

WYBÓR OPROGRAMOWANIA PRZY WIELU KRYTERIACH

Paweł Bobryk*
Ignacy Kaliszewski†**

* Wyższa Szkoła Informatyki Stosowanej i Zarządzania, Wydział Informatycznych Technik Zarządzania, studia II stopnia, specjalność Komputerowe Wspomaganie Zarządzania.

** Wyższa Szkoła Informatyki Stosowanej i Zarządzania, Wydział Informatycznych Technik Zarządzania;

† Instytut Badań Systemowych, Polska Akademia Nauk
Newelska 6, 01-447 Warszawa

W artykule przedstawiono zastosowanie analizy wielokryterialnej do wyboru oprogramowania, zgodnie z metodologią, przekazywaną studentom Wydziału Informatycznych Technik Zarządzania, w Wyższej Szkole Informatyki Stosowanej i Zarządzania, w ramach przedmiotu Komputerowe Wspomaganie Podejmowania Decyzji.

Słowa kluczowe: analiza decyzyjna, wielokryterialne problemy decyzyjne, wybór oprogramowania.

1. Wprowadzenie

Na Wydziale Informatycznych Technik Zarządzania Wyższej Szkoły Informatyki Stosowanej i Zarządzania w Warszawie, na II stopniu studiów, studenci specjalności Komputerowe Wspomaganie Zarządzania mają w programie przedmiot Komputerowe Wspomaganie Podejmowania Decyzji, obejmujący wykład i zajęcia laboratoryjne. Przedmiot ten wykładany jest przez drugiego z autorów, według skryptu akademickiego Kaliszewskiego i Podkopaeva (2010).

W niniejszym artykule przedstawimy zagadnienie wyboru oprogramowania przy istnieniu wielu kryteriów. Artykuł jest oparty na pracy magisterskiej pierwszego z autorów, której drugi z autorów był promotorem (Bobryk, 2018). Będziemy starali się uzasadnić tezę, którą podzielamy, że metodologia analizy wielokryterialnej przekazywana studentom w ramach przedmiotu może wzbogacić analizy biznesowe przeprowadzane w odniesieniu do rzeczywistych problemów decyzyjnych, występujących w działalności gospodarczej. Polem konkretnych rozważań jest tu inżynieria oprogramowania.

2. Problem wyboru oprogramowania

Dzisiejszy rynek firm tworzących oprogramowanie jest trudniejszy niż ten sprzed 20 lat. Jest wiele firm, jest duży popyt, są wysokie wymagania odbiorców oprogramowania, jest wiele nowych rozwiązań. Na przykład 30 lat temu program na ogół działał na jednym komputerze i sporadycznie tylko łączył się z Internetem. Obecnie programy to raczej złożone systemy informatyczne, które współpracują w czasie rzeczywistym w połączeniu z innymi systemami, działają na wielu platformach: telefonach komórkowych, komputerach, tabletach i mogą być dostępne z każdego miejsca na świecie.

Jest oczywiste, że złożoność systemów informatycznych, w porównaniu do pojedynczych programów, dramatycznie wzrosła. Tym samym wzrosła również liczba możliwych do zastosowania konfiguracji systemów. Przystępując do budowy systemu informatycznego musimy zdecydować:

- jaką architekturę systemu zastosujemy?
- jakie bazy danych zastosujemy?
- w jakim języku napisany zostanie system?
- za pomocą jakiego oprogramowania można zrealizować funkcje systemu w najlepszy możliwy sposób?

W dalszym ciągu pracy skupimy się wyłącznie na problemie wyboru oprogramowania. Będziemy posługiwać się pojęciem *technologii* i *stosy technologii*.

W procesie tworzenia oprogramowania, wykorzystywane są różne komponenty. Dla przykładu, przyjrzyjmy się komponentom, wykorzystywanym w konstruowaniu systemów internetowych. JavaScript, HTML, CSS są przedstawicielami grupy języków przeznaczonych do tworzenia oprogramowania systemów internetowych; MySQL, MongoDB są systemami baz danych; RestApi, SOAP są standardami do tworzenia interfejsów aplikacji do komunikacji pomiędzy systemami informatycznymi; OAuth jest standardem wykorzystywanym przy autoryzacji; PGP jest programem do szyfrowania wiadomości. Wymienione elementy tworzą komponenty systemów informatycznych. Takie komponenty przyjęło się nazywać w slangu informatycznym *technologiami*. Zgodnie z definicją Słownika Języka Polskiego PWN, *technologia* to metoda przeprowadzania procesu produkcyjnego oraz przetwórczego lub dziedzina techniki zajmująca się opracowywaniem nowych metod produkcji wyrobów lub przetwarzania surowców. Taką definicję technologii można zastosować do komponentów systemów informatycznych, ponieważ każdy taki komponent odpowiada pewnej metodzie lub metodom postępowania, zgodnie z którymi należy je konstruować, aby poprawnie działały w oprogramowaniu systemu. Dla przykładu, SOAP zawiera dokładny opis zasad komunikacji między systemami informatycznymi, standard opisuje to krok po kroku.

