



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

División de estudios de posgrado e investigación

Maestría en ciencias de la computación



TESIS

DESARROLLO DE AGENTE VIRTUAL INTELIGENTE VERBAL PARA LA INTERACCIÓN CON USUARIOS DE ACUERDO CON SUS PREFERENCIAS.

Que para obtener el grado de
Maestro en ciencias de la computación

Presenta:

I.S.C. Héctor Arin Núñez Sánchez

G13071831

Director de Tesis:

Dr. Nelson Rangel Valdez

Codirectora de Tesis:

Dra. María Lucila Morales Rodríguez

Cd. Madero

mayo 2021

Autorización de Tesis



Instituto Tecnológico de Ciudad Madero
Subdirección Académica
División de Estudios de Posgrado e Investigación

Cd. Madero, Tam. 18 de mayo de 2021

OFICIO No. : U.025/21
ASUNTO: AUTORIZACIÓN DE
IMPRESIÓN DE TESIS

C. HECTOR ARÍN NÚÑEZ SÁNCHEZ
No. DE CONTROL G13071831
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su Examen de Grado de Maestría en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

"DESARROLLO DE AGENTE VIRTUAL INTELIGENTE VERBAL PARA LA INTERACCIÓN CON USUARIOS DE ACUERDO CON SUS PREFERENCIAS"

El Jurado está integrado por los siguientes catedráticos:

PRESIDENTE:	DRA. LAURA CRUZ REYES
SECRETARIO:	DR. HÉCTOR JOAQUÍN FRAIRE HUACUJA
VOCAL:	DR. NELSON RANGEL VALDEZ
SUPLENTE:	DRA. MARÍA LUCILA MORALES RODRÍGUEZ
DIRECTOR DE TESIS:	DR. NELSON RANGEL VALDEZ
CO-DIRECTOR DE TESIS:	DRA. MARÍA LUCILA MORALES RODRÍGUEZ

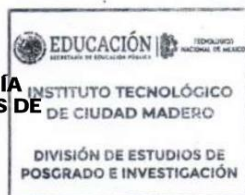
Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE

Excelencia en Educación Tecnológica
"Por mi patria y por mi bien"

MARCO ANTONIO CORONEL GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

c.c.p.- Archivo
MACG 'mdcoa'



Av. 1° de Mayo y Sor Juana I. de la Cruz S/N Col. Los Mangos,
C.P. 89440 Cd. Madero, Tam. Tel. 01 (833) 357 48 20, ext. 3110
e-mail: depi_cdadero@tecnm.mx
tecnm.mx | cdmadero.tecnm.mx



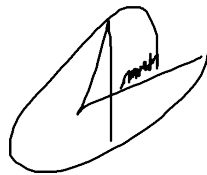
Declaración de Originalidad

Yo, Hector Arin Nuñez Sanchez en mi calidad de autor, declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derecho de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

Además, en caso de infracción a los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirector de tesis, así como al Instituto Tecnológico de Cd. Madero y sus autoridades.

Febrero de 2021, Cd. Madero, Tamaulipas.



I.S.C. Hector Arin Nuñez Sanchez

Agradecimientos

Quiero agradecer a mis padres, que desde el comienzo de mis estudios me han apoyado de toda forma posible, ellos han sido parte de mi fuerza para seguir adelante, su cariño y consejos me han llevado hasta donde me encuentro actualmente y este logro no solamente es mío, si no, también de ellos.

Agradezco a mi asesor de tesis el Dr. Nelson Rangel Valdez por permitirme ser su tesista, por brindarme su tiempo, apoyo y por siempre motivarme para sacar lo mejor de mí.

A los doctores que estuvieron en todo el camino brindando su apoyo y conocimientos de la mejor forma posible, siempre estaré agradecido con sus enseñanzas.

A mis amigos y compañeros que mutuamente nos apoyamos para sacar lo mejor de nosotros, en constantes ocasiones nos motivamos para seguir adelante y compartimos muy buenos momentos de desestrese, compañerismo, diversión, entre otros.

Y por último quiero agradecer a mi hijo y a mi pareja que en los momentos más difíciles me hicieron desistir de rendirme, su compañía y apoyo me hicieron seguir adelante no importaba la adversidad que encontrara, sobra decir que sin ellos este logro no se hubiera completado y por cual este trabajo y éxito, al igual que de mis padres es completamente de ellos.

Tabla de contenido

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 RESUMEN.....	1
1.2 INTRODUCCIÓN.....	3
1.3 JUSTIFICACIÓN.....	4
1.4 OBJETIVOS.....	5
1.4.1 <i>Objetivo General</i>	5
1.4.2 <i>Objetivos específicos</i>	5
1.5 PROBLEMA DE INVESTIGACIÓN.....	6
1.6 ALCANCES Y LIMITACIONES.....	7
1.7 ORGANIZACIÓN DEL DOCUMENTO.....	8
CAPÍTULO 2. MARCO TEÓRICO.....	9
2.1 AGENTE INTELIGENTE.....	9
2.1.1 <i>Componentes o características</i>	9
2.1.2 <i>Arquitecturas de un agente</i>	11
Arquitectura deliberativa.....	11
Arquitectura Reactiva.....	11
Arquitectura híbrida.....	12
2.2 TOMA DE DECISIONES.....	13
2.2.1 <i>Problema de toma de decisiones en grupo</i>	13
2.2.2 <i>Modelos outranking</i>	14
2.2.3 <i>ELECTRE</i>	14
2.3 INOSGA-II.....	16
CAPÍTULO 3. ESTADO DEL ARTE.....	18
3.1 ENFOQUES DEL MANEJO DE PREFERENCIAS.....	18
3.2 MODELOS DE DECISIÓN GRUPAL EN AGENTES.....	19
3.3 ANÁLISIS INTEGRAL DEL ESTADO DEL ARTE.....	20
CAPÍTULO 4. METODOLOGÍA.....	22
4.1 ANÁLISIS Y SELECCIÓN DE ARQUITECTURA.....	22
4.2 DESARROLLO DEL MODELO DE DECISIÓN GRUPAL BASADO EN MÉTODOS DE OUTRANKING.....	23
4.3 INTEGRACIÓN DEL MODELO DE PREFERENCIAS EN LA ARQUITECTURA.....	23
4.4 IMPLEMENTACIÓN.....	24
CAPÍTULO 5. PROPUESTA DE SOLUCIÓN.....	25
5.1 ANÁLISIS Y SELECCIÓN DE ARQUITECTURA.....	25
5.1.1 <i>Análisis de arquitecturas</i>	25
5.1.2 <i>Análisis comparativo de arquitecturas reactivas, deliberativas e híbridas</i>	25
5.2 ARQUITECTURA PROPUESTA.....	27
5.3 MODELO DE DECISIÓN GRUPAL PROPUESTO.....	29
5.4 INTEGRACIÓN DEL MODELO DE PREFERENCIAS EN LA ARQUITECTURA.....	30
5.5 IMPLEMENTACIÓN.....	31
5.5.1 <i>ANDROID</i>	32

5.5.2	<i>Servidor</i>	34
5.5.3	<i>Modelo de integración de preferencias (JAVA)</i>	35
CAPÍTULO 6. EXPERIMENTACIÓN Y RESULTADOS		37
6.1	DESCRIBIR EL DISEÑO EXPERIMENTAL.....	37
6.1.1	<i>Configuración del entorno de experimentación</i>	38
6.1.2	<i>Diseño experimental general</i>	38
6.2	CASO DE ESTUDIO	39
6.3	DESCRIPCIÓN DE LA INSTANCIA	39
6.4	GENERADOR ALEATORIO DE INSTANCIAS	40
6.5	OBTENCIÓN DE MEJORES COMPROMISOS.....	43
6.6	OBTENCIÓN DE CONSENSO.....	44
6.7	INDICADOR DE CALIDAD DE CONSENSO.....	44
6.8	RESULTADOS OBTENIDOS.....	45
6.9	ANÁLISIS DE RESULTADOS.....	46
CAPÍTULO 7. CONCLUSIONES		47

Índice de figuras

FIGURA 1. PROCESO DE TOMA DE DECISIONES.....	13
FIGURA 2. ARQUITECTURA PROPUESTA.....	27
FIGURA 3. CAPA REACTIVA	28
FIGURA 4. CAPA DELIBERATIVA.....	28
FIGURA 5. CAPA MODELADORA.....	29
FIGURA 6. FLUJO DEL PROCESO DE INTEGRACIÓN DE PREFERENCIAS	31
FIGURA 7. DIAGRAMA DE MÓDULOS DEL PROYECTO	32
FIGURA 8. DIAGRAMA DE CLASES ANDROID	33
FIGURA 9. IMPLEMENTACIÓN DE LA APLICACIÓN	34
FIGURA 10. CÓDIGO COMUNICACIÓN POR JSON	35
FIGURA 11. DIAGRAMA DE CLASES JAVA	36
FIGURA 12. RESULTADO FINAL DE LA INTEGRACIÓN DE PREFERENCIAS EN EL BOT.....	36
FIGURA 13. COMPARACIÓN DE RESULTADOS.....	46

Índice de tablas

TABLA 1. UMBRALES EMPLEADOS EN ELECTRE	15
TABLA 2. MODELOS DE DECISIÓN GRUPAL EN AGENTES.....	19
TABLA 3, CARACTERÍSTICAS DE ARQUITECTURAS	25
TABLA 4. DEBILIDADES DE ARQUITECTURAS EN EL PROYECTO	26
TABLA 5. EJEMPLO GMD.....	37
TABLA 6. PARÁMETROS DE EXPERIMENTACIÓN.	38
TABLA 7. PARÁMETROS DE INSTANCIA	39
TABLA 8. INFORMACIÓN CARACTERÍSTICA DE LAS INSTANCIAS	40
TABLA 9. ALTERNATIVAS EN LAS INSTANCIAS	43
TABLA 10. RESULTADOS OBTENIDOS	46

Capítulo 1. Introducción

1.1 Resumen

Los procesos de consenso para la resolución de problemas dentro de la toma de decisiones buscan alcanzar un alto nivel de acuerdo antes de llegar a tomar una decisión. La participación de más de un tomador de decisión o grupos en dichos problemas es cada vez más frecuente de observar, en la vida diaria o en distintas organizaciones, donde es común encontrar a un moderador que guíe el proceso de decisión para alcanzar un mayor nivel de consenso. Esto ha llevado a crear herramientas o sistemas que apoyen los procesos de decisión hasta alcanzar un alto nivel de acuerdo, sin embargo, existe todavía una gran área mejora en lo que respecta al consenso y al manejo de múltiples usuarios, lo que hace necesario la búsqueda de nuevas técnicas, metodologías o el desarrollo de sistemas que apoyen al consenso mediante arquitecturas escalables, que faciliten computacionalmente el manejo de grupos.

Dentro de esta contribución, se presentan los resultados de un sistema de apoyo al consenso basado en una arquitectura híbrida, para un problema de selección de viajes en grupos, que incorpora un modelo de integración de preferencias con el objetivo de maximizar la calidad de satisfacción de los decisores.

Abstract

The consensus processes for solving problems within decision making seek to reach a high level of agreement to reach a decision. The participation of more than one decision maker or groups in these problems is increasingly observed, in daily life or in different organizations, where it is common to find a moderator to guide the decision process to reach a higher level of consensus. This has led to the creation of tools or systems that support decision processes until reaching a high level of agreement, however, there is still a great area of improvement in respect of consensus and the management of multiple users, which makes it necessary to search for new techniques or methodologies, as well as the development of systems that support consensus-based scalable architectures, which computationally facilitate group management.

Within this contribution, the results of a support system based on a hybrid architecture are presented, for a group travel model selection problem, which incorporates a preference integration system with the aim of maximizing the quality of satisfaction. of decision-makers.

1.2 Introducción

Los problemas centrados en la toma de decisiones y la cooperación entre grandes grupos es frecuentemente observado en organizaciones, familias o grupos de trabajo, en los que se trata de alcanzar una solución de un conjunto de alternativas, sin embargo, la mayoría de los problemas de decisión tratan de identificar la mejor o las mejores soluciones considerando simultáneamente múltiples criterios en competencia y estrictamente en conflicto [1]. De acuerdo con Fernández [2] la teoría de toma de decisiones proporciona un paradigma general en relación con ambientes complejos, inciertos y conflictivos, de tal manera que operan con un modelo de preferencias provenientes de agentes humanos, por lo cual, el problema radica en la introducción de más de un DM (tomador de decisiones) que propone un desafío más complejo para el consenso dentro de la selección de la mejor solución.

Para mejorar la obtención de resultados, los procesos de alcance de consenso donde expertos discuten y modifican sus preferencias, son guiados por un moderador para alcanzar un alto nivel de acuerdo antes de tomar una decisión, como resultado de ello, existen diferentes modelos de consenso dentro la literatura, de los cuales se han implementado múltiples sistemas de apoyo (SAC), donde algunos de ellos reemplazan a la figura del moderador humano [3], sin embargo estos aún carecen de un alto nivel de consenso y del manejo de múltiples usuarios.

