

Hacia una biblioteca de software MADM multiplataforma

J. Francisco Figueroa-Perez^a, Juan C. Leyva-Lopez^b, Edgar O. Pérez-Contreras^c, Alan Ramírez-Noriega^d, Yobani Martínez Ramírez^e.

^a Universidad Autónoma de Sinaloa, juanfco.figueroa@uas.edu.mx, Fuente de Poseidón y Ángel Flores s/n, Los Mochis, Sinaloa 81223, México

^b Universidad Autónoma de Occidente, juan.leyva@uadeo.mx, Blvd. Lola Beltrán y Blvd. Rolando Arjona s/n, Culiacán, Sinaloa 80120, México

^c Universidad Autónoma de Sinaloa, edgar.perez@uas.edu.mx, Fuente de Poseidón y Ángel Flores s/n, Los Mochis, Sinaloa 81223, México

^d Universidad Autónoma de Sinaloa, alandramireznoriega@uas.edu.mx, Fuente de Poseidón y Ángel Flores s/n, Los Mochis, Sinaloa 81223, México

^e Universidad Autónoma de Sinaloa, yobani@uas.edu.mx, Fuente de Poseidón y Ángel Flores s/n, Los Mochis, Sinaloa 81223, México

Resumen

La Toma de Decisiones Multiatributo (MADM) aborda problemas de decisión sobre un conjunto limitado de alternativas disponibles que se caracterizan por múltiples atributos. Los métodos MADM se utilizan para resolver problemas que implican una selección de entre un número finito de alternativas. Un elemento importante para su aplicación a problemas reales está relacionado con la existencia de software que los implemente. Este artículo describe una nueva biblioteca de software MADM (MADM-SL) que se puede utilizar para el desarrollo de nuevas aplicaciones multicriterio. La biblioteca proporciona un conjunto de algoritmos MADM que implementan diversos métodos multicriterio y puede ser extendida a otros métodos. Las evaluaciones de la biblioteca muestran resultados satisfactorios y señalan algunas oportunidades de mejora.

Palabras clave— Librería de Software MADM, Toma de Decisiones Multiatributo, Toma de Decisiones Multicriterio.

Abstract

Multiple Attribute Decision Making (MADM) refers to making preference decisions over available alternatives which are characterized by multiple attributes. MADM methods are used to solve problems involving selection from among a finite number of alternatives. An important element for its application to real problems is related to the existence of user friendly software implementing them. This paper offers a description of a novel MADM Software Library (MADM-SL) which can be used for the development of new multicriteria applications. The library supply a suite of MADM algorithms that offer the functionality of their respective implemented multicriteria method. Evaluations of the library show satisfactory results and point to some opportunities for improvement.

Keywords— Multiple Attribute Decision Making, Multiple Criteria Decision Making, MADM Software Library.

1. INTRODUCCIÓN

Los problemas de Toma de Decisiones Multiatributo son muy diversos y se pueden encontrar en prácticamente cualquier área. MADM es la rama más conocida de la Toma de Decisiones Multicriterio (MCDM) y se refiere a tomar decisiones de preferencia evaluando y priorizando un conjunto limitado de alternativas basadas en múltiples atributos en conflicto [1]. Los métodos MADM se utilizan para encontrar la solución óptima, seleccionar una sola opción o clasificar las opciones. Algunos métodos MADM bien conocidos incluyen AHP, SAW, TOPSIS, VIKOR, ELECTRE, PROMETHEE, SMART, REGIME, ORESTE, QUALIFLEX, entre otros [2], [3].

En esa misma línea, una biblioteca de software es una colección de activos de software escritas en un lenguaje de programación que tiene una interfaz bien definida mediante la cual se pueden invocar desde otros sistemas informáticos [4].

Este artículo describe el desarrollo de una biblioteca de software multiplataforma que proporciona un conjunto de algoritmos de métodos MADM que se pueden utilizar para el desarrollo de nuevas aplicaciones multicriterio importando directamente la biblioteca en el software bajo desarrollo.