Stosem technologii będziemy nazywać zbiór współpracujących ze sobą technologii. Różne stopy technologii stanowią warianty decyzyjne (w skrócie warianty) w rozpatrywanym tu procesie decyzyjnym.

3. Proces tworzenia oprogramowania

Zgodnie z najlepszymi praktykami inżynierii oprogramowania, proces tworzenia oprogramowania składa się z następujących etapów.

I. Analiza i projektowanie

Na tym etapie ustalane są:

- cele oprogramowania,
- wymagania dla oprogramowania,
- technologie programistyczne,
- architektura oprogramowania,
- protokoły komunikacyjne,
- zasoby niezbędne dla wytworzenia oprogramowania.

II. Implementacja

Jest to etap, na którym wytwarzane jest oprogramowanie zgodnie z wymaganiami technologiami. To pole do działania programistów i menadżerów zespołów programistycznych.

III. Testowanie

Na tym etapie sprawdzana jest poprawność działania oprogramowania i jego zgodność z wymaganiami. Oprogramowanie, jeśli nie spełnia przyjętych założeń, powraca do etapu implementacji. Zespoły testujące powinny być rozłączne z zespołami programującymi.

IV. Wdrożenie

Na tym etapie wytworzone oprogramowanie uruchamiane jest w środowisku produkcyjnym (w odróżnieniu od etapu testowania, który przeprowadzany jest w środowisku testowym). Również z tego etapu oprogramowanie może być skierowane ponownie na etap implementacji celem eliminacji zidentyfikowanych usterek.

V. Utrzymanie

Etap ten rozpoczyna się po stwierdzeniu, że oprogramowanie w środowisku produkcyjnym działa stabilnie, innymi słowy: oprogramowanie wchodzi w fazę serwisowania.

VI. Wprowadzanie zmian

Etap ten występuje tylko wtedy, gdy pojawia się potrzeba zmian wymagań co do funkcji spełnianych przez oprogramowanie (na ogół konieczność wprowadzenia zmian ma miejsce, co jest konsekwencją zmieniających się uwarunkowań, w których oprogramowanie pracuje: biznesowych, technologicznych, prawnych).

Naszym zdaniem, dla sukcesu procesu wytworzenia oprogramowania najważniejszy jest etap analizy. Etap ten, mianowicie, przesądza w zasadniczy sposób o przebiegu dalszych etapów procesu. Mnogość dostępnych technologii, w kontekście dalej wspomnianych aspektów innych aspektów decyzyjnych, powoduje, że etap analizy jest bardzo złożony. Ustalenie (wybór) celów, wymagań, technologii, architektury, protokołów i zasobów jest problemem wieloaspektowym, tzn. należy wziąć pod uwagę wiele aspektów związanych z tworzeniem i funkcjonowaniem oprogramowania, takich jak, na przykład: wydajność, niezawodność, dojrzałość, stabilność, wymagania zasobowe, dostępność programistów o różnych specjalizacjach, koszty implementacji, koszty utrzymania, koszty wdrożenia i wiele innych.

4. Warianty stosu technologii

W tym rozdziale artykułu przedstawimy sposób tworzenia wariantów stosu technologii, będących następnie przedmiotem właściwego wyboru. W pierwszym kroku zostanie przedstawiona struktura wariantu stosu. Następnie przejdziemy do opisu listy technologii, na podstawie której generowane są warianty oraz opiszemy mechanizm tworzenia wariantów – stosów technologii.

4.1. Struktura stosu technologii

Przedmiotem decyzji w opisywanym problemie jest wybór stosu technologii, opartego na architekturze trójwarstwowej. Struktura stosu składa się z warstwy prezentacji, warstwy przetwarzania oraz warstwy składowania. Elementami stosu technologii są poszczególne technologie, przeznaczone do realizacji elementów wspomnianych trzech warstw: języki programowania, systemy operacyjne i systemy baz danych.

W warstwie składowania wyróżnia się tzw. logikę A (główny mechanizm odpowiedzialny za realizację zadań warstwy składowania), logikę B (dodatkowy

Wybór oprogramowania przy wielu kryteriach

mechanizm warstwy składowania), bazę A (główna technologia bazy danych), bazę B (dodatkowa technologia bazy danych).

Stosy technologii (warianty decyzyjne) generowane są na podstawie dostępnych informacji o technologiach. Tabela 1, na następnej stronie, przedstawia fragment listy wykorzystanych technologii, obejmujący jednakże wyłącznie języki programowania.