Algunos de los softwares que utilizan un proceso de toma de decisión y tratan de emular el comportamiento humano son los denominados agentes, por lo cual, en este documento se propone un modelo de integración de preferencias basado en métodos outranking, los cuales se utilizan para resolver problemas de ordenamiento o de decisión donde se dispone de información parcial [4], para integrar las preferencias de todos los DMS en un solo conjunto de parámetros que logre alcanzar un mayor índice de aceptación, e integrar dicho modelo dentro de una arquitectura de agente que actúe como un sistema de apoyo al consenso.

1.3 Justificación

La toma de decisiones es un proceso complejo y una de las actividades más comunes de los seres humanos, en la cual constantemente se pueden encontrar situaciones en las que existen varias alternativas, logrando ir de un problema de decisión unipersonal a uno multi-experto o de decisión en grupo. Igualmente considerando a los atributos que se valoran sobre cada alternativa se puede hablar de problemas de decisión multicriterio, por lo tanto, en cualquiera de los dos casos dependiendo del conocimiento que los expertos tengan sobre las alternativas del problema y de los atributos a valorar, el modelo de decisión puede variar.

De acuerdo con [5] el apoyo a la toma de decisiones grupales se vuelve intensamente difícil debido a la presencia de múltiples actores, cada uno con sus propias percepciones sobre la forma en que se debe manejar el problema y tomar la decisión. Desde esta perspectiva, un grupo de decisión puede entenderse como la reducción de diferentes preferencias individuales en un conjunto de datos que expresan una sola preferencia colectiva [6]. A pesar de los logros alcanzados gracias a modelos y SAC propuestos, los procesos de consenso aun presentan algunas limitaciones, como la necesidad de manejar grandes grupos [7], en los que se pueden encontrar expertos con opiniones muy diferentes entre sí. Como consecuencia de ello, el proceso de consenso puede suponer un alto costo de dificultad y procesamiento. De tal modo que dotar a los sistemas actuales con una metodología que evalúe y considere las preferencias para unir la información de cada individuo en uno, posiblemente podría mejorar la calidad de la elección a tomar por el tomador de decisiones que la implemente.

Este trabajo contribuye a la toma de decisiones grupales mediante un modelo de preferencias basado en relaciones de outranking para la toma de decisiones de un grupo, integrado a una arquitectura de agente de software que apoye como el mediador o moderador en el proceso de toma de decisión.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar arquitectura de agente inteligente verbal que integre decisiones de un grupo colaborativo a través de un modelo de preferencias basado en relaciones de outranking.

1.4.2 Objetivos específicos

- Analizar y seleccionar la arquitectura que mejor se adapte al agente a desarrollar
- Desarrollar modelo de decisión grupal basados en métodos de outranking.
- Integrar modelo de preferencias de decisión grupal a la arquitectura de interacción entre usuario y agente.
- Validar desempeño de agentes implementados a partir de la arquitectura definida

1.5 Problema de investigación

El proceso de consenso es un proceso dinámico e iterativo, normalmente coordinado por una figura humana conocida como moderador, encargada de supervisar y guiar a los expertos durante dicho proceso [3], sin embargo, el resultado final (mejor alternativa) puede no ser satisfactoria para todos los expertos, en vista de que cada experto posee una estructura de preferencias diferente. Teniendo en cuenta este problema, se busca integrar las preferencias de cada decisor en un solo modelo de preferencias, que encuentre una alternativa que satisfaga a todos los expertos o decisores maximizando el nivel de consenso, en virtud de que este proceso es complejo para el ser humano se pretende hacer uso de un agente que implemente dicho modelo.

El problema se plantea formalmente de la siguiente forma:

Dados:

1. Un conjunto de datos $D = \{d_1, d_2, d_3, \dots, d_i\}$ que describen las condiciones de un problema P .
2. Un conjunto acciones potenciales $A = \{a_1, a_2, a_3, \dots, a_j\}$, que desea hacer un grupo de usuarios sobre los datos de P .
3. Un conjunto $U = \{u_1, u_2, u_3, \dots, u_k\}$ de usuarios que dan su opinión sobre las acciones en A mediante una estructura de preferencias.
4. Un conjunto consistente de criterios $F = \{f_1, f_2, f_3, \dots, f_l\}$, que caracterizan las alternativas de acción sobre los datos de P .
5. Un umbral de consenso mínimo deseado por el grupo, $\mu \in [0,1]$
6. Un moderador(agente) que guie a los participantes a la mejor solución posible.

Se busca:

1. Obtener un solo conjunto de preferencias que integre la de todos los usuarios
2. Una alternativa o solución con un alto nivel de consenso

1.6 Alcances y limitaciones

Limitaciones:

- I. La estrategia de solución solo se basa en métodos de outranking de la familia ELECTRE.
- II. El agente de software no usa lenguaje natural, solamente ciertas frases preestablecidas.
- III. Cada usuario considera el mismo conjunto de alternativas
- IV. Todos los usuarios tienen el mismo grado de importancia

Alcances:

- I. El problema por resolver es la integración de preferencias.
- II. El agente de software atiende al menos 2 personas.
- III. El prototipo de agente de software se realizó en una plataforma móvil (Android)
- IV. Solo se seleccionó e implemento una arquitectura.
- V. La configuración se basa solo en el proceso de selección de solución

1.7 Organización del documento

El presente documento se divide en ocho capítulos;

Capítulo 2 se definen los conceptos teóricos que se utilizan en esta investigación

Capítulo 3 muestra los elementos importantes dentro de la literatura que se revisó como: enfoques del manejo de preferencias, modelos de decisión grupal en agentes y modelos de integración de preferencias, así como un breve análisis comparativo con respecto a este trabajo.

Capítulo 4 describe la metodología empleada para resolver el problema de investigación.

Capítulo 5 se describe la propuesta de solución, así como los detalles de su implementación

Capítulo 6 presenta el diseño de experimentos realizado, los resultados y análisis experimentación.

Capítulo 7 se muestran las conclusiones obtenidas a partir de los resultados y se presentan líneas de trabajo futuras.

Capítulo 2. Marco Teórico

En este capítulo se presentan los aspectos teóricos relacionados con los problemas de toma de decisión, el proceso de toma de decisiones y cómo influye el modelo de preferencias del TD en los problemas de consenso grupal.

2.1 Agente inteligente

El concepto de agente es natural para describir sistemas inteligentes y adaptativos en un escenario distribuido y colaborativo. [11] Un agente inteligente se denomina un software que cumple una o más tareas determinadas con un grado de independencia y una capacidad de interactuar con el entorno en el que se encuentra de forma que pueda desenvolverse de forma efectiva en sus tareas sin requerir de la intervención humana.

Como software un agente posee todas las funciones de un sistema de información que puedan ser programadas, pero lo que le da su calidad de agente es una estructura que identifica cuáles son las sentencias determinadas para cumplir sus objetivos y cuáles hacen parte de su composición como entidad. Un agente [11] es un sistema informático situado en algún entorno, que percibe mediante sensores (entradas sensibles de su entorno), y que a partir de tales percepciones determina (mediante técnicas de resolución de problemas) y ejecuta acciones (de forma autónoma y flexible) que le permiten alcanzar sus objetivos y pueden cambiar el entorno.

2.1.1 Componentes o características

El hecho de que un agente sea una entidad situada en un entorno significa que recibe información de entrada de sensores situados en el entorno y a partir de esta información realiza acciones que de alguna forma implican cambios en el entorno. La autonomía se refiere a que el agente debe ser capaz de actuar sin intervención de humanos o de otros agentes. Que un agente sea capaz de actuar de forma flexible para conseguir sus objetivos implica que el agente debe ser [10]:

- **Reactivo:** capaz de responder en un tiempo adecuado a cambios en el entorno en el que se encuentra situado.
- **Proactivo:** capaz de exhibir un comportamiento oportunísimo, dirigido a obtener sus metas y tomar la iniciativa cuando sea apropiado.
- **Social:** capaz de interactuar con otros agentes (humanos o no) a través de un lenguaje de comunicación entre agentes.

Otras propiedades que se les suelen atribuir en mayor o menor medida a los agentes son:

- **Continuidad temporal:** se considera un agente como un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- **Racionabilidad:** el agente siempre realiza “lo correcto” a partir de los datos que percibe del entorno.
- **Adaptabilidad:** el agente aprende y puede cambiar su comportamiento basándose en este aprendizaje.
- **Movilidad:** capacidad de un agente de trasladarse a través de una red telemática.
- **Veracidad:** asunción de que un agente no comunica información falsa a propósito.

2.1.2 Arquitecturas de un agente

Arquitectura deliberativa

Son aquellas arquitecturas que utilizan modelos de representación simbólica del conocimiento. Suelen estar Basadas en la teoría clásica de planificación, donde existe un estado inicial de partida, un conjunto de planes y un estado objetivo a satisfacer. En estos sistemas parece aceptada la idea de que un agente contenga algún sistema de planificación que sea el encargado de determinar que paso debe de llevar a cabo para conseguir su objetivo [10].

Por lo tanto, un agente deliberativo (o con una arquitectura deliberativa) es aquel que contiene un modelo simbólico del mundo, explícitamente representado, en donde las decisiones se toman utilizando mecanismos de razonamiento lógico basados en la concordancia de patrones y la manipulación simbólica.

Arquitectura Reactiva

Los agentes reactivos se caracterizan porque pueden operar rápida y efectivamente sin la necesidad de procesar una representación simbólica del entorno, ya que representan una categoría de agentes que no posee modelos simbólicos externos del entorno en el que se encuentran [10]. Los agentes reactivos toman decisiones basadas totalmente en el presente, sin hacer uso de lo que ha ocurrido en el pasado, porque no conocen su historia [11]. Este tipo de agentes actúan siguiendo un esquema estímulo-respuesta según el estado actual del entorno en el que están embebidos. La característica más importante de los agentes reactivos es el hecho de que los agentes son relativamente simples e interactúan con otros agentes de manera sencilla.

Arquitectura híbrida

Las arquitecturas híbridas combinan componentes de tipo reactivo con componentes de tipo deliberativo. La parte reactiva interacciona con el entorno y reacciona rápidamente a los eventos que en él se producen, sin invertir tiempo en realizar razonamiento, mientras que la parte deliberativa planifica y se encarga de la parte de toma de decisiones, es decir, realiza tareas a un nivel de abstracción superior[9][10].

Las arquitecturas híbridas pretenden aprovechar los beneficios que proporcionan las arquitecturas reactivas y los beneficios que proporcionan las deliberativas, ya que para la mayoría de los problemas no es adecuada ni una arquitectura puramente deliberativa ni una arquitectura puramente reactiva. Típicamente los sistemas híbridos se diseñan siguiendo una arquitectura jerárquica en capas, en la que las capas más bajas son principalmente reactivas y las capas más altas son principalmente deliberativas.

Diferentes niveles de abstracción:

- Reactivo:
 - Nivel bajo
 - Se toman decisiones en base a los datos recopilados por el agente
- Conocimiento:
 - Nivel intermedio
 - Se centra en el conocimiento que posee del medio
 - Normalmente utiliza una representación simbólica del medio
- Social:
 - Nivel más alto
 - Maneja aspectos sociales del entorno, incluyendo tanto información de otros agentes, como deseos, intenciones, etc.

2.2 Toma de decisiones

Se entiende como toma de decisiones al proceso de selección entre cursos alternativos de acción, basado en un conjunto de criterios, para alcanzar uno o más objetivos [12].

La toma de decisiones es el término que generalmente se asocia con las primeras 5 etapas del proceso de resolución de problemas. Así, la toma de decisiones se inicia al identificar y definir el problema, y termina con la elección de una alternativa, este es el acto de tomar una decisión como se muestra en la Figura 1.

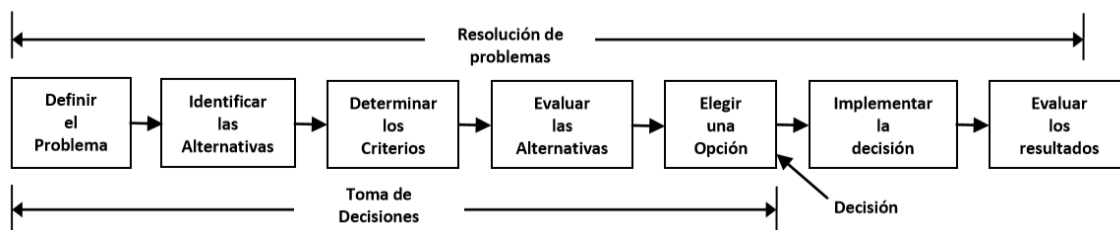


Figura 1. Proceso de toma de decisiones

2.2.1 Problema de toma de decisiones en grupo

Los problemas de Toma de Decisiones en Grupo (o TDG) se caracterizan por la existencia de múltiples expertos, con diferentes puntos de vista, que deben alcanzar una solución común a un problema de toma de decisión [13]. Formalmente, un problema de TDG se compone de los siguientes elementos [26] [27]:

- Un conjunto $X = \{x_1, \dots, x_n\}$ ($n \geq 2$) de alternativas o posibles soluciones al problema.
- Un conjunto $E = \{e_1, \dots, e_m\}$ ($m \geq 2$) de individuos o expertos, que dan su opinión sobre las alternativas en X mediante una estructura de preferencia.

