UNIVERSIDAD DE MÁLAGA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

TESIS DOCTORAL

Big Data Optimization: Algorithmic Framework for Data Analysis Guided by Semantics

E.T.S.I. Informática
R.D. 99/2011

Autor

Cristóbal Barba González

Directores

Dr. José F. Aldana Montes
Departamento
Lenguajes y Ciencias de la Computación
Universidad de Málaga

Dr. José Manuel García Nieto
Departamento
Lenguajes y Ciencias de la Computación
Universidad de Málaga

Noviembre 2018

Departamento de Lenguajes y Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

Los Dres. **José F. Aldana Montes**, Profesor Catedrático del Departamento de Lenguajes y Ciencias de la computación de la Universidad de Málaga, y **José M. García Nieto**, Doctor del Departamento de Lenguajes y Ciencia de la Computación de la Universidad de Málaga,

Certifican

que, D. **Cristóbal Barba González**, Ingeniero en Informática por la Universidad de Málaga, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo sus direcciones, el trabajo de investigación correspondiente a su Tesis Doctoral por Compendio titulada:

Big Data Optimization: Algorithmic Framework for Data Analysis Guided by Semantics

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo. Y para que conste a efectos de lo establecido en la legislación vigente, autorizamos la presentación de esta Tesis Doctoral en la Universidad de Málaga.

En Málaga, noviembre del 2018

Fdo: Dr. José F. Aldana Montes

Dr. José Manuel García Nieto

Contents

Acknowledgements

First, I would like to thank my supervisors Prof. José F. Aldana Montes and Dr. José Manuel García Nieto for their guidance and support throughout these years of developing the work that is part of this thesis. I would especially like to thank them for giving me the opportunity of doing a PhD thesis and trust on me during all these years.

I also wish to thank everyone who accepted to be part of my thesis committee, for agreeing so quickly, and making it all so easy for me. In addition I thank my external evaluators for all their insightful corrections that improved my manuscript.

My sincere thanks to Prof. Kaisa Miettinen and Dr. Vesa Ojalehto for the lovely stay I had in those four months at the University of Jyväskylä. They gave me a huge opportunity for living an excellent experience and also improving my research.

I would like to mention all the people I have been working with all these years in the Khaos research group, both doctors and students. Most of them I am glad to call friends, as we have shared lots of fun and hard work over the years. I would especially like to thank Prof. Antonio Nebro and Dr. José Manuel García Nieto, both have taught me how to do research and I do think I would not have made it without them.

I would also like to thank other members of the Grupo de Ingeniería del Software de la Universidad de Málaga (GISUM) who I know personally and have shared personal experiences with them. I want to thank all the staff at the Ada Byron research centre for their help in the day-to-day work.

My family has been a great pillar of support in my life so I want to to thank all my family, especially my parents, my three sisters, nephews and brothers-in-law, thanks for always being there and supporting me in difficult times.

Finally, I would like to thank the Spanish Minister of Economic and Competitiveness for supporting my research with the grant BES-2015-072209.

Resumen

Introducción

En las últimas décadas el aumento de fuentes de información en diferentes campos de la sociedad desde la salud hasta las redes sociales ha puesto de manifiesto la necesidad de nuevas técnicas para su análisis, lo que se ha venido a llamar el *Big Data*. Los problemas clásicos de optimización no son ajenos a este cambio de paradigma, como por ejemplo el problema del viajante de comercio (TSP), ya que se puede beneficiar de los datos que proporciona los diferentes sensores que se encuentran en las ciudades y que podemos acceder a ellos gracias a los portales de *Open Data*. En esta tesis se ha desarrollado un nuevo framework, jMetalSP, para la optimización de problemas en el ámbito del Big Data permitiendo el uso de fuentes de datos externas para modificar los datos del problema en tiempo real. Por otro lado, cuando estamos realizando análisis, ya sea de optimización o machine learning en Big Data, una de las formas más usada de abordarlo es mediante workflows de análisis. Estos están formados por componentes que hacen cada paso del análisis. El flujo de información en workflows puede ser anotada y almacenada usando herramientas de la Web Semántica para facilitar la reutilización de dichos componentes o incluso el workflow completo en futuros análisis, facilitando así, su reutilización y a su vez, mejorando el proceso de creación de estos. Para ello se ha creado la ontología *BIGOWL*, que permite trazar la cadena de valor de los datos de los workflows mediante semántica y además ayuda al analista en la creación de workflow gracias a que va guiando su composición con la información que contiene por la anotación de algoritmos, datos, componentes y workflows.

Debido al gran avance que existe día a día en las tecnologías de información, las organizaciones se han tenido que enfrentar a nuevos desafíos que les permitan analizar, descubrir y entender más allá de lo que sus herramientas tradicionales reportan sobre su información. Existen numerosas fuentes de información como son, las aplicaciones disponibles en internet (redes sociales, georeferenciamiento, etc.); la información obtenida en el campo de la medicina y biología con la secuenciación del ADN y otros datos analíticos, etc. Todos estos datos se conocen como *Big Data* y hace necesario el uso de nuevos enfoques para el análisis de estos, debido principalmente, a que las técnicas tradicionales usadas hasta ahora (Machine Learning y Optimización) no están diseñadas ni optimizadas para manejar grandes volúmenes de información [1].

El término Big Data, por tanto, hace referencia a aquellos datos que no pueden ser procesados o analizados usando las técnicas tradicionales [2]. El análisis del Big Data permite extraer información a partir de estos datos. Multitud de soluciones software alrededor del proyecto Apache Hadoop van resolviendo las diferentes problemáticas a través de este proyecto y otros complementarios. Algunos de estos proyectos son:

- ***Apache Hadoop***. Que integra el framework MapReduce y el sistema de archivos HDFS.
- ***Apache Spark***. Framework que permite realizar tareas de transformación y de acción sobre los datos en streaming de manera eficiente.

Motivación

De acuerdo con Gartner [1] y la asociación europea Big Data Value (BDVA)[2], hay un reto en el campo del Big Data acerca de la construcción de aplicaciones que inyecte el conocimiento del dominio (problema, algoritmo, dato, etc.), así como el contexto, en el proceso de análisis. Por contexto entendemos toda la (meta)información relevante que permita interpretar los resultados del análisis. Esto tendrá como consecuencia la accionabilidad de dichos resultados. Así facilitará la interpretación de estos datos; permitirá integrarlos fácilmente con otros datos estructurados; facilitará la integración del sistema de análisis del Big Data con otros sistemas y habilitará la interconexión de algoritmos. En esta tesis hemos abordado este reto creando BIGOWL, una ontología en la que se ha definido toda la semántica necesaria para poder definir cualquier problema de análisis en el ámbito del Big Data, así como los algoritmos y datos a utilizar para dicho análisis.

Uno de los principales problemas que nos encontramos a la hora de trabajar con Big Data es la eficiencia de los algoritmos de optimización a la hora de procesar grandes volúmenes de información, sobre todo en tiempo de cómputo, por ello es necesario realizar algoritmos optimizados para problemas de Big Data. Existen dos grandes tipos de problemas de optimización los llamados mono-objetivo y los multi-objetivo. Este trabajo se ha centrado en los problemas multi-objectivo ya que existen muchos en Big Data. Para ello hemos desarrollado *jMetalSP* que es una librería de algoritmos de optimización adaptados para Big Data, el cual está basado en jMetal [3], una herramienta para la optimización multi-objetivo y Apache Spark [4]. Dentro de los algoritmos multi-objetivo desarrollados en jMetalSP, nos hemos centrado en los algoritmos dinámicos y en los interactivos, ya que facilitan reducir el espacio de búsqueda de soluciones de un problema y, por tanto, reducir el tiempo de cómputo quedándose sola con las áreas preferidas por el analista (Decision Maker). Además, permiten abordar procesamiento en streaming (Velocidad), actuar ante cambios en el problema, datos o entorno (Variabilidad) y acercar el proceso de optimización al Decision Maker (Valor, Veracidad).

Los problemas de procesamiento que dan lugar al Big Data provienen de la cantidad de datos, que es desproporcionadamente grande y del origen de la información, que es muy variado (fuentes, formato, etc.). Estos problemas asociados al Big Data se entienden en cinco dimensiones (las denominadas “cinco uves”): Volumen, Velocidad, Variedad, Variabilidad y Veracidad. El Big Data trae consigo la posibilidad de encontrar información relevante en los nuevos tipos de datos emergentes la capacidad de abordar cuestiones que hasta no hace mucho eran imposibles de plantear. El análisis del Big Data puede revelar información importante que anteriormente permanecía oculta debido al elevado coste de procesar ciertos datos, tales como las tendencias sociales de opinión, tendencias de consumo de ciertos productos comerciales, orientación y seguimiento publicitario, etc.

En general, la generación de un análisis en Big Data es compleja ya que entran en juego diferentes aplicaciones o algoritmos, concretamente diferentes componentes, que se tienen que interconectar entre ellos y tienen que ser compatibles, es decir, los datos de salida de uno son los de entrada del siguiente. En este sentido, esta tesis se propone que el uso de la semántica apoyada con tecnologías de la Web Semántica puede ayudar en la anotación de toda la información necesaria para que componentes distintos puedan interactuar entre ellos, facilitando así la interconexión entre componentes, su reutilización y la trazabilidad de la cadena de valor de los datos. La información extraída a través del análisis del Big Data se debe poner en contexto para permitir su aprovechamiento de cara a la toma de decisiones o cualquier otra actividad en la que se pretenda explotar. En esta tesis se propone que el uso de la semántica puede llenar este vacío.

La problemática se encuentra en dar estructura a esta información para poder ser integrada en

[1] https://www.gartner.com/doc/3656517/adopt-datadriven-approach-consolidating-infrastructure

[2] http://www.bdva.eu/

los componentes. De hecho, la automatización de este proceso es uno de los grandes retos en los que las tecnologías relacionadas con los estándares de la Web Semántica podrían tener un papel relevante.

Hoy en día se están desarrollando una serie de plataformas para el análisis del Big Data que además ofrecen una conexión con otros sistemas para la asistencia en la toma de decisiones. No obstante, hasta donde llega nuestro conocimiento, estos frameworks no incluyen una anotación semántica de sus elementos (componentes, algoritmos, métodos, interfaces, etc.) que facilite la reutilización y la composición de workflows de análisis del Big Data. Tampoco aprovechan la semántica específica del dominio de aplicación de forma que se facilite el desarrollo de los workflows de análisis, la anotación de los datos y los metadatos y/o de los resultados de los algoritmos, además de la generación de nuevos operadores adaptados que utilizan la estructura semántica del dominio del problema. Esto provoca que:

1. Los algoritmos de análisis del Big Data no son fácilmente reutilizables, no son capaces de explotar la semántica y, por tanto, la creación de workflows "inteligentes" es complicado y requiere desarrollo ad hoc.

2. No existe un procedimiento estandarizado para conectar los resultados del análisis del Big Data con otros sistemas.

Todas estas razones nos llevan a plantear la principal hipótesis de este proyecto: *la "anotación semántica" de los componentes software que constituyen un framework para el análisis del Big Data puede actuar como nexo de unión, tanto para la generación de nuevas propuestas algorítmicas que utilicen este conocimiento semántico, como para su final conexión con otros sistemas.* Se reutiliza así la idea fundamental de la Web Semántica en el contexto del Big Data orientado a la optimización.

Por otro lado, hemos desarrollado un framework para el análisis del Big Data, centrándonos en la optimización multi-objectivo para Big Data, más concretamente en metaheurísticas dinámicas y/o interactivas. Este framework, junto con otros de Machine Learning (WEKA, Scikit Learn Python, MLlib Spark, etc.), se usarán como casos de uso para comprobar que la anotación de sus componentes es capaz de cubrir completamente sus especificaciones.

Los algoritmos que se han desarrollado en el framework se usarán en diferentes problemas reales, como son: la bioinformática o la búsqueda de la ruta óptima en tráfico. Hemos integrado el framework jMetal, como base, que proporciona una gran batería de algoritmos de optimización multi-objetivo, junto con el modelo de programación MapReduce [5] y el framework Spark [4].

Otro de nuestros objetivos es realizar una aplicación que nos permita evaluar algoritmos interactivos que se desarrollen en esta tesis y que nos permita compararlos ya que solo existe uno en el estado de arte actual [6].

Objetivos

Esta tesis tiene como objetivo el integrar técnicas y resultados del análisis del Big Data con una capa de metadatos (de los datos objetos del análisis, de las técnicas de análisis y del dominio donde éstas se aplican) para romper las barreras de acceso y aplicabilidad relacionados con las tecnologías de análisis del Big Data. Como enfoque principal el desarrollo de herramientas para dicho análisis, nos centramos en la optimización en Big Data. Concretamente los principales objetivos son:

1. Definir un modelo ontológico para la anotación semántica de algoritmos de análisis del Big Data.

   (a) Desarrollar una clasificación semántica de los algoritmos, componentes de procesamiento y visualización.

   (b) Diseñar una metodología para anotar la funcionalidad genérica de los algoritmos (tipo de algoritmos, entradas, salidas, transformación de los datos).

   (c) Diseñar una metodología para anotar las entradas, salidas y tipo de algoritmos según una ontología de dominio en caso de ser algoritmos específicos de un dominio de aplicación.

2. Desarrollo de una plataforma para la optimización en problemas Big Data.

   (a) Estudio de nuevos algoritmos de análisis en Big Data (dinámicos y/o interactivos).

   (b) Desarrollo de mecanismos para la evaluación de algoritmos interactivos.

   (c) Diseñar la estructura del repositorio para incluir no sólo los algoritmos, sino también sus anotaciones semánticas relativas a su funcionalidad y los eventos que generan o consumen.

3. Validación de la plataforma con casos de uso reales y académicos.

   (a) Problema del viajante de comercio en la ciudad de Nueva York con streaming Open Data.

   (b) Inferencia en Redes génicas (E.coli, Yeast).

   (c) Familia de problemas multi-objectivo DTLZ.

   (d) Familia de problemas dinámicos y multi-objetivo FDA.

Metodología y Plan de Trabajo

El método a utilizar es una adaptación del método Investigación en Acción de Avison, et al. [7]. Se trata de un método cualitativo utilizado para validar los trabajos de investigación mediante su aplicación a proyectos reales. En la Conferencia sobre Procesamiento de Información de 1998 se declaró a los métodos cualitativos como métodos de investigación apropiados para el campo de los sistemas de información [7], y los difundió en [8]. Proponemos un método de investigación genérico, basado en la propuesta de Bunge [9] y formado por etapas que, dada su generalidad, pueden aplicarse a cualquier tipo de investigación. Por tanto, utilizaremos un método de validación práctica, especialmente apropiado para la investigación en ingeniería y específicamente para validar aquellos resultados, en los que la aplicación de un método científico tradicional (inductivo o hipotético-deductivo) no son directamente aplicables. El método a seguir para la resolución y validación del problema concreto que nos ocupa (método de Investigación en Acción) no es un proceso lineal, sino que va avanzando mediante la compleción de ciclos. Al comenzar cada ciclo se ponen en marcha nuevas ideas, que son puestas en práctica y comprobadas hasta el inicio del siguiente ciclo [10]. Este proceso cíclico, en el que hemos ido probando y refinando cada uno de los resultados obtenidos, será nuestro método de validación.

La metodología Software que se va a seguir estará inspirada en este método, ya que se tendrán como referencia modelos ágiles como Scrum [3], basados en iteraciones de breve duración.

En concreto las tareas realizadas, para cada ciclo, son:

- ***Observación***. Estudio pormenorizado del problema a tratar, realizando un estudio del estado del arte e identificando posibles riesgos antes de afrontar la tarea.

- ***Formulación de hipótesis***. Declaración de la hipótesis que queremos llevar a cabo en dicho ciclo, se dividirá en pequeñas tareas abordables en la duración del ciclo.