Kat.	Nazwa	Społ	D	Koszt	Pełniona funkcja						
					P	S	O	IA	IB	bA	bB
1	Java	1353781	22	6774	0	0	0	1	1	0	0
1	C	267333	45	6733	0	0	0	1	1	0	0
1	C++	547721	32	6800	0	0	0	1	1	0	0
1	Python	868865	26	6516	0	0	0	1	1	0	0
1	C\#	1165182	17	6774	0	0	0	1	1	0	0
1	JavaScript	1528074	22	5430	0	0	0	1	1	0	0
1	R	215798	24	5733	0	0	0	1	1	0	0
1	PHP	1152101	22	5081	0	0	0	1	1	0	0
1	Swift	174610	3	6833	0	0	0	1	1	0	0
1	Objective-C	281806	3	6533	0	0	0	1	1	0	0
1	Assembly language	26606	68	6893	0	0	0	1	1	0	0
1	Perl	58301	30	4000	0	0	0	1	1	0	0
1	Ruby	190209	22	6516	0	0	0	1	1	0	0
1	Delphi/Object Pascal	139	31	5430	0	0	0	1	1	0	0
1	Go	26595	8	6893	0	0	0	1	1	0	0

Tabela 1. Fragment listy wykorzystanych technologii, obejmujący wyłącznie języki programowania. Skróty: Kat. – kategoria, Społ – społeczność, D – Dojrzałość, P – warstwa prezentacji, S – system operacyjny, O – obsługa poleceń z warstwy prezentacji, IA – logika A, IB – logika B, bA – baza A, bB – baza B.

Atrybuty technologii

I. Kategoria:

- 1 - język programowania,
- 2 - gui,
- 3 - serwer http,
- 4 - baza danych,
- 5 - system operacyjny.

II. Nazwa - nazwa własna technologii.

- III. Społeczność - wskaźnik wielkości społeczności organizującej się wokół technologii. Wartość ta jest obliczana na podstawie liczby wątków związanych z daną technologią na portalu `stackoverflow.com`. Portal `stackoverflow.com` jest jednym z większych portali skupiających społeczność ludzi związanych z technologiami wytwarzania oprogramowania.
- IV. Dojrzałość - wiek technologii obliczany na podstawie roku udostępnienia technologii.
- V. Koszt - szacunkowy koszt pracy. Wielkość ta uwzględnia tylko koszty pracy, nie uwzględnia kosztów związanych z zakupem oprogramowania. Dane pochodzą z kilku portali internetowych i zostały uśrednione.
- VI. Pełniona funkcja - wszystkie te atrybuty mają wartość 0 lub 1, informującą czy dana technologia może być wykorzystana dla:
- warstwy prezentacji,
 - systemu operacyjnego,
 - obsługi poleceń z warstwy prezentacji,
 - logiki A,
 - logiki B,
 - bazy A,
 - bazy B.

4.2. Tworzenie wariantów – stosów technologii

Na podstawie listy technologii tworzone są wszystkie możliwe warianty stosu technologii. Następnie z tak utworzonego zbioru wariantów usuwane są (ze względów niezawodnościowych) te warianty, dla których wybrana logika A jest taka sama jak wybrana logika B oraz te warianty, dla których wybrana baza A jest taka sama jak wybrana baza B. Dla każdego wariantu sumowane są wartości atrybutów Społeczność, Dojrzałość i Koszt tych technologii, które wchodzi w skład tego wariantu. Te trzy sumaryczne atrybuty liczbowe przyjmujemy za kryteria oceny (typu „im więcej, tym lepiej”) wytworzonych wariantów stosu technologii.

O ile sumaryczne kryterium Koszt nie budzi wątpliwości, to kryteria Społeczność i Dojrzałość, w kontekście sposobu ich tworzenia (sumowanie wartości odpowiednich atrybutów poszczególnych technologii wchodzących w skład wariantu stosu technologii) może być uznane za nadmierne uproszczenie. Uważamy jednak, że sposób ten w przypadku tych dwóch wielkości prawidłowo oddaje relacje pomiędzy wariantami technologii.

Wybór oprogramowania przy wielu kryteriach

Tabela 2 przedstawia pewien podzbiór pełnej listy technologii wraz z ich atrybutami, natomiast Tabela 3 przedstawia wszystkie warianty możliwe do wygenerowania z podzbioru technologii z Tabeli 2.

Kat.	Nazwa	Społ	D	Koszt	Pełniona funkcja						
					P	S	O	IA	IB	bA	bB
1	Java	1353781	22	6774	0	0	0	1	1	0	0
1	C++	547721	32	6800	0	0	0	1	1	0	0
2	wxWidgets	2745	25	6733	1	0	0	0	0	0	0
3	Apache	77477	22	7500	0	0	1	0	0	0	0
3	Nginx	28819	13	7500	0	0	1	0	0	0	0
4	MariaDB	56	8	5666	0	0	0	0	0	1	1
5	Linux	158049	26	10000	0	1	0	0	0	0	0

Tabela 2. Wybrany podzbiór pełnej listy technologii wraz z ich atrybutami. Oznaczenia jak w Tabeli 1.

Społ	D	Koszt	P	S	O	IA	IB
2945833	117	37781	wxWidgets	Linux	Apache	Java	Java
2139773	127	37807	wxWidgets	Linux	Apache	Java	C++
2139773	127	37807	wxWidgets	Linux	Apache	C++	Java
1333713	137	37833	wxWidgets	Linux	Apache	C++	C++
2897175	108	37781	wxWidgets	Linux	Nginx	Java	Java
2091115	118	37807	wxWidgets	Linux	Nginx	Java	C++
2091115	118	37807	wxWidgets	Linux	Nginx	C++	Java
1285055	128	37833	wxWidgets	Linux	Nginx	C++	C++

Tabela 3. Wszystkie możliwe do wygenerowania warianty stosu technologii z listy technologii z Tabeli 2. Oznaczenia jak w Tabeli 1. Kolumny bA i bB, zawierające we wszystkich pozycjach tę samą bazę danych, zostały pominięte.