El TDG busca integrar las distintas estructuras de preferencias E en una configuración (conjunto de preferencias) que exprese las opiniones de todos los individuos.

2.2.2 Modelos outranking

Entre los principales métodos de decisión multicriterio, se encuentran los métodos de outranking, ayudan al tomador de decisiones en la resolución de problemas donde muchos puntos de vista deben ser tomados en cuenta, debido a que todas las partes afectadas por el proceso de decisión participan en el mismo y suministran una gran cantidad de información, usan como mecanismo básico el de las comparaciones binarias de alternativas, es decir comparaciones dos a dos de alternativas, criterio por criterio.

De esta forma puede construirse un coeficiente de concordancia C_{ik} asociado con cada par de alternativas (a_i, a_k) .

Estos métodos surgen como consecuencia de las dificultades prácticas de otros métodos de decisión multicriterio, como el de función de valor, el cual requiere condiciones teóricas muy fuertes y restrictivas y una información muy precisa sobre las preferencias del decisor.

Dentro de los métodos de outranking cabe distinguir los métodos de la familia ELECTRE [31] y los de la familia PROMETHEE.

2.2.3 ELECTRE

Uno de los métodos outranking más aceptados y aplicados en problemas reales [28][29][30] por citar algunos, son los métodos ELECTRE (Elimination Et Choix Traduisant la REalité). Los cuales fueron desarrollados originalmente por Roy [31], y en el transcurso de los años se han ido anexando nuevos métodos. Los métodos ELECTRE están basados en la relación de outranking, las diferentes versiones se apoyan en los mismos conceptos fundamentales y difieren en diversas características y en el tipo de problema de decisión al que se aplican. Existen seis versiones de los métodos ELECTRE: I, II, III, IV, TRI (con sus variantes) e IS.

Según Figueira et al.,[35] las preferencias en los métodos ELECTRE se modelan usando relaciones binarias de outranking, S , cuyo significado es “al menos tan bueno como”. Considerando acciones a y b , pueden ocurrir cuatro situaciones:

1. I. aSb y no bSa , es decir, aPb (a es estrictamente preferido a b).
2. II. bSa y no aSb , es decir, bPa (b es estrictamente preferido a a).

3. III. aSb y bSa , es decir, aIb (a es indiferente a b).
4. IV. No aSb y no bSa , es decir, aRb (a es incomparable a b).

De acuerdo con Figueira et al., [35] los métodos basados en outranking se basan en dos grandes conceptos que de alguna manera manejan razones a favor o en contra de una situación de outranking, estos conceptos son los siguientes:

- Concordancia. Para una superación de aSb para ser validada, la mayoría de los criterios deberían estar a favor de esta afirmación.
- Discordancia. Cuando se cumple la condición de concordancia, ninguno de los criterios de la minoría debe oponerse demasiado a la afirmación aSb .

Los métodos ELECTRE comprenden dos procedimientos principales [35], los cuales son:

- Procedimiento de agregación: Construye una o posiblemente varias relaciones de superación, este procedimiento tiene como objetivo comparar de manera integral cada par de acciones.
- Procedimiento de explotación: Deriva recomendaciones a partir de los resultados obtenidos en la primera fase.

Los métodos ELECTRE también hacen uso de los umbrales descritos en la Tabla 1.

<i>Umbral</i>	<i>Descripción</i>
w_i	Representa el peso del criterio i .
p_i	Representa el umbral de preferencia. Es el valor numérico por el cual el DM prefiere una alternativa sobre otra.
q_i	Representa el umbral de indiferencia. Es el valor numérico por el cual el DM es indiferente ante dos alternativas.
v_i	Representa el umbral de veto. Es el valor numérico el cual hace un bloqueo en la relación de superación entre dos alternativas en el criterio i , es decir, opone fuertemente a la afirmación de aSb .

Tabla 1. Umbrales empleados en ELECTRE.

2.3 INOSGA-II

El antecedente de este método fue llamado NOSGA (Non-Outranked-Sorting Genetic Algorithm) y fue inspirado en el algoritmo NSGAI, pero ejerciendo presión selectiva hacia la frontera no estrictamente superada, y buscando Soluciones no dominadas, ver Balderas et al. [19] [20]. El objetivo de este método es aumentar la presión selectiva hacia las soluciones más satisfactorias buscando soluciones no estrictamente superadas con buenos valores en la frontera débilmente superada (o WNS) y de flujo neto (FNS).

En comparación con NOSGA, las principales diferencias son:

- a) encontrar la frontera no estrictamente superada, NOSGA resuelve un problema de optimización de criterio único;
- b) NOSGA-II se basa en una caracterización teórica mejorada de las soluciones más satisfactorias para los MOP;
- c) el uso de Q y K en NOSGA-II;
- d) como consecuencia de los puntos b) y c), NOSGA-II promete un aumento de la presión selectiva hacia la mejor parte de la frontera de Pareto.

El pseudocódigo de NOSGAI se muestra en el Algoritmo 1.

Algoritmo 1. NOSGAI

PROCEDIMIENTO NOSGA-II (L, Numero de generaciones)

0. Inicializar población P
1. *Generar población aleatoria de tamaño L*
2. *Evaluar valores objetivos*
3. *Evaluar σ en $P \times P$*
4. *Para cada $x \in P$, calcular carta $(S_o)_x$*
5. *Generar frentes de igual valor de carta (S_o)*
6. *Asignar a los frentes un rango(nivel) basado en la carta (S_o)*
7. *PARA cada rango C_i HACER*
8. *PARA cada $x \in C_i$ calcular carta (W_{ci}) basado en carta $(F_{ci})_x, \eta$*
9. *Fin PARA*
10. *Generar población hijos Q de tamaño L*
11. *Realizar selección por torneo binario*
12. *Realizar recombinación y mutación*
13. *PARA I = 1 a número de generaciones HACER*
14. *Asignar $P' = P \cup Q$*
15. *Evaluar σ en $P' \times P'$*
16. *PARA cada padre e hijo en P' HACER*
17. *Calcular carta S_o*
18. *Asignar rango (nivel) basado en carta (S_o)*
19. *Calcular η*
20. *Ciclo (interior) agregando soluciones a la*
21. *Próxima generación hasta que se hayan encontrado L individuos*
22. *Fin Para*
23. *Reemplazar P por los L individuos encontrados*
24. *Generar población hijos Q de tamaño L*
25. *Realizar selección por torneo binario*
26. *Realizar recombinación y mutación*
27. *Fin FOR*
28. *Fin PROCEDIMIENTO*

Algoritmo 1. Pseudocódigo de INOSGA- II.

Capítulo 3. Estado del arte

En este capítulo se presentan los trabajos relacionados con la investigación, las aportaciones que servirán como apoyo para distinguir las diferencias entre el tema propuesto y las relaciones que estos presenten con la investigación.

3.1 Enfoques del manejo de preferencias

De acuerdo con Coello [16] existen diferentes enfoques en el manejo de preferencias, las cuales son:

Logros del objetivo: La propuesta era extender el MOGA para acomodar la información de metas como un criterio adicional a la no dominancia para asignar rangos a la población.

Funciones de utilidad: La idea era clasificar un conjunto de soluciones del problema de optimización multiobjetivo (por sus siglas en inglés, MOP) en lugar de clasificar explícitamente los atributos del problema. La información de preferencia también se incorpora en los criterios de supervivencia utilizados por la EA.

Relaciones de Preferencia: Se propusieron el uso de la relación de preferencia binaria que pueden ser expresadas cualitativamente para luego traducirlos en términos cuantitativos (es decir, pesos) para limitar la búsqueda de un EA.

Outranking: Proponen el uso de PROMETHEE (*Preference Ranking Organization Method for Enrichment Evaluation*) combinado con un-EA. Los métodos PROMETHEE pertenecen a la familia de los enfoques de superación (tales como ELECTRE) presentado por Bernard Roy. Los métodos incluyen dos fases:

- La construcción de una relación de superación en los diferentes criterios u objetivos del problema
- La explotación de esta relación para dar una respuesta a la opción del problema de optimización multiobjetivo

Dentro de la lectura se menciona que, al hacer uso de cualquier enfoque para manejar preferencias en un algoritmo evolutivo, existen diversas cuestiones para tomar en cuenta, como preservar la dominancia, la transitividad, la escalabilidad y el hecho de que puede existir más de un tomador de decisión. Se optó por la incorporación de preferencia a través de métodos de superación, ya que, algoritmos como *NOACO*, *H-MCSGA*, *NOSGA*, *O NOSGA-II* han utilizado estos métodos con buenos resultados en tiempo de ejecución y de calidad, otra de las razones por la cual este método se eligió es por uso del valor sigma como referencia al establecer que una solución es mejor o casi tan buena como otra.

3.2 Modelos de decisión grupal en Agentes

Existe dentro de la literatura un cierto número de modelos que tratan de implementar un agente con la capacidad de manejar preferencias de un grupo. Muchos de los proyectos encontrados implementaban diferentes metodologías dentro del proceso de decisión, sin embargo, no todos implementaban un agente con esta metodología, en su mayoría eran solo modelos de preferencias y sistemas de software de apoyo al consenso. La Tabla 2 muestra algunos de los trabajos que se revisaron, de los cuales, las siguientes columnas describen parte de sus características: la primera especifica si el modelo utiliza métodos de superación, la segunda detalla si está integrada a una arquitectura, la tercera columna representa si hay un agente implementado, la cuarta columna si la cantidad de objetivos dentro del problema es más de uno y por último el nombre del algoritmo utilizado para el manejo de preferencias.

<i>Autor</i>	<i>Outranking</i>	<i>Integración con arquitectura</i>	<i>Agente</i>	<i>Cant de criterios >1</i>	<i>Aprendizaje de preferencias</i>
<i>Lee, W. P., At [32]</i>	-	✓	✓	-	<i>AG</i>
<i>Chai & Liu [33]</i>	-	✓	-	✓	<i>FUZZY</i>
<i>Goletsis, At. ,[34]</i>	✓	-	✓	✓	<i>ELECTRE Y PROMETHEE</i>
<i>Rouyendegh & Erkan [8]</i>	✓	-	-	✓	<i>ELECTRE FUZZY</i>
<i>Propuesta de tesis</i>	✓	✓	✓	✓	<i>ELECTRE</i>

Tabla 2. Modelos de decisión grupal en agentes

Aunque los modelos encontrados presentan similitudes con el modelo propuesto, la diferencia que caracteriza a este, es la integración e implementación dentro de una arquitectura de agente inteligente.

Modelo ELECTRE FUZZY

ELECTRE FUZZY es un modelo basado en outranking que utiliza números triangulares difusos $\widetilde{w}_j = (l_j, m_j, u_j)$ para representar la integración de los pesos provenientes de diferentes decisores.

Este modelo solo considera los pesos de los decisores para cada criterio, primero construye los pesos de los decisores en números triángulos difusos normalizados, para después construir una matriz de decisión asociando a cada alternativa un valor para cada uno de los criterios involucrado, esta matriz se normaliza y de manera similar a los métodos de ELECTRE se calculan los índices de concordancia y discordancia, dejando en claro que estos pasos siguen una estrategia parecida, estos índices se integran y se establece una relación entre las alternativas evaluadas.

Este modelo trata de integrar las preferencias de todos los decisores que han participado, usando *outranking* y número difusos triangulares, sin embargo, no considera todos los umbrales de ELECTRE y es necesario normalizar los valores para integrar los datos, lo cual requiere devolver los resultados al formato original, estas dos características pueden suponer una desventaja al consenso de la solución obtenida, debido a que no considera cierta información de relevancia por los umbrales omitidos, y a que los valores finales al ser transformados pierdan cifras o datos relevantes.

3.3 Análisis integral del estado del arte

Dentro de la literatura se han revisado programas y modelos de decisión de grupal basado en outranking que apoyen a los problemas de consenso, dentro de estos ELECTRE un modelo de comparación binaria permite evaluar dos alternativas de acuerdo con los parámetros que lo definen (veto, preferencia, etc.) obteniendo $\sigma(A_1, A_2)$ como una medida de que tan cierta es la frase “ A_1 es al menos tan buena como A_2 ”.