[3] `https://www.scrum.org`

- ***Recogida de observaciones.*** Obtención de resultados como consecuencia de la realización de las tareas del paso anterior.

- ***Contrastes de hipótesis.*** Estudio de las observaciones recogidas y comprobación si se han cumplido nuestras hipótesis de partida.

- ***Demostración o refutación de hipótesis.*** Aceptación o rechazo de la hipótesis, si es necesario una modificación de la misma, empieza un nuevo ciclo con estos cambios.

Para alcanzar los objetivos siguiendo esta metodología, se definieron los siguientes pasos del plan de trabajo:

1. Análisis de la tarea, estudiando el problema que queremos abordar, tecnología a usar y búsqueda de otras propuestas en el estado del arte.

2. Diseño de la tarea, estudio de detalles técnicos y patrones de diseño a usar, ya sea para aplicarlo a la implementación de nuevos algoritmos o la anotación de componentes de BIGOWL (ontología propuesta).

3. Implementación del diseño anteriormente indicado.

4. Validación de los resultados obtenidos mediante el uso de herramientas externas como pueden ser estadísticos, razonadores de ontología, etc.

5. Difusión y publicación de revistas y congresos internacionales de impacto o relevancia científica.

Publicaciones

Cabe destacar que, como elemento central de esta tesis, tenemos la optimización en Big Data, la cual se apoya en otras tecnologías, como por ejemplo, los modelos de la web semántica para la anotación de los componentes de los workflows con el fin de mejorar el proceso de creación de los mismos y definir la cadena de valor de sus datos, o la optimización multi-objetivo junto con los procedimientos en streaming para crear la optimización multi-objetivo dinámica con el fin de poder abordar optimizaciones con grandes volúmenes de datos.

Con el fin de poner en relieve la relevancia de esta tesis se muestra a continuación la lista de las publicaciones realizadas durante su desarrollo:

1. ***Fine Grain Sentiment Analysis with Semantics in Tweets*** [11]. Este artículo se realizó una primera aproximación al mundo del Big Data añadiéndole semántica al análisis de los datos. El trabajo consistió en el estudio de los sentimientos de los aficionados durante el torneo universitario de baloncesto de Estados Unidos *Big 12 Men's Basketball Championship.* En este artículo se comprobó de manera empírica, nuestra hipótesis, la semántica mejora y facilita el análisis en Big Data. El análisis consitió en, usando tripletas RDF con la información de los jugadores, equipos, entrenadores, ciudades y estados de los equipos se filtraban los tweets referentes a cada uno. Una vez filtrados y clasificados se pudo realizar el análisis de sentimiento en cada tweet. Por tanto, cabe destacar que la semántica en este artículo guió la clasificación de los tweets por equipo y facilitó así su estudio por separado.

2. ***Dynamic Multi-Objective Optimization With jMetal and Spark: a Case Study*** [12]. El objetivo de este artículo fue comprobar que el uso de un framework de Big Data como Spark facilita la lectura de datos en streaming para su optimización. Es decir, se comprobó que es posible dotar con datos reales y en streaming a un problema de optimización.

3. ***Un Framework para Big Data Optimization Basado en jMetal y Spark*** [13]. En este trabajo se presentó una primera aproximación al framework para la optimización en Big Data, en el que se utiliza Spark como motor de computación distribuida y jMetal como motor de optimización. Es decir, los algoritmos son capaces de evaluar en paralelo las soluciones en cada iteración de la metaheurística.

4. ***jMetalSP: a framework for dynamic multi-objective big data optimization*** [14]. En este artículo se presentó jMetalSP, el famework basado en jMetal para optimizar los problemas de Big Data. jMetalSP está pensado para resolver problemas de optimización dinámica con algoritmos dinámicos mediante el análisis de múltiples fuentes de datos en streaming. jMetalSP dispone de algoritmos dinámicos multiobjetivos y/o interactivos (algunos de esos algoritmos tienen las dos caraterísticas). Como hemos indicado anteriormente, este framework combina jMetal con Apache Spark con el objetivo de soportar ejecución paralela, fuentes de datos en streaming y por tanto Big Data. Una de las características más destacable de jMetalSP es que permite que los algoritmos dinámicos detecten cambios en el problema y reaccionen de acuerdo con ellos (por ejemplo, aplicando una estrategia de reinicio). Al código de jMetalSP se puede acceder de forma online [4] para que pueda ser utilizado libremente por la comunidad científica.

5. ***Multi-Objective Big Data Optimization with jMetal and Spark***[15]. En este trabajo se presentó una versión inicial de nuestro framework jMetalSP y de la plataforma utilizada para la ejecución del software desarrollado durante esta tesis. En este trabajo se estudió dos escenarios distintos: primero, el uso de Spark como motor para evaluar las soluciones de una metaheurística en paralelo y, segundo, el estudio de la influencia del acceso a una gran cantidad de datos en cada evaluación en algoritmos metaheurísticos. En lugar de centrarnos en un problema de optimización particular, hemos definido un escenario genérico, en el que se modifica un problema estándar para aumentar artificialmente su tiempo de cálculo y para leer datos del sistema de archivos Hadoop (HDFS). Además, se realizó un estudio para evaluar, mediante el tiempo de cómputo, el rendimiento de Spark al ejecutar el paso de evaluación de la población de una metaheurística. El otro estudio que se realizó en este artículo fue la medición tanto del tamaño de los datos cómo del número de archivos que era capaz de soportar la plataforma diseñada para esta tesis, así pudimos comprobar como se comportaba Spark con diferentes tamaños de ficheros o diferentes números de archivos. Además, se estudió cual era el rendimiento de la plataforma cuando se modificaba el número de nodos a usar, pudiendo estudiar así la escalabilidad del sistema desde un punto de vista de nodos activos y tiempo de cómputo.

6. ***Design and Architecture of the jMetalSP Framework*** [16]. Este trabajo tuvo como objetivo la presentación de una nueva arquitectura para jMetalSP en el que, ya no es tan dependiente de Spark, sino que se hace flexible a cualquier motor de ejecución paralela. Se añadieron además nuevos algoritmos dinámicos como Dynamic SMPSO y Dynamic MOCell, así como la familia de problemas dinámicos FDA para su evaluación. Se incorporaron los mecanismos, para poder realizar de forma sencilla y común, la transformación de metaheurísticas estáticas en dinámicas

7. ***InDM2: Interactive Dynamic Multi-Objective Decision Making Using Evolutionary Algorithms*** [17]. En este artículo se presentó InDM2, que es el primer algoritmo del estado del arte que combina la optimización dinámica de múltiples objetivos, la toma de decisiones de criterios múltiples y la interactividad. Una característica clave en InDM2 es el mecanismo para visualizar, en tiempo de optimización, las aproximaciones de la región de

[4] https://github.com/jMetal/jMetalSP/

interés que se están generando a lo largo del proceso de búsqueda de soluciones, junto con el punto de referencia que indica estas preferencias. InDM2 puede incorporar cualquier algoritmo evolutivo basado en puntos de referencia para manejar la información de preferencia de forma interactiva. Otra característica importante de InDM2 es que permite la incorporación de estrategias para reaccionar a los cambios tanto en el problema como en el punto de referencia. Como InDM2 está desarrollado en jMetalSP, está disponible para la comunidad científica. El rendimiento de InDM2 se validó con tres DMOP de referencia como son los de la familia de problemas FDA y, además con un problema del mundo real, consistente en una versión dinámica del problema del viajante de comercio (TSP) basado en datos reales de tráfico proporcionados por el Departamento de Tráfico de la ciudad de Nueva York.

8. ***BIGOWL: Knowledge Centered Big Data Analytics*** [18]. En este estudio se presentó BIGOWL que fue diseñado e implementado para la representación y consolidación de conocimiento en el análisis en Big Data. BIGOWL contiene un conjunto amplio de conceptos, atributos y relaciones que se han tomado del ecosistema Big Data. El modelo semántico capturó toda la semántica necesaria para guiar el diseño inteligente de los flujos de trabajo (workflows) de análisis de Big Data y mejorar su rendimiento. El modelo semántico se evaluó con dos casos de uso realistas: el cálculo de enrutamiento en tiempo real en tráfico urbano y la clasificación clásica con árboles de decisión. Con estos dos casos de uso evaluamos la ontología desde un punto de vista de la optimización Big Data y del análisis de minería de datos.

9. ***Análisis de datos de acelerometría para la detección de tipos de actividades*** [19]. Se realizó un estudio de viabilidad para la clasificación de actividades físicas obtenidas mediante pulseras de acelerometría en pacientes con problemas cardiovasculares. Es decir, en este estudio se ha buscado demostrar la viabilidad para clasificar los datos (de un conjunto de datos de 4 Tb) usando redes neuronales profundas en el contexto del Big Data.

10. ***Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods*** [20]. Este trabajo ha sido recientemente presentado en la XV Conferencia Internacional Parallel Problem solving for Nature (PPSN 2018), celebrada en Coimbra, Portugal en septiembre de 2018. Surgió de la idea de evaluar algoritmos interactivos.

    Artificial Decision Maker basado en PSO (ADM-PSO) ha sido desarrollado para evaluar métodos interactivos basados en puntos de referencia. Puede calcular puntos de referencia tomando como base la información obtenida durante el proceso de optimización hasta el momento del cálculo. El decision maker (DM) a través de puntos de referencia indica los valores deseables de la función objetivo para dirigir de manera interactiva el proceso de solución. Sin embargo, al analizar el comportamiento de estos métodos, un tema crítico es cómo involucrar sistemáticamente a los DM humanos. En este estudio, presentamos un DM artificial, que reutiliza la dinámica de la optimización del algoritmo PSO para guiar la generación de puntos de referencia, por lo tanto, reemplazando al humano en la obtención de dichas preferencias. Usamos el ADM-PSO para comparar métodos interactivos. Para evaluar el ADM-PSO se usan los problemas de referencia DTLZ con diferentes objetivos y se comparan los algoritmos iterativos R-NSGA-II y WASF-GA. El experimento y los resultados mostraron que el ADM-PSO propuesto es capaz de evaluar de forma eficiente los algoritmos interactivos si lo comparamos con el otro ADM del estado del arte. Cabe destacar que nuestra versión obtuvo resultados similares en un menor número de iteraciones.

11. ***Extending the Speed-constrained Multi-Objective PSO (SMPSO) With Reference Point Based Preference Articulation*** [21]. Este trabajo fue presentado en la conferen-

cia internacional *12th International Symposium on Intelligent Distributed Computing (IDC 2018)*, celebrado en Bilbao, España, en octubre de 2018. En este artículo presentamos una nueva versión interactiva del algoritmo SMPSO, usando el framework jMetalSP. SMPSO con puntos de referencia puede centrarse solo en un área del espacio de búsqueda del problema especificado por el analista mediante los puntos de referencias, mejorando así los tiempos de ejecución ya que se reduce mucho el espacio de búsqueda.

12. ***Scalable Inference of Gene Regulatory Networks with the Spark Distributed Computing Platform*** [22]. Este trabajo fue presentado en la conferencia internacional *12th International Symposium on Intelligent Distributed Computing (IDC 2018)*, celebrado en Bilbao, España, en octubre de 2018. En este trabajo abordamos el problema de las inferencias en Redes Reguladoras Genéticas (GRN), que es un complejo problema de optimización que involucra el procesamiento de modelos S-System, que incluyen gran cantidad de datos de expresión genética de cientos (incluso miles) de genes en múltiples series temporales. La hipótesis de este trabajo fue usar el paralelismo de Spark en la evaluación de las soluciones de la metaheurística. En este enfoque, hemos utilizado el algoritmo MOCell con Spark para evaluar cada solución de la población en paralelo.

    Las pruebas realizadas mostraron que el enfoque propuesto obtiene importantes reducciones en el tiempo de cálculo de varios días (10) a horas (8,3) cuando se enfrenta a inferencias complejas de GRN de gran tamaño.

    Además, se obtuvo una mejora notable en el rendimiento y el consumo de recursos con una configuración del MOCell en paralelo en el rango de 20 y 50 nodos, ya que finalizó todas las tareas de ejecución en menos de medio día logrando, por tanto, un 50 % de eficiencia.

    Hicimos experimentos con datos de benchmarking realistas de DREAM3 para evaluar la propuesta y demostraron que MOCell era competitivo y, a menudo, supera los procedimientos de inferencia de GRN del estado de arte.

13. ***About Designing an Observer Pattern-Based Architecture for a Multi-Objective Metaheuristic Optimization Framework*** [23]. Este trabajo versó sobre la presentación de una nueva arquitectura para jMetal usando el patrón observador en el que se puede componer metaheurísticas mediante componentes independientes y reusables, por tanto, extendemos la idea de la composición de workflow a la composición de algoritmos. Se presentó en la conferencia internacional *12th International Symposium on Intelligent Distributed Computing (IDC 2018)*.

14. ***Algoritmo Evolutivo Multi-Objetivo para la Toma de Decisiones Interactiva en Optimización Dinámica*** [24]. En este trabajo se presentó una versión para la comunidad científica en español del algoritmo multi-objetivo dinámico e interactivo InDM2. Fue presentado en la conferencia nacional *XIII Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2018).*

Conclusiones

Esta tesis propone y analiza nuevas metaheuriticas dinámicas para abordar problemas de optimización de Big Data, basadas en la extensión de técnicas multi-objetivo y en la identificación de inconvenientes y posibles oportunidades de mejora, como la interactividad o la capacidad de lectura de datos en streaming.

Esta tesis proporciona una herramienta software para la implementación y experimentación de algoritmos de optimización en Big Data (jMetalSP), así como para la anotación semántica

(BIGOWL) de análisis en Big Data. El modelo semántico propuesto se materializa mediante un repositorio RDF, consultas programáticas y reglas de razonamiento lógico.

Resumiendo, las principales contribuciones de esta tesis se pueden enumerar de la siguiente manera:

1. ***Diseño y análisis de algoritmos dinámicos multi-objetivos***. Tomando como punto de partida el algoritmo NSGA-II, hemos desarrollado una metodología para la transformación de metaheurísticas estáticas en dinámicas. Para comprobar su valía, hemos diseñado e implementado la versión dinámica de NSGA-II en nuestro framework jMetalSP. Dynamic NSGA-II es por tanto, la metaheurística utilizada para la evaluación del comportamietno de nuestro diseño de algoritmos dinámicos con datos en streaming. Además, también se usa para evaluar la plataforma de Big Data desarrollada para la realización de las pruebas a lo largo de la tesis. Nuestro objetivo es evaluar el rendimiento de ejecución de una metaheurística basada en nuestro framework en tres escenarios: ejecución en paralelo, acceso en paralelo a datos y una combinación de ambos. Hemos llevado a cabo experimentos para medir el rendimiento de la infraestructura propuesta en un entorno basado en máquinas virtuales en un clúster local compuesto por hasta 100 nodos. Los resultados nos llevan a obtener conclusiones interesantes sobre el esfuerzo computacional y a proponer pautas cuando se enfrentan problemas de optimización de Big Data. En primer lugar, nuestro enfoque es capaz de mejorar tiempos de computación de más de una semana a solo medio día, al abordar tareas de optimización complejas y lentas. Este rendimiento se ha obtenido en el ámbito de un entorno de cómputo local (virtualizado) con recursos limitados. En segundo lugar, en aquellos experimentos donde solo nos centramos en la administración de datos sin gastar esfuerzo computacional adicional en la evaluación de las soluciones, el rendimiento general del modelo paralelo generalmente se ve afectado cuando se usa una gran cantidad de nodos (más de 50) debido a la sobrecarga de la red y la creciente transferencia de datos entre nodos. Finalmente, se obtiene una compensación notable entre los requisitos de rendimiento y recursos con una configuración del clúster en el rango de 50 a 100 nodos. Hemos diseñado y desarrollado la versión dinámica de los algoritmos NSGA-II, R-NSGA-II, NSGA-III, MOCell, SMPSO y WASF-GA.