Wśród wygenerowanych wariantów, przedstawionych w Tabeli 3, jest tylko jedna technologia dla warstwy prezentacji (wxWidgets), jeden system operacyjny (Linux), dwie różne technologie dla warstwy obsługi poleceń (Apache i Nginx) i dwie różne technologie dla warstwy logiki (Java i C++), oraz jedna baza danych. Tabela przedstawia również, dla każdego wariantu, sumaryczne wartości odpowiednich atrybutów.

Przykład powyższy jest czysto ilustracyjny, chociażby dlatego że, jak to było wcześniej powiedziane, nigdy nie są tworzone warianty stosu technologii z wykorzystaniem jednej technologii baz danych. Co więcej, każdy konkretny przypadek praktyczny może nieść ze sobą specyficzne uwarunkowania i ograniczenia, które muszą być dodatkowo uwzględniane na etapie tworzenia wariantów.

5. Wspomaganie wielokryterialnych problemów decyzyjnych

5.1. Wprowadzenie

Wybór stosu technologii jest typowym wieloatrybutowym (a po wyborze atrybutów, które będą kryteriami wyboru - wielokryterialnym) problemem decyzyjnym.

W ostatnich latach nastąpił rozwój praktycznych metod wielokryterialnego wspomaganie podejmowania decyzji, a także wzrost zainteresowania ich praktycznymi zastosowaniami. Przykładami takich konkretnych zastosowań są: infrastruktura dróg (Żabicki, Gardziejczyk, 2014), wybór projektów inwestycyjnych (Kukułka, Wirkus, 2017), wybór technologii (Mucha i in., 2012). Zastosowania dotyczą także wielu innych dziedzin, w tym takich jak: zarządzanie zasobami leśnymi i wodnymi, planowanie zagospodarowania terenów, zagadnienia logistyczne i transportowe, konstruowanie maszyn i urządzeń, planowanie terapii nowotworowej, czy wreszcie handel i marketing.

W odróżnieniu od problemów optymalizacyjnych, w których pojęcie optymalnego wariantu jest jednoznacznie zdefiniowane (wybieramy wariant o najlepszej wartości kryterium), w problemach wielokryterialnych na ogół nie istnieje wariant o najlepszych wartościach wszystkich kryteriów. Możemy jedynie ograniczyć pole wyboru do wariantów, dla których nie istnieją warianty, względem wszystkich kryteriów nie gorsze, a względem co najmniej jednego kryterium lepsze (czyli nie istnieją dla tego wariantu inne warianty, które, we wskazanym tu sensie, są względem niego dominujące). Takie warianty nazywamy Pareto optymalnymi.

Zatem, wynikiem procesu decyzyjnego w przypadku problemu wielokryterialnego jest wariant wybrany spośród wariantów Pareto optymalnych, bo pomijanie wariantów zdominowanych to kwestia elementarnej racjonalności. Będzie to zawsze wybór subiektywny, uwarunkowany indywidualnymi cechami decydenta: wiedzą, doświadczeniem, ale także przekonaniem, systemem wartości, egoizmem bądź altruizmem, presją czasu, aktualnym nastrojem, panującymi trendami, etc. Te czynniki nie dają się sformalizować, stąd dotychczasowe niepowodzenia metod starających się ująć preferencje decydenta w postaci funkcji (funkcji użyteczności), co pozwoliłoby sprowadzić problem wyboru wariantu do zadania optymalizacyjnego.

Pozostaje zatem stworzyć mechanizm, który w możliwie jak największym stopniu odsłaniałby relacje pomiędzy wariantami Pareto optymalnymi i powodował, że decydent podejmowałby decyzje w sposób najbardziej świadomy. Odpowiedzią na to są metody wielokryterialnego wspomaganie decyzji, w których decydent identyfikuje wariant najbardziej preferowany (spośród wariantów Pareto optymalnych) w wyniku przeglądu, pełnego lub częściowego, wariantów Pareto optymalnych.

Za Herbertem Simonem (Simon, 1977) przyjmuje się, że proces podejmowania decyzji składa się z czterech faz: formułowanie, modelowanie, wybór wariantu, przegląd. W tym artykule koncentrujemy się na wyborze wariantu, natomiast więcej na temat pozostałych faz można znaleźć w źródłowej pracy (Simon, 1977), a także w pracach Kaliszewskiego i Podkopaeva (2010) oraz Kaliszewskiego i innych (2016).

5.2. Model formalny problemu wyboru wariantu

Następujący ogólny model formalny ujmuje wszystkie wielokryterialne problemy decyzyjne, niezależnie od ich natury, a więc także rozważany w tym artykule problem wyboru stosu technologii.

Elementami modelu są: warianty dopuszczalne x , ich zbiór X_0 , k kryteriów zadanych funkcjami $f_l(x)$, $l = 1, \dots, k$, $k \geq 2$. Przyjmujemy, że wszystkie kryteria są typu „im więcej, tym lepiej”, to znaczy, że gdy rozpatrujemy je indywidualnie, interesują nas ich maksymalne wartości.