Sin embargo, en un grupo de *DMs* (Tomadores de decisiones) cada *DM* tiene su propio modelo de preferencias por lo que al utilizar electre, cada uno tiene un conjunto propio de parámetros característicos. Electre fuzzy el modelo mencionado anteriormente, trata de tomar los parámetros de cada *DM* e unirlos usando **números triangulares difusos** para construir el parámetro unificado de todos los *DMs* (en este caso peso). Se puede observar que la integración de los diferentes pesos de varios decisores produce una incertidumbre, la cual en el modelo de Rouyendegh and Erkan [8] es tratada a través de números difusos. Dada la relación existente en la literatura para abordar la incertidumbre con intervalos y la capacidad de los agentes para la toma de decisiones, se considera un medio adecuado para definir un modelo de preferencias grupales.

Capítulo 4. Metodología

La metodología de trabajo utilizada consta de tres módulos: análisis y selección de arquitectura, desarrollo del modelo de decisión grupal e integración del modelo de preferencias en la arquitectura.

4.1 Análisis y selección de arquitectura

Para resolver el problema propuesto el objetivo del módulo es seleccionar una arquitectura de agente que cumpla con las características solicitadas para la evaluación del modelo de preferencias, se divide en 4 fases: revisión del estado del arte, análisis de arquitecturas de agentes, identificación de la arquitectura mejor adaptada al problema y el análisis de los componentes de la arquitectura seleccionada. Las actividades asociadas con estas fases son las siguientes:

1. Revisión en la literatura de agentes
2. Revisión de métodos de comparación y medidas desempeño en arquitecturas
3. Análisis del método SAAM (*Software Architecture Analysis Method*)
4. Implementación del método SAAM

La descripción de cada una de las actividades es la siguiente:

La revisión de la literatura de agentes se llevó a cabo con el objeto de expandir el conocimiento previo de agentes inteligentes e identificar las arquitecturas para ser evaluadas.

Dentro de la revisión de los métodos de comparación y medidas de desempeño el objetivo fue encontrar una metodología que permitiera comparar las arquitecturas encontradas, para la selección de la mejor candidata.

Uno de los métodos encontrados para la comparación de arquitecturas fue el método SAAM, el análisis de este método consto de encontrar trabajos donde se implementará y se identificaran los pasos e índices de desempeño que se utilizaron para la comparación.

En la implementación del método SAAM se evaluaron los tres tipos de arquitecturas en agentes (deliberativa, reactiva e híbrida) y se seleccionó la más adecuada de acuerdo con los índices propuestos.

4.2 Desarrollo del modelo de decisión grupal basado en métodos de outranking

El objetivo del módulo es desarrollar un modelo de decisión grupal que integre las preferencias de todos los participantes dentro del grupo usando métodos outranking. Este módulo se divide en 3 fases: análisis de modelos de decisión grupal, identificación de características y adaptación del modelo electre para el manejo de preferencias de un grupo. Las actividades asociadas a este módulo son las siguientes:

1. Análisis de diferentes modelos de integración de preferencias.
2. Identificación de las características de los modelos revisados.
3. Análisis de los elementos del modelo a propuestos
4. Modelo de integración de preferencias propuesto

La descripción de las actividades son las siguientes:

La primera actividad se basó en el análisis de modelos de preferencias en grupos, que utilizan métodos outranking como funciones para la integración de preferencias, así como la identificación de los elementos y funcionamientos en la segunda actividad. Una vez terminado la identificación de las características, se revisó y propuso un modelo de integración de preferencias con uso de intervalos dentro de la tercera y cuarta actividad.

4.3 Integración del modelo de preferencias en la arquitectura

Dentro de este módulo el objetivo principal es la integración del modelo de preferencias desarrollado en la arquitectura para llegar a esta meta, el módulo se divide en 2 fases: diseño del módulo de toma de decisiones basado en modelos outranking y la integración del modelo de preferencias en la arquitectura. Las actividades asociadas a este módulo son las siguientes:

1. Diseño gráfico del modelo de integración de preferencias
2. Diseño del módulo de toma de decisiones
3. Integración del módulo de toma de decisiones dentro de la arquitectura

Las actividades están enfocadas en el diseño del módulo grafico de integración de preferencias dentro de la arquitectura de agente seleccionada.

4.4 Implementación

El último paso dentro de la metodología aplicada fue la implementación integrada de todas las tecnologías diseñadas, El objetivo de este módulo fue probar el funcionamiento correcto del proyecto completo e identificar posibles fallas, para al final corregir cualquier inconveniente. Una actividad adicional e importante dentro del módulo fue el diseño de la experimentación utilizada y la interpretación de los resultados obtenidos.

Capítulo 5. Propuesta de Solución

5.1 Análisis y selección de arquitectura

5.1.1 Análisis de arquitecturas

La Tabla 3 resume las características principales de las arquitecturas mencionadas anteriormente. Tomando en cuenta solo las características, la arquitectura híbrida y la deliberativa coinciden con el agente a proponer.

Arquitecturas	Características principales
Agentes deliberativos	<ul style="list-style-type: none">• Elementos centralizados (planificador)• Representación interna del mundo exterior, esquema <i>estimulo</i> → <i>manipulacion simbolica(razonamiento logico)</i> → <i>respuesta</i>• No implica un agente simple solo: Se pueden construir arquitecturas deliberativas para coordinar comportamientos• Agentes complejos
Agentes reactivos	<ul style="list-style-type: none">• Sin elementos centralizados• Sin representación interna del entorno, esquema <i>estimulo</i> → <i>respuesta</i>• No mantienen historia pasada ni estado• Agentes simples y fáciles de implementar• Interacción con otros agentes muy sencilla
Agentes híbridos	<ul style="list-style-type: none">• Arquitectura en capas• Técnicas reactivas para acciones simples y de bajo nivel y deliberativas para comportamientos complejos

Tabla 3, Características de arquitecturas

5.1.2 Análisis comparativo de arquitecturas reactivas, deliberativas e híbridas

Tomando como referencias los tres tipos de arquitecturas encontradas fue necesario determinar la más adecuada para el problema a resolver. En base a esto, se buscaron estrategias para la comparación de arquitecturas, una de las estrategias encontradas y utilizadas para el análisis comparativo fue el método SAAM (Software Architecture

Analysis Method)[1][2], el cual utiliza el agrupamiento de situaciones que reflejen los atributos de calidad de interés en la arquitectura llamados escenarios, como criterio para evaluar sus componentes. El resultado de este análisis de acuerdo con los escenarios propuestos se muestra en la Tabla 4.

Arquitectura	Debilidades
<p align="center">Arquitectura reactiva Subsumpción</p>	<ul style="list-style-type: none"> • La arquitectura solo funciona con los valores asignados. • No hay aprendizaje. • No existe razonamiento. • Acciones predeterminadas.
<p align="center">Arquitectura deliberativa BDI</p>	<ul style="list-style-type: none"> • Tiempo de respuesta lento debido a los diferentes módulos que trabajan en la generación y razonamiento de las opciones.
<p align="center">Arquitectura híbrida Touring Machines</p>	<ul style="list-style-type: none"> • Su lado reactivo solo funciona mediante valores asignados.

Tabla 4. Debilidades de arquitecturas en el proyecto

Considerando las debilidades y cualidades de las arquitecturas analizadas, Touring machines poseen varias características que se ajustan al agente, como su implementación en ambientes multi-agentes, su modulo para el manejo y control de actividades, y su capacidad para reaccionar de manera reactiva o deliberativa dependiendo la situación, haciendo menos pesada la comunicación con el usuario, por lo cual la arquitectura propuesta fue desarrollada en base a esta arquitectura.

5.2 Arquitectura propuesta

Para el desarrollo de este trabajo se propone una arquitectura híbrida basada en Turing machines, la cual se encuentra dividida en 3 módulos principales como se muestra en la Figura 2.

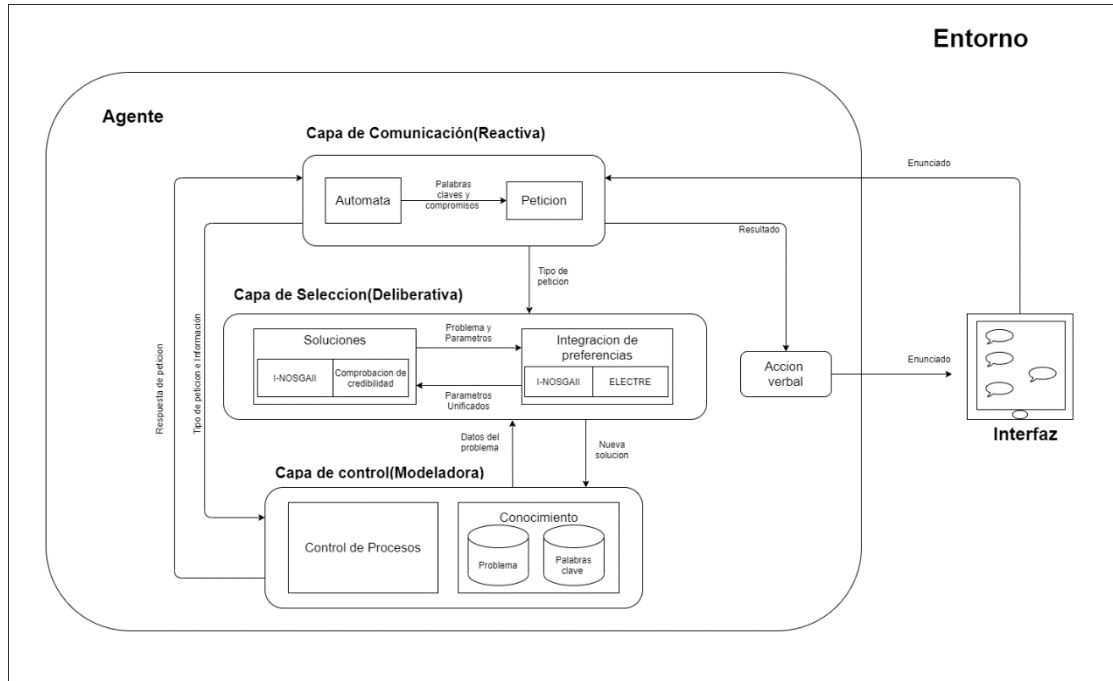


Figura 2. Arquitectura propuesta

Los componentes de la arquitectura se explican en el resto de este apartado.

Capa de comunicación

Se muestra en la Figura 3. Esta capa representa la parte reactiva de la arquitectura, su función es procesar los enunciados de entrada mediante el chat de la interfaz, donde a través del submódulo autómata, se reconocen y obtienen las palabras claves e información necesaria por el siguiente submódulo. Una vez procesados los enunciados, las palabras claves determinan dentro del submódulo petición la acción a ocurrir dentro del agente, estas acciones son peticiones HTTP que permiten comunicarse con el servidor y enviar (Post), obtener (Get), eliminar (Delete) o actualizar (Update), la información de instancia dentro de la base datos o si bien empezar el proceso deliberativo del agente.

Capa de Comunicación(Reactivá)



Figura 3. Capa reactiva

Capa de selección

Se encarga de llevar a cabo el proceso deliberativo del agente, es decir, en esta capa se encuentra el modelo de preferencias grupales, el cual sigue una serie de pasos para poder integrar las preferencias de los usuarios en un solo conjunto de parámetros, que expresen las opiniones conjuntas del grupo. La Figura 4 muestra el esquema de esta capa.

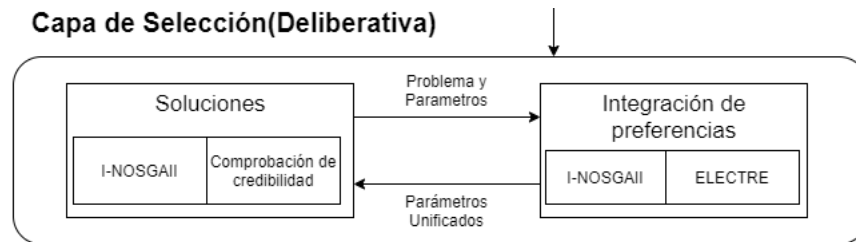


Figura 4. Capa deliberativa

Capa de modelado

La capa de control se encarga de administrar las actividades, conocimientos y procesos del agente y del grupo de decisores (ver Figura 5). El módulo de control de procesos captura y controla las posibles interacciones entre los usuarios, gestiona los procesos y coordina las actividades como el paso de mensajes entre el agente mediador y el grupo.

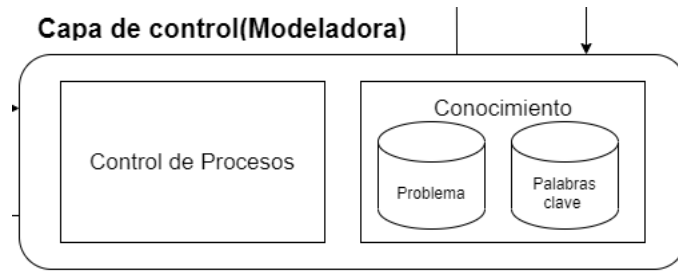


Figura 5. Capa modeladora

5.3 Modelo de Decisión Grupal Propuesto

Haciendo uso de outranking, se propone un problema de optimización como el modelo de decisión grupal propuesto para llevar a cabo el proceso de consenso. Esta sección define el problema propuesto. formalmente.

