \delta(cena \times pr_{sk_i} - \left(\sum_{j=1} k_j \right) \times pr_{sk_i}) \end{aligned} \quad (5)$$

jeśli przyjmiemy, że

$$\begin{aligned} a = & (cena \times pr_{sk_1} - \left(\sum_{j=1} k_j \right) \times pr_{sk_1}) \dots \\ ..b = & (cena \times pr_{sk_i} - \left(\sum_{j=1} k_j \right) \times pr_{sk_i}) \end{aligned}$$

otrzymamy w ten sposób wielomian, w którym współczynniki a, b są nam znane¹⁰, a pary współczynników α, β jednego krańca grafu muszą być mniejsze lub równe 1.¹¹

Za Sydsaeter, 2005 zauważmy, że $\frac{\partial zysk}{\partial \alpha}, \frac{\partial zysk}{\partial \beta} \dots \frac{\partial zysk}{\partial \delta}$ - czyli pierwsze pochodne cząstkowe po α, β, δ z tak zdefiniowanej funkcji - dadzą nam informację, o ile wzrośnie zysk przedsiębiorstwa w przypadku zwiększenie obłożenia tej trasy. Dlatego dla każdego produktu pr_n istnieje możliwość wybrania takiej z tras $< d_1, d_2, d_3 \dots d_m >$, która będzie maksymalizowała wynik przedsiębiorstwa - jak w równaniu 1.

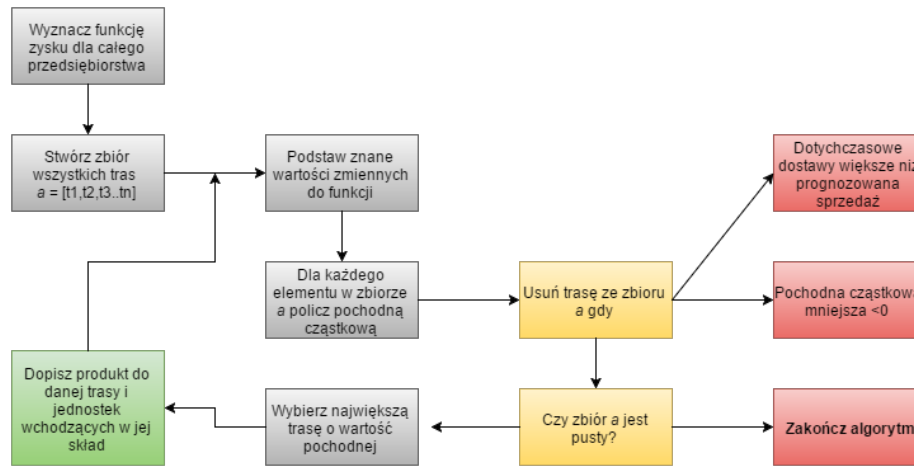
Algorytm będzie więc działał następująco:

⁸Zakładamy, że dzięki *Predictive analytics* na podstawie poprzednich obserwacji mamy dokładne prognozy dotyczące sprzedaży w $t + 1$

⁹Żeby ograniczyć złożoność obliczeniową staramy się unikać pętli, która sprawdza każdą kombinację, zamiast tego szukając bardziej wyrafinowanych sposobów

¹⁰ pr_{sk_1} zostały wyprognozowane modelowaniem predykcyjnym, a k_j to znana nam suma funkcji kosztów w każdej jednostce

¹¹Warunkiem nie jest "równe 1", ponieważ może istnieć taki udział α w przedziale $< 0, 1 >$ powyżej którego przez nieliniowość funkcji kosztów dostawa może być nieopłacalna



Rysunek 2: Proponowany algorytm optymalizacyjny

2 Model

2.1 Koncepcja modelu

W ramach pracy zbudowany został model wieloagentowy, który symuluje lokalny rynek na wybrany produkt. W modelu agentami są