2. ***Diseño y análisis de algoritmos multi-objetivos interactivos***. En esta tesis proponemos dos algoritmos interactivos multiobjetivo. Por un lado, presentamos por primera vez en el estado del arte, un algoritmo interactivo multi-objetivo dinámico, llamado InDM2, *es una metaheurística interactiva de optimización multiobjetivo para resolver problemas dinámicos de optimización multiobjetivo.* Esta propuesta permite al DM cambiar interactivamente la región de interés que desea aproximar dando y actualizando un punto de referencia que contiene sus preferencias. InDM2 incorpora un algoritmo evolutivo basado en puntos de referencia como optimizador de base, que actualmente incluye los algoritmos WASF-GA y R-NSGA-II, lo que de hecho permite cambiar los puntos de referencia durante el proceso de optimización. Para ayudar al DM, las aproximaciones obtenidas por el algoritmo se muestran en una ventana gráfica. Un componente clave de InDM2 es que su diseño interno permite especificar diferentes estrategias de reinicio que se aplicarán cuando se detecten cambios en el punto de referencia y/o en la configuración del problema, haciéndolo más versátil. Analizamos InDM2 para la resolución de DMOP (usando la famila de problemas multi-objetivos y dinámicos FDA) y una versión dinámica del problema de optimización combinatoria de dos objetivos Traveling Salesman Problem, creada con datos de tráfico transmitidos en tiempo real de la ciudad de Nueva York. InDM2 puede reaccionar y adaptarse cuando cambian el problema y los puntos de referencia. Además, puede manejar preferencias de forma interactiva, siendo capaz de generar una aproximación ajustada a las preferencias dadas (es decir, la región de interés), en tiempo real, mientras que el problema también cambia al mismo tiempo. Por otro lado, presentamos SMPSO/RP, esta es una extensión de la metaheurística

SMPSO que incorpora un mecanismo de articulación de preferencias basado en la indicación de puntos de referencia. Nuestro enfoque permite cambiar los puntos de referencia de forma interactiva y evaluar las partículas del enjambre en paralelo. Comparamos SMPSO/RP con otras metaheurísticas interactivas como gSMS-EMOA, gNSGA-II y WASF-GA. Los resultados mostraron que SMPSO/RP logra el mejor rendimiento general al indicar puntos de referencia alcanzables e inalcanzables. También hemos medido las reducciones de tiempo que se han logrado al ejecutar el algoritmo en una plataforma de procesador de múltiples núcleos.

3. ***Diseño y análisis de un Artificial Decision Maker para evaluar metaheurísticas interactivas***. Presentamos un artificial decision maker (ADM) para la articulación de preferencias en forma de puntos de referencia guiados por PSO. Este enfoque, denominado ADM-PSO, permite comparar EMO basados en puntos de referencia interactivos sin involucrar DM humanos. El enfoque propuesto se evaluó con los problemas de referencia de DTLZ con varios objetivos y utilizando R-NSGA-II y WASF-GA como métodos interactivos basados en puntos de referencia para ser comparados. Los resultados experimentales mostraron que ADM-PSO es útil y eficiente en comparación con el otro ADM del estado del arte. Además nuestro enfoque ofrece un mecanismo bio-inspirado y flexible para capturar el contexto actual de un ADM en procesos de solución interactivos.

4. ***jMetalSP***. Hemos diseñado e implementado un software para desarrollar y analizar algoritmos multi-objetivos dinámicos e interactivos. jMetalSP puede gestionar el proceso de streaming, por lo que es capaz de manejar los datos en tiempo real. Todas las metaheurísticas utilizadas en esta tesis han sido desarrolladas usando jMetalSP. De esta manera, cualquier investigador puede reproducir fácilmente los resultados presentados aquí. Además, como jMetalSP está basado en jMetal, tiene todos los algoritmos, operadores, tipo de soluciones, etc., heredados de él. Hemos diseñado una metodología para cubrir fácilmente cualquier metaheurística estática de jMetal a dinámica para jMetalSP. Por todos estos motivos, consideramos jMetalSP como una contribución importante de esta tesis.

5. ***Diseño y análisis de un modelo ontológico para la optimización en Big Data***. Proponemos un enfoque ontológico, llamado BIGOWL para proporcionar un marco conceptual para la anotación de análisis de Big Data. Para demostrar la ventaja de utilizar BIGOWL, se han desarrollado dos casos de estudios, que consisten en: primero, el procesamiento de datos de tráfico en tiempo real para la optimización de rutas en el entorno urbano; y segundo, un problema académico de clasificación usando minería de datos. En ambos casos la plataforma de ejecución puede realizarse tanto en un cluster local o en la nube. Los resultados obtenidos de estos dos casos de uso revelaron que el enfoque BIGOWL es útil cuando se integra el dominio del conocimiento en el análisis de un problema analítico específico. En concreto, el conocimiento integrado se utiliza para guiar el diseño de workflows analíticos de Big Data, ya que va recomendando los siguientes componentes que han de usarse para la realización completa del análisis que se desea realizar, desde la captura de datos hasta su validación final.

6. ***Diseño de una metodología para anotar workflow analíticos de datos***. En esta tesis doctoral, capturamos toda la semántica necesaria para guiar el diseño inteligente de workflow analíticos en Big Data y mejorar su rendimiento. Por lo tanto, siguiendo la ontología BIGOWL, es posible anotar algoritmos analíticos, conjuntos de datos, problemas y workflows en el contexto de Big Data. El procedimiento de anotación y consulta semántica involucra todos los conceptos clave en el proceso analítico: algoritmos, características tecnológicas/de plataforma y atributos del dominio del conocimiento del problema. Además de consultas automáticas mediante el leguaje de consultas SPARQL. Indicar también que diseñamos una

serie de reglas SWRL para razonar nueva información que no se expresa explícitamente en la base de conocimiento. Esta nueva información indicará, cuando corresponda y entre otros, si un workflow analítico está compuesto correctamente o no.

Resumiendo, durante esta tesis hemos realizado una gran cantidad de contribuciones al campo de la optimización en Big Data. Desde el punto de vista algorítmico, se han desarrollado y analizado nuevas técnicas que abordan diferentes cuestiones. Desde el punto de vista de las aplicaciones, hemos abordado varios problemas de ingeniería pertenecientes a diferentes áreas, mostrando la utilidad de nuestras propuestas para abordar los problemas que podrían surgir en la academia y la industria. Y desde el punto de vista semántico, hemos desarrollado una ontología que se ocupa de la anotación de workflows analíticos.

Trabajos Futuros

Como líneas de investigación futuras en general, planeamos continuar esta propuesta de análisis de algoritmos de optimización Big Data. En particular, uno de los temas que encontramos de particular interés es el estudio de algoritmos de optimización multi-objetivo dinámicos e interactivos para entornos Big Data. Nuestro objetivo es llevar a cabo este tipo de estudios, ya que este tipo de metaheurísticas pueden abordar el problema del mundo real, ya que son capaces de detectar cambios en el problema y hacer frente a ellos. La interactividad ayuda al analista a agregar su información preferida y facilitar la búsqueda en el espacio de la solución del problema. Hoy en día, aparecen algunos estudios que apuntan en esta dirección [25, 26], donde la idea es combinar algoritmos tradicionales con operadores interactivos para agregarles interactividad. También nos gustaría continuar trabajando en el diseño de nuevos ADM. Es conceptualmente intuitivo y directo, aunque abre una línea prometedora de investigación futura. Los trabajos futuros que se quieren abordar como continuación de esta tesis los podemos resumir de la siguiente manera:

- Queremos agregar semántica en algoritmos multi-objetivos dinámicos e interactivos para resolver problemas de optimización en Big Data. A través de la semántica, podemos agregar el dominio del problema al algoritmo interactivo, mediante puntos de referencia, y mejorar así el proceso de búsqueda de soluciones.

- Planeamos explorar las posibilidades de usar diferentes metaheurísticas como DE, CMA-ES y GA para la generación de puntos de referencia en lugar de PSO en nuevos ADM.

- Otra investigación futura es probar el ajuste de parámetros en PSO (y otras metaheurísticas), por ejemplo, φ_1 y φ_2, para controlar la influencia del punto de referencia actual (mejor global) y o el historial local de partículas, por lo tanto, inducir el comportamiento de ADM en términos de mecanismos de intensificación/diversificación.

- Queremos llevar a cabo más comparaciones de iEMOs de última generación, para probar sus comportamientos en un marco de ejecución controlado y computacionalmente justo.

- También planeamos aplicar los algoritmos propuestos en esta tesis para resolver nuevos problemas en el mundo real. En concreto, estamos interesados en aplicarlos para resolver problemas más complejos relacionados con las ciencias de la vida, como la bioinformática y la agroalimentación.

- Tomando como punto de partida el artículo *Building and Using an Ontology of Preference-Based Multiobjective Evolutionary Algorithms* [27], queremos mejorar BIGOWL añadiéndole algoritmos multi-objetivo basados en preferencias.

- De la misma manera que hemos anotado jMetalSP con BIGOWL, planeamos anotar bibliotecas de Machine Learning como: Spark MLlib, BIGML, Weka, etc.

- Queremos aplicar la idea de usar semántica para diseñar workflows en el diseño de algoritmos, es decir, podemos usar la semántica para anotar los componentes que conforman a los algoritmos de optimización (operadores de selección, cruce, mutación, etc.) y a través de reglas de razonamiento, generar nuevos diseños de algoritmos. Actualmente, están apareciendo algunos estudios que abordan la generación de algoritmos de diseño automático, como [28, 29, 30, 31, 32, 33].

Chapter 1

Introduction

1.1 Motivation

Over the past decade the rapid rise of creating data in all domains of knowledge such as traffic, medicine, social network, industry, etc, has highlighted the need for enhancing the process of analyzing large data volumes, in order to be able to manage them with more easiness and in addition, discover new relationships which are hidden in them. *Big Data* is the the approach when an extremely large data volume are analyzed. One of the main reasons why Big Data has emerged, is because classical algorithms are not able to manage this huge amount of data due to they were not designed for this purpose. For example, definition of storing and accessing data is missing in this kind of classical algorithms.

The exploitation of Big Data in various sectors has a potential socio-economic impact far beyond the specific European Big Data market. To this end, in 2014, it was launched *Big Data value association* (BDVA)[1], that is the private counterpart to the EU Commission to implement the Big Data Value Public Private Partnership (BDV PPP) programme. BDVA has over 190 members all over Europe with a well-balanced composition of large and small and medium-sized industries as well as research organizations. With the goal of specifying the challenges of Big Data in the period 2016-2020, BDVA creates the *Strategic Research and Innovation Agenda (SRIA 4.0)* [34] that describes, as a model, a research and innovation roadmap for Big Data in Europe [35].

This thesis tries to fall in the challenges of SRIA (*Data Management, Data Processing Architectures, Data Analytics and Data Visualization and User Interaction*) so we cope with them as explained in next Chapters and publications. Furthermore, this work is aligned within the scope of the Spanish Ministry of Education and Science's project (TIN2014-58304-R) (I am supported by Grant BES-2015-072209 (Spanish Ministry of Economy and Competitiveness)) and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación). The project, which is aligned with this PhD Thesis, is called *Perception*, whose goal is to make a real contribution in Big Data analysis, in order to facilitate the process of analysis throughout the use of semantics during the design of analytic workflows.

Optimization problems, which are commonly found in current industry, are not unrelated to this trend, therefore Multi-Objective Optimization Algorithms (MOA) should bear in mind this new scenario. This means that, MOAs have to deal with problems, which have either various data sources (typically streaming) of huge amount of data. Indeed these features, in particular, are found in Dynamic Multi-Objective Problems (DMOPs), which are related to Big Data optimization problems. Mostly with regards to velocity and variability. When dealing with DMOPs, whenever

[1] `http://www.bdva.eu/`

there exist changes in the environment that affect the solutions of the problem (i.e., the Pareto set, the Pareto front, or both), therefore in the fitness landscape, the optimization algorithm must react to adapt the search to the new features of the problem [36]. This means that dynamic multi-objective optimization metaheuristic ought to be able to detect when the problem changes and to apply a strategy to cope with the changes, meaning they have to be interactive with the context [37].

A Decision Maker(DM) is a person who is an expert in the domain of knowledge of the optimization problem and can express his/her preference information to choose a single, the most preferred solution when working with multi-objective optimization problems, this fact helps the algorithm to bound the area where searching the preferred solution. DM issues a decision to choose between multiple criteria about the problem, which are usually in conflict with each other. Many decision and planning problems involve multiple conflicting objectives that should be considered simultaneously. Such problems are generally known as Multiple Criteria Decision Making (MCDM) problems [38].

This process makes easier for the DM to learn about the problem being considered, and about their own and others' values and judgments. The final outcome of a MCDM process should provide DM with information to help to identify the most preferred solution [39].

In interactive methods, the idea is to form a solution pattern, where the DM is allowed to guide the search for better solutions (according to his/her preferences), and then repeat this pattern until a solution preferred by the DM is found.

DM's preferences can be included in interactive MOAs through different preference information modeling [40] [41]. Nevertheless, in this thesis, we have modeled this information through *reference points* since, it is straightforward and popular practice when we are working with real world problems.

In addition, the benchmarks used for analyzing the performance of interactive MOAs methods based on the reference point approach for communicating preference information are not trivial because, usually they involve human decision makers into interactive solution processes, which makes the performance of interactive methods dependent on the performance of humans using them. Consequently it becomes necessary to develop an artificial decision maker that can be able of assessing interactive methods without the participation of a human decision maker [6].

Big Data analytics are long and complex processes therefore, with the aim of simplify them, a series of steps are carried out through. A typical analysis is composed of data collection, data manipulation, data analysis and finally result visualization.

In this sense, *analytic workflows* can be seems as a network of service operations connected together by data links describing how the outputs of some operations are to be fed into the inputs of others. Consequently, workflows are useful when a process is made up of a row of tasks more or less complex [42].

In the process of creating a Big Data workflow the analyst should bear in mind the semantics involving the problem domain knowledge and its data. What is more, the semantic of the algorithm, which is used to resolve the problem, is a key issue when deciding what kind of algorithm is able to tackle with the characteristics of the problem. At this point, one of our scientific hypothesis is as follows: "*The semantic annotation of Big Data sources, components and algorithms can acts as a link to capture and incorporate the domain knowledge to guide and enhance the analytical processes*". In addition, the semantic annotation can provide the background for reasoning methods based on axiomatic and rule logic recommendations.

To this end, ontology is the standard way for describing the knowledge about a domain. In this PhD Thesis, the description of the domain of Big Data analysis such as, problems, data, algorithms, workflows, etc, is carried out using Web Semantic technology. As mentioned above, the process of developing a workflow is not trivial, so an analyst, who knows the problem domain, but perhaps, barely knows the different type of algorithms for analyzing data (optimization, data mining, deep learning, etc.) or even what components are compatible with each other, could require assistance in the moment of choosing them. An ontological approach, which has all this knowledge, will be a useful help in the process of designing new workflows. Consequently, with all those data annotated, through semantic, we are describing their data value chain.

As a global target of this PhD Thesis, we are interested in investigating the use of the semantic in the process of Big Data analysis, not only focused on machine learning analysis, but also in optimization, because it is of special interest for the industry (SRIA 4.0), although it has not been widely studied in the past. When we define a workflow, in order to analyze a Big Data problem, semantics helps us in the process of creating it, that is to say, guiding the expert (in the knowledge domain), in the creation of a component composition from the knowledge annotate through semantic.