Dado un conjunto m decisores, denotado por $DM = \{DM_1, DM_2, \dots, DM_m\}$, un modelo de caracterización denotado por DM_i , para cada $0 \leq i \leq m$, definido a través de los parámetros $P_i = (W, V, \eta, d, \lambda)$, un mejor compromiso, denotado por BC_i , para cada DM_i , y un conjunto ROI_i de aproximación a la ROI de cada DM_i de donde BC_i se obtuvo, se busca encontrar el mejor consenso $CDM = (W, V, \eta, d, \lambda)$ tal que al hacer uso del modelo de preferencias caracterizado por los valores de parámetros establecidos en CDM , satisfaga el mayor número posible de los BC_i provistos por los DM_s individuales.

En donde:

n : Número de criterios u objetivos que definen cada mejor compromiso BC_i

W : Vector de n pesos de los objetivos.

V : Vector de n umbrales de veto

η : Vector de n umbrales de credibilidad

δ : Umbrales de credibilidad del outranking

λ : Umbral de concordancia

Formalmente, el modelo se define de acuerdo con la Ecuación 1.

$$\text{Maximizar } f_1 = \sum_{i=1}^m \text{sat}(CDM, BC_i, ROI_i) \quad (1)$$

Sujeto a:

$$CDM \in \Omega$$

Donde:

- Ω Es la región factible para CDM (valores válidos para los parámetros, p.e. suma de pesos igual a 1)
- $NSS(s)$ en ROI_i es el número de soluciones r en ROI_i tal que: [s *Dominates* r] OR [$\sigma(s, r) > \delta$ && $\sigma(r, s) < 0.5$].
- Una solución $s \in ROI_i$ se considera BC_i si posee la fuerza máxima $NSS(.)$ de entre todas las soluciones $r \in ROI_i$, donde $s \langle \rangle r$
- $\text{sat}(CDM, BC_i, ROI_i)$ es la función que evalúa en 1 si BC_i se mantiene como mejor compromiso en ROI_i usando CDM , y 0 en caso contrario.
- BC_i es el mejor compromiso de DM_i
- ROI_i región de interés de DM_i tal que $BC_i \in ROI_i$

5.4 Integración del modelo de preferencias en la arquitectura

La Figura 6 presenta de manera gráfica el flujo de proceso de integración de preferencias. El flujo del proceso de integración comienza con la recopilación de los datos de instancia en la interfaz del chat de Android. Haciendo uso de frases o tokens reconocidos por el agente, este puede responder dependiendo lo que solicite el usuario, una vez obtenidos los datos de entrada que representan la instancia, estos se envían a la memoria del agente, es decir, a la base de datos, en este momento es cuando empieza el proceso deliberativo, la ejecución del modelo de preferencias grupales, el cual con los datos obtenidos (instancia completa) empieza un proceso de dos fases:

1. La fase uno comienza con el uso de *INOSGAI* para obtener los mejores compromisos por cada usuario en el chat, utilizando las estructuras preferenciales de cada uno.
2. La segunda fase hace uso de las soluciones obtenidas en la fase uno, y encuentra un conjunto de parámetros que aumenten el índice de credibilidad de las soluciones obtenidas, de tal manera que estas expresen mayor satisfacción dentro del grupo.

El proceso termina con el resultado siendo mostrado como respuesta al consenso por medio del chat, este flujo se puede observar.

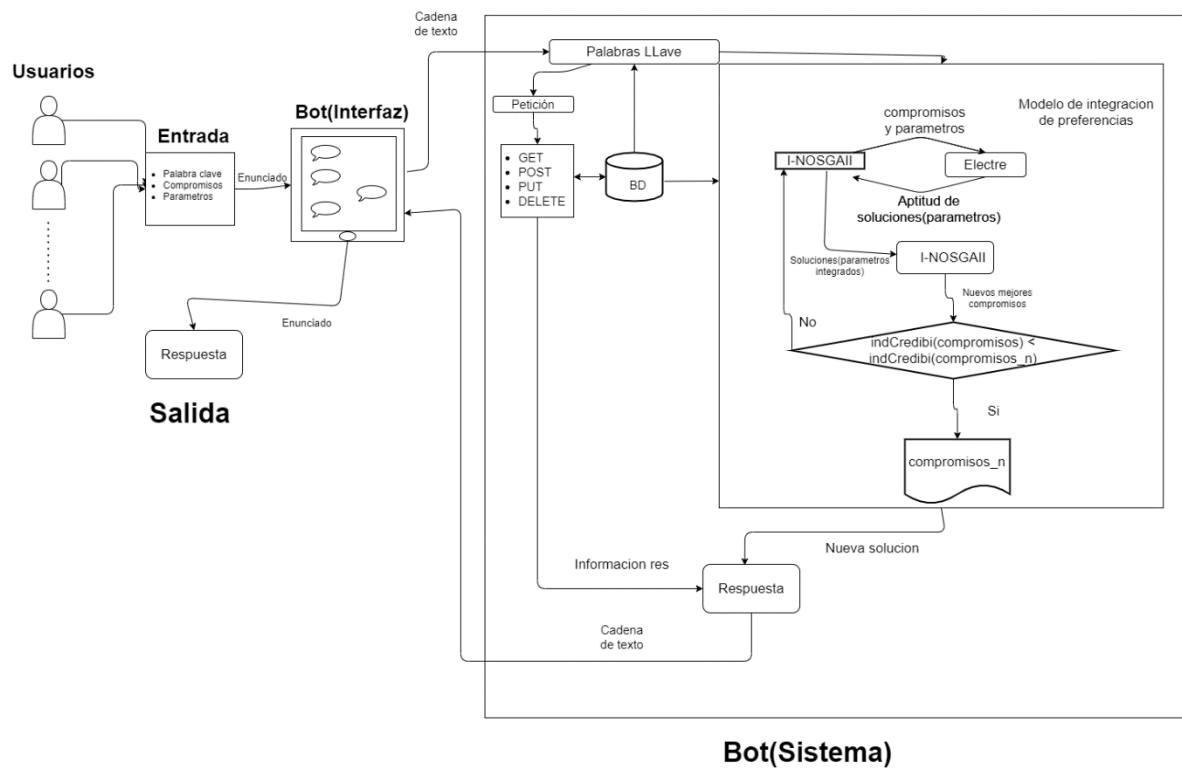


Figura 6. Flujo del proceso de integración de preferencias

5.5 Implementación

El proyecto completo está constituido por 3 módulos importantes los cuales se observa en la Figura 7, cada módulo cumple con las funciones que el agente requiere, desde el proceso de comunicación hasta el proceso de control y deliberación.

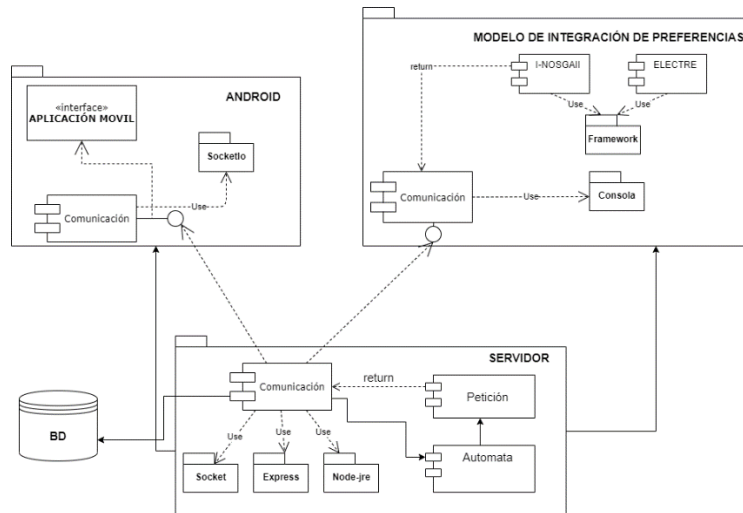


Figura 7. Diagrama de módulos del proyecto

5.5.1 ANDROID

Las funciones dentro del módulo de Android son las siguientes:

- **Recopilar Información.** La información será introducida mediante un chat, todas las sentencias o enunciados deberán tener un orden obligatorio.
- **Mostrar Resultados.** Los resultados o respuestas por parte del agente serán mostrados en el chat, dependiendo de las sentencias que haya recibido el agente
- **Conexión con servidor.** La información recaudada será enviada al servidor para el procesamiento de datos y la recepción de respuesta.

La Figura 8 muestra las clases desarrolladas dentro del módulo del chatBot. Empezando desde el *MainActivity* donde se implementa el reconocimiento de voz para la entrada de datos y siguiendo con la clase *Automata* para el reconocimiento de tokens o palabras clave. Las peticiones http se realizan por medio las clases *Select* o *Insert*, obteniendo o insertando la información de acuerdo con las palabras claves obtenidas y finalizando con la impresión en pantalla por medio de la clase *Imprimir_Chat*.

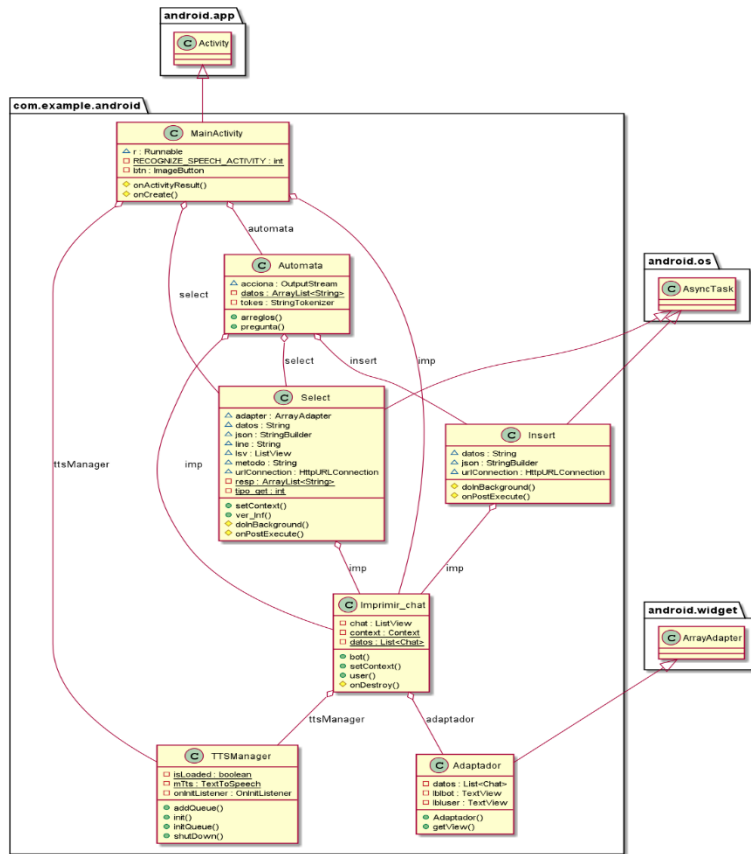


Figura 8. Diagrama de clases Android

Seguendo el flujo descrito anteriormente, la ejecución del proyecto comienza desde la recopilación de los parámetros de preferencias de cada decisor, los cuales se pueden mostrar en la Figura 9.

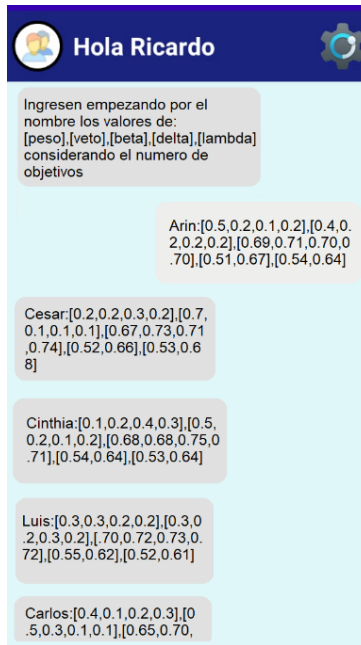


Figura 9. Implementación de la aplicación

5.5.2 Servidor

El servidor es la parte del agente que comunica los módulos de Android y de java, se puede tomar como el módulo de control dentro del agente en donde se transmite la información entre los módulos la Figura 10 muestra una pequeña parte del funcionamiento de comunicación del servidor. Las funciones más importantes que este realiza se detallan a continuación.