- **klienci**, których definiują unikalne cechy ¹² wpływającą na podejmowane przez niego decyzje
- **przedsiębiorstwo**, sprzedające *produkt* na rynku. Przedsiębiorstwo przy tym ma złożoną strukturę, tj. zamiast działać jako indywidualny agent, składa się z współpracujących ze sobą agentów
 - fabryk
 - magazynów
 - sklepów
 - zarządu, pełniącego funkcje koordynującą
- **konkurencji**, również sprzedającej na rynku swoje produkty, ale pasywnej w stosunku do symulowanego przedsiębiorstwa. ¹³
- **produktów**, które z oczywistych względów nie podejmują decyzji, jednak mają swoją charakterystykę wpływającą na decyzje innych agentów (przede wszystkim konsumentów) oraz przemieszczają się w ramach przedsiębiorstwa.

¹²Są to między innymi wiek, zarobki, wykształcenie, zainteresowania - zostanie to dokładnie opisane w dalszej części pracy

¹³Tj. konkurencja nie zmienia decyzji podjętych przed rozpoczęciem gry, i w założeniu ma stanowić jedynie alternatywę dla konsumentów

Żeby dobrze odwzorować kluczowy aspekt lokalizacji i drogi w łańcuchach dostaw, symulowany rynek jest osadzony w *wirtualnym mieście*, czyli każdy agent ma swoją lokalizację w macierzy o wymiarach $x \times y$. Lokalizacja wpływa na działania agenta - klient kupi produkt tylko w sklepie w pobliżu, a dostawa z magazynu do sklepu będzie tym droższa, im bardziej oddalone będą od siebie.

W każdej jednostce czasu t klienci z prawdopodobieństwem p będą potrzebować symulowany produkt, więc odwiedzając bliski sklep, wybiorą jeden z produktów oferowanych przez przedsiębiorstwo i konkurencję. Symulacja wyboru opiera się na danych o preferencjach konsumenckich zebranych w ankiecie na próbie 127 badanych. Ponieważ każdy konsument-agent ma swoje unikalne cechy, w symulowanym procesie wyboru metodą drzewa klasyfikacyjnego przyporządkowujemy wybór, jakiego prawdopodobnie dokonał by jego odpowiednik w świecie rzeczywistym.¹⁴

Ponieważ konsument wybiera produkt tylko z gamy dostępnych w sklepie, przed rozpoczęciem tury przedsiębiorstwo musi podjąć szereg decyzji o m.in.

- poziomie produkcji
- wolumen dostaw do każdego ze sklepów
- rozdzieleniu wolumenów pomiędzy części przedsiębiorstwa¹⁵
- jaką trasą powinny zostać wysłane dostawy

Każda z tych decyzji będzie miała wpływ na przychody¹⁶ oraz koszty firmy. Celem pracy jest zbudowanie algorytmu, który na podstawie dotychczasowej historii transakcji pozwoli symulowanemu przedsiębiorstwu przewidzieć potencjalną sprzedaż w czasie $t + 1$ i ze zbioru możliwych sterowań (wyżej wymienionych decyzji) $< u_{t+1} >$ wybierze takie, które będą maksymalizować zysk.

2.2 Zastosowane narzędzia

Model został zbudowany w języku programowania Python 2.7, z wykorzystaniem następujących bibliotek:

Biblioteka	Źródło	Zastosowanie
Sympy	www.sympy.org	Wykorzystanie do obliczeń symbolicznych
scikit-learn	scikit-learn.org	Wykorzystanie bibliotek metod statystycznych

Kod programu dostępny jest pod adresem github.com/hubertguzera/master-thesis

¹⁴Oczywiście, o wiele lepsze byłoby oparcie pracy o prawdziwe historie transakcji, jednak jest to niemożliwe ze względu na dużą poufność tych danych

¹⁵Tj. ile z całkowitego wolumenu ma wyprodukować fabryka A , a ile fabryka B

¹⁶Przy założeniu stałej ceny, będą to przede wszystkim *utraczone koszty* w przypadku wyczerpania się zapasów w sklepie

2.3 Struktura programu

Program podzielony jest na trzy części, jak zaprezentowano na rysunku 3

- Pierwsza odpowiada za stworzenie, w drodze losowań, środowiska w ramach którego toczy się symulacja, wraz z agentami i macierzą lokalizacji.¹⁷
- Druga część przez t_n jednostek czasu egzekwuje decyzje podjęte przez przedsiębiorstwo i symuluje zachowania konsumenckie, zwracając na koniec wyniki (przychody i zysk) przedsiębiorstwa w danej jednostce czasu t .
- Trzecia po t_n rund symulacji aplikuje algorytm optymalizacyjny, który poprzez prognozowanie sprzedaży na krańcach grafu w t_{n+1} , szuka najlepszych decyzji o alokacji produktów.

