Raport Badawczy

Research Report

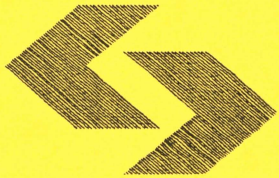
RB/46/2002

System informatyczny do wspomagania podejmowania decyzji wykorzystujący oceny preferencji decydenta

O. Hryniewicz, P. Nycz

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:
Prof. dr hab. inż. Olgierd Hryniewicz

Warszawa 2002

System informatyczny do wspomagania podejmowania decyzji wykorzystujący oceny preferencji decydenta.

Celem projektu jest przetestowanie oraz wdrożenie metod podejmowania decyzji wykorzystujących oceny preferencji decydenta. Niniejsze opracowanie jest poświęcone implementacji systemu informatycznego przeznaczonego do wspomagania decyzji wykorzystującego wymienione wyżej techniki. Program został napisany w systemie C++ Builder firmy Inprice corp. Źródłem informacji oraz podstawą teoretyczną projektu jest artykuł O. Hryniewicza oraz P. Nycza „Komputerowe wspomaganie decyzji wykorzystujące oceny preferencji decydenta".

Celem programu jest dostarczenie użytkownikowi narzędzia, które wspomagałoby oraz ułatwiałoby jego pracę w procesie podejmowania decyzji. Dołożono starań aby program był przyjazny użytkownikowi, od którego wymaga się tylko minimalnej wiedzy informatycznej związanej z obsługą komputera.

W procesie podejmowania decyzji można wyszczególnić trzy etapy

- definiowania problemu (wprowadzanie danych).
- optymalizacji (podejmowania decyzji).
- prezentacji wyników.

Chcąc uczynić pracę z programem jak najbardziej naturalną program także podzielono na trzy moduły odpowiadające poszczególnym etapom rozwiązywania problemu decyzyjnego.

Każdy z modułów jest odrębnym elementem programu, co systematyzuje pracę użytkownika oraz ułatwia poruszanie się po systemie.

Definiowanie problemu.

Przystępując do rozwiązywania nowego problemu musimy zdefiniować nowy projekt (problem decyzyjny). W tym celu należy za pomocą menu PLIK utworzyć nowy projekt (komenda „Nowy Projekt"). Po wykonaniu polecenia ukaże się okno dotyczące projektu. Zawiera ono kilka zakładek, za pomocą których pogrupowano poszczególne parametry.

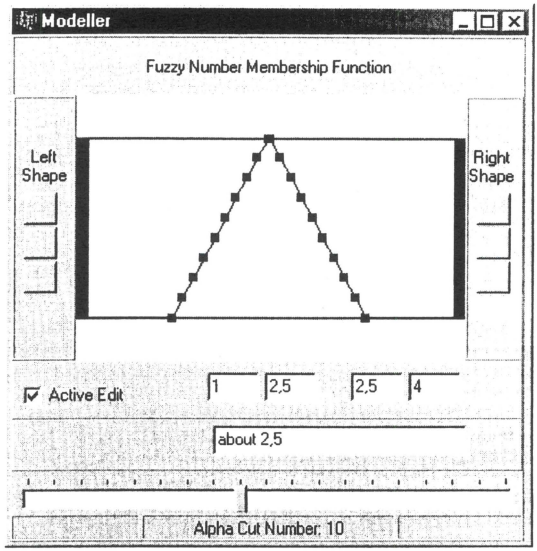
Zakładka „Informacje ogólne" zawiera pola do wypełnienia danymi takimi jak nazwa projektu oraz krótkiej notki na jego temat (np. data, przeznaczenie, autor projektu).

Informacją wejściową do poszukiwania rozwiązań preferowanych jest informacja o miarach preferencji przypisywanych różnym wartościom zmiennej decyzyjnej biorąc pod uwagę różne kryteria.

Parametry dotyczące zmiennych decyzyjnych umieszczono na osobnej zakładce nazwanej „Zmienne decyzyjne". Definiując zmienną decyzyjną umieszczamy w niej informacje takie jak jednostki, nazwę, krótki komentarz itp.