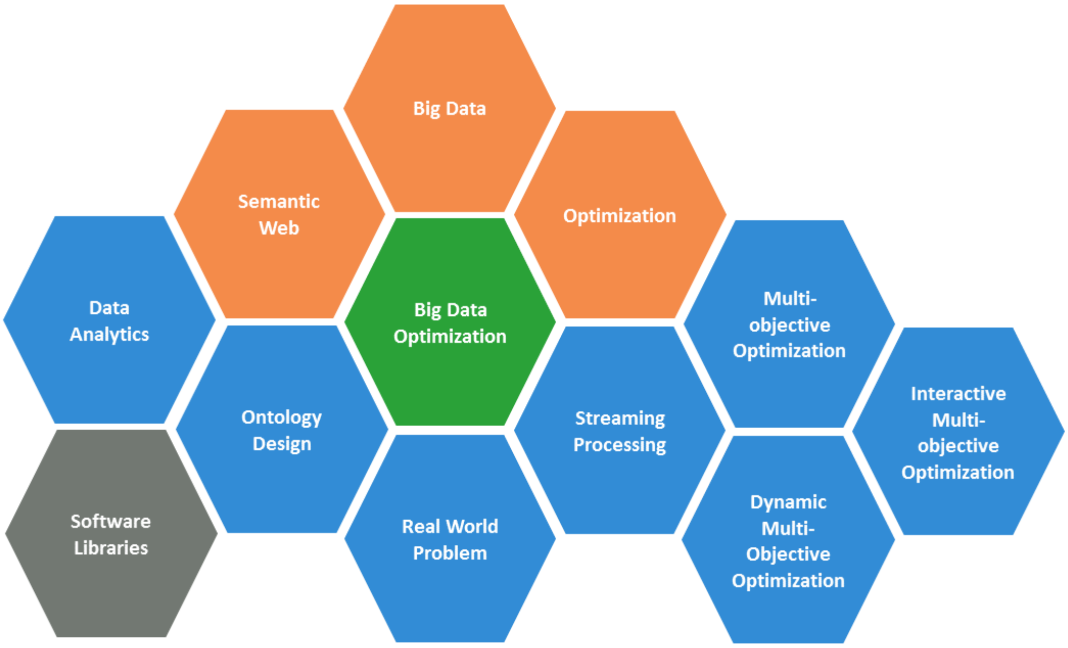


Figure 1.1: Conceptual block involving the Big Data optimization that we have covered in this PhD Thesis.

Figure 1.1 illustrates the conceptual block structure involving this PhD Thesis. It is worth noting that the main aim in this work is to cope with Big Data optimization problems. The other concepts around Big Data optimization are related to techniques or technologies for tacking it, such as streaming processing or dynamic multi-objective optimization.

Our motivation in this thesis is twofold. These steps are in fact part of the present PhD Thesis plan. First, we are interested in designing a conceptual framework using Web Semantics technology, in order to improve the design of Big Data workflows. This framework helps to analysts, who are expert on knowledge domain of Big Data problems, to design Big Data workflows so as to cope with aforementioned problems. To this point, we have designed the ontology *BIGOWL*. This ontology is designed to cover a wide vocabulary of terms concerning Big Data analytics workflows, including their components and how they are connected, from data sources to the analytics visualization. Furthermore, a semantic approach has been implemented to annotate all the involved meta-data from multiple data sources, processing components and analytic algorithms. The meta-data are integrated following the BIGOWL structure and stored in a common RDF repository. Second, we are aimed at solving Big Data optimization problems, so for this purpose we have designed and developed the framework *jMetalSP*. This is a software platform for dynamic multi-objective Big Data optimization, which combines the features of the jMetal framework for multi-objective optimization metaheuristics with the Apache Spark cluster computing system. Therefore, jMetalSP tackles dynamic Big Data optimization problems in Hadoop/Spark clusters with different data sources. In addition, jMetalSP contains interactive multi-objective metaheuristics for giving to DMs the possibility of adding their preferences in multi-objective algorithms.

In order to assess our proposals, we have used real world problems such as, Travelling Salesman Problem (using real-data), or Gene Regulatory Network Analysis.

Finally, as deliverable of this thesis, all that work has also led the development of a software in the scope of Big Data optimization, jMetalSP framework and semantic approach such as BIGOWL ontology and RDF repository are freely available. Besides, a number of realistic problem instances, simulation rule files, software scripts, and free web sites have been generated and are available.

1.2 Objectives and Phases

This PhD Thesis focuses on Big Data analytics and more specifically in Big Data optimization, as well as the benefits of making use of the web semantic in the process of designing a workflow.

For the sake of clarity, we list the main objectives of this thesis as follows:

- Analyze current state of Big Data analytic and identify the most important deficiencies it shows on complex analysis, focusing on optimization.
- Analyze current ontology versions in the process of design analytic workflows and identify their deficiencies, especially in the scope of Big Data optimization.
- Explore on state of metaheuristics in Big Data optimization and identify the most important deficiencies, focusing on dynamic multi-objective optimization methods and interactive multi-objective algorithms.
- Design of new Big Data optimization algorithm proposals by means of managing dynamic changes on the Big Data problems, deal with different data source, most of them streaming data at the same time an finally, applying multi-objective optimization methods in Big Data context.
- Develop innovative approaches that enhance the performance of current Big Data optimization techniques from the perspective of the quality of the solutions produced, or from the perspective of the computational effort required to reach them. Demonstrate their effectiveness through statistically assessed experimental evaluation.
- Design and develop new Artificial Decision Maker approach for assessing interactive multi-objective algorithms.
- Validate our findings in previous points for solving academic and real-world problems.

In order to fulfill these thesis objectives, the work has been carried out as follows. First, we have surveyed the concepts Big Data analytics and, in concrete, Big Data optimization. Then, we have focused on identifying deficiencies of Big Data optimization algorithms. After this, we have proposed advanced design of dynamic multi-objective optimization algorithms and as well, interactive multi-objective metaheuristics. Second, we have identified semantic approach for Big Data Analytic. Then we have focused on designing a semantic approach in Big Data optimization.

Finally we have applied our methods and semantic approach for solving academic problems (dynamic FDA, ZDT, DTLZ families problems) and aforementioned real-world problems taken from different disciplines of Engineering.

1.3 Thesis Contributions

The contributions of this thesis are mainly devoted to the research field of Big Data optimization. However, additional interesting contributions can be also highlighted in the research fields of bioinformatics or traffic management. To summarize, the main contributions of this thesis are shown below:

- This thesis addresses a key challenge nowadays, that is to adapt current metaheuristics to cope with Big Data optimization [43]. As a result, we have developed jMetalSP, which is a Big Data optimization framework.
- Include semantic in the design of workflow whose aim is to indicate the steps so as to analyze a Big Data problem, thus describing the data value chain of those workflows and their components. As a result, BIGOWL ontology has been created [18].
- We have tackled real-world complex problems with dynamic multi-objective algorithms. In concrete, we have focused in this thesis on traffic problem.
- In view of the capacity of parallelism of jMetalSP, we tackle the bioinformatic problem, Scalable Inference of Gene Regulatory Network.
- It has been designed a methodology in order to transform any multi-objective algorithm from jMetal in a dynamic multi-objective algorithm in jMetalSP. All of them are well-known algorithms.
- We have designed and developed two interactive multi-objective methods (iEMO), such as InDM2 [17] and SMPSO/RF [21] using jMetalSP.
- It has been developed an Automatic Decision Maker so as to be able of assessing iEMOs, thus we have a benchmark for this kind of algorithms.
- We have published three articles in journal indexed in the Journal of Citation Report (JCR), in addition, seven articles in international conferences and three in national conference.

1.4 Thesis Organization

This thesis work is highly oriented towards the design and analysis of dynamic multi-objective metaheuristics for Big Data optimization, and this is reflected into the outline of this document. In particular, it is structured in four chapters , following this introduction.

The current Chapter contains an introduction to the work done, presenting the motivation to carry it out, the objectives that have been sought, the phases that have been followed to achieve those objectives and the main contributions of the thesis.

Chapter 2 focuses on describing the principles about all the concepts covered in this thesis, such as, general concepts of optimization, multi-objective optimization, dynamic algorithm, interactive metaheuristics, artificial decision maker, Big Data, Big Data optimization and Semantic Web. The goal of this chapter is given a general vision of the technologies and background concepts covered in this thesis.

Chapter 3 includes a full description about the software created in this PhD Thesis, it has a full description of the framework developed for dealing with Big Data optimization, jMetalSP. This chapter contains, as well, a description of BIGOWL and the cluster where we run all the tests and the methods used for assessing them. Furthermore, this chapter analyzes the performance of the techniques proposed in this thesis when facing problems with different size data or computational effort. And finally, the information about where we can find all the software developed in this thesis is given. It is shown a table with the repositories used for storing the deliverables.

Chapter 4 contains all the published work that supports this thesis with a summary of each one of them.

Finally, Chapter 5 is intended to present the main conclusions and the future research lines that can be opened by this study.

Chapter 2

Context and Fundamentals

2.1 Introduction

This Chapter provides a general vision to Big Data. Specifically the basic concept from the point of view of Big Data optimization. First, Big Data basic concepts are described and then we continue with Big Data Analytics. Finally we describe the main fundamentals of optimization, focusing on Dynamic Multi-Objective optimization, thus giving an insight into the kind of optimization carried out in this thesis.

2.2 Big Data

Over the last decades, the amount of data created has been growing year on year due to Internet of things and Social Networks. Consequently under this explosive increase of global data, the term *Big Data* is mainly used to describe enormous dataset. Compared with traditional datasets, Big Data typically includes masses of unstructured data that need more real-time analysis [1].

Nowadays, data related to the service of Internet companies grow rapidly, for instance, Google processes more than 24 petabytes of data per day, or 400 millions of tweets are sent every day [44]. However, not only Internet companies deal with Big Data, but also traditional research fields such as: transportation, biology, medicine, business activities or even national security are involved in Big Data [45]. The birth of Big Data cannot avoid mentioning another current popular term - social networks - and the relation between the two is obvious [45]. All of this, demonstrates how important is Big Data in today's society.

2.2.1 Definition and Features of Big Data

Big Data is an abstract concept, that aggregates several different definitions. For instance, one of them, Kim et al. describes the term *Big Data* as the huge volumes of (digital) data that are collected from large variety of sources that are too large, raw, or unstructured for analysis through conventional database techniques [2]. Other definitions focus on the fact that the effort to manage Big Data are higher than traditional approach. For example, Chen et al.[1] define it as, the data that cannot be processed or analyzed using traditional techniques; consequently, we not only need to change the way of dealing with Big Data, but also how we apply the new techniques, so as to cope with the huge amount of data. In fact, one of the main benefits of this new approach is that Big Data provides opportunities for discovering new values, helps us to gain an in-depth understanding of the hidden values.

As, a matter of fact, Big Data have been defined as early as 2001, in the report of Doug Laney [46] was defined challenges and opportunities brought about by increased data with a 3 V's model, i.e., the increase of Volume, Velocity, and Variety. However, nowadays Big Data definition have to take Value and Veracity into consideration because data truthfulness as well, so we talk about the 5 V's: Volume, Velocity, Variety, Value and Veracity [47].

The following is a more detailed explanation of these Big Data features.

- ***Volume.*** Over time, systems and humans being are constantly creating new data, resulting in an incredible volume of data. For instance, more than 5 billion individuals are using various mobiles devices [48] hence, the volume each year is growing more and more.
- ***Velocity.*** Velocity means the timeliness of Big Data, must be rapidly and timely conducted, thus information is flowing at high speed and should be dealt with a sensible way.
- ***Variety.*** Organized and unorganized information are producing a variety of data types. It is worth mentioning that there exist various types of data, which include semi-structured and unstructured data such as audio, video, web page, and text, as well as traditional structured data.
- ***Veracity.*** Veracity includes questions of trust and uncertainty with regards to data and the outcome of analysis of that data, thus, identifying and verifying inconsistent information is significant, to accomplish faithful study. Creating faith in Big Data is a big challenge to manage even more variety of available data.
- ***Value.*** The main goal of Big Data is to get utility value in terms of analyzing or discovering new features from the original data.

Recently the 5 V's were changed as 7 V's as *Variability* and *Visualization* were added to the list of big V's due to the increasing importance of data visualization and their format.

2.2.2 Reference Model

This thesis is aligned with some of the challenges proposes in *Big Data Value (BDV) Reference Model* [34]. The BDV Reference Model has been developed by the Big Data Value Association (BDVA), taking into account input from technical experts and stakeholders along the whole Big Data Value chain.

The BDV Reference Model defines the following priority areas in Big Data:

- ***Data Visualization and User Interaction***. Data visualization plays a key role in effectively exploring and understanding Big Data. Data generated from data analytics processes need to be presented to users. Common tasks that allow users to gain a better understanding of Big Data include scalable zooms, dynamic filtering and annotation. All the challenges of this priority are related to improve the ways of visualizing data, showing them with different scales and, what is more, the users must be able to interact with those data.
- ***Data Analytics***. The progress of data analytics is not only key for turning Big Data into value, but also for making it accessible to the wider public. Data analytics will have a positive influence on all parts of the data value chain and increase business opportunities through business intelligence and analytics, while bringing benefits to both society and citizens. The next generation of analytics will be required to cope with a vast amount of information from different types of sources, with differentiated characteristics, levels of trust and frequency of updating. Data analytics will have to provide insights into the data in a cost-effective and

economically sustainable way. This priority indicates that semantic will improve the analysis of Big Data because provide an interpretation of the data and with this meta-data algorithms will be able of improving their performances. The implementation of new frameworks (with semantic) is key in this new type of analysis.

- ***Data Processing Architectures***. The parallel need for real-time and large data volume capabilities is a key challenge for Big Data processing architectures. There exist a number of advances in Big Data analytics to support the dimension of Big Data volume. In a separate development, stream processing has been enhanced in terms of analytics on the fly to cover the velocity aspect of Big Data. Data processing architectures must address: heterogeneity of the data, scalability, real-time processing, decentralization of data storage and parallelism executions, supporting large running time and consume of memory.

- ***Data Protection***. Data protection and anonymization are major issues in the areas of Big Data and data analytics, for that, the application of legal privacy regulations, to guarantee the confidentiality of individuals' who are represented in the data is key. Data protection becomes crucial in the context of large-scale sensitive data processing. The challenges identified in this priority area are: generic data, easy use to use and enforceable data protection approach, maintain robust data privacy and calibrate risk-based approaches.

- ***Data Management***. Big Data applications have to be able to cope with the size and speed of data delivered in diverse formats and at distributed locations. Large amounts of data are being made available in a variety of formats, like: ranging from unstructured to semi-structured to structured formats, such as: text, images, Web 2.0 data, sensor data, mobile data, geo-spatial data and multimedia data. As challenge, this priority points out that semantic annotation can help in data management because let the interoperability and the integration of data with heterogeneous format.