2.4 Generowanie mapy

W celu odpowiedniego odwzorowania kluczowego aspektu lokalizacji i dróg w łańcuchach dostaw, a przy tym wzorując się na podejściu zastosowanym w *modelu segregacji Schellinga*, Schelling 1971, agenci osadzeni są w przestrzeni, reprezentowaną przez macierz klas o wymiarach (x,y). Dodatkowo, lokalizacje będą połączone drogami, wymuszając na agentach poruszanie się tylko w obrębie ścieżek. Dzięki temu, w modelu będziemy mogli wiernie odwzorować wpływ odległości i wyboru trasy na efektywność procesów logistycznych, oraz zależność wyników sklepu od zamieszkującej okolicę populacji.

Wszystkie te informacje przechowywane są w klasie *świat*, która zawiera wszystkie informacje dotyczące wirtualnego otoczenia przedsiębiorstwa - czyli także konsumentów. Klasa *świat* ma następujące właściwości

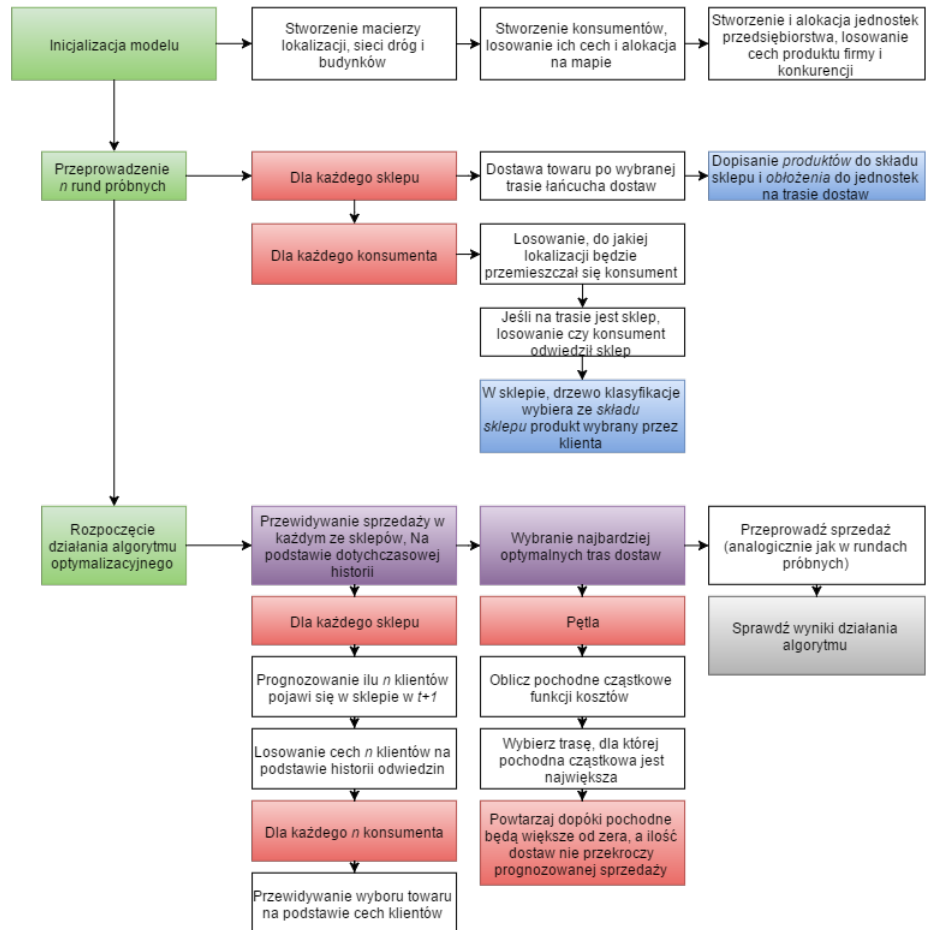
```
class swiat:
    mapa = [] # macierz lokalizacji o wymiarach x,y
    ludnosc = [] # macierz instancji klasy konsument
    nodes = {} # słownik sieci drog
```