Model ma następującą postać:

$$\begin{aligned} & \text{vmax } f(x) \\ & x \in X_0, \end{aligned}$$

gdzie vmax oznacza operację wyznaczenia wszystkich wariantów Pareto optymalnych (zgodnie z narratywną definicją podaną powyżej). Wyznaczanie numeryczne wszystkich wariantów Pareto optymalnych (co może być pracochłonne lub nawet, w przypadku, gdy zbiór wariantów X_0 jest nieskończony, niemożliwe) należy rozumieć w ten sposób, że istnieje metoda postępowania pozwalająca na wyznaczenie dowolnego z takich wariantów.

Wyznaczenia pojedynczego wariantu Pareto optymalnego dokonuje się poprzez rozwiązanie odpowiednio skonstruowanego problemu optymalizacyjnego z funkcją zbudowaną na wszystkich kryteriach. Zwykle do tego celu wykorzystuje się funkcję liniową ważoną

$$\sum_{l=1}^k \lambda_l f_l(x),$$

gdzie $\lambda_l > 0$, $l = 1, \dots, k$, są parametrami funkcji (wagami). Wariantem Pareto optymalnym jest wariant, który przy ustalonych wagach **maksymalizuje** tę funkcję (przy założeniu, że wszystkie kryteria są typu „im więcej, tym lepiej”). Funkcja ta jest jednak ułomna, mianowicie za jej pomocą nie można, w ogólnym przypadku, wyznaczyć wszystkich wariantów Pareto optymalnych (por. Kaliszewski, 2016; Kaliszewski i Podkopaev, 2010; Kaliszewski i in., 2016). Taka ułomność może się objawić na przykład w przypadku problemu wyboru stosu technologii.

Dlatego też posłużymy się ważoną funkcją Czebyszewa, o której wiadomo, że za jej pomocą można wyznaczyć wszystkie warianty Pareto optymalne (Kaliszewski, 2016; Kaliszewski i Podkopaev, 2010; Kaliszewski i in., 2016). Zdefiniujemy element \hat{y} , którego współrzędnymi są:

$$\hat{y}_l = \max_{x \in X_0} f_l(x), \quad l = 1, \dots, k.$$

Ważona funkcja Czebyszewa ma postać

$$\max_l \lambda_l (\hat{y}_l - f_l(x)), \quad (5.1)$$

gdzie $\lambda_l > 0$, $l = 1, \dots, k$, są parametrami funkcji (wagami). Wariantem Pareto optymalnym jest wariant, który dla ustalonych wag **minimalizuje** tę funkcję na zbiorze wariantów dopuszczalnych X_0 (przy założeniu, że wszystkie kryteria są typu „im więcej, tym lepiej”). Co więcej, dla każdego wariantu Pareto optymalnego istnieją takie wagi $\lambda_l > 0$, $l = 1, \dots, k$, że wariant ten minimalizuje funkcję Czebyszewa na zbiorze X_0 . Jednak, żeby to ostatecznie stwierdzenie było prawdziwe, należy zastąpić we wzorze (5.1) element \hat{y} elementem y^* , gdzie $y_l^* = \hat{y}_l + \epsilon$, $\epsilon > 0$, $l = 1, \dots, k$, natomiast ϵ jest pewną małą wartością. Takie podstawienie dokonuje się na etapie obliczeń optymalizacyjnych i nie zmienia ono w żaden istotny sposób przedstawionego poniżej sposobu sterowania procesem wyboru wariantu decyzyjnego.

5.3. Sterowanie procesu wyboru wariantu preferencjami decydenta

Mając zdefiniowany model wielokryterialnych problemów decyzyjnych, a także sposób wyznaczania wariantów Pareto optymalnych, pokażemy teraz jak sterować procesem wyboru wariantów Pareto optymalnych za pomocą preferencji decydenta.

Zacznijmy od spostrzeżenia, że element \hat{y} , zwany często elementem idealnym, niesie dla decydenta następującą informację:

- współrzędna $\hat{y}_l = \max_{x \in X_0} f_l(x)$, $l = 1, \dots, k$, określa maksymalną wartość kryterium l jaką można osiągnąć na zbiorze wariantów decyzyjnych,
- dowolny element $f(x)$, $x \in X_0$, przedstawia sobą zestaw ustępstw co do wartości poszczególnych kryteriów względem ich wartości maksymalnych; wielkość tych ustępstw wynosi

$$\hat{y}_l - f_l(x), \quad l = 1, \dots, k.$$

Skoro decydent, nie mogąc wybrać w ogólnym przypadku wariantu o ocenie równej \hat{y} , musi iść na ustępstwa (kompromisy) co do wartości $f_l(x)$, $l = 1, \dots, k$,

(czyli wartości funkcji kryterium l dla wariantu x), to można zakładać, że będzie chciał to robić w sposób świadomy:

„Jeżeli nie jestem w stanie uzyskać \hat{y} (czyli nie jestem w stanie uzyskać maksymalnych możliwych wartości kryteriów równocześnie), to godzę się (muszę się zgodzić) na ustępstwa, ale niech ustępstwa te będą dokonywane według wskazanych przeze mnie proporcji”.