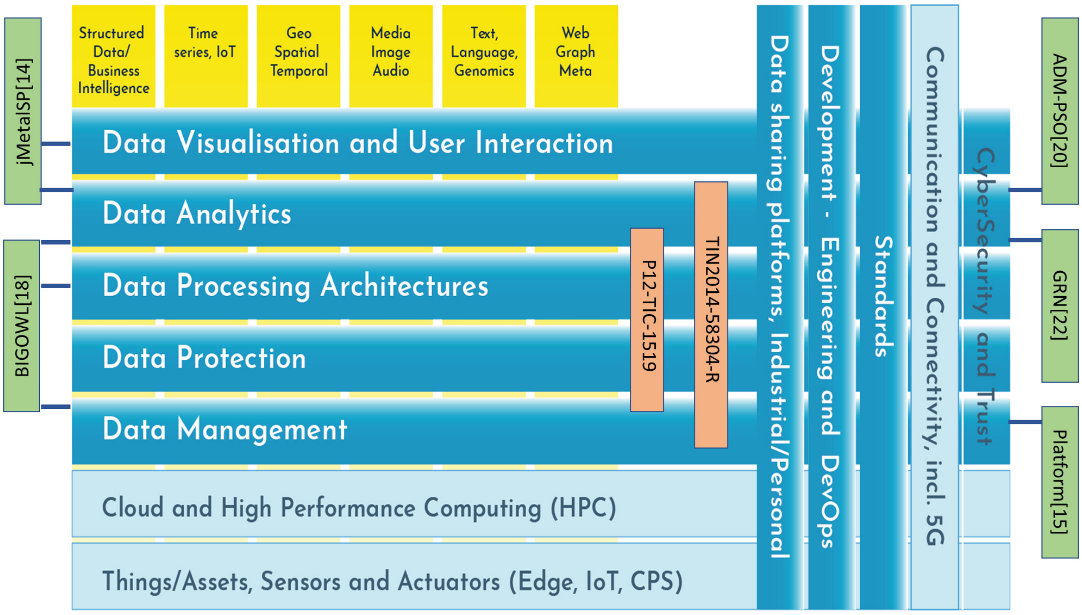

jMetalSP[14]
BIGOWL[18]
Structured Data/ Business Intelligence
Time series, IoT
Geo Spatial Temporal
Media Image Audio
Text, Language, Genomics
Web Graph Meta
Data Visualisation and User Interaction
Data Analytics
Data Processing Architectures
Data Protection
Data Management
Cloud and High Performance Computing (HPC)
Things/Assets, Sensors and Actuators (Edge, IoT, CPS)
P12-TIC-1519
TIN2014-58304-R
Data sharing platforms, Industrial/Personal
Development - Engineering and DevOps
Standards
Communication and Connectivity, incl. 5G
CyberSecurity and Trust
ADM-PSO[20]
GRN[22]
Platform[15]

Figure 2.1: BDVA Big Data model for priority areas in Big Data

In Figure 2.1 is depicted the priority areas defined by BDVA in the fourth version of Strategic Research and Innovation Agenda (SRIA) document. In green are shown the articles that support this thesis, and in light pink the research project that support it as well.

In this regard, the priority areas where this thesis falls are:

- ***Data Management***. We have designed a semantic model where we define how to annotate components of Big Data analytic, and consolidate them following data management standards. In addition, we have designed how to integrate data from different components, thus we tackle heterogeneous data format.

- ***Data Processing Architectures***. We have defined a virtualization platform for dealing with Big Data analysis, we follow the remarks indicated in SRIA document related to scalability and decentralization. For instance we have used the platform: Spark cluster, HDFS file system, etc.

- ***Data Analytics***. The main goal of this thesis is to cope with analysis in Big Data, so we define a framework (jMetalSP) for Big Data optimization and furthermore, the semantic for improving the aforementioned analysis in Big Data context.

- ***Data Visualization and User Interaction***. Since jMetalSP supports interactive algorithms, we let users (Decision Makers) add their preferences in the algorithms and furthermore they are able to see how the Pareto front changes over the time following their preferences. Consequently users can do data visualization and then choosing their preferred areas in the Pareto front.

2.2.2.1 Big Data Technology Ecosystem

Big Data is linked to a wide range of technologies whose aims are diverse, from data collection and processing to data analysis and visualization. Bellow we describe a set of key technologies involved in Big Data ecosystem.

Cloud computing

The National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce defines Cloud computing [49] as follows: *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.* The main objective of cloud computing is to use huge computing and storage resources under concentrated management, consequently both technologies, Big Data and cloud computing, give each other feedback and are growing together. On one hand the development of cloud computing provides solutions for the storage and processing of Big Data. On the other hand, the emergence of Big Data also accelerates the development of cloud computing.

Big Data uses cloud computing as a service. From this combination emerges the three most popular cloud paradigms include: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [50].

Internet of Things

The Cluster of European research projects on the Internet of Things (IoT)[51] defines it as *'Things' are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real and physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention.*

Given their inherent nature, IoT has become rapidly in an important Big Data's data source, there exists numerous applications whose data sources are from IoT, Figure 2.2 illustrates the family of applications regarded with IoT [52].

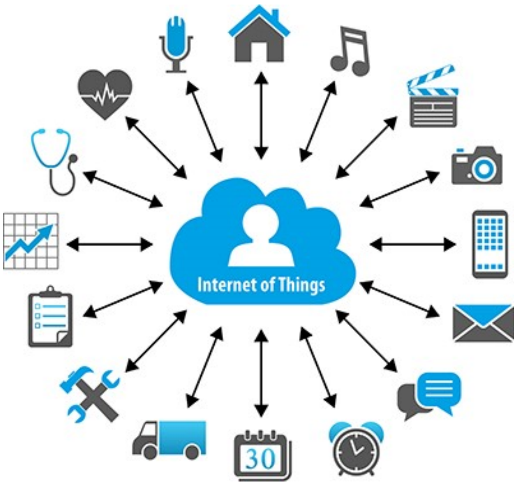


Figure 2.2: Internet of things applications and cloud computing.

NoSQL

Not Only SQL (*NoSQL*) refers to the familiar group of non-relational data management systems. The main feature of this databases is the fact that are not built primarily on tables, and generally do not use SQL for data manipulation. NoSQL database are useful when working with a huge quantity of (un)structured data or semi-structured.

NoSQL systems are distributed, non-relational databases designed for large-scale data storage and for massively-parallel data processing across a large number of commodity servers [53].

Classic relational database management systems (RDBMS) are unable to handle the rapid growth of the data with different (or without) structure of information, and that is the reason why NoSQL databases come up.

Han *et al.* [54] classifies NoSQL databases in three types regard with how it is defined their data model, *key-value*, *clumn-oriented* and *document.*

- ***Key-value***. These database store items as alpha-numeric identifiers (keys) and associate values in simple. The values may be various from simple text strings to more complex like lists or sets. Data searches can usually only be performed against keys, not values, and are limited to exact matches. Examples of this type of database are: Redis [55], Tokyo Cabinet-Tokyo Tyrant [56] or Scalaris [57].

- ***Column-oriented***. The data model of this kind of database is defined as rows and columns, although we could think this database share same model as RDBMS this is not really, because Column-oriented databases have an architecture with data compression, shared-nothing, massively parallel processing. As Column-oriented databases we could highlight the following: HBase [58], HadoopDB [59], Apache Cassandra [60] or Google's Bigtable [61].

- ***Document***. Document database and key-value share very similar structure however, the Value of document database is semantic, and is stored in JSON or XML format therefore, document stores support more complex data than the key-value stores. Documents can be any type of traditional document such as: articles, Microsoft Words, etc or other type such as: register, list, etc. Some instances of document database are: MongoDB [62] or CouchDB [63].

Hadoop and MapReduce

In the early years of 2000 Google started off the development of a distributed file system called *Google File System* (GFS) [64] whose main goal was to tackle with the issue of managing enormous amount of data using computer clusters. Only one year later, *Apache* created the project *Hadoop* that integrates the components: *MapReduce* and *Hadoop Distributed File System* (HDFS), which are inspired by GFS.

Hadoop is an open source project developed by the Apache Software Foundation under the Apache v2 license, which provides a framework to enable the distributed processing of large data sets on clusters built with a generic hardware. The processing phase [5] is divided into many fragments of work, which can be executed or re-executed, independently, in each of the nodes of the cluster. This framework is able to work with hundreds of nodes and with data quantities of the order of petabytes.

In essence, Hadoop consists of two elements: a distributed file system called HDFS (Hadoop Distributed File System) and a data processing engine, which implements the Map/Reduce programming paradigm (Hadoop MapReduce). Both, Map/Reduce and HDFS, are designed in such a way that if any type of failure (breakdown, power supply problems, etc.) come up in any of the nodes that make up the cluster, the Hadoop framework by its own is capable of solving the problem. The implementation of this feature is managed using another available node that can replace the task of the one that has suffered the breakdown. This is possible because Hadoop makes replicas of the data stored in different nodes with the idea of being able to solve these possible problems during the execution. This feature of Hadoop makes the system very robust, since before any unforeseen the application is able to react and find another way to continue with the tasks assigned to it.

In addition, as Hadoop has been gaining adoption and maturity, it does not refer to a single tool, but to a family of applications around HDFS and MapReduce. Some of these applications are Hive, HBase, Kafka and Zookeeper.

The main benefits of MapReduce are, first its scalability and second, the variety of data it can process, such as: files, database tables, websites (Web crawling), etc. To complete this stage of data processing it is common to use another component that is responsible for coordinating the workflow of the MapReduce tasks, since even the simplest MapReduce tasks consist of a large number of map functions and reduces, so it is advisable to rely on a workflow coordinator.

HDFS is a distributed file system designed to run on commodity hardware. HDFS not only stores application data but also metadata, which are stored on dedicated server, called the NameNode. Application data are stored on other servers called DataNodes. All servers are fully connected and communicate with each other using TCP-based protocols.

HDFS has a master/worker architecture, according to which, an HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like any other file system, such as: list, copy, delete and so on. It also determines the mapping of blocks to DataNodes.

The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode. DataNodes do not use data protection mechanisms such as RAID to make the data durable. Instead, like GFS, the file content is replicated on multiple DataNodes for reliability.

When *Google* realized (later *Yahoo!*) that standard algorithms were unable to deal with the amount of data stored on HDFS or GFS they developed a new programming model for processing and generating Big Data sets with a parallel, distributed algorithm on a cluster, this model is known as MapReduce.

MapReduce is a framework or programming model that supports parallel programming on large collections of data, typically it is used in the resolution of algorithms that can be parallelized, furthermore can be used in order to manage or analyze large dataset, reaching even petabytes. The MapReduce programming model was originally created by Google to process large amounts of data. It is inspired by the functions map and reduce in functional programming, however, it operates differently in this context. It is worth mentioning that the name of the framework is due to the two functions that form it, $map()$ and $reduce()$. The Map and Reduce functions are both defined with respect to structured data in tuples of the type (key, value).

- ***Map***. When an input arrives, the master split it in smalls pieces which can be run in the workers and each one sends the result back to the master, the aforementioned results are listed and sorted following the keys calculated. The formulation of this function is: $Map(k_1, v_1) \rightarrow list(k_2, v_2)$

- ***Reduce***. The master node collects all received responses and combines them to generate the output. Its formulation is the following, $Reduce(k_2, list(v_2)) \rightarrow list(v_3)$

Therefore, MapReduce transforms a list of pairs (key, value) into a list of values, these values, which are obtained during the reduction phase, can be simple data, that is, a single data or a tuple again (key , value). If the result is a tuple, this data can again pass through the MapReduce framework to be re-executed if necessary. One of the main features of this framework is that it has a great tolerance to failures, because the master node is constantly checking that the worker nodes it has are active and if it is not, if it has assigned any tasks to those worker nodes, they are redistributed to others that if they are active.

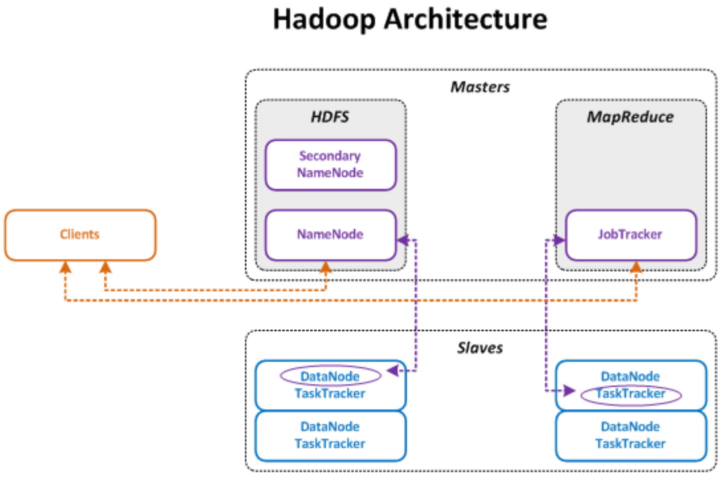


Figure 2.3: Hadoop architecture can be categorised into three - Client Machines, Master Servers and Worker Servers.

Figure 2.3 shows the process in Hadoop architecture, on the one hand task tracker instances execute the process (one in each node) and on the other hand, master has the job tracker process whose goal is allocated a process among other nodes and even itself.

2.2.3 Apache Spark

Apache Spark [4] is an open-source cluster computing framework that was originally developed in the AMPLab at University of California, Berkeley.

The Spark framework allows for reusing a working set of data across multiple parallel operations. This includes many iterative machine learning algorithms as well as interactive data analysis tools. Therefore, this framework supports these applications while retaining the scalability and fault tolerance of MapReduce. In order to achieve that goals, Spark is based on the concept of Resilient Distributed Datasets (RDD), which are collections of elements that can be operated in parallel on the nodes of a cluster by using two types of operations, transformations (e.g., map, filter, union, etc.) and actions (e.g., reduce, collect, count, etc.). Apache Spark exploits in-memory computation for solving iterative algorithms and can be run in traditional clusters such as Hadoop or Hadoop Yarn or even its own cluster, *Standalone cluster*.

Other features of Spark are a fault-tolerant and general-purpose cluster computing system, which provides APIs in Java, Scala, Python, and R. Spark is efficient at iterative computations and is thus well-suited for the development of large-scale machine learning applications, moreover, it has an optimized engine that supports general execution graphs. It is worth mentioning that Spark is able to manage streaming data, and in this thesis we use Spark not only for paralleling process but also gathering streaming data.

Different Big Data frameworks such as Apache Hadoop and Spark have been designed to allow the efficient application of data mining methods and machine learning algorithms in different domains. Based on these Big Data frameworks, others libraries have been thought to develop new efficient versions of machine learning algorithms, such as Mahout [65] and SparkMLib [66].

Since 1.6 version, Spark supports Dataframe and Dataset which are a distributed data collection and let a better managing of them in fact, this way of collection will be the future of Spark at the expense of RDD.

We use spark as core in our Big Data optimization framework (jMetalSP) for the following reasons:

1. Spark has better performance, when manages large amount of data, from a point of view of running time or memory usage than Hadoop [67, 15].

2. Spark allows parallel processing, consequently jMetalSP is able to run parallel algorithms.

3. Spark supports streaming process, this is key in our framework because we develop dynamic multi-objective problem that are changing over the time due to their real-time data are changing.

4. Spark is compatible with a tone of Big Data technologies and API such as: HDFS, Kafka, Flume, Twitter API, ZERO MQ, etc. In addition, it is easy to integrate with other Big Data technologies or data source.

5. The installation of Spark is trivial and it is compatible with all version of Unix operation systems.

6. Spark supports Java, Scala, Python and R as programming language.

7. Spark has a huge working community of contributors, therefore is always updated with the last technology.

8. Spark has Apache license so, we can use in our research projects.

We have assessed Spark solving dynamic multi-objective optimization problems [15, 12, 14] with good results from a point of view of scalability, running time and memory usage.

2.2.4 Data Types for Big Data

When we define the types of data which are present in Big Data we can sum them up in three types.

- ***Structured data***. Structured Data is any set of data values conforming to a common schema or type [68], that is to say, this kind of data refers to information with a high degree of organization, such as: sensor data, web log data, financial data (stock-trading data), to sum up, all the data that can be stored in a data base with structure.
- ***Semi-structured and complex data***. Semi-structured data are becoming more and more prevalent, Semi-structured data contain semantic tags (some organization), but do not conform to the structure associated with typical relational databases [69], for instance, emails, XML files, markup languages, etc.
- ***Unstructured data***. Unstructured data have internal structure but they are not structured via predefined data models or schema [70]. This sort of data can be stored in File System repositories like HDFS or even in NoSQL data base. Some examples of unstructured data are, text files, data from social networks, MP3, photos, videos, etc.