Każdy element macierzy mapa o wymiarach (x,y) jest instancją klasy lokalizacja o następujących właściwościach

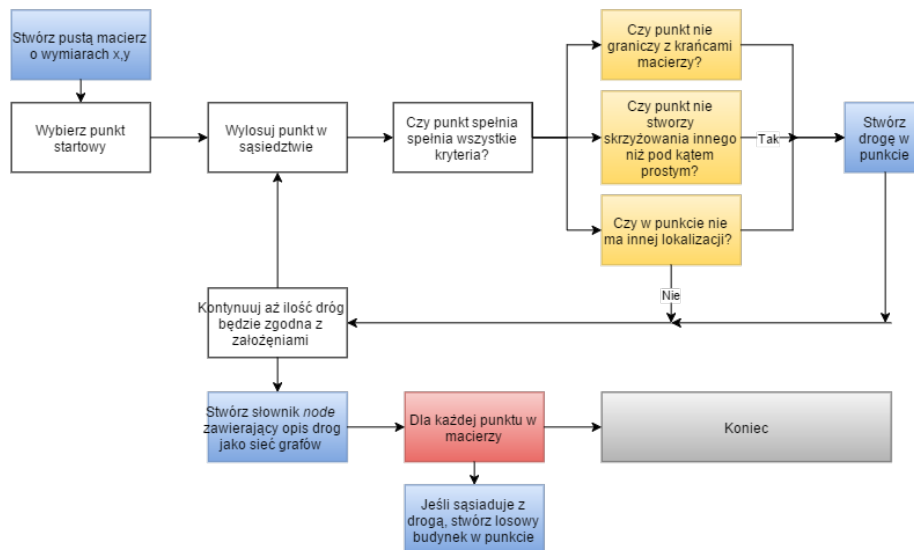
```
class lokalizacja:
    typ = string # typ budynku w lokalizacji
    x = integer # lokalizacja x macierzy mapa
    y = integer # lokalizacja y macierzy mapa
    droga = [] # lokalizacja drogijazdowej
```

które generowane są według algorytmu widocznego na rysunku 4

¹⁷Model może pominąć ten etap i wczytać pregenerowany świat w celu sprawdzenia różnych scenariuszy w statycznym świecie (*ceteris paribus*).