Przykład: *„pogorszeniu wartości wybranego kryterium o jedną jednostkę niech towarzyszy pogorszenie pozostałych kryteriów o dwie jednostki.”*

Wektor o współrzędnych dodatnich, których wartości określają proporcje ustępstw, nazywamy wektorem ustępstw. Wektory ustępstw będziemy dalej oznaczać przez τ .

Półprostą o równaniu

$$y = \hat{y} - t\tau, \quad t \geq 0,$$

będziemy nazywać półprostą kompromisu. Półprosta kompromisu jest zbiorem punktów, które zachowują zadane przez decydenta proporcje ustępstw (wektor ustępstw τ) w wartościach kryteriów, poczynając od wartości określonych przez współrzędne elementu y^* .

Istnieją dwa sposoby zadawania kierunku ustępstw: bezpośredni, polegający na jawnym wskazaniu wektora τ , lub pośredni, za pomocą wskazania tzw. punktu bazowego. Punktem bazowym może być dowolny element y przestrzeni R^k , spełniający warunek $y_l < \hat{y}_l$, $l = 1, \dots, k$. Wówczas $\tau = \hat{y} - y$.

Niezależnie od sposobu zadawania kierunku ustępstw, jego określenie definiuje półprostą kompromisu. Te dwa sposoby wskazywania kierunku ustępstw reprezentują dwa całkowicie różne podejścia do pozyskiwania informacji od decydenta.

W przypadku jawnego wskazywania kierunku ustępstw τ , preferencje są wyrażane w formie proporcji, akceptowanych przez decydenta ustępstw. Taki sposób wyrażania preferencji określamy jako atomistyczny, gdyż odwołuje się on do informacji szczegółowej (niezagregowanej).

W przypadku wskazywania punktu bazowego, preferencje są wyrażane w formie preferowanych wartości funkcji $f_l(x)$, $l = 1, \dots, k$, co pośrednio definiuje akceptowane przez decydenta ustępstwa. Ten sposób wyrażania preferencji określamy jako holistyczny, gdyż odwołuje się on do informacji zagregowanej.

Minimalizacja funkcji (5.1) na zbiorze X_0 z parametrami (wagami) $\lambda_l = (\tau_l)^{-1}$, $l = 1, \dots, k$, wyznacza warianty Pareto optymalne, dla których odpowiadające im $f(x)$ leżą na lub, w pewnym sensie, najbliższej półprostej kompromisu, a zatem odpowiadają preferencjom decydenta, wyrażonym w formie wektorów τ (por. Kaliszewski, 2016; Kaliszewski i Podkopaev, 2010; Kaliszewski i in., 2016).

6. Model decyzyjny wyboru stosu technologii - przypadku użycia

W tym rozdziale przedstawimy wyniki działania modułu wspomagania decyzyjnego, opracowanego w ramach pracy dyplomowej przez pierwszego z autorów, na przykładzie problemu wyboru oprogramowania dla systemów informatycznych klasy CRM i ERP. W obu przypadkach ogólny model (5.1) określony jest dwoma kryteriami (wybranymi atrybutami technologii): „Społeczność” i „Dojrzałość” oraz zbiorem wszystkich wariantów stosu technologii (zbiór X_0), przy czym sposób tworzenia tych wariantów został opisany w rozdziale 4. Ponadto, w przypadku systemów informatycznych klasy ERP zbiór X_0 jest ograniczony dodatkowym ograniczeniem kosztowym.

Opisany tu model został zaimplementowany w autorskim programie.

6.1. System klasy CRM

Przedstawimy obecnie wyniki działania modelu decyzyjnego na przykładzie systemów informatycznych (oprogramowania) klasy CRM.

Systemy informatyczne klasy CRM (ang. *customer relationship management*) są systemami wspierającymi procesy zachodzące pomiędzy klientem a firmą. Systemy te wspomagają pracę takich działów w firmie jak: obsługa klienta, sprzedaż, marketing, działy zarządcze. Systemy CRM powinny wspierać wszystkie etapy kontaktu z klientem, począwszy od fazy pierwszego kontaktu, a skończywszy na usługach posprzedażowych (Januszewski 2017).

Technologie, wykorzystane do stworzenia systemu powinny być na tyle popularne, aby można było zlecić projekt dowolnej firmie programistycznej. Pozwala to przyjąć, subiektywnie i bez wykluczania innych możliwych opcji, wartości współrzędnych wektora ustępstw: $\tau_1 = 0.9$ (relatywnie duża gotowość do ustępstw w zakresie kryterium „Społeczność”); $\tau_2 = 0.1$ (relatywnie mała gotowość do ustępstw w zakresie kryterium „Dojrzałość”). Po wprowadzeniu wartości tych parametrów, program wygenerował wyniki, przytoczone w Tabeli 4.

Liczba wariantów: 3643200.

Ocena idealna (dojrzałość, społeczność): $\hat{y} = (273; 4475316)$

Wariant idealny: nie istnieje.