Para desarrollar la MADM-SL se utilizó el lenguaje de programación C++ y su evaluación se llevó a cabo valorando las métricas del producto de software obtenidas con la herramienta C and C++ Code Counter (CCCC).

2. ANTECEDENTES

Hoy en día existen varias herramientas de software MCDM que implementan algunos conocidos métodos MADM o MODM. Se puede encontrar una revisión sobre ellas en el sitio web de la Sociedad Internacional de MCDM [5], el cual enlista y describe una amplia gama de este tipo de software que se muestra en la Tabla 1:

Tabla 1: Herramientas de software MCDM

Software	Orientación
ASMO	Problemas de optimización no lineal multiobjetivo
BENSOLVE	Problemas de optimización de vectores lineales
Comprehensive Package for Multi-objective Integer Programming	Problemas de optimización de vectores lineales
Decisionarium	Soporte interactivo de decisiones multicriterio
DESDEO	Framework para optimización interactiva multiobjetivo
DEXi	Modelado cualitativo de decisiones multiatributo
Diviz	Software con algoritmos MCDA / M complejos

ElectioVis	Aplicación multimétodo MCDA.
Entscheidungsnavi ("decision navigation")	Herramienta de apoyo a la toma de decisiones basada en MAUT
FITradeoff	Procedimiento de elicitación para modelos aditivos multicriterio en MAVT
FLO	Software para resolver problemas de ubicación de uno o varios objetivos
GUIMOO	Software de optimización multi- objetivo
IDSS Software	Software MCDM de la Universidad de Poznan, Polonia
IND-NIMBUS	Implementación interactiva del método NIMBUS
IRIS and VIP	Análisis interactivo de inferencia de parámetros para problemas de clasificación multicriterio
MCDA Package for R	Paquete para la toma de decisiones multicriterio para el lenguaje R
MEM - Multiplex Electionis Methodus	Método para el problema multicriterio de elección
PriEsT P	Herramienta para toma de decisiones con AHP
Onlinemoco.com	Software para resolver enteros mixtos multiobjetivo
DecideI	Software MCDM y analítico probabilístico
Bubble Chart Pro OPTIM	Software que integra gráficos de burbujas con SMART
DEFINITE/ BOSDA	Toolbox de varios métodos de evaluación
D-Sight	Herramienta interactiva para toma de decisiones basada en PROMETHEE
Helision	Software MCDM basado en investigación sobre representación de orden
MACBETH for MCDA	Medición de atractivo mediante una técnica de evaluación basada en categorías
MAMCA	Herramienta que permite la inclusión de múltiples actores en el proceso de evaluación y toma de decisiones
modeFRONTIER	Herramienta dedicada a la optimización multiobjetivo y al diseño multidisciplinario
1000Minds	Software para toma de decisiones multicriterio

Fuente: sitio web de la MCDM [5]

Otras alternativas disponibles en Internet incluyen Decision Radar [6], Expert choice [7], Superdecisions [8], HIPRE 3+ [9], Web-HIPRE [10] y Decision Deck [11]. Estas herramientas implementan diferentes métodos MCDM, tanto MADM como MODM, entre los que se encuentran MOLP, NSGA III, AHP, ELECTRE III y MAVT. El software desarrollado incluye programas de escritorio,