Opisane we wspomnianym artykule podejście do wyznaczania optymalnych decyzji wymaga od użytkownika określenia zbiorów rozmytych opisujących nieprecyzyjne ograniczenia. W opisywanym tu programie komputerowym procedurę określania miar preferencji jest rozbito na dwa etapy. W fazie wstępnej użytkownik określa swoje preferencje poprzez podanie liczby rozmytej o postaci trapezowej zapisanej w formacie (A,B,C,D), gdzie odcinek (A,D) jest podstawą trapezu, a dla wartości zmiennej z przedziału (B,C) funkcja przynależności rozpatrywanego zbioru rozmytego przyjmuje wartość równą jeden. Jak łatwo zauważyć, szczególnymi przypadkami trapezowych liczb rozmytych są liczby rozmyte trójkątne, liczby rozmyte prostokątne (służące do opisu „ostrych" ograniczeń), a także liczby nierozmyte (rzeczywiste). Przewidywane jest wprowadzenie takiego rozwiązania, by użytkownik na tym etapie mógł również formułować oceny w sposób werbalny, na przykład korzystając z wbudowanego słownika określeń. Na przykład, może określić swoje preferencje używając sformułowania: preferuję rozwiązanie o wartości około 5. Dane o rozmytych ograniczeniach zapisywane są w specjalnie zaprojektowanym arkuszu, który może podlegać edycji. W drugiej fazie definiowania swoich preferencji użytkownik może zmodyfikować postać wprowadzonej funkcji przynależności wykorzystując do tego celu specjalny edytor graficzny nazwany „MODELLER"-em.

Narzędzie to jest uruchamiane przez podwójne kliknięcie myszą na nazwę zmiennej decyzyjnej. Wprowadzona w fazie wstępnej trapezowa liczba rozmyta zostaje wyświetlona na ekranie edytora. MODELLER może pracować w dwóch trybach: prezentacji oraz edycji. W pierwszym z nich wyświetlana jest funkcja przynależności rozmytego ograniczenia jedynie w celu wizualizacji liczby. W drugim trybie – edycji, istnieje możliwość modyfikacji funkcji przynależności zbioru do wymaganych kształtów. Zmiany trybu pracy dokonuje się przez jednokrotne kliknięcie myszą na kontrolkę typu „checkbox" opisaną „Active Edit". Wprowadzona w fazie wstępnej trapezowa liczba rozmyta tym razem zostaje wyświetlona na ekranie edytora wraz z punktami stanowiącymi granice jej α-cięć. Punkty te można przeciągać w poziomie przy pomocy myszy, modyfikując w ten sposób funkcję przynależności opisującą rozpatrywane rozmyte ograniczenie. Omawiany edytor funkcji przynależności przedstawiono na poniższym rysunku.



Edytor graficzny funkcji przynależności

Po dokonaniu modyfikacji kształtu funkcji przynależności opisującej ograniczenie zostaje ona zapisana w postaci zbioru swoich α-cięć. W arkuszu wyświetlane są wyłącznie zmodyfikowane wartości (A,B,C,D), ale sama funkcja przynależności nie musi być już liczbą rozmytą o postaci trapezowej. Wykorzystując oba rodzaje edycji liczb rozmytych możemy opisać interesujące nas rozmyte ograniczenia będące opisem miary preferencji odnośnie wartości analizowanej zmiennej decyzyjnej.

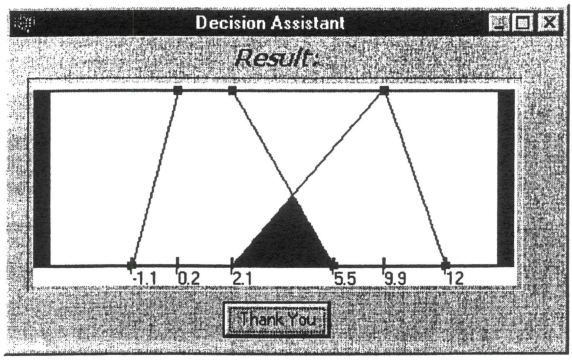
Poszukiwanie rozwiązania zdefiniowanego problemu (optymalizacja).

Kolejnym krokiem po zdefiniowaniu problemu jest wyznaczenie jego rozwiązania. Po wprowadzeniu opisu rozmytych ograniczeń program poszukuje zbioru rozwiązań optymalnych w sensie maksyminowym. Szczegóły opisano w wyżej wspomnianym artykule. Czas niezbędny do otrzymania wyniku jest bardzo różny i zależy od ilości zdefiniowanych zmiennych decyzyjnych, zasobów komputera itp.