2.3 Big Data Analytics

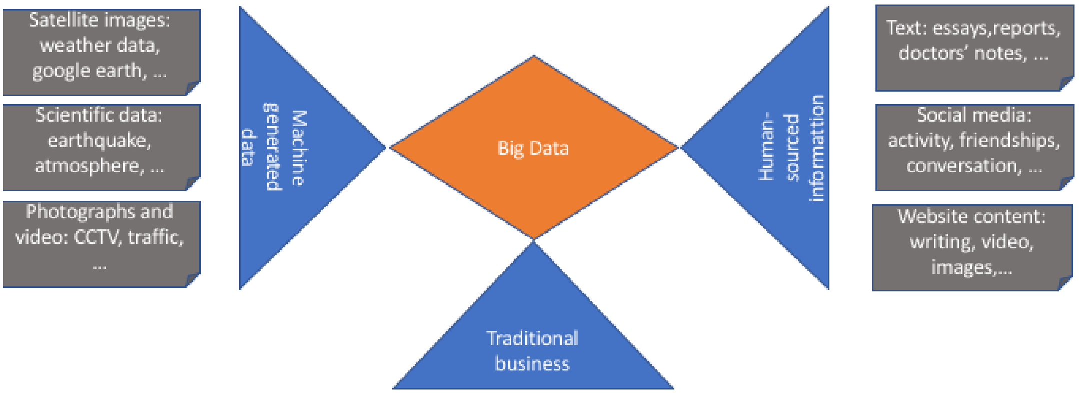

Satellite images: weather data, google earth, ...
Scientific data: earthquake, atmosphere, ...
Photographs and video: CCTV, traffic, ...
Machine generated data
Big Data
Traditional business
Human-sourced information
Text: essays,reports, doctors' notes, ...
Social media: activity, friendships, conversation, ...
Website content: writing, video, images,...

Figure 2.4: Big Data Sources.

Big Data Analytic reflect the challenges of data that are vast, unstructured, and fast moving to be managed by traditional methods. Input to Big Data systems comes from various sources. Traditionally, the largest amount of data were created by business, such as: paper, audio or photographs records. However, nowadays, there exists plenty of emerging sources thank to new technologies, for instance, different fields, which are creating new data in real-time. Furthermore social networks are a powerful data source, Figure 2.4 shows a briefly view of data sources.

Big Data analytics have become increasingly important in both the academic and the business communities over the past years. Gleaning meaningful information and competitive advantages from massive amounts of data has become increasingly important to organizations globally. Trying to efficiently extract the meaningful insights from such data sources quickly and easily is challenging. Thus, analytics has become inextricably vital to realize the full value of Big Data to improve their business performance and increase their market share.

2.3.1 Data Mining

Since Big Data is not only large, but also varied and fast growing many technologies and analytical techniques are needed in order to attempt extracting relevant information. There are a myriad of analytic techniques that could be employed when attacking a Big Data project. Which ones are used, depends on the type of data being analyzed, the technology available to you, and the research questions you are trying to solve. One of the most used due to the amount of variants it offers is *Data Mining.*

Brown et al. [71] defines Data Mining as the "combining methods from statistics and machine learning with database management" in order to pinpoint patterns in large datasets. Data mining is an area that has taken much of its inspiration and techniques from machine learning (and some, also, from statistics).

Data mining can be divided into two main classes of tools: model building and pattern discovery. Model building is a high level global descriptive summary of datasets, which in modern statistics include: regression models, cluster decomposition and Bayesian networks. Models describe the overall shape of the data. Pattern is a local structure, in a possibly vast search space, describing data with an anomalously high density compared with that expected in a baseline model. Patterns are usually embedded in a mass of irrelevant data [72].

However, in this PhD Thesis we focus on Big Data optimization because, as we said in Chapter 1, it is of special interest for the industry (SRIA 4.0) and furthermore, it has not been widely studied yet.

2.3.2 Big Data Optimization

Currently, the growing interest in Big Data applications [43], where many of them require to process large amounts of data coming at great speed from streaming data sources, brings new opportunities to apply dynamic multi-objective optimization. The rationale is that both, Big Data and multi-objective optimization are found in many disciplines, such as: transportation, economics, agri-food, mobility, and medicine; so it is foreseeable that they converge in a near future leading to multi-objective Big Data optimization problems hence, Big Data optimization includes the high dimensionality of data, the dynamical change of data, and the multi-objective of problems and algorithms.

In recent years, the content of the Big Data has been increasing over time, and the goal of Big Data analytic also changes with time. The algorithm not only should be able to handle the dynamical changing data, but also adjusting the target of data analytic. One of the foremost characteristics of Big Data is that the data came from different sources, so as to create together the large dataset. Generally, more than one objective needs to be satisfied at the same time in these large dataset. The most traditional methods can only be applied to continuous and differentiable functions, and have to perform a series of separate runs to satisfy different objectives [73].

In the last decade, a number of approaches consisting in adapting metaheuristic techniques to work in parallel on Hadoop ecosystems have been proposed. These proposals are related to data mining or data management applications, like: a swarm intelligence method to optimize the feature selection in Big gene expression datasets [74], data partitioning in Big Databases [75], dimension reduction in Big Data analytics [76], pattern detection with Artificial Immune Algorithms [77], a parallel MapReduce evolutionary algorithm for graph inference [78], and a parallel artificial ant colony optimization for task scheduling in clusters environments [79]. Most of these approaches are based on the MapReduce programming model. However, MapReduce entails a series of disadvantages that make it unsuitable to be properly integrated in global optimization techniques in general, and metaheuristics in particular. These drawbacks are mostly related to: high latency queries, non-iterative programming model, and weak real-time processing. Therefore, new challenging approaches are demanded to integrate Big Data based technologies with global optimization algorithms in order address all these issues. In this thesis we present a new framework in order to cope with all that issues, namely jMetalSP.

Big Data analytics and specifically Big Data optimization, are divided into four components: handling large amount of data, handling high dimensional data, handling dynamical data, and multi-objective optimization. Most real-world Big Data problems can be modelled as a large scale, dynamical, and multi-objective problems.

2.4 Fundamentals of optimization

In this section, the principles of optimization are outlined and basic concepts are formally defined. We focus on dynamic multi-objective algorithms, *metaheuristics*, for dealing with Big Data optimization problems.

Optimization in the sense of finding the best solution, or at least a good enough solution, for a problem is a vital importance field in the real world and, particularly, in Big Data problems. We are constantly solving optimization problems, as searching the shortest path to go from some place to another, organizing our activity schedule, etc. Generally, these problems are small enough, so it is possible to solve them by ourselves without additional help. However, as these problems get larger and more complex, computer assistance is inevitable to solve them.

We start giving a formal definition about the concept of optimization. Assuming, without loss of generality, the minimization case, we can define an *optimization problem* as follows:

Definition 1 (Optimization problem)**.** *An optimization problem is formalized as a pair* (S, f)*, where* $S \neq \emptyset$ *represents the solution space (or search space) of the problem, while* f *is a function named* objective function *or* fitness function*, that is defined as:*

$$f : S \rightarrow \mathbb{R} \ . \tag{2.1}$$

Therefore, solving an optimization problem consists in finding a solution, $i^* \in S$*, that satisfies the following inequality:*

$$f(i^*) \leq f(i), \qquad \forall \ i \in S \ . \tag{2.2}$$

Assuming the case of maximization or minimization does not restrict the generality of the results, as it is possible to establish an equality between maximization and minimization problems as follows [80, 81]:

$$\max\{f(i)|i \in S\} \equiv \min\{-f(i)|i \in S\} \ . \tag{2.3}$$

Depending on the domain to which S belongs, we can define *binary* ($S \subseteq \mathbb{B}^*$), *integer* ($S \subseteq \mathbb{N}^*$), *continuous* ($S \subseteq \mathbb{R}^*$), or *heterogeneous* ($S \subseteq (\mathbb{B} \cup \mathbb{N} \cup \mathbb{R})^*$) optimization problems.

On the other hand, multi-objective optimization is needed whenever there are several conflicting objectives functions to be optimized simultaneously.

We present the formulation of a multi-objective optimization problem as

$$\begin{array}{ll} \text{minimize} & \{f_1(x), f_2(x), ..., f_k(x)\} \\ \text{subject to} & x \in S \subset \mathbb{R}^n \end{array} \tag{2.4}$$

with $k \geq 2$ conflicting objective functions $f_i : S \rightarrow \mathbb{R}$ and where x is a vector of decision variables from the feasible set S. We can denote an objective vector by $z = f(x) = (f_1(x), f_2(x), ..., f_k(x))^T$.

In multi-objective optimization, our aim is to optimize the values of several objectives at the same time however, usually there exist no single point within the search space where all the objectives reach their individual optima. In other words, if we want to gain something, we must give away something else.

In general, problem (2.4) has many optimal solutions with different trade-offs. These optimal solutions are called Pareto optimal solutions, the so-called Pareto optimal solutions forming a Pareto optimal set. With the goal of being more specific, in (2.4), a design variable vector $x^{'} \in S$ and the corresponding objective vector z are called Pareto optimal if there does not exist another $x \in S$ such as $f_i(x) \leq f_i(x^{'})$ for all $i = 1, ..., k$ and $f_j(x) < f_j(x^{'})$ for at least one index j.

Therefore, a set of all Pareto optimal solution in S is called a Pareto optimal set and the image of the Pareto optimal set in $f(S)$ is called a Pareto optimal front as shows Figure 2.5.

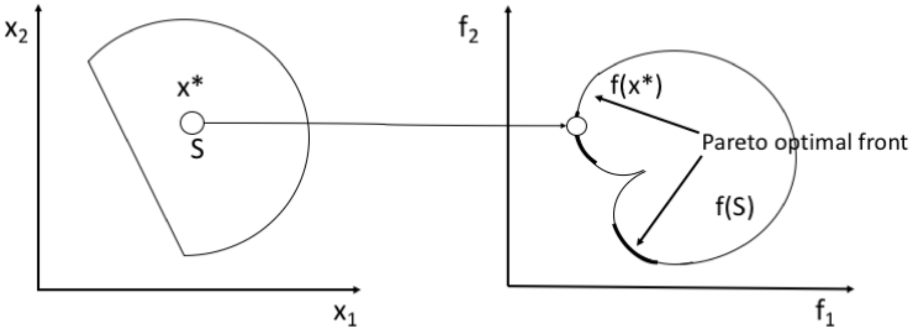


Figure 2.5: Pareto optimal front.

The precedent definition applies for global Pareto optimality however, also local Pareto optimality may be defined. A decision vector $x^* \in S$ is locally Pareto optimal if there exists a neighborhood $N(x^*)$ of x^* such that x^* is Pareto optimal in $N(x^*) \cap S$. The objective vector $f(x^*)$ is locally Pareto optimal if the corresponding point x^* is locally Pareto optimal.

The concept of dominance is related to Pareto optimality. In Equation 2.4, an objective vector z^1 is said to *dominate* another vector z^2 if $z_i^1 \leq z_i^2$ for all $i = 1, ..., k$, and the inequality is strict for least on index i. Furthermore, an objective vector z^1 is *non-dominated* if there does not exist another objective vector z^2 such that z^2 dominates z^1. Therefore, Pareto optimal points are non-dominated points. Figure 2.6 illustrates the concepts of dominance and non-dominance. It illustrates graphically both concepts, Pareto dominance and Non-dominance. Specifically, it shows two distinct sets of solutions computed for a multi-objective problem where the two objective functions, f_1 and f_2, are to be minimized. Since both objectives are equally important, it is not trivial to decide which solution is better. Considering the previous definitions, we can say that a is better than b in the picture on the left as $f1(a) < f1(b)$ and $f2(a) < f2(b)$, for instance., a is better in all the objective functions; thus, we say that *a dominates b* ($a \prec b$). The same can be said with respect to a and c: $f1(a) < f1(c)$ and $f2(a) < f2(c)$, thus $a \prec c$. Let's compare now solutions b and c. In this case, we can observe that c is better than b in the f_1 objective function, but b is better than c in $f2(f2(b) < f2(c))$. According to the concept of dominance, as we have seen above, we cannot say that b dominates c, nor c dominates b. In this case, the solutions are said to be non-dominated with respect each other. In the right side graphic of Figure 2.6, we show four non-dominated solutions, where none can be said to be better than the others. Thus, solving a MOP consists in computing the set of solutions that dominates every other point in the solution space; this means that the solutions in that set are optimal for that problem

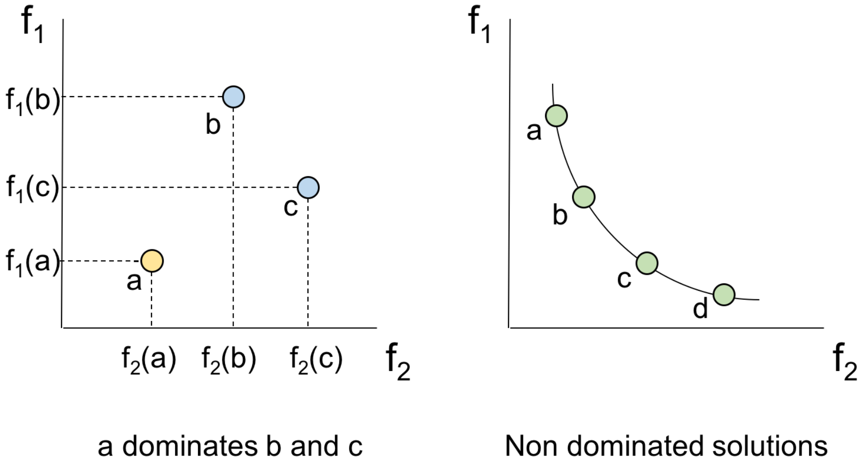


Figure 2.6: Dominance in multi-objective optimization: (left) solution 'a' dominates 'b' and 'c',(right) non-dominated solutions.

The limits of objective function values in the Pareto optimal front (called Pareto front) are defined by *ideal* and *nadir* objective vectors. An ideal objective vector $z^* \in \mathbb{R}^k$ gives lower bounds for the objective functions, and it is obtained by minimizing each objective function individually subject to the constraints. A vector strictly better than z^* can be called an *utopian objective vector* z^{**} therefore, we set $z_i^{**} = z_i^* - \epsilon$ for $i = 1, ..., k$, where ϵ is a small positive integer. Finally, a *nadir objective vector* z^{nad} is the upper bounds of objective function values in the Pareto optimal set, it is not easy to calculate; consequently, these values are usually only approximated for instance, by using pay-off tables [82, 83]. Examples of ideal, utopian and nadir objective vectors are shown in Figure 2.7. Since there exist many Pareto optimal solutions, a decision maker is needed for choosing one solution among them. A decision maker (DM) is a person who is an expert in the domain of the multi-objective optimization problem and can express her/his preference information to choose the most preferred solution. A very popular type of preference articulation in multi-objective methods is based on *reference points* [38, 84].

Reference point is denoted by $\mathbf{q}_t = (q_{t,1}, \ldots, q_{t,k})^T$. It is said to be *achievable* for problem 2.4, if $\mathbf{q}_t \in Z + \mathbb{R}_+^k$ (where $\mathbb{R}_+^k = \{\mathbf{y} \in \mathbb{R}^k \mid y_i \geq 0 \text{ for } i = 1, \ldots, k\}$), that is, if either $\mathbf{q}_t \in Z$ or if $\mathbf{q}_t$ is dominated by a Pareto optimal objective vector in Z. Otherwise, the reference point is said to be *unachievable*, that is, not all of its components can be achieved simultaneously.

Reference point represents the region of interest in the form of desirable objective function values, therefore, a reference point is regarded as an intuitive way to express preferences.