aplicaciones basadas en la web, herramientas basadas en matlab, código fuente, frameworks y servicios web. Entre el software que podría utilizarse para el desarrollo de nuevas aplicaciones multicriterio (código fuente, frameworks o webservices) encontramos los siguientes cuatro: 1) Un paquete para programación entera multiobjetivo; 2) DESDEO, un framework de código abierto para la optimización interactiva multiobjetivo; 3) Un paquete MCDA para el lenguaje R que apoya el proceso de ayuda a la toma de decisiones de criterios múltiples y 4) Los servicios web XMCDM, un conjunto de servicios web MCDA que están disponibles para ser utilizados a través de una aplicación propietaria conocida como Diviz o por aplicaciones de terceros. De la revisión de las herramientas anteriores, encontramos que solo una tiene características afines a nuestro trabajo: los Servicios Web XMCDM del Decision Deck Consortium [11]. Su propuesta es interesante, sin embargo, tiene algunos inconvenientes: se prioriza el uso del lenguaje R en sus algoritmos, el cual es interpretado y poco conocido fuera de entornos de investigación, están orientados principalmente a usarse en aplicaciones del sistema operativo Linux, muchos de ellos están fragmentados en las etapas de los métodos que implementan y no en todos los casos los códigos fuente de los algoritmos están disponibles. Así, nuestro trabajo se distingue de éste por el hecho de que los algoritmos están implementados en el conocido lenguaje C++ lo que permite lograr una velocidad de ejecución más eficiente cuando se requiere, son independientes del sistema operativo, implementan métodos completos y todos los códigos fuente están disponibles, lo que permite a un desarrollador ampliarlos o mejorarlos cuando sea necesario.

3. BIBLIOTECA DE SOFTWARE MADM

Los métodos MADM se pueden clasificar en cinco categorías según sus características: scoring, que consiste en evaluar las alternativas mediante operaciones aritméticas básicas; basados en la distancia, cuyo principio básico es obtener la distancia entre cada alternativa y un punto específico; comparación por pares, que comparan alternativas con respecto a un criterio subjetivo; outranking, que consiste en establecer una relación de preferencia sobre un conjunto de alternativas que indica el grado de dominio entre ellas; utilidad/valor, que definen expresiones que determinan el grado de satisfacción de los criterios [12]. La biblioteca propuesta proporciona un conjunto de clases en la que cada una implementa la funcionalidad específica del método MADM asociado. Fue desarrollada en C++, se puede compilar para múltiples plataformas y se puede usar para crear nuevas aplicaciones multicriterio de cualquier tipo. El código fuente de la biblioteca está disponible para su descarga como software libre en <https://github.com/jfFiguroa23/MADM-SL>. Tiene una licencia GPL-3.0 que brinda la libertad de usarlo, estudiarlo, compartirlo y modificarlo. El diagrama de jerarquía de clases MADM-SL se muestra en la figura 1. Fue diseñado de acuerdo con la clasificación

criterio (0,2, 0,2, 0,2, 0,2, 0,2), umbrales de preferencia (50, 24, 1, 24 20), umbrales de indiferencia (25, 15, 0, 12, 10), umbrales de veto (100, 60, 2, 48, 90) y dirección (1, 1, 1, 1, 1; donde 1 = maximizar, 0 = minimizar):

-F,SD,SC,RM,E;
PRJ1,-14,90,0,40,100;
PRJ2,129,100,0,0,0;
PRJ3,-10,50,0,10,100;
PRJ4,44,90,0,5,20;
PRJ5,-14,100,0,20,40;
W,0,2,0,2,0,2,0,2,0,2;
P,50,24,1,24,20;
Q,25,16,0,12,10;
V,100,60,2,48,90;
D,1,1,1,1,1;

Por otro lado, en lo que concierne al ejemplo de código en C++ que utiliza las bibliotecas, este se muestra en las siguientes líneas:

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <stdlib.h>
4. #include <cstring>
5. #include <iostream>
6. #include <iomanip>
7. #include <algorithm>
8. #include <string>
9. #include <fstream>
10. #include <streambuf>
11. #include "ElectreIII.hpp"
12. #include "../LIBRERIAS/utilerias/utilerias.hpp"
13. 13.
14. using namespace std;
15. char NOMBREARCHIVO[300];
16.
17. string CargarArchivo(char sNombreArchivo[])
18. {
19.     ifstream t(sNombreArchivo);
20.     stringstream buffer;
21.     buffer << t.rdbuf();
22.     string sTmp = buffer.str();
23.     findAndReplaceAll(sTmp, "\n", "");
24.     return sTmp;
25. }
26.
27. int main(int argc, char** argv) {
28.     ElectreIII E1(5,5,-1);
29.     #ifdef _WIN32
30.     strcpy(NOMBREARCHIVO,"D:\\artesanias.csv");
31.     #else
32.     strcpy(NOMBREARCHIVO,"/artesanias.csv");
33.     #endif
34.     E1.IngresarDatos(CargarArchivo(NOMBREA
RCHIVO));
35.     E1.Construir();

```

```

36.     cout << "RESULTADO FLUJO NETO:\n" <<
E1.ExplotarFlujoNeto() + "\n";
37.     cout << "RESULTADO DESTILACION:\n" <<
E1.ExplotarDestilacion() + "\n";
38.     return 0;
39. }

```

Para utilizar el algoritmo ELECTRE III de la biblioteca primero se debe crear un objeto del tipo de la clase *ElectreIII* como en 28 pasándole como parámetro al constructor de la clase el número de alternativas, criterios y un valor de corte. Enseguida se carga la matriz de desempeño a través del método *IngresarDatos()* tal como se hace en la línea 34. La matriz debe ser una cadena de texto y tener el formato mostrado en la sección anterior. Después se llama al método *Construir()* el cual genera las matrices de concordancia, discordancia y credibilidad del método las cuales pueden ser consultadas si así se requiere a través de los métodos *ObtenerMatrizConcordancia()*, *ObtenerMatrizDiscordancia()* y *ObtenerMatrizCredibilidad()*. Por último, para obtener el resultado final con este método se dispone de dos métodos de explotación, *ExplotarFlujoNeto()* y *ExplotarDestilacion()*, los cuales se pueden utilizar de forma independiente o en conjunto según se requiera.

6. EVALUACIÓN

La MADM-SL se evaluó utilizando la herramienta CCCC para el análisis de código fuente para obtener algunas métricas del producto de software [16]. Las métricas consideradas en la evaluación permitieron conocer aspectos relacionados con el volumen y complejidad de cada módulo y sus funciones. El resumen de métricas a nivel de proyecto y de procedimiento se muestra en las Tablas 2 y 3.

Tabla 2: Resumen de proyecto MADM-SL

Métrica	General	Por módulo
Número de módulos	6	
Líneas de código	809	134.833
Número ciclomático de McCabe	190	31.667
Líneas de comentarios	107	17.833
Líneas de código por línea de comentario	7.561	
Complejidad ciclomática por línea de comentario	1.776	

Fuente: elaboración propia

Tabla 3: Resumen de métricas de procedimiento de MADM-SL

Nombre de módulo	LDC	CCM	LDCO	L_C	M_C
ELECTREIII	645	175	86	7.500	2.035
MetodoMulticriterio	8	0	0	-----	-----
Superacion	79	11	19	4.158	0.579

Fuente: elaboración propia

Notas: LDC = Líneas de código, CCM = Complejidad Ciclomática McCabe, LDCO = Línea de Comentarios, L_C = Líneas de Código por Línea de Comentarios, M_C = Complejidad Ciclomática por Línea de Comentarios.

Las puntuaciones obtenidas se encuentran dentro de los rangos considerados como “normales” en todos los aspectos evaluados, según lo indica la herramienta CCCC. Entre los resultados obtenidos podemos destacar el Número Ciclomático de McCabe, que es relativamente alto (31.6 por módulo) y representa un "programa complejo y de alto riesgo" que es difícil de probar, comprender y modificar. Creemos que esto se puede mejorar hasta cierto punto con una refactorización, aunque también que está relacionado con la complejidad inherente del algoritmo implementado.