Do celów obliczeniowych została wykorzystana specjalnie zaprojektowana i zaimplementowana biblioteka funkcji i procedur umożliwiająca dokonywanie operacji arytmetycznych z użyciem zbiorów rozmytych. Poświęcono jej cały podrozdział zamieszczony na końcu niniejszego opracowania.

Prezentacja wyników.

Po zakończeniu obliczeń ukaże się okno z zapytaniem skierowanym do użytkownika dotyczącym prezentacji wyników. W przypadku udzielenia odpowiedzi twierdzącej ukaże się okno nazwane „DECISION ASSISTANT", na którym ukazuje się oznaczone na czerwono znalezione rozwiązanie problemu. Prezentacja wyników analizy w postaci graficznej ułatwia wybór właściwej decyzji. Na poniżej zamieszczonym rysunku przedstawiona została konstrukcja zagregowanej miary preferencji na podstawie informacji o poszczególnych rozmytych ograniczeniach.



W dalszych etapach rozwoju systemu planuje się zwracanie wyników przez system w formie zdań języka naturalnego, przez co system stanie się jeszcze bardziej przyjazny użytkownikowi.

Moduł obliczeniowy umożliwiający dokonywanie operacji arytmetycznych na zbiorach rozmytych.

Moduł obliczeniowy został zaprojektowany w technologii obiektowej, która obecnie jest najczęściej stosowanym podejściem przy tworzeniu systemów informatycznych. Do implementacji użyto języka C++, który jest bardzo elastyczny oraz dostępny na wielu platformach sprzętowych i sofware'wych co umożliwia przenoszenie kodu i używanie go przy budowie nowych systemów.

Podstawową jednostką danych jest liczba rozmyta opisana funkcją przynależności. Jest ona zdefiniowana w postaci klasy „TFuzzySet" na podstawie której tworzy się poszczególne instancje obiektów - liczb rozmytych. Na funkcję przynależności zbioru rozmytego składają się α – cięcia, które są jak wynika z ich definicji - przedziałami. Do reprezentacji przedziałów stworzono klasę TInterval. Poniżej pokrótce zaprezentowano kolejno klasę TFuzzySet oraz TInterval.

Klasa danych typu „TFuzzySet" zawiera następujące pola:

- FalphaCutNum – typu całkowitego przeznaczona jest do przechowywania ilości α – cięć, które reprezentują zbiór rozmyty.
- MSFunction – typu „TInterval", gdzie „TInterval" jest klasą definiującą typ danych przedziałowych. Pole to jest tablicą wskaźników do obiektów – przedziałów reprezentujących funkcję przynależności.
- FStatus – pole typu całkowitego. Przeznaczone jest do przechowywania stanu obiektu. W przypadkach krytycznych gdy nastąpiło niepowodzenie, np. z powodu wystąpienia dzielenia przez zero, z tego pola można odczytać jego przyczynę. Wyszczególniono następujące stany:
  a) ST_OK. – Wszystkie operacje zostały wykonane poprawnie.
  b) ST_DIV_BY_ZERO – błąd dzielenia przez zero.
  c) ST_FLOAT_OVERFLOW – błąd przepełnienia- występujący przypadku przekroczenia granicy reprezentowalności liczbą zmienno-przecinkową.
  d) ST_FLOAT_UNDERFLOW – błąd niedomiaru występujący w przypadku liczb bardzo małych (bliskich zeru). Sygnalizacja problemów z reprezentacją bardzo małych wartości.
  e) ST_BAD_LCUT_INDEX – odwołanie się do α – cięcia o nieprawidłowym indeksie.

f) ST_BAD_LEFT_BOUND – Próba ustalenia lewej granicy α – cięcia w nieprawidłowy sposób.

g) ST_BAD_RIGHT_BOUND Próba ustalenia lewej granicy α – cięcia w nieprawidłowy sposób.

Klasę „TFuzzySet" wyposażono w następujące metody:

- **Konstruktory**
  a) Konstruktor domyślny (bezparametrowy), tworzący liczbę rzeczywistą równą zero. W tym wypadku wartości wszystkich α – cięć zostają ustawione na wartość zerową.
  b) Konstruktor o czterech parametrach, które ograniczają liczbę rozmytą trapezoidalną (A,B,C,D) – patrz rozdział definiowanie problemu.