However, such methods are regarded as ad hoc methods, i.e., methods where the DM cannot be replaced by a value function [83, 85].

2.4.1 Classification of Multi-objective Optimization Methods

To solve the optimization problem 2.4, efficient search or optimization algorithms are needed. There exists many multi-objective optimization algorithms which can be classified in many ways, depending on the focus and characteristics.

On one hand, multi-objective optimization methods can be classified according to the role of the decision maker in the solution process [86]:

- *No-preference methods*: This methods are usually used when there is no DM available or

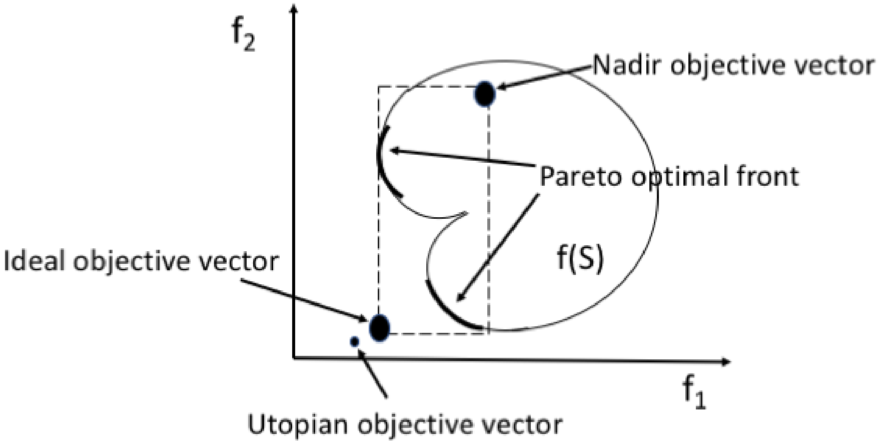


Figure 2.7: Ideal, utopian and nadir objective vectors in the objective space

.

when no preference information from the DM is available consequently, the opinions of the DM are not taken into consideration.

- *A posteriori methods*: In this methods the exploration is made as wide as possible to generate as many compromise solutions as possible. It is, then, when the decision-making process by the expert takes place. Precisely, because of this approach, these *a posteriori* techniques are being used in the field of metaheuristics and, particularly, in the field of evolutionary computing [87, 88]. More specifically, the most advanced algorithms apply *a posteriori* techniques based on the concept of *Pareto Optimality* [89].
- *A priori methods*: In this methods, the DM specifies his/her preferences before the solution process. The main withdraw of this method is that DM does not necessarily know beforehand whether the solutions will be realistic. Lexicographic ordering [90] and goal programming [91, 92] are examples of a priori methods.
- *Interactive methods*: In interactive methods, the DM articulates preference information interactively and thus directs the solution process progressively. Only part of the Pareto optimal set has to be generated and evaluated, and based on this data the DM can further adjust his/her preferences as the solution process continues. The procedure is iterated until the DM is satisfied with the Pareto optimal solution. The advantage of using interactive methods is that the DM can guide the solution process and simultaneously learn about the different trade-offs between different solutions. In the last decades, many interactive methods have been proposed, such as: step method [93], reference point method [94], satisficing trade-off method [95], NIMBUS method [96], etc. Recently, several evolutionary algorithm based interactive methods have also proposed, such as progressively interactive evolutionary multi-objective algorithm [97, 17, 98, 99] or interactive territory defining evolutionary algorithm [100].

On the other hand, there exists other ways for classifying the multi-objective methods, such as *static* or *dynamic* multi-objective optimization or, from a point of view of the number of population, multi-population [101] or one population, and even from a different perspective, optimization algorithms can be classified into *trajectory-based* and *population-based.* A trajectory-based algorithm typically uses one solution at a time, which will trace out a path as the iterations continue.

Whereas, population-based algorithms such as Particle Swarm Optimization (PSO), use multiple agents which will interact and trace out multiple paths [102].

2.4.2 Multi-Objective Optimization Methods

The techniques used to solve MOPs are not usually restricted to finding a single solution, but a set of compromise solutions between the multiple conflicting objectives, since there is usually no solution that simultaneously optimizes all objectives. Two stages can therefore be distinguished when addressing this type of problem: on the one hand, the optimization of several objective functions involved and, on the other hand, the decision-making process on which compromise solution is most appropriate [87].

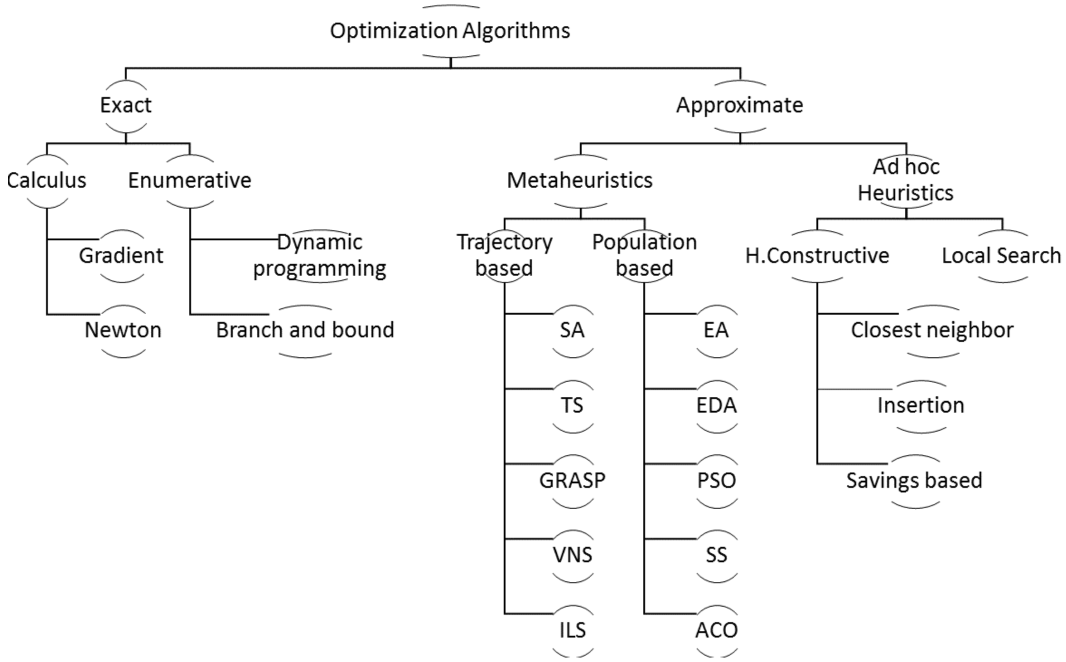


Figure 2.8: General classification of optimization techniques

A simple classification of the optimization methods used throughout the history of Computer Science is shown in Figure 2.8. In a first approach, the techniques can be classified into Exact and Approximate. Exact techniques, which are based on the mathematical finding of the optimal solution, or an exhaustive search until the optimum is found, guarantee the optimality of the obtained solution. However, these techniques present some drawbacks. The time they require may be very large, especially for NP-hard problems. Furthermore, it is not always possible to find such an exact technique for every problem. This makes exact techniques lead to poor performance, since both their time and memory requirements can become unreasonably high for large scale problems. For this reason, approximate techniques have been widely used by the international research community in the last few decades. These methods sacrifice the guarantee of finding the optimum in favor of providing some satisfactory solution within reasonable times. Among approximate algorithms, we can find two types: ad hoc heuristics, and *metaheuristics*, which we focus on this thesis.

Metaheuristics are approximate algorithms that combine basic heuristics methods in higher level frameworks aimed at efficiently and effectively exploring a search space, Blum et al. [103]

summarize metaheuristics characteristics as:

- Metaheuristics are strategies that guide the search process.
- The goal is to explore, in an efficiently way, the search space so as to find optimal solutions or the nearest solution to the optimal.
- There exist a ton of metaheuristics techniques from simple local search procedures to complex learning process.
- Metaheuristics algorithms are approximate and typically non-deterministic.
- They usually incorporate mechanisms to avoid getting stuck in confined areas of the search space.
- The basic concepts of metaheuristics permit an abstract level description.
- Metaheuristics are not problem-specific.
- Metaheuristics may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy.

Formally, a metaheuristic is defined as a tuple of elements that, depending on how they are defined, leads to a particular technique or another. This formal definition has been developed in [104] and subsequently extended in [105].

Definition 2 (Metaheuristic)**.** *A metaheuristic $\mathcal{M}$ is a tuple composed by the following eight components:*

$$\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle \ , \tag{2.5}$$

where:

- *$\mathcal{T}$ is the set of elements that are handled by the metaheuristic. This set contains the search space and in most cases it coincides with it.*
- *$\Xi = \{(\xi_1, D_1), (\xi_2, D_2), \ldots, (\xi_v, D_v)\}$ is a set of v pairs. Each pair is composed by a state variable of the metaheuristic and by the domain of this variable.*
- *μ is the number of solutions with which $\mathcal{M}$ operates in one step.*
- *λ is the number of new solutions that are generated in each iteration of $\mathcal{M}$.*
- *$\Phi : \mathcal{T}^\mu \times \prod_{i=1}^{v} D_i \times \mathcal{T}^\lambda \to [0, 1]$ represents the operator that creates new solutions from the existent ones. This function must fulfill for all $x \in \mathcal{T}^\mu$ and for all $t \in \prod_{i=1}^{v} D_i$,*

$$\sum_{y \in \mathcal{T}^\lambda} \Phi(x, t, y) = 1 \ . \tag{2.6}$$

- *$\sigma : \mathcal{T}^\mu \times \mathcal{T}^\lambda \times \prod_{i=1}^{v} D_i \times \mathcal{T}^\mu \to [0, 1]$ is a function that allows to select those solutions that will be handled in the following iteration of $\mathcal{M}$. This function must fulfill for all $x \in \mathcal{T}^\mu$, $z \in \mathcal{T}^\lambda$ and $t \in \prod_{i=1}^{v} D_i$,*

$$\sum_{y \in \mathcal{T}^\mu} \sigma(x, z, t, y) = 1 \ , \tag{2.7}$$

$$\begin{aligned} &\forall y \in \mathcal{T}^\mu, \sigma(x, z, t, y) = 0 \vee \\ &\quad \vee \sigma(x, z, t, y) > 0 \wedge \\ &(\forall i \in \{1, \ldots, \mu\} \bullet (\exists j \in \{1, \ldots, \mu\}, y_i = x_j) \vee (\exists j \in \{1, \ldots, \lambda\}, y_i = z_j)) \ . \end{aligned} \tag{2.8}$$

- $\mathcal{U} : \mathcal{T}^{\mu} \times \mathcal{T}^{\lambda} \times \prod_{i=1}^{v} D_i \times \prod_{i=1}^{v} D_i \to [0,1]$ represents the update procedure of the state variable of the metaheuristic. This function must fulfill for all $x \in \mathcal{T}^{\mu}$, $z \in \mathcal{T}^{\lambda}$ and $t \in \prod_{i=1}^{v} D_i$,

$$\sum_{u \in \prod_{i=1}^{v} D_i} \mathcal{U}(x, z, t, u) = 1 \ . \tag{2.9}$$

- $\tau : \mathcal{T}^{\mu} \times \prod_{i=1}^{v} D_i \to \{false, true\}$ is a function that decides the termination of the algorithm.

The above definition reflects the typical stochastic behavior of metaheuristic techniques. In particular, the Φ, σ, and $\mathcal{U}$ functions must be interpreted as conditional probabilities. For example, the value of $\Phi(x, t, y)$ is interpreted as the probability that the child vector $y \in \mathcal{T}^{\lambda}$ is generated since at the moment the set of individuals with which the metaheuristic works is $x \in \mathcal{T}^{\mu}$ and its internal state is defined by the state variables $t \in \prod_{i=1}^{v} D_i$. It can be seen that the constraints imposed on the functions Φ, σ y $\mathcal{U}$ allow to consider them as functions that return these conditional probabilities.

Definition 3 (Metaheuristic state)**.** *Let* $\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle$ *be a metaheuristic and* $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{\mu}\}$ the set of variables that will store the solution set with which the metaheuristic works. We will use the notation $first(\Xi)$ to refer to the state variable set of the metaheuristic, $\{\xi_1, \xi_2, \ldots, \xi_v\}$. A state s of the metaheuristic is a pair of functions $s = (s_1, s_2)$ with

$$s_1 : \Theta \to \mathcal{T}, \tag{2.10}$$

$$s_2 : first(\Xi) \to \bigcup_{i=1}^{v} D_i \ , \tag{2.11}$$

where s_2 satisfies

$$s_2(\xi_i) \in D_i \ \ \forall \xi_i \in first(\Xi) \ . \tag{2.12}$$

We will denote with $\mathcal{S}_{\mathcal{M}}$ the set of all states of a metaheuristic $\mathcal{M}$.

Finally, once defined the state of a metaheuristic, we can define its dynamic.

Definition 4 (Metaheuristic dynamic)**.** *Let* $\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle$ be a metaheuristic and $\Theta = \{\theta_1, \theta_2, \ldots, \theta_{\mu}\}$ the set of variables that will store the solution set with which the metaheuristic works. We will use the notation $\overline{\Theta}$ for the tuple $(\theta_1, \theta_2, \ldots, \theta_{\mu})$ and $\overline{\Xi}$ for the tuple $(\xi_1, \xi_2, \ldots, \xi_v)$. We will extend the state definition so that it can be applied to element tuples. Then, we define $\overline{s} = (\overline{s}_1, \overline{s}_2)$ where

$$\overline{s}_1 : \Theta^n \to \mathcal{T}^n \ , \tag{2.13}$$

$$\overline{s}_2 : first(\Xi)^n \to \left(\bigcup_{i=1}^{v} D_i \right)^n \ , \tag{2.14}$$

and besides that

$$\overline{s}_1(\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_n}) = (s_1(\theta_{i_1}), s_1(\theta_{i_2}), \ldots, s_1(\theta_{i_n})) \ , \tag{2.15}$$

$$\overline{s}_2(\xi_{j_1}, \xi_{j_2}, \ldots, \xi_{j_n}) = (s_2(\xi_{j_1}), s_2(\xi_{j_2}), \ldots, s_2(\xi_{j_n})) \ , \tag{2.16}$$

for $n \geq 2$. We will say that r is a successor state of s if $t \in \mathcal{T}^{\lambda}$ exists such that $\Phi(\overline{s}_1(\overline{\Theta}), \overline{s}_2(\overline{\Xi}), t) > 0$ and besides that

$$\sigma(\overline{s}_1(\overline{\Theta}), t, \overline{s}_2(\overline{\Xi}), \overline{r}_1(\overline{\Theta})) > 0 \;\; y \tag{2.17}$$

$$\mathcal{U}(\overline{s}_1(\overline{\Theta}), t, \overline{s}_2(\overline{\Xi}), \overline{r}_2(\overline{\Xi})) > 0 \;\; . \tag{2.18}$$

We will denote with $\mathcal{F}_{\mathcal{M}}$ the binary relation "being a successor of" defined in the states set of a metaheuristic $\mathcal{M}$. *That is,* $\mathcal{F}_{\mathcal{M}} \subseteq \mathcal{S}_{\mathcal{M}} \times \mathcal{S}_{\mathcal{M}}$, and $\mathcal{F}_{\mathcal{M}}(s, r)$ if r is a successor state of s.