5. CONCLUSIÓN

Los tomadores de decisiones se enfrentan en la actualidad a decisiones difíciles a diario en las que deben considerar simultáneamente una gran número de alternativas y criterios, lo que hace patente la necesidad de herramientas que les apoyen. Así, este trabajo presenta una novedosa biblioteca multiplataforma de métodos MADM (MADM-SL) que proporciona un conjunto de algoritmos que implementan computacionalmente diversos métodos multiatributo, la cual puede usarse como base para construir nuevo software de apoyo a la toma de decisiones.

Los resultados de una revisión del software disponible que implementa métodos MADM muestran la existencia de diferentes desarrollos de este tipo, los cuales incluyen programas de escritorio, aplicaciones basadas en la web, programas basados en matlab, códigos fuente, frameworks y servicios web. Sin embargo, entre las herramientas que pueden usarse para construir nuevos software, solo se encontró una con características similares a la propuesta en este trabajo, pero que carece de varios elementos importantes que aquí se contemplan, tales como la independencia de la plataforma, el licenciamiento gratuito y la disponibilidad completa de los códigos fuente de los algoritmos, lo que la hace muy apropiada para integrarse libremente y desarrollar nuevas aplicaciones multicriterio con flexibilidad.

Las evaluaciones preliminares de la MADM-SL son alentadoras, pero muestran que aún queda trabajo por hacer para mejorar varios aspectos de la biblioteca. En particular, nuestros pasos futuros incluirán abordar las oportunidades de mejora encontradas en la sección anterior, la adición gradual de nuevos algoritmos de cada categoría y el desarrollo de

una arquitectura basada en microservicios para exponer la biblioteca libremente en Internet.

6. REFERENCIAS

- [1] L. Zhang, “Multi-attribute Decision Making,” in *Encyclopedia of Quality of Life and Well-Being Research*, A. C. Michalos, Ed. Dordrecht: Springer Netherlands, 2014, pp. 4164–4166.
- [2] A. Alinezhad, J. Khalili, and others, *New methods and applications in multiple attribute decision making (MADM)*, vol. 277. Springer, 2019.
- [3] G.-H. Tzeng and J.-J. Huang, *Multiple attribute decision making: methods and applications*. CRC press, 2011.
- [4] S. Atkinson and A. Mili, “Software Libraries,” *Encycl. Electr. Electron. Eng. Ed. John Webster. John Wiley Sons*, 1999.
- [5] I. S. on MCDM, “International Society on MCDM,” 2021. - (accessed Jul. 12, 2021).
- [6] S. Balaei, “Decision Radar,” 2021. - (accessed Jul. 12, 2021).
- [7] E. C. Inc, “Expert Choice.” - (accessed Jul. 12, 2021).
- [8] C. D. Foundation, “Super Decisions CDF,” 2021. -
- [9] M. R. I. O. (MFO) and F. K. (FIZ), “HIPRE 3+,” 2021. - (accessed Jul. 12, 2021).
- [10] R. P. Hämäläinen and M. Jyri, “Web-Hipre,” 2021. - (accessed Jul. 12, 2021).
- [11] D. D. Consortium, “Decision Deck,” 2021. <http://www.decision-deck.org/project/> (accessed Oct. 07, 2021).
- [12] W. A. Treadwell, “Fuzzy set theory movement in the social sciences,” *Public Adm. Rev.*, pp. 91–98, 1995.
- [13] C. Kahraman, *Fuzzy multi-criteria decision making: theory and applications with recent developments*, vol. 16. Springer Science & Business Media, 2008.
- [14] M. Rogers, M. Bruen, and L.-Y. Maystre, *ELECTRE and Decision Support: Methods and Applications in Engineering and Infrastructure Investment*. Boston, MA: Springer US, 2000.
- [15] J. R. Figueira, S. Greco, B. Roy, and R. Słowiński, “ELECTRE methods: main features and recent developments,” in *Handbook of multicriteria analysis*, Springer, 2010, pp. 51–89.
- [16] Arnold Stephen, “C and C++ Code Counter(CCCC).” 2020.