Rysunek 3: Struktura programu



Rysunek 4: Algorytm tworzenia dróg i lokalizacji

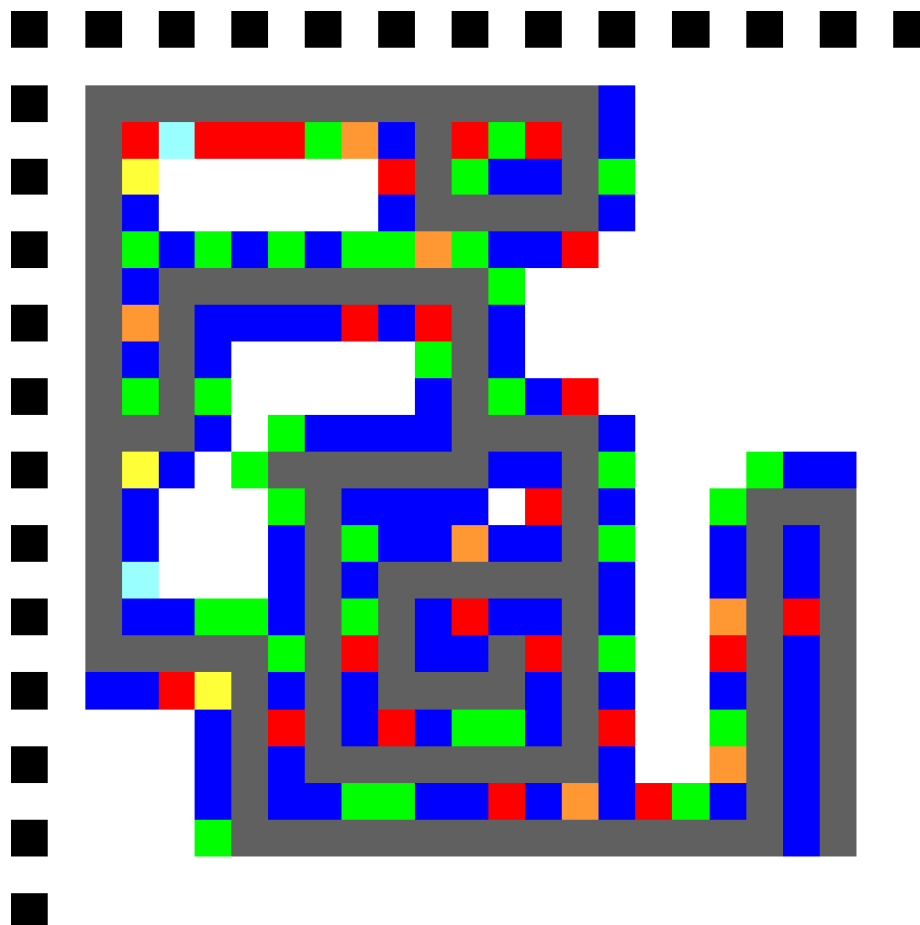
Każda mapa wygenerowana przez program spełniać będzie następujące założenia

- Drogi krzyżują się i skręcają tylko pod kątem prostym. Poza skrzyżowaniami, drogi nie mają w sąsiedztwie innych dróg
- Inne lokalizacje (domy, sklepy etc.) mogą występować tylko w bezpośrednim sąsiedztwie drogi
- Drogi stanowią ciągłą linię, dzięki czemu nie ma punktu, do którego nie dałoby się dojechać z dowolnego miejsca startowego
- W regionie 2 punktów od skraju mapy nie są generowane ani drogi, ani lokalizacje ¹⁸
- Gęstość sieci dróg oraz prawdopodobieństwo występowania zakrętów jest predefiniowana przez użytkownika

2.4.1 Algorytm wyszukiwania drogi

Odległości pomiędzy zadanymi punktami w modelu są wyszukiwane dynamicznie, na podstawie algorytmu wyszukiwania drogi i zliczaniu ilości punktów w

¹⁸Jest to zabezpieczenie algorytmu, który w odległości 2 pkt od skraju mapy ma 0 proc. szansy na poprowadzenie ścieżki dalej - ponieważ w przypadku iterowania na skrajach mapy niektóre funkcje (jak sprawdzenie sąsiadujących punktów) mogą odnieść się do współrzędnych poza mapą, powodując błąd programu



Rysunek 5: Przykładowa mapa - szary to drogi, niebieski - domy mieszkalne, czerwony i zielony - biurowce, pomarańczowy - sklepy, żółty - magazyny, błękitny - fabryki

zwracany przez niego łańcuchu. Algorytm oparty jest na metodach wyszukiwania ścieżek w grafach, dzięki założeniu, że każdy droga o współrzędnej (x,y) na mapie jest punktem grafu, który może sąsiadować z punktami o współrzędnych $(x-1,y), (x+1,y), (x,y+1), (x,y-1)$ ¹⁹, o ile również są drogami. Informacje o punktach i sąsiadujących przechowywane są w zmiennej `nodes`, która jest słownikiem, dla każdego klucza - punktu na mapie - przechowuje informacje o sąsiadujących punktach, np. $(3,2) = [(3,3)(4,3)]$.²⁰