- Peticiones
 - De acuerdo con las palabras claves encontradas, el servidor guardara la información, responderá con una frase preestablecida o iniciara el proceso de solución
- Comunicación *BD* Y *MIP* (Modelo de integración preferencias)
 - El servidor es el canal de comunicación entre la aplicación Android, la base de datos y el *MIP*. Guarda y obtiene la información de la *BD*, este también ejecuta el modelo de integración y recibe la respuesta dentro de un *JSON* para al final mandar su respuesta a la aplicación Android

```

app.post('/preferencias', (req, res) => {
  var jsonDMSData = {
    decisor: req.body.decisor,
    pesos: req.body.pesosDMS,
    correo: req.body.vetosDMS
  };
  User.insertUsuario(userData, (err, data) => {
    if (data && data.insertId) {
      res.status(200).json({
        success: true,
        msg: "Se inserto la informacion con exito",
        data: data
      });
    } else {
      res.status(500).json({
        success: false,
        msg: "Error al insertar datos de entrada"
      });
    }
  });
});
});

```

Figura 10. Código comunicación por JSON

5.5.3 Modelo de integración de preferencias (JAVA)

Este módulo ejecuta el proceso de integración de preferencias, dentro del agente ejecuta el algoritmo deliberativo, este se ejecuta para llegar a la toma de decisión final. Las Funciones de este módulo de describen a continuación:

- Lectura de instancias
 - Realiza la lectura de instancia
- Integrar preferencias
 - Al ser ejecutado por el servidor el proceso de integración de preferencias, comienza tratando de unificar en un solo conjunto de parámetros las preferencias de los decisores, utilizando los mejores compromisos trata de encontrar mediante la optimización el conjunto de parámetros unificados que aumente el índice de credibilidad(aptitud) de las soluciones.
- Conexión con el servidor
 - Recibe argumentos por el servidor y devuelve un JSON con el resultado

El diagrama en la Figura 11 es una presentación de las clases dentro del modelo de integración. Las clases *ConsensoProblem* y *TravelingProblem* definen los problemas descritos en las primeras diapositivas y la clase *ConsensoInstance* obtiene los datos de la

instancia, ambas clases utilizan en teoría la misma instancia solo difieren en el caso en donde *ConsensoProblem* utiliza las *ROIs* de los *DMs*.

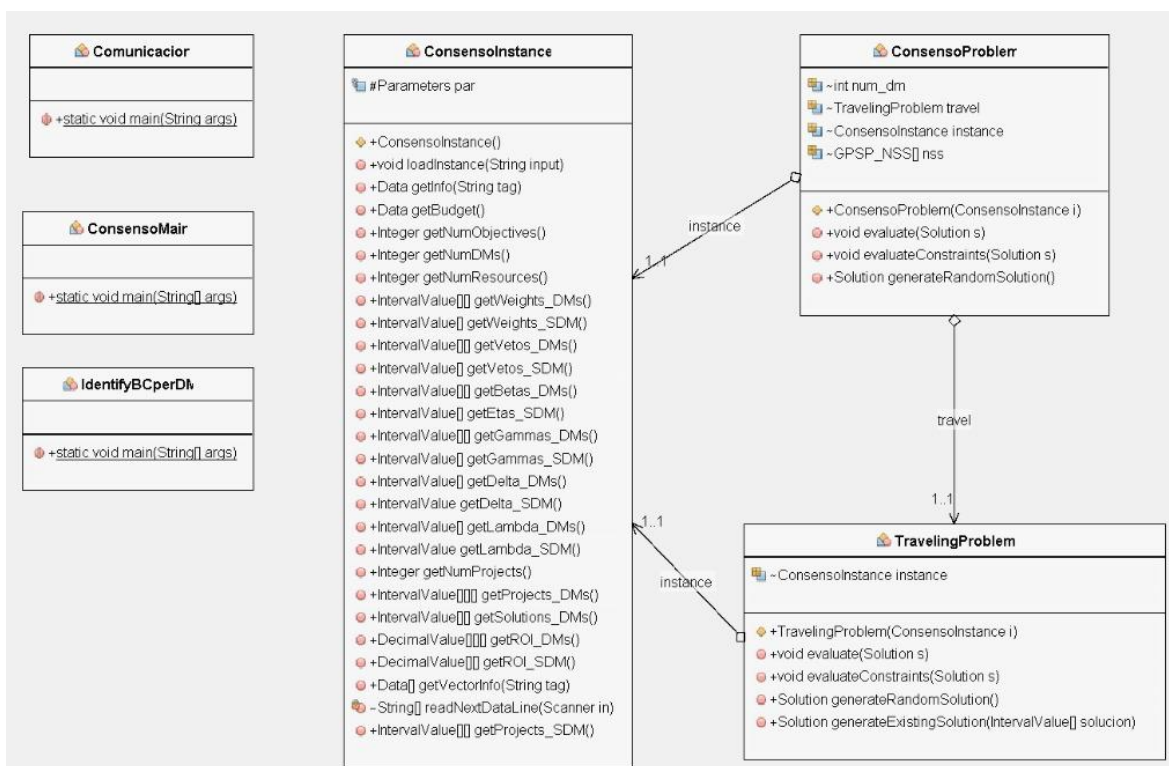


Figura 11. Diagrama de clases JAVA

Al terminar este reporta el resultado mediante la aplicación, como se ve en la Figura 12.

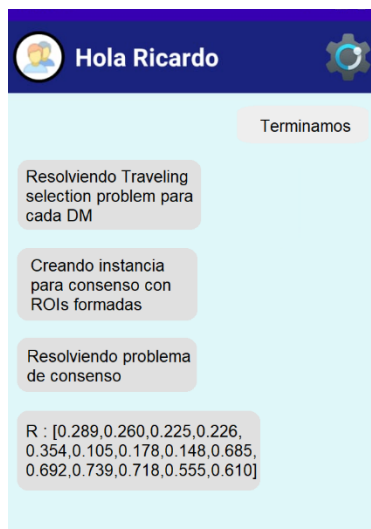


Figura 12. Resultado final de la integración de preferencias en el Bot

Capítulo 6. Experimentación y resultados

6.1 Describir el diseño experimental

Para determinar qué tan bueno fue la implementación del modelo desarrollado, se define una métrica de calidad que compara la cantidad de DMs satisfechos a partir de los resultados obtenidos por el modelo de consenso propuesto (CMD) y el modelo de consenso genérico (GMD).

Este trabajo de investigación compara la propuesta de la sección 5.2 contra un modelo que forma el consenso a partir de los valores extremos (GMD), ejemplo en la tabla 5.

	Pesos w		
	w_1	w_2	w_3
DM₁	0.2	0.7	0.1
DM₂	0.1	0.6	0.3
DM₃	0.4	0.5	0.1
GMD	[0.1-0.4]	[0.5-0.7]	[0.1-0.3]

Tabla 5. Ejemplo GMD

El método propuesto obtiene el mejor consenso usando el siguiente método algorítmico:

1. Leer la información de preferencias de los DMs a través del agente $\{W, V, \eta, \delta, \lambda\}$
2. Con la información preferencial de cada agente se generan los mejores compromisos BC_i para cada DM_i , y se obtiene la ROI_i que es el último frente reportado de donde se extrajo BC_i .
3. Con los mejores compromisos de todos los agentes se resuelve el problema de optimización propuesto en la Sección 5.2, el cual reporta un conjunto de soluciones $\mathbf{s}^*_{\text{mdg}} = \{s^*_{\text{mdg},1}, \dots\}$ con el mismo valor de la función objetivo (es decir el mismo número de DMs satisfechos).

GMD define el consenso como s_{gen}^* los intervalos formados por los valores extremos de cada parámetro de entre los distintos DMs.

Condiciones del Experimento: Por simplicidad el experimento analizará el desempeño de la propuesta para hacer consenso cuando varían dos elementos preferenciales clave, pesos y vetos. Para propósitos del experimento entonces el resto se mantiene fijo entre las instancias.

6.1.1 Configuración del entorno de experimentación

La experimentación se realizó en una maquina con las siguientes características: Marca *HP* con un procesador Intel(R) Core (TM) i5-9300H CPU con 2.40 GHz, Memoria RAM de 16GB y Sistema Operativo Windows 10.

6.1.2 Diseño experimental general

Dentro de la experimentación se proponen 5 instancias, donde varían la cantidad de decisores, de alternativas y los valores contenidos dentro de estas, con la finalidad de observar la variación del nivel de consenso obtenido al utilizar los parámetros de los participantes, los generados por el algoritmo INOSGA-II y los obtenidos mediante el uso de intervalos extremos. El objetivo es analizar y comparar la satisfacción de los decisores haciendo uso del cálculo de sigma para determinar con que parámetros se obtiene un mejor consenso la siguiente tabla muestra los parámetros de experimentación. La Tabla 6 muestra los parámetros que se utilizaran en el experimento.

Parámetro	Cantidad
Decisores	5
Casos de estudio	1
Cantidad de corridas	30
Cantidad de evaluaciones BC	10000
Cantidad de evaluaciones Consenso	100000

Tabla 6. Parámetros de experimentación.

6.2 Caso de estudio

Es un problema de selección de viajes basado en cartera de proyectos y visto desde el caso de estudio, en donde se tiene a un grupo de personas (Familia, Empresa, etc.), escoger un conjunto de destinos, cada uno con la siguiente información: precio (p_i), calificación (c_i), atracciones (a_i) y estancia (e_i) asociados, sin exceder un presupuesto (P) y una cantidad máxima de días de estancia (DM_{max}).

En otras palabras, consiste en seleccionar un grupo de destinos con el objetivo de maximizar: la calificación promedio, la cantidad de destinos y de atracciones, siempre que no supere el presupuesto máximo propuesto y los días de estancia.

El objetivo de la resolución de este problema es obtener los mejores compromisos y ROIs de cada persona dentro del grupo y utilizarlos en el problema de consenso.

6.3 Descripción de la instancia

La Tabla 7 describe los parámetros de las instancias utilizadas en la experimentación.

Parámetro	Descripción
Budget	Presupuesto Disponible
MaxDays	Representa el número de días máximos a viajar
NumObjectives	Número de objetivos
NumDMs	Número de DM
NumResources	Número de restricciones
$\{W, V, \eta, \delta, \lambda, SDM\}$	Representa una matriz de tamaño $m \times n$ donde m representa el índice de DM y n el valor de peso, veto, eta, gamma y lambda de los DMs por cada objetivo de los destinos
NumProjects	Número de destinos
DMS_Projects	Representa una matriz de tamaño $m \times n$ donde m representa el índice de cada destino y $n + 1$ representa los valores del presupuesto y sus objetivos por cada $m - esimo$ destino
Flag	Variable que contiene el valor TRUE si la instancia contiene las ROIs de los DMs y FALSE en el caso contrario
Num_Sol	Número de soluciones en la ROI
DMS_ROI	Representa una matriz que contiene las mejores soluciones de los DMs
SDM_ROI	Representa una matriz que contiene las mejores soluciones del SDM

Tabla 7. Parámetros de instancia

6.4 Generador aleatorio de instancias

Las instancias generadas se producirán aleatoriamente y tendrán las características mostradas en la Tabla 8.

Descripción	Valor
Número de DMs	5
Cantidad de Instancias	30
Presupuesto Fijo	[100000, 100000]
Número Máximo de Días Fijo	60
Número de Objetivos Fijo	4
Número de Restricciones Fijo:	2
Pesos Aleatorios Siguiendo método	[RandomWeights]
Vetos Aleatorios Siguiendo método	[RandomVetos]
Betas Fijos para cada DM y cada objetivo	[0.51 0.67]
Gammas Fijos para cada DM y cada objetivo	[0.51, 0.67]
Deltas Fijos para cada DM	[0.51, 0.67]
Lambdas Fijos para cada DM	[0.51, 0.67]

Tabla 8. Información característica de las instancias

En la Tabla 8, el valor [0.51, 0.67] se define como un valor característico para DMs con perfiles relajado a semi-conservador.

La Figura 13 contiene las 100 alternativas fijas que se utilizan dentro de las instancias las columnas están ordenadas de la siguiente manera: costo (Restricción), Estancia días (Obj 1), Destinos (Obj2), Calificación (Obj3), Atracciones (Obj4).