Obliczenie y^* : $y^* = (274; 4475317)$ ($\varepsilon = 1$)

Wyznaczenie wariantów Pareto optymalnych dla zadanego τ :

wariant 1
technologie: HTML, CSS, Linux, Node.js, Java, JavaScript, Oracle, MySQL,
wartości kryteriów dla wariantu 1:
koszt: 51134, dojrzałość: 174, społeczność: 4475316,

wariant 2
technologie: HTML, CSS, Linux, Node.js, Java, JavaScript, MySQL, Oracle,
wartości kryteriów dla wariantu 2:
koszt: 51134, dojrzałość: 174, społeczność: 4475316,

wariant 3
technologie: HTML, CSS, Linux, Node.js, JavaScript, Java, Oracle, MySQL,
koszt: 51134, dojrzałość: 174, społeczność: 4475316,

wariant 4
technologie: -> HTML, CSS, Linux, Node.js, JavaScript, Java, MySQL, Oracle,
wartości kryteriów dla wariantu 4:
koszt: 51134, dojrzałość: 174, społeczność: 4475316.

Tabela 4. Wyniki dla systemu klasy CRM.

Program generuje wszystkie warianty Pareto optymalne dla zadanego τ ; w tym przypadku takich wariantów jest 4.

Zgodnie z wymaganiami dla systemów klasy CRM, wszystkie wybrane (Pareto optymalne) warianty stosu technologii umożliwiają realizację współpracy z klientami za pomocą przeglądarki internetowej (technologia HTML, JavaScript, CSS). Wszystkie warianty wskazują też na system Linux z Node.js jako serwer (co jest możliwe). Logika systemu, w zależności od wariantu, to Java lub JavaScript, natomiast baza danych to Oracle lub MySQL.

Na tym etapie widać, że model decyzyjny zaproponował bazę Oracle lub MySQL, które są bazami relacyjnymi. Można to uznać za błędny wynik, ponieważ nie ma sensu korzystać z dwóch baz relacyjnych. Lepsze wydawałoby się rozwiązanie łączące bazę relacyjną z bazą np. obiektową. Podobna zależność pojawia się w wariantach drugim, w którym technologia JavaScript jest wykorzystywana jako technologia pomocnicza, natomiast Java jako główna. W przypadku, kiedy dla oprogramowania serwerowego wybieramy Node.js, to lepszym wyborem byłby JavaScript dla technologii podstawowej. Te dwa przykłady wskazują na mnogość możliwych dodatkowych warunków, które powinny być uwzględniane w modelu wyboru stosu technologii. Na tym etapie dodatkowe ograniczenia wspomnianego rodzaju nie były uwzględniane – zostały one pominięte, aby przedstawiony problem decyzyjny był możliwie jak najczytelniejszy.

Przyjęty model wyboru oprogramowania jest modelem statycznym i nie uwzględnia kolejności zakupu poszczególnych modułów oprogramowania, które to

zakupy mogą być rozłożone w czasie. Dodatkowa analiza uzyskanych rozwiązań wskazuje, że uwzględniając wartość bieżącą projektu, wariant pierwszy będzie wariantem droższym, ponieważ już na samym początku musimy zainwestować w zakup bazy Oracle, która jest bazą podstawową (baza A) i dlatego musi być dostępna od samego początku eksploatacji systemu. Natomiast w drugim wariantcie, w którym bazą podstawową (baza A) jest baza MySQL, baza ta jest dostępna za darmo.

6.2. System ERP

Systemy klasy ERP (ang. *enterprise resource planning*) w swym pierwotnym założeniu służą do efektywnego planowania i zarządzania zasobami w przedsiębiorstwie. W wyniku ich rozwoju są one obecnie stosowane do całościowego zarządzania przedsiębiorstwem (Słownik Terminologii Logistycznej, 2006).

Podobnie jak w przypadku systemów klasy CRM, przyjmujemy, subiektywnie i bez wykluczania innych możliwych opcji, wartości współrzędnych wektora ustępstw: $\tau_1 = 0.7$; $\tau_2 = 0.3$. W tym przypadku uzupełnimy model o ograniczenie na koszt projektu: koszt $\leq 40\,000$ [zł]. Po wprowadzeniu wartości tych parametrów, program wygenerował wyniki, przytoczone w Tabeli 5.

Liczba wariantów: 3643200.

Liczba wariantów po uwzględnieniu ograniczenia na koszt projektu: 400

Ocena idealna (dojrzałość, społeczność): $\hat{y} = (194; 3177622)$

Wariant idealny: nie istnieje.

Obliczenie y^* : $y^* = (195; 3177623)$ ($\varepsilon = 1$)

Wyznaczenie wariantów Pareto optymalnych dla zadanego τ :

wariant 1

technologie: HTML, CSS, Linux, Node.js, JavaScript, Perl, MongoDB, SQLite,

wartości kryteriów dla wariantu 1:

koszt: 39360, dojrzałość: 148, społeczność: 3177622,

wariant 2

technologie: HTML, CSS, Linux, Node.js, JavaScript, Perl, SQLite, MongoDB,

wartości kryteriów dla wariantu 2:

koszt: 39360, dojrzałość: 148, społeczność: 3177622,

wariant 3

technologie: HTML, CSS, Linux, Node.js, Perl, JavaScript, MongoDB, SQLite,

koszt: 39360, dojrzałość: 148, społeczność: 3177622,

wariant 4

technologie -> HTML, CSS, Linux, Node.js, Perl, JavaScript, SQLite, MongoDB,

wartości kryteriów dla wariantu 4:

koszt: 39360, dojrzałość: 148, społeczność: 3177622.