Algorytm, udostępniony przez Python Foundation²¹ ma następujące cechy

- jest rekurencyjny
- nie jest losowy
- nie gwarantuje znalezienia najkrótszej trasy

```
def find_path(graph, start, end, path=[]):
    path = path + [start]
    if start == end:
        return path
    if not graph.has_key(start):
        return None
    for node in graph[start]:
        if node not in path:
            newpath = find_path(graph, node, end, path)
            if newpath: return newpath
    return None
```

2.5 Agenci, ich rodzaje i właściwości

2.5.1 Konsumentci

Idąc za Kamiński, w modelu stosujemy modelowanie rynku za pomocą heterogeniczne konsumentów. Stąd, każdy z konsumentów ma swoją unikalną charakterystykę, która wpływa na jego wybory. Ponadto, ponieważ każdy klient ma przypisany dom, pracę i znajomych i mieszka w towarzystwie osób sobie podobnych²², pula lokalizacji w ramach których się porusza się zamknięta. Dzięki temu, odwziewiedlamy zjawisko ze świata rzeczywistego, że konsumenci zazwyczaj robią zakupy w ograniczonej liczbie sklepów będących po drodze bądź niedaleko.

¹⁹Punkty $(x-1,y-1), (x+1,y-1), (x+1,y+1), (x-1,y+1)$ wykluczamy przez wcześniejsze założenie, że drogi krzyżują się tylko pod kątem prostym

²⁰Pewnym ograniczeniem jest, że jako punkty grafu definiujemy tylko drogi, tak więc szukając trasy z punktu A do punktu B de facto szukamy trasy z drogi przy punkcie A do drogi przy punkcie B.

²¹<https://www.python.org/doc/essays/graphs/>

²²Jest to założenie inspirowanie !!!, i osiągnięte w podobny sposób - po wstępnej alokacji konsumentów do lokalizacji przeprowadzamy n rund, w których jest szansa na przeprowadzkę do miejsca o bardziej podobnym profilu mieszkańców

Jest to bardzo istotny warunek funkcjonowania modelu, ponieważ losowy dobór klientów uniemożliwiłby modelowanie predykcyjne.

Każdy będzie definiowany w klasie o następujących właściwościach

```
class konsument:
    plec = string
    wiek = integer
    zarobki = integer
    zainteresowania = array # macierz trzech string
    znajomi = array # trzy relacje z innymi agentami
    wykształcenie = integer # 5-stopniowa skala
    okazja = boolean # wskazanie czy wychodzi z domu
    domx = integer # współrzędna x domu na mapie
    domy = integer # współrzędna y domu na mapie
    pracax = integer # współrzędna x pracy na mapie
    pracay = integer # współrzędna y pracy na mapie
```

Wartości dla każdego z konsumentów są losowane niezależnie na podstawie rozkładów publikowanych przez Główny Urząd Statystyczny oraz danych firmy Sedlak&Sedlak²³ w celu zagwarantowania odzwierciedlenia struktury społeczeństwa. Ze względu na zastosowanie prawdopodobieństw warunkowych dla niektórych cech (np. zarobki są zależne od wykształcenia) istnieje pomiędzy nimi współliniowość.²⁴ Przykładowy rozkład pokazany jest na rysunku 6

2.5.2 Przedsiębiorstwo

Jak wskazano w rozdziale 1, przedsiębiorstwo nie jest modelowane jako jedna jednostka, a zamiast tego stosowane jest podejście wieloagentowe - każda z jednostek organizacyjnych jest samodzielnym, niezależnym agentem. Stąd, klasa firma jest tylko klasą przechowującą dane o wszystkich jednostkach wchodzących w skład przedsiębiorstwa, z następującymi właściwościami.