	REST-1	OBJ1	OBJ2	OBJ3	OBJ4
Alternativa 1	[7563,7563]	[6,6]	[1,1]	[2,2]	[44,44]
Alternativa 2	[5426,5426]	[6,6]	[1,1]	[7,7]	[81,81]
Alternativa 3	[3515,3515]	[8,8]	[1,1]	[7,7]	[12,12]
Alternativa 4	[9824,9824]	[7,7]	[1,1]	[4,4]	[57,57]
Alternativa 5	[8229,8229]	[5,5]	[1,1]	[4,4]	[12,12]
Alternativa 6	[6499,6499]	[7,7]	[1,1]	[4,4]	[16,16]
Alternativa 7	[4868,4868]	[4,4]	[1,1]	[4,4]	[10,10]
Alternativa 8	[7999,7999]	[3,3]	[1,1]	[8,8]	[98,98]

Alternativa 9	[8063,8063]	[5,5]	[1,1]	[7,7]	[18,18]
Alternativa 10	[6197,6197]	[8,8]	[1,1]	[9,9]	[48,48]
Alternativa 11	[8525,8525]	[5,5]	[1,1]	[8,8]	[11,11]
Alternativa 12	[3217,3217]	[7,7]	[1,1]	[4,4]	[95,95]
Alternativa 13	[4557,4557]	[7,7]	[1,1]	[6,6]	[44,44]
Alternativa 14	[8335,8335]	[5,5]	[1,1]	[8,8]	[13,13]
Alternativa 15	[5174,5174]	[5,5]	[1,1]	[2,2]	[31,31]
Alternativa 16	[4288,4288]	[3,3]	[1,1]	[5,5]	[41,41]
Alternativa 17	[5636,5636]	[5,5]	[1,1]	[6,6]	[48,48]
Alternativa 18	[4347,4347]	[5,5]	[1,1]	[9,9]	[93,93]
Alternativa 19	[6472,6472]	[6,6]	[1,1]	[8,8]	[49,49]
Alternativa 20	[4544,4544]	[4,4]	[1,1]	[7,7]	[33,33]
Alternativa 21	[6746,6746]	[7,7]	[1,1]	[2,2]	[25,25]
Alternativa 22	[6372,6372]	[7,7]	[1,1]	[2,2]	[98,98]
Alternativa 23	[8188,8188]	[7,7]	[1,1]	[3,3]	[65,65]
Alternativa 24	[5002,5002]	[8,8]	[1,1]	[5,5]	[51,51]
Alternativa 25	[5676,5676]	[4,4]	[1,1]	[5,5]	[24,24]
Alternativa 26	[7135,7135]	[3,3]	[1,1]	[8,8]	[59,59]
Alternativa 27	[5938,5938]	[7,7]	[1,1]	[5,5]	[97,97]
Alternativa 28	[3688,3688]	[8,8]	[1,1]	[9,9]	[29,29]
Alternativa 29	[6106,6106]	[6,6]	[1,1]	[7,7]	[50,50]
Alternativa 30	[4012,4012]	[5,5]	[1,1]	[8,8]	[73,73]
Alternativa 31	[5235,5235]	[6,6]	[1,1]	[2,2]	[89,89]
Alternativa 32	[6955,6955]	[8,8]	[1,1]	[9,9]	[82,82]
Alternativa 33	[5708,5708]	[7,7]	[1,1]	[2,2]	[34,34]
Alternativa 34	[7810,7810]	[6,6]	[1,1]	[1,1]	[12,12]
Alternativa 35	[3043,3043]	[3,3]	[1,1]	[4,4]	[94,94]
Alternativa 36	[6761,6761]	[7,7]	[1,1]	[1,1]	[47,47]
Alternativa 37	[7286,7286]	[8,8]	[1,1]	[2,2]	[87,87]
Alternativa 38	[6234,6234]	[5,5]	[1,1]	[5,5]	[31,31]
Alternativa 39	[6878,6878]	[8,8]	[1,1]	[6,6]	[64,64]
Alternativa 40	[4991,4991]	[3,3]	[1,1]	[9,9]	[87,87]
Alternativa 41	[3606,3606]	[8,8]	[1,1]	[1,1]	[81,81]
Alternativa 42	[8296,8296]	[3,3]	[1,1]	[1,1]	[26,26]
Alternativa 43	[9872,9872]	[6,6]	[1,1]	[3,3]	[36,36]
Alternativa 44	[7949,7949]	[6,6]	[1,1]	[7,7]	[81,81]
Alternativa 45	[9271,9271]	[5,5]	[1,1]	[2,2]	[94,94]
Alternativa 46	[4618,4618]	[7,7]	[1,1]	[8,8]	[20,20]

Alternativa 47	[6102,6102]	[7,7]	[1,1]	[6,6]	[14,14]
Alternativa 48	[4539,4539]	[3,3]	[1,1]	[5,5]	[15,15]
Alternativa 49	[3512,3512]	[3,3]	[1,1]	[4,4]	[80,80]
Alternativa 50	[5828,5828]	[5,5]	[1,1]	[5,5]	[43,43]
Alternativa 51	[5672,5672]	[5,5]	[1,1]	[6,6]	[87,87]
Alternativa 52	[3581,3581]	[6,6]	[1,1]	[3,3]	[32,32]
Alternativa 53	[5152,5152]	[3,3]	[1,1]	[2,2]	[69,69]
Alternativa 54	[4115,4115]	[5,5]	[1,1]	[5,5]	[64,64]
Alternativa 55	[3603,3603]	[7,7]	[1,1]	[2,2]	[10,10]
Alternativa 56	[7253,7253]	[6,6]	[1,1]	[5,5]	[98,98]
Alternativa 57	[9345,9345]	[6,6]	[1,1]	[9,9]	[15,15]
Alternativa 58	[4475,4475]	[3,3]	[1,1]	[8,8]	[30,30]
Alternativa 59	[5167,5167]	[6,6]	[1,1]	[2,2]	[22,22]
Alternativa 60	[6671,6671]	[3,3]	[1,1]	[5,5]	[50,50]
Alternativa 61	[8633,8633]	[8,8]	[1,1]	[7,7]	[69,69]
Alternativa 62	[4819,4819]	[3,3]	[1,1]	[3,3]	[15,15]
Alternativa 63	[9202,9202]	[7,7]	[1,1]	[4,4]	[78,78]
Alternativa 64	[4559,4559]	[8,8]	[1,1]	[3,3]	[97,97]
Alternativa 65	[7522,7522]	[8,8]	[1,1]	[4,4]	[40,40]
Alternativa 66	[4587,4587]	[6,6]	[1,1]	[4,4]	[39,39]
Alternativa 67	[3609,3609]	[3,3]	[1,1]	[4,4]	[85,85]
Alternativa 68	[4608,4608]	[6,6]	[1,1]	[4,4]	[94,94]
Alternativa 69	[6845,6845]	[3,3]	[1,1]	[1,1]	[25,25]
Alternativa 70	[5591,5591]	[5,5]	[1,1]	[1,1]	[75,75]
Alternativa 71	[7753,7753]	[4,4]	[1,1]	[2,2]	[56,56]
Alternativa 72	[3239,3239]	[6,6]	[1,1]	[4,4]	[51,51]
Alternativa 73	[3216,3216]	[7,7]	[1,1]	[1,1]	[84,84]
Alternativa 74	[6462,6462]	[6,6]	[1,1]	[2,2]	[10,10]
Alternativa 75	[6145,6145]	[7,7]	[1,1]	[1,1]	[70,70]
Alternativa 76	[4223,4223]	[3,3]	[1,1]	[9,9]	[17,17]
Alternativa 77	[5842,5842]	[5,5]	[1,1]	[5,5]	[61,61]
Alternativa 78	[8732,8732]	[7,7]	[1,1]	[5,5]	[92,92]
Alternativa 79	[8735,8735]	[4,4]	[1,1]	[1,1]	[33,33]
Alternativa 80	[6969,6969]	[8,8]	[1,1]	[1,1]	[76,76]
Alternativa 81	[3456,3456]	[5,5]	[1,1]	[2,2]	[85,85]
Alternativa 82	[7777,7777]	[3,3]	[1,1]	[4,4]	[17,17]
Alternativa 83	[9806,9806]	[3,3]	[1,1]	[7,7]	[12,12]
Alternativa 84	[3457,3457]	[7,7]	[1,1]	[6,6]	[55,55]

Alternativa 85	[6457,6457]	[6,6]	[1,1]	[5,5]	[79,79]
Alternativa 86	[3030,3030]	[5,5]	[1,1]	[7,7]	[33,33]
Alternativa 87	[9871,9871]	[7,7]	[1,1]	[9,9]	[17,17]
Alternativa 88	[3007,3007]	[8,8]	[1,1]	[2,2]	[68,68]
Alternativa 89	[6616,6616]	[5,5]	[1,1]	[5,5]	[45,45]
Alternativa 90	[3444,3444]	[8,8]	[1,1]	[7,7]	[71,71]
Alternativa 91	[4627,4627]	[8,8]	[1,1]	[2,2]	[52,52]
Alternativa 92	[8763,8763]	[7,7]	[1,1]	[4,4]	[83,83]
Alternativa 93	[9094,9094]	[3,3]	[1,1]	[7,7]	[74,74]
Alternativa 94	[4605,4605]	[7,7]	[1,1]	[8,8]	[85,85]
Alternativa 95	[8331,8331]	[7,7]	[1,1]	[5,5]	[29,29]
Alternativa 96	[5044,5044]	[7,7]	[1,1]	[9,9]	[30,30]
Alternativa 97	[9850,9850]	[3,3]	[1,1]	[1,1]	[31,31]
Alternativa 98	[6126,6126]	[4,4]	[1,1]	[6,6]	[11,11]
Alternativa 99	[3631,3631]	[5,5]	[1,1]	[3,3]	[22,22]
Alternativa 100	[4536,4536]	[3,3]	[1,1]	[2,2]	[75,75]

Tabla 9. Alternativas en las instancias

6.5 Obtención de mejores compromisos

Los mejores compromisos son obtenidos mediante el algoritmo INOSGA-II, resolviendo el caso de estudio descrito en el punto anterior. Haciendo uso del Framework MS-DOSS, se implementa INOSGA-II para obtener la ROI y los mejores compromisos de cada DM. El proceso sería el siguiente.

Resolver con I-NOSGA-II para cada $DM0_i$:

- Instances01_DM0i.txt
 - Instances01_DM0i_CORRIDA01.txt
 - ...
 - Instances01_DM0i_CORRIDA30.txt
- Instances02_DM0i.txt
 - Instances01_DM0i_CORRIDA01.txt
 - ...
 - Instances01_DM0i_CORRIDA30.txt

La ROI_i es el conjunto de 30 corridas por instancia para cada DM, usando estos resultados se obtiene el mejor compromiso BC_i , es decir las soluciones s en ROI_i con el mayor $NSS(s)$, que es el número de soluciones s que cumplen:

$$s \text{ domina } r \ || \ \sigma(s, r) > \text{umbral} \ \&\& \ \sigma(r, s) < 0.05$$

Considerando los datos preferencias del Decisor $DM0_i$. La estrategia para calcular $NSS(s)$ la llamamos $\text{ComputeNSS}(ROI_i, DM0_i)$;

6.6 Obtención de consenso

Al igual que en la obtención de mejores compromisos este hace uso de MS-DOSS, dentro del cual se da el problema de optimización de consenso. Este paso busca obtener un conjunto de parámetros de preferencias que incrementen el consenso dentro del grupo, es decir, que la solución obtenida tenga un índice de credibilidad mayor a 0.5 y que sea mejor que las demás soluciones. El proceso es el siguiente:

- Se crea la `Instances01_DM_BC.txt` que tendrá los BC_i de cada DM.
- Con I-NSOGA resolviendo el modelo propuesto en [`Instances01_DM_BC.txt`], se obtiene un conjunto de soluciones.
- Se selecciona la solución CMD que maximice la función objetivo (si hay empates se elige una al azar) esta será la solución respuesta.
- Se identifica BC a partir de $\text{ComputeNSS}(ROI_i, CMD)$, si la solución que se obtiene es la misma que usando $\text{ComputeNSS}(ROI_i, DM0_i)$, entonces se cuenta como satisfecho, y se acumula para CMD.

6.7 Indicador de calidad de consenso

Para comparar la calidad entre ambas estrategias y demostrar la efectividad de la propuesta se usó la métrica M1.

- M1. $NSS(s)$ = número de soluciones r en ROI_i que [$s \text{ Dominates } r$] OR [$\sigma(s, r) > \delta \ \&\& \ \sigma(r, s) < 0.5$]. s es una solución obtenida por la propuesta, o bien el consenso definido genéricamente.

Los parámetros preferenciales son los obtenidos por los modelos en consideración. A partir de esto, se identifica la solución o soluciones con el valor máximo de NSS, y si se mantiene el Original BC_i se dice que satisface al DM_i .

6.8 Resultados Obtenidos

En este apartado se muestran los resultados obtenidos del modelo de consenso propuesto (CMD) y el modelo genérico por extremos (GMD). Para cada Instancia, la Tabla 10 reporta la cantidad de DMs satisfechos, que cumplen con el indicador de calidad de consenso explicado en el punto anterior, y resalta de color verde las celdas donde mostro resultados superiores a GMD.