Tabela 5. Wyniki dla systemu klasy ERP.

Jest to bardzo podobny wynik jak w przypadku systemu CRM. I tym razem program wygenerował cztery warianty równoważne pod względem wartości kryteriów. Wynika to stąd, że w obu przypadkach mamy tę samą listę technologii z takimi samymi kryteriami. Zmieniły się natomiast wartości współrzędnych wektora ustępstw oraz zostało dodane do modelu ograniczenie na koszt systemu. Różnice te sprawiły, że model decyzyjny wyznaczył inne technologie w dwóch kategoriach: logika i baza danych.

7. Podsumowanie

W pracy przedstawiona została możliwość wykorzystania metod wielokryterialnego wspomaganie podejmowania decyzji do problemu wyboru stosu technologii. Podane dwa przykłady ilustrują zaledwie jedną iterację procesu wyboru wariantu, który ma doprowadzić decydenta do wyboru wariantu najbardziej preferowanego, zgodnie z jego preferencjami. Przedstawiony sposób sterowania procesem wyboru wariantów Pareto optymalnych pozwala w analogiczny sposób realizować kolejne iteracje procesu decyzyjnego.

Zastosowano metodę opartą o wykorzystanie funkcji Czebyszewa w celu sprowadzenia modelu wielokryterialnego do postaci klasycznego problemu optymalizacyjnego. Dzięki takiemu podejściu uzyskano pewność, że żaden Pareto optymalny stos technologiczny nie zostanie apriorycznie wyeliminowany z rozważań.

Przedstawione podejście, tak jak też i żadne inne, nie jest w stanie ani zobjektywizować ani zautomatyzować procesu podejmowania decyzji. Ciągłe suwerenem takiego procesu, z jego wszystkimi ograniczeniami i subiektywnym postrzeganiem rzeczywistości, pozostaje człowiek. W opinii autorów tego artykułu, twórcom i propagatorom metod formalnych nie pozostaje nic innego niż uznać ten fakt z należąą pokorą.

Bibliografia

- Bobryk P. (2018) Metody wielokryterialne wspomaganie decyzji oraz ich zastosowanie przy wyborze technologii tworzenia oprogramowania. Praca magisterska, Wyższa Szkoła Informatyki Stosowanej i Zarządzania, Wydział Informatycznych Technik Zarządzania, Warszawa.
- Januszewski A. (2017) *Funkcjonalność Informatycznych Systemów Zarządzania*, t. 1. Wydawnictwo Naukowe PWN.
- Kaliszewski I. (2016) On variant selection mechanisms in interactive MCDA - engineering versus reverse engineering. *Journal of Multi-Criteria Decision Analysis*, **23**(1-2), 40-48.

- Kaliszewski I., Podkopaev D. (2010) *Komputerowe wspomaganie podejmowania decyzji: ujęcie wielokryterialne*. Skrypt Wyższej Szkoły Informatyki Stosowanej i Zarządzania, Warszawa.
- Kaliszewski I., Miroforidis J., Podkopaev D. (2016) *Multiple Criteria Decision Making by Multiobjective Optimization - A Toolbox*. Springer.
- Słownik Terminologii Logistycznej* (2006) Biblioteka Logistyka, ILiM, Poznań.
- Żabicki P., Gardziejczyk W., (2014) Wybrane aspekty analizy wielokryterialnej w projektowaniu obejść drogowych. *Budownictwo i Architektura* **13**(1), 213-222.
- Mucha Z., Mikosz J., Generowicz A. (2012) Zastosowanie analizy wielokryterialnej do wyboru technologii w małych oczyszczalniach ścieków. *Środowisko*, Wydawnictwo Politechniki Krakowskiej, 4, 145-155.
- Kukułka A., Wirkus M. (2017) Metody wielokryterialne wspomaganie decyzji oraz ich zastosowanie w opracowaniu metody oceny niepotokowych procesów produkcyjnych. *Innowacje w zarządzaniu i inżynierii produkcji*, t. 1, Ryszard Knosala (red.) Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 612-623.
- Simon H. (1977) *The New Science of Management Decision*. Prentice Hall.

SOFTWARE SELECTION UNDER MULTIPLE CRITERIA

Paweł Bobryk^{*}
Ignacy Kaliszewski^{†}**

^{*} Warsaw School of Information Technology, Faculty of Information Technology Management, master programme in Computer Aided Management Support.

^{**} Warsaw School of Information Technology,
Faculty of Information Technology Management;

[†] Systems Research Institute, Polish Academy of Sciences.

We present an application of multiple criteria decision analysis to software selection, as it is taught to students of the Faculty of Information Technology Management of the Warsaw School of Information Technology, during the course “Computer Aided Management Support”.

Keywords: decision analysis, multiple criteria problems, software selection