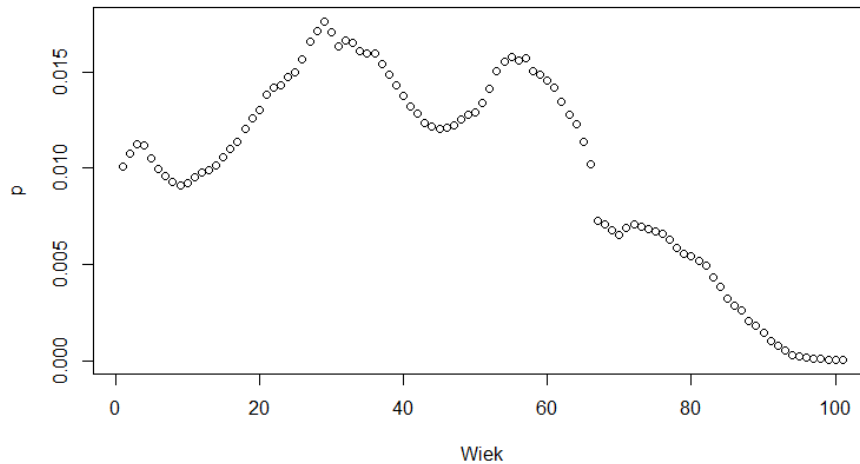
```
class firma:
    fabryki = array # macierz klasy fabryki
    magazyny = array # macierz klasy magazyny
    sklepy = array # macierz klasy sklepy
    trasy = array # macierz przechowująca trasy dostaw
    produkt = class produkt
    cena = integer # cena produktu
```

Fabryki, magazyny i sklepy mają wspólnych zestaw cech podstawowych

```
class fabryka, magazyn, sklep:
```

²³Raporty firmy Sedlak&Sedlak służyły do zbudowania tabeli prawdopodobieństwa wystąpienia danego wynagrodzenia w zależności od płci i wykształcenia. Reszta danych oparta na GUS

²⁴Chociaż współliniowość może być problemem przy modelowaniu, będziemy sobie z nią radzić na późniejszym etapie



Rysunek 6: Przykładowy rozkład prawdopodobieństwa zmiennej wiek, na podstawie GUS

```

nazwa = string
lokalizacja = array # macierz zmiennej x i y
droga = array # lokalizacja drogi dojazdowej
oblozenie = integer # wolumen przechodzacy przez jednostke
self.koszt = integer # wspolczynnik kosztu
self.efekt_skali = integer # wspolczynnik efektu skali
symbol = sympy.symbol

```

przy czym sklepy mają dodatkowe zmienne i funkcje przechowujące informacje o inwentarzu, klientach odwiedzających sklep w danej jednostce czasu t oraz historii transakcji -

```

class fabryka , magazyn , sklep :
    klienci = array #klienci odwiedzajacy w t
    klienci_historycznie= array # wszyscy klienci sklepu
    sklad = dict # ilosc produktow w sklepie
    sprzedaz= dict # statystyki sprzedazy w t
    przewidywana_sprzedaz= integer

```

2.5.3 Produkt

Opierając się na argumentacji, wedle którego każdy produkt charakteryzuje się cechami wpływającymi na prawdopodobieństwo jego zakupu przez konsumentów, nasz produkt definiujemy przez zestaw dowolnych cech, definiowanych jako

factors, skalę ocen bądź zmienne binarne, które odróżniają go od produktów konkurencji.²⁵ Produkt jest więc klasą przechowującą jednowymiarową macierz z cechami produktu $[x_1, x_2, x_3]$

2.5.4 Konkurencja

W założeniach przyjmujemy, że konkurencja jest pasywna - tj. nie podejmuje działań ani decyzji w trakcie trwania symulacji. Wynika to z odmiennego celu badania, którym jest analiza działania algorytmów optymalizacyjnych - nagle zmiany sprzedaży spowodowane np. obniżeniem ceny przez konkurencję spowodowałyby wątpliwości interpretacyjne i są zbędne. Konkurencja jest za to potrzebna do stworzenia alternatywnych dla symulowanego produktu, o odmiennych cechach i przyciągających klientów o specyficznych charakterystykach.

Dlatego do symulowania konkurencji wystarczy stworzenie n innych instancji klasy *produkt* i dodanie ich do zmiennej *skad* klasy *sklep*. Informacje o produktach konkurencji będą przechowywane w klasie *rynek*

2.5.5 Trasy

Klasa *trasy* przechowuje wszystkie możliwe kombinacje łańcuchów dostaw, składające się z instancji klas *fabryka*, *magazyn*, *sklep* oraz *scieek* pomiędzy nimi²⁶. Długość ścieżki bezpośrednio wpływa na koszt łańcucha - koszt wzrasta liniowo w stosunku do długości ścieżki.