Instancia	Métrica 1	
	CMD	GMD
I01	4	1
I02	1	1
I03	3	3
I04	0	0
I05	0	0
I06	1	1
I07	2	1
I08	3	3
I09	2	1
I10	3	3
I11	2	0
I12	3	1
I13	1	0
I14	2	2
I15	1	1
I16	2	0
I17	2	2
I18	1	1
I19	0	0
I20	1	1
I21	1	0

I22	2	0
I23	3	1
I24	3	2
I25	2	1
I26	2	1
I27	1	1
I28	0	0
I29	1	0
I30	2	1

Tabla 10. Resultados obtenidos

6.9 Análisis de resultados

Los resultados obtenidos por el modelo de consenso en comparación con el modelo genérico por extremos se muestran en la Figura 14, el eje “x” refleja la instancia en el que se registraron los resultados y el eje “y” la cantidad de DMs satisfechos. De cinco decisores participando se obtuvo el 80 % de los DMs satisfechos como mejor resultado, y en el 50% de los casos el modelo propuesto gano, quedando el otro 50% en empate, aunque por sí solos en algunos experimentos no se obtuvieron una cantidad considerable de decisores satisfechos, en ningún caso perdieron contra el modelo GMD, el mejor resultado de este fue del 60% de decisores satisfechos con resultados más pobres en las demás instancias, por lo cual, podemos asumir que el modelo CMD tiene una clara ventaja sobre el modelo GMD.

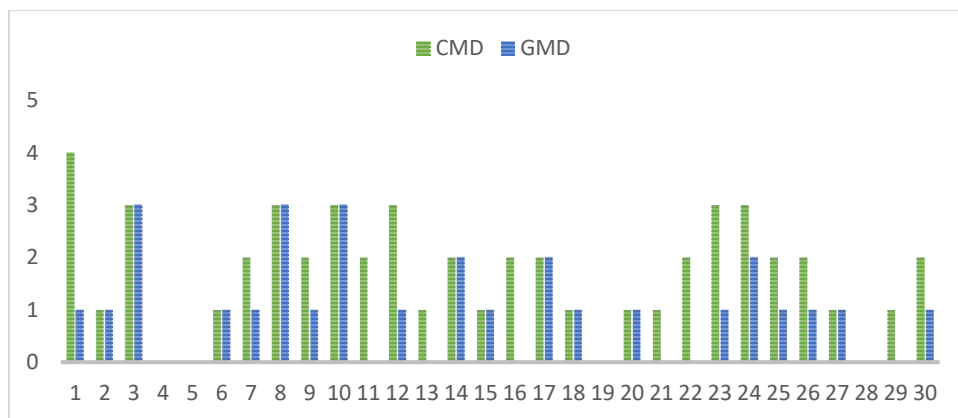


Figura 13. Comparación de resultados

Capítulo 7. Conclusiones

El presente trabajo de investigación desarrolla una propuesta para la integración de preferencias en grupos de investigadores. El consenso es logrado a través de la combinación exitosa de relaciones de superación e intervalos en un algoritmo de optimización que busca satisfacer el mayor número de decisores posibles. Las principales contribuciones de este trabajo son:

- Un modelo genérico para consenso basado en intervalos extremos que asocia un número de intervalo cuyos límites engloban todos los posibles valores deseados por los decisores en el grupo;
- Un modelo de optimización basado en outranking e intervalos que maximiza el número de decisores satisfechos en el consenso;
- Una estrategia de solución a través de un enfoque evolutivo que resuelve el modelo de optimización;
- Una arquitectura para agentes que integra un modelo de preferencias, intervalos, y algoritmos de optimización para el manejo de consenso;
- Un análisis del desempeño de la estrategia de solución propuesta.

A través de los resultados obtenidos del diseño experimental definido para este trabajo de investigación se puede llegar a las siguientes conclusiones:

- La incertidumbre crece cuando se trata de integrar valores preferenciales para lograr consenso en grupos de decisión;
- En la literatura se busca aumentar el nivel de satisfactibilidad en procesos de consenso a través de sistemas de apoyo y modelos que manejen la incertidumbre, sin embargo, todavía hay áreas de oportunidad de mejora, como la abordada en este trabajo que busca satisfacer el consenso mediante intervalos y relaciones de outranking combinados en un modelo de optimización;
- Una de las formas de tratar la incertidumbre es por medio de intervalos extremos, sin embargo, el modelo genérico, sigue presentado una baja cantidad de decisores satisfechos;

- El número de sistemas de apoyo a la decisión que emplea el uso de outranking como parte del proceso de solución al consenso es muy reducido;
- El desarrollo de sistemas de apoyo al consenso basados en outranking e intervalos obtiene mejores resultados que los modelos genéricos;
- Se puede integrar el modelo propuesto basado en outranking e intervalos para manejo de preferencias grupales (y consenso) dentro de la arquitectura Touring Machines;
- Los resultados obtenidos muestran que el modelo propuesto tiene una ventaja sobre el modelo genérico usando los intervalos extremos, por lo cual, es posible integrar las preferencias de los decisores en una sola estructura que aumente la satisfactibilidad de los DMs en el proceso, y esto se demuestra al no perder en ningún experimento dentro de la comparación en la sección anterior y de generar un mayor número de decisores satisfechos;
- Las arquitecturas híbridas encajan mejor con los sistemas de apoyo a la toma de decisiones en grupos debido a que estos agentes están trabajando en entornos más complejos que tienen múltiples usuarios que deben compartir información de manera consensuada o bien que existen módulos que pueden tener conflictos en la información que comparten entre ellos y que deben estandarizar.

Finalmente, y después del análisis de los resultados finales, se observaron nuevas preguntas de investigación como: ¿Podemos construir los umbrales de preferencias mediante los mejores compromisos ingresados por los usuarios?, ¿Es posible guiar la construcción de las preferencias de los usuarios hasta alcanzar una decisión unánime?, y ¿En qué otros sistemas de agentes es posible integrar un modelo de preferencias grupales? las cuáles abren líneas de investigación para trabajo futuro.

Referencias

- [1] Castro Rivera, J. A. (2018). Modelado de la personalidad en modelos preferenciales multicriterio a través de agentes virtuales inteligentes.
- [2] Fernandez, E., & Olmedo, R. (2005). An agent model based on ideas of concordance and discordance for group ranking problems. *Decision Support Systems*, 39(3), 429-443.
- [3] Palomares, I., Quesada, F. J., & Martínez, L. Sistema de Apoyo al Consenso para Problemas de Toma de Decisión en Grandes Grupos basado en una Plataforma Multi-Agente.
- [4] Budán, M.C., Simari, G.R., & Costaguta, R. (2011). Especificación e implementación de agentes inteligentes para el soporte a la toma de decisiones. Mayo 10, 2020, de *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*.
- [5] Corchado, J. (2005). Modelos y Arquitecturas de Agente.
- [6] Fernández, E., López, E., López, F. & Coello, C. (2010, septiembre 3). Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Journal ElSevier*, 1, 13.
- [7] Ferretti, E., Sosa, C., Aguirre, G., Loyola, J., Cagnina, L. & Errecalde, M. (mayo 11, 2016). Toma de decisiones y aprendizaje en agentes artificiales inteligentes. Mayo 11, 2020, de *Red de Universidades con Carreras en Informática (RedUNCI)* Sitio web: <http://sedici.unlp.edu.ar/handle/10915/52730>
- [8] Rouyendegh, B.D., Erkan, T.E. (2013) An application of the fuzzy electre method for academic staff selection. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 23(2), 107-115.
- [9] Ferguson, I. A. (1992). Touring machines: Autonomous agents with attitudes. *Computer*, 25(5), 51-55.

- [10] Iglesias, C. (1998). Definición de una metodología para el desarrollo de Sistemas Multi-Agente. Universidad Politécnica de Madrid, España.
- [11] Iglesias, C. A. (1997). Fundamentos de los agentes inteligentes. Informe Técnico UPM/DIT/GSI, 16, 97
- [12] Delgado-Hernández, X. S., Morales-Rodriguez, M. L., Rangel-Valdez, N., Cruz-Reyes, L., & Castro-Rivera, J. (2020). Development of Conversational Deliberative Agents Driven by Personality via Fuzzy Outranking Relations. *International Journal of Fuzzy Systems*, 1-15.
- [13] Berry, P., Peintner, B., Conley, K., Gervasio, M., Uribe, T., & Yorke-Smith, N. (2006, May). Deploying a personalized time management agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems* (pp. 1564-1571).
- [14] Balderas Jaramillo, Fausto & Fernández, Eduardo & Santillán, Claudia & Cruz Reyes, Laura & RangelValdez, Nelson & Morales-Rodríguez, María. (2018). A Grey Mathematics Approach for Evolutionary Multiobjective Metaheuristic of Project Portfolio Selection. 10.1007/978-3-319-71008-2_27.
- [15] Fernandez, E., & Olmedo, R. (2005). An agent model based on ideas of concordance and discordance for group ranking p
- [16] Coello, C.A.C. Handling Preferences in evolutionary Multiobjctive optimization: A Survey. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pp. 30 – 37, (2000).
- [17] Fernandez, E., Lopez, E., Mazcorro, G., Olmedo, R., & Coello, C. A. C. (2013). Application of the non-outranked sorting genetic algorithm to public project portfolio selection. *Information Sciences*, 228, 131-149.
- [18] Fernández, E., López, E., López, F. & Coello, C. (2010). Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Journal ElSevier*, 1, 13.

- [19] Balderas, F., Fernandez, E., Gomez-Santillan, C., Rangel-Valdez, N.*, Cruz, L. (2019). An Interval-Based Approach for Evolutionary Multi-Objective Optimization of Project Portfolios. *International Journal of Information Technology & Decision Making*. Vol. 18, No. 4 (2019) 1317–1358. DOI: <https://doi.org/10.1142/S021962201950024X>
- [20] Balderas, F., Fernandez, E., Gomez, C., Cruz-Reyes, L., Rangel-Valdez, N. (2016). Metaheuristic Optimization of Project Portfolios using an Interval-Based Model of Imprecisions. *International Journal of Combinatorial Optimization*, vol. 7(3):101 – 118.
- [21] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the FiBh International Conference on Genetic Algorithms*, pages 416-423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.
- [22] Garrison W. Greenwood, Xiaobo Sharon Hu, and Joseph G. D’Ambrosio. Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings. In Richard K. Belew and Michael D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 437-455. Morgan Kaufmann, San Mateo, California, 1997.
- [23] Dragan Cvetkovic and Ian C. Parmee. Genetic Algorithm-based Multi-objective Optimisation and Conceptual Engineering Design. In *Congress on Evolutionary Computation - CEC99*, volume 1, pages 29-36, Washington D.C., USA, 1999. IEEE.
- [24] Dragan Cvetkovic and Ian C. Parmee. Designer’s preferences and multi-objective preliminary design processes. In Ian C. Parmee, editor, *Proceedings of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM’2000)*, pages 249-260. PEDC, University of Plymouth, UK, Springer London, 2000.
- [25] Brahim Rekiek, Pierre De Lit, Fabrice Pellichero, Thomas L’Eglise, Emanuel Falkenauer, and Alain Delchambre. Dealing With User’s Preferences in Hybrid Assembly Lines Design. In *Proceedings of the MCPL ’2000 Conference*, 2000)

- [26] C.T.L. Butler and A. Rothstein. *On Conflict and Consensus: A Handbook on Formal Consensus Decision Making*. Takoma Park, 2006.
- [27] J. Kacprzyk. *Group decision making with a fuzzy linguistic majority*. *Fuzzy Sets and Systems*, 18(2):105-118, 1986.
- [28] Almeida, A. T. (2005). *Multicriteria modelling of repair contract based on utility and ELECTRE I method with dependability and service quality criteria*. *Annals of Operations Research*, 138(1), pp. 113-126.
- [29] Goletsis, Y., Psarras, J., & Samouilidis, J. E. (2003). Project ranking in the Armenian energy sector using a multicriteria method for groups. *Annals of Operations Research*, 120(1-4), pp. 135-157.
- [30] Mavrotas, G., Diakoulaki, D., & Capros, P. (2003): Combined MCDA–IP approach for project selection in the electricity market. *Annals of Operations Research*, 120(1-4), pp. 159-170.
- [31] Roy, B. (1968). Classement et choix en présence de points de vue multiples (la méthode de ELECTRE), *RIRO* 8, 57-75.
- [32] Lee, W. P., Liu, C. H., & Lu, C. C. (2002). Intelligent agent-based systems for personalized recommendations in Internet commerce. *Expert Systems with Applications*, 22(4), 275-284.
- [33] Chai, J., & Liu, J. N. (2010, December). Towards a reliable framework of uncertainty-based group decision support system. In *2010 IEEE International Conference on Data Mining Workshops* (pp. 851-858). IEEE.
- [34] Goletsis, Y., Psarras, J., & Samouilidis, J. E. (2003). Project ranking in the Armenian energy sector using a multicriteria method for groups. *Annals of operations research*, 120(1), 135-157.
- [35] Figueira, J. R., Greco, S., Roy, B., & Słowiński, R. (2010). ELECTRE Methods: Main Features and Recent Developments. *Handbook of Multicriteria Analysis*, 103, 51–89. https://doi.org/10.1007/978-3-540-92828-7_3