- **Ustalanie wartości funkcji przynależności.**
  a) SetMSFunction(int,double,double) – funkcja przyjmuje trzy argumenty, kolejno: numer α – cięcia, lewą granicę przedziału, prawą granicę przedziału.
  b) SetMSFunction(int,TInterval) - funkcja przyjmuje dwa argumenty, kolejno: numer α – cięcia, obiekt typu TInterval reprezentujący przedział.
- **Odczytywanie wartości funkcji przynależności**
  a) GetMSFunction(int) - jako argument przyjmuje numer α – cięcia, zwraca natomiast wskaźnik do obiektu typu TInterval który reprezentuje odpowiedni przedział.
  b) GetAlphaCutNum() – zwraca ilość α – cięć.
  c) ChangeAlphaCutsNum(int) – Zmienia ilość α – cięć na ilość podaną jako argument.
  d) GetMSFunValue(double) – zwraca wartość funkcji przynależności w punkcie podanym jako argument.
- **Operacje arytmetyczne.**
  Podczas implementacji operacji arytmetycznych skorzystano z właściwości języka C++, który umożliwia przeciążanie operatorów. Tą metodą zrealizowano operacje
  a) (+) –dodawania.
  b) (-) –odejmowania.
  c) (*) –mnożenia.
  d) (/) –dzielenia.

- **Operacje poszukiwania maksimum oraz minimum**

  Zaimplementowano operacje Max oraz Min, które umożliwiają wyznaczanie rozwiązań zadań optymalizacji metodą maksminową.

  a) Max(TFuzzySet,TFuzzySet) – argumentami są obiekty klasy TFuzzySet a wynikiem wskaźnik do obiektu tego samego typu będący maksimum zbiorów podanych jako argumenty.

  b) Min(TfuzzySet,TFuzzySet) - argumentami są obiekty klasy TFuzzySet a wynikiem wskaźnik do obiektu tego samego typu będący minimum zbiorów podanych jako argumenty.

Klasa TInterval.

- **Pola w zawarte w klasie:**

  a) FUp – górna granica przedziału.

  b) FLow – dolna granica przedziału.

- **Funkcje ustawiające granice przedziału:**

  a) SetBounds(TInterval)– Ustawianie przedziału na wzór innego przedziału.

  b) SetBounds(double,double)–ustawianie kolejno dolnego i górnego przedziału.

- **Konstruktory**

  a) TInterval()- domyslny – ustawia obydwie granice przedziału na wartość równą 0.

  b) TInterval(TInterval) – tworzy przedział identyczny z podanym jako argument.

  c) TInterval(double):- ustawia obydwie granice na wartość równą argumentowi.

  d) TInterval(double,double).- ustawia dwie granice podane jako argumenty.

- **Operatory arytmetyczne**

  Funkcje arytmetyczne zaimplementowano zgodnie z zasadami tzw. arytmetyki przedziałowej. Dostępne są funkcje realizujące następujące operacje

  a) (+)-dodawania.

  b) (-) – odejmowania.

  c) (*)-mnożenia.

  d) (/)-dzielenia.

- **Funkcje dodatkowe**

  a) IntervalInInterval(TInterval) – jeżeli przedział zawiera przedział podany jako argument metoda zwraca wartość logicznej prawdy.

  b) IntervalInInterval(double,double) – przeznaczenie funkcji jak wyżej. Argumenty podane są jako granice przedziału.

c) Intersection(TInterval,TInterval) – wyznacza przedział będący częścią wspólną przedziałów podanych jako argumenty.

d) InInterval(double) – funkcja zwraca wartość logicznej prawdy jeżeli podany argument – punkt znajduje się w przedziale.

e) GetIntLen() – zwraca długość przedziału.

Podsumowanie.

Pojawienie się metod wspomagania decyzji umożliwiających pracę z nieprecyzyjnymi ograniczeniami może wspomóc w efektywny sposób pracę decydentów. Do powszechnego zastosowania ich w praktyce niezbędne jest odpowiednie oprogramowanie, które pozwoli w realnym czasie otrzymywać wyniki obliczeń. Tworzony system ma umożliwić wdrożenie oraz przetestowanie zaproponowanych metod wspomagania decyzji w praktyce. Moduł obliczeniowy dostarcza narzędzia programistom do tworzenia oprogramowania wykorzystującego metody oparte na teorii zbiorów rozmytych. Całość można potraktować jako część większego przedsięwzięcia-sytemu informatycznego prz eznaczonego do wspomagania decyzji z uwzględnieniem preferencji decydenta.