2.6 Symulowanie decyzji konsumenckich

Jak wskazano na 3, podczas wizyty każdego z wirtualnych konsumentów w sklepie symulujemy jego decyzję co do zakupu towaru. Ta symulacja jest bardzo ważna dla modelu, i co warto odnotować, nie jest tym samym co późniejsze przewidywanie "prognozowanej sprzedaży". W tej części chcemy, aby decyzje konsumentów w modelu jak najbardziej przypominały decyzje klientów w analogicznych sytuacjach w świecie rzeczywistym.

Opierając się na Sagan, 2011, wiemy, że jesteśmy w stanie zdefiniować kluczowe cechy konsumenta i produktu jako zbiór zmiennych. Stąd, w grze eksperymentalnej przeprowadzonej na potrzeby pracy²⁷, poproszono uczestników o stwierdzenie, jakie produkty z dostępnej listy kupi klient o charakterystyce wylosowanej przez program. Po odpowiedzi udzielonej przez gracza, predefiniowane, jakościowe cechy produktu były transponowane na wartości liczbowe²⁸ i

²⁵Ich istotność nie jest w tym momencie ważna, ponieważ nawet jeśli w zbiorze znajdzie się cecha mająca mały wpływ na decyzje konsumentów, zostanie ona wyeliminowana na etapie tworzenia modelu bądź drzewa klasyfikacyjnego ze względu na brak istotności statystycznej współczynnika

²⁶Jako ścieżkę rozumiemy sekwencję punktów z drogami, jakie trzeba przebyć od fabryki do magazynu i od magazynu do sklepu

²⁷Gra dostępna jest pod adresem <http://serwer1418288.home.pl/test/piwo/zapisy.php>

²⁸Na przykład, piwo Grolsh jest drogie, klasy premium i jest lager, stąd otrzyma zapis $[5,1,1]$, a tani smakowy Redd's $[3,0,0]$.

wraz z ilościowymi cechami klienta i informacją *kupił/niekupił* zapisywany na serwerze SQL.

W programie dane te służą do budowy drzewa klasyfikacyjnego, które - ponieważ dane liczbowe są w identycznej formie- dla każdego *query* o klient i produkt zwraca prawdopodobieństwo zakupu. Stąd, dla każdego agenta możemy zbudować listę prawdopodobieństw zakupu każdego z towarów dostępnego na rynku, i w ten sposób losowaniem symulować decyzje konsumenckie. To podejście gwarantuje wysokie podobieństwo z wyborami realnych konsumentów ze względu na -

- zastosowaniu podejścia, w którym budujemy bazę danych decyzji konsumenckich, w którym ze względu na wysoką ilość rekordów jest duża szansa, że istnieje zapis o decyzjach klienta o bardzo podobnej charakterystyce
- baza danych jest generowana przez ludzi, przez co zapisy mają wysoką jakość, rozumianą jako zgodność z wyborami w świecie rzeczywistym
- wybór drzewa klasyfikacyjnego jako metody i wysoki n rekordów pozwala na bardzo pożądaną w tej sytuacji *overfit*, który jak opisuje James powoduje, że dostajemy bardzo dokładne odwzorowanie zbioru uczącego²⁹

²⁹Chociaż to by było bardzo niepożądane w modelach predykcyjnych, tutaj bardzo nam odpowiada, ponieważ chcemy odwzorować zbiór uczący

3 Efekty działania algorytmu optymalizacyjnego

- 3.1 Charakterystyka badanego środowiska
- 3.2 Przewidywanie decyzji konsumentów
- 3.3 Wyniki przedsiębiorstwa przy braku optymalizacji
- 3.4 Optymalizacja przy stałych cenach i braku efektu skali
- 3.5 Optymalizacja przy stałych cenach i istnieniu efektu skali
- 3.6 Optymalizacja przy zmiennych cenach i istnieniu efektu skali

Literatura