Definition 5 (Metaheuristic execution)**.** *A metaheuristic $\mathcal{M}$ execution is a finite or infinite sequence of states, $s_0, s_1, \ldots$ in which $\mathcal{F}_{\mathcal{M}}(s_i, s_{i+1})$ for all $i \geq 0$ and besides that:*

- *if the sequence is infinite $\tau(s_i(\overline{\Theta}), s_i(\overline{\Xi})) = false$ is satisfied for all $i \geq 0$ and*
- *if the sequence is finite $\tau(s_k(\overline{\Theta}), s_k(\overline{\Xi})) = true$ if satisfied for the last state s_k and, besides, $\tau(s_i(\overline{\Theta}), s_i(\overline{\Xi})) = false$ for all $i \geq 0$ such that $i < k$.*

2.4.3 Multi-Objective Optimization in Big Data

In [106] authors point optimization in Big Data as one of most important opportunities and challenges in this field. Big Data optimization is an innovative research area where traditional problems and consequently traditional metaheuristics, have to take into account the newly amount of data created in their context. For instance, The Travelling Salesman Problem [107] (TSP) a few years ago would only be created using static data of a city and would not modify with real-time changes in the city (traffic jam, work on roads and so on). However, currently thanks to the *Open Data* websites, we can modify in real-time the data of the problem by streaming, and as result, obtain better and realistic routes.

It is therefore desirable that optimization algorithms for resolving Big Data optimization manage streaming data sources, and generally, because of the nature of the problem, more than one objective needs to be satisfied at the same time, consequently, the algorithms have to be able to resolve multi-objective problems. Therefore, they have to be *Dynamic Multi-Objective algorithms.* However, we cannot forget the fact that decision makers may give the algorithm their preferences in order to find a preferred solution, so usually we need *Interactive Dynamic Multi-Objective algorithms* [17].

2.4.3.1 Dynamic Multi-Objective Problems

Dynamic optimization problems (DOPs) include dynamic single-objective optimization problems (DSOPs) and dynamic multi-objective optimization problems (DMOPs).

In recent years, most researchers have expressed their interest in dynamic multi-objective problems due to most real-world optimization problems are dynamic [108, 109, 110, 111].

DOPs, which are a special class of dynamic problems that *are solved online by an optimization algorithm as time goes by*, furthermore, in Equation 2.19 is shown the formal description of DOP, whose features are dependent on the time [112, 113].

In other words, a DOP is a problem where the objective function or the restrictions change with time. The simplest method for solving these problems is ignoring dynamics, considering each change as the arrival of a new optimization problem and re-optimizing, but it is often impractical, as it is shown in [114].

Therefore, the aim of the metaheuristics for dealing with DOPs is no longer to locate a stationary optimal solution, but to track its movement through the solution and time spaces as closely as possible. Traditional algorithms in their dynamic versions have been adapted to cope with such dynamic scenarios enhancing their ability for tracking moving optima [115].

$$\begin{aligned} &\text{minimize} \quad \{f_1(x,t), f_2(x,t), ..., f_k(x,t)\} \\ &\text{subject to} \quad x \in S \subset \mathbb{R}^n \\ &\text{and} \quad t \in T \end{aligned} \tag{2.19}$$

with $k \geq 2$ conflicting objective functions $f_i : S \to \mathbb{R}$ and where x is a vector of continuous decision variables from the feasible set S. We can denote an objective vector by $f(x,t) = (f_1(x,t), f_2(x,t), ..., f_k(x,t))^T$ and, finally, t is the time.

2.4.3.2 Dynamic Multi-Objective Algorithms

The goal of the metaheuristics for dealing with DOPs is no longer to locate a stationary optimal solution, but to track their progression through the space and time as closely as possible.

The general assumption is that the problem after a change is somehow related to the problem before the change, and thus an optimization algorithm needs to learn from its previous search experience as much as possible to hopefully advance the search more effectively, especially in real-world problems.

According to Cruz *et al.* [115] *Evolutionary Techniques* and their variants have been the most widely used methods to solve these DOPs. Dynamic multi-objective methods have to take into account a series of points when are designed, for instance, how the algorithms maintain of diversity in the population or how react on changes in the problem or in the data or even themselves. Other important point is how the algorithms manage the memory, below are described these features in more details [116].

- ***Maintenance of diversity***. It is essential for the algorithms in order to be able to adapt to changes more easily keep certain level of diversity in the population, because if the algorithms converge quickly this changes can be gone unnoticed. Random Immigrants GA (RIGA) [117] is an example of this strategy. Every iteration RIGA replaces part of the population by randomly generated individuals. This introduces new individuals in every time step and avoids the convergence of the whole population to a narrow region of the search space. This strategy is used in jMetalSP how we will see in Chapter 3.

- ***React on changes***. There exist plenty of different ways of changing the algorithm or problem environment, such as: evaluations, the dynamic problem changes due to the time or streaming data can change the state of the problem, when this occurs the dynamic algorithm has to be aware and react. In general, this detection is performed reevaluating the population as well as replacing part of the population by randomly generated individuals [118]. Although there exists other techniques [119, 117] in this thesis we focus on replacing and evaluating the population. In any case, when a change in the environment occurs, the best fitness areas are probably others and the proper detection of such changes can allow to track the good solutions or to vary the intensification of the search giving more weight to other areas.

- ***Use of Memory***. When a dynamic algorithm has a good manage of memory is able to store good (partial) solutions that had found during the process and consequently, enhances its performance as the problem in a future could convergence to this area of the solutions space [120, 121].

- ***Multiple populations***. Dynamic metaheuristics have been using multiple population strategy in order to enhance the diversity in dynamic environments. As would be expected, evolutionary algorithms have employed this kind of strategy due to their very nature [122, 123]. Furthermore, PSO methods have used this strategy as well [124, 125].

In this thesis, we propose as well, a dynamic interactive multi-objective algorithm (*InDM2*) [17], which tackles simultaneously two important issues: dynamic optimization and preference incorporation, using an interactive evolutionary approach. This method not only deals with the challenges involved related with the incorporation of the DM's preference, but also the difficulties associated with handling a changing environment. In fact, a dynamic MOP could be seen as a sequence of static MOPs, where the migration from a MOP to a new one is triggered by the occurrence of an environmental change. The latter could modify the PF, the Pareto set, the fitness landscape, etc. The MOEA should react to these changes by launching some adaptation actions that can cope with changes once they are detected.

2.4.4 Interactive Multi-Objective Algorithm

As mentioned earlier, in multi-objective optimization, the main aim is to find the most preferred solution to the DM. From a point of view of the role of the decision maker in the solution process, and there exists, three type of methods to do: no-preference, a posteriori and a priori methods. When we describe interactive multi-objective algorithms, we are referring to a priori methods.

In iMOAs, the decision maker works together with an analyst or an interactive computer program. DM specifies her/his preference information a priori, consequently, the algorithms can be tuned to search and generate solutions in desirable regions, reflecting the preference information of the DM, thereby approximating a specific region of the Pareto optimal front. This approach brings multiple advantages related to time and efficiency as, interactive approach can alleviate the need for generating the entire Pareto optimal front and is useful when a DM is aware of his/her aspiration. In addition, the entire process of generating a part of the Pareto optimal front is able to be iterated with new preference information from the DM, until a satisfactory solution is found by the DM. That is to say, a solution pattern is formed and repeated several times, and after every iteration, information is given to the decision maker and he/she is asked to provide some other type of information. The information given to and asked from the decision maker must be readily understandable. Finally, he/she decides, which one of the obtained Pareto optimal solutions is the most desired. In Figure 2.9 is shown how an iEMO interacts with DM so as to resolve an optimization problem.

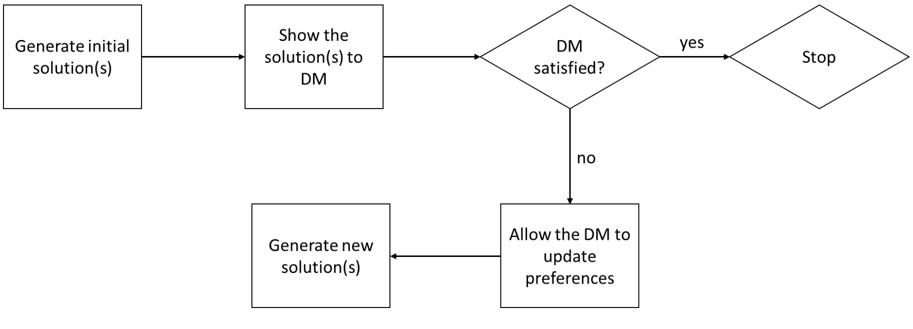

Generate initial solution(s)
Show the solution(s) to DM
DM satisfied?
yes
Stop
no
Allow the DM to update preferences
Generate new solution(s)

Figure 2.9: Phase diagram of an interactive method with a decision maker

Most of the interactive multi-objective algorithms are evolutionary multi-objective algorithms, which have been proposed in conjunction with MCDM based approaches, such as light beam search [126] or reference direction method [127]. More recently, several interactive EMO algorithms have been proposed [97, 128, 129, 130].

Interactive multi-objective optimization methods based on a reference point are very popular techniques [83, 38, 84] not only in current research, but also in industry as they allow decision makers (DMs) to specify information about their preferences in an intuitive manner to guide the operation of the optimization algorithms. As a consequence, the DM is able to learn progressively (at each iteration) about the set of (approximated) solutions in the Pareto front of a complex problem, hence reducing one's cognitive load [38].

Using jMetalSP, the framework that has been proposed in this thesis, we have designed an interactive algorithm based on SMPSO [131]. This algorithmic proposal is called SMPSO with reference points (SMPSO/RP) [21]. This is an extension of SMPSO based on the idea of reference point archives. The algorithm has external archives with an associated reference point so that only solutions that are dominated by the reference point or that dominate it are added. SMPSO/RP manages several reference point archives, so it can help to focus the search on one or more regions of interest. Furthermore, the algorithm allows interactively changing the reference points during its execution. Additionally, the particles of the swarm can be evaluated in parallel.

2.4.4.1 Artificial Decision Maker

In order to understand strengths and weaknesses of optimization algorithms, it is important to have access to different types of test problems, well defined performance indicators and analysis tools.

An important drawback about interactive multi-objective methods is, it is not trivial to assess their performance as it is necessary a DM (usually a human being) so as to bring the preference information, that is to say, making tests and comparisons of interactive multi-objective optimization methods is hampered by the necessity of involving DMs in tests, due to the fact that employ a DM make the test more expensive and furthermore, DMs because they are human being, get tired, and then appears difficulties owing to inconsistency of human nature and variability in their performance.

As noted in [37, 84], only few interactive multi-objective optimization methods have been extensively tested, which means that information about the quality of most of the methods cannot be called reliable, besides the conclusions of the authors of the methods and results are intuitive, consequently this test are not valid for testing that kind of algorithms due to the bias. In order to overcome the deficiency of tests and comparisons of interactive methods, one can use artificial DMs understood as techniques of generating preference information, which will be able to assess any type of interactive method.

In [6] three main concepts of an artificial DM are defined, as well as their interaction with an interactive method, as follows:

- ***Steady part***. This part can be defined as the previous DM knowledge, consequently, this knowledge is static during all the solution process. This includes accumulated experience and the core preferences, which do not change in time.

- ***Current context***. It is the actual situation in the solution process in a particular time, therefore this knowledge changes over time. This includes: the knowledge about the problem accumulated by the DM during the solution process, level of tiredness which can affect concentration, and the probability of making mistakes.

- ***Preference information***. The way as the DM express his/her choice during the solution process is defined as the Preference information. Therefore, this information expressed by the DM during the solution process guides the method toward solutions that are more preferred by the DM.

The artificial DM is defined by three different parts: first, the steady part, which does not change in time; second, a mechanism of representing and updating the current context as the solution process continues; and finally, the mechanism of generating the preference information based on the steady part and the current context.

In this thesis, we focus on interactive algorithm based on reference points. As a result we have developed an artificial decision maker driven by Particle Swarm Optimization algorithm (ADM-PSO) [20]. The main purpose is to assess iEMOs in our Big Data optimization platform.

This method helps us to evaluate automatically any interactive multi-objective algorithm based on reference points as, mimics the behavior of a human DM who adjusts preferences based on obtained information about derived solutions, and demonstrates randomness in the behavior in responses to the uncertainty about the Pareto optimal set.

ADM-PSO uses an interactive multi-objective optimization algorithm (WASFGA, R-NSGA-II) so as to optimize a multi-objective optimization problem, the outcome of this optimization is a Pareto front, which is the input of a reference point method to calculate a new one, as it is shown in Figure 2.10.

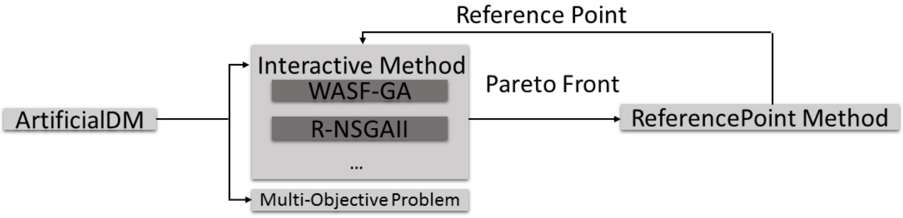


Figure 2.10: Artificial Decision Maker schema.

As said, the preference information is assumed to be given as a *reference point*. ADM-PSO is a new approach based on nature-inspired swarm algorithm (particle swarm optimization) to generate the reference point, which takes into account all the no dominated solutions generated so far (not only one solution as in [6]), in order to obtain a new reference point, and hence, improving with more knowledge the current context component.

With PSO we provide to the ADM with dynamic response, since the *particles* of the swarm represent a reference point in the objective space of the problem considered, that is to say, we take the features of bio-inferred dynamic particles to decide on new reference points. The core idea of this mechanism is to select the "global best" among PSO solutions as a new reference point. In other words, the PSO finds, in the objective space of the problem, the nearest reference point to the point consisting of the initial aspiration levels (asp). We sum up the features of this new approach as follows:

- **Higher convergence** towards initial aspiration level. This fact allows the ADM to achieve a final (and stable) reference point quicker than the other ADM.

- **Improve the current context** using all the solutions provide by the interactive method, not only the nearest one as in other ADM [6], consequently this allows the algorithm to explore thoroughly the Pareto front in the objective space.

- **Application of PSO's social behavior techniques** to mimic the DM's activity. This kind of methods have been used with good performance in other decision making process, for instance, in stock market domain [132] or [133].

2.5 Semantic Web

Semantic web was presented by Berners-Lee et al. in [134] and few years later came up a new revision of its definition [135]. There is a fundamental idea in web semantic, *knowledge representation*, that is key in this work. The core idea is to represent all the knowledge of the Big Data analytic with semantic web technology (through ontology) and, consequently, we will be able of using reasoners or SPARQL queries in order to fetch this information or deduce new knowledge from them.

Via Semantic web models we define the concepts related to Big Data analytic, below it is described this models and tools that has been used in this work.

- ***Ontology***. Following the definition on [136] and [137], an ontology provides a formal representation of the real world. It defines an explicit description of concepts in a domain of discourse (classes or concepts), properties of each concept describing various features and attributes of the concept (properties) and restrictions on properties, that is to say, ontologies define data models in terms of classes, subclasses, and properties. Ontologies are part of the W3C standard stack of the Semantic Web[1]. An ontology together with a set of individual instances of classes constitutes a knowledge base and offer services to facilitate interoperability across multiple, heterogeneous systems and databases.

- ***RDF***. Resource Description Framework [138] is a W3C recommendation that defines a language for describing resources on the web. Resources are described in terms of properties and property values using RDF statements. Statements are represented as triples, consisting of a subject, predicate and object. RDF Schema (RDFS) [139] describes vocabularies used in RDF descriptions.

- ***OWL***. The Ontology Web Language is used to define ontologies on the Web, which extends RDF and RDFS, but adding a vocabulary. From a formal description, OWL is equivalent to a very expressive description logic DL, where an ontology corresponds to a Tbox [140]. In this sense, OWL-DL is syntactic description that gives maximum expressiveness while retaining computational completeness and decidability [141]. In this work, we use OWL-DL syntax summarized in Table 2.1 to formalize the proposed ontology.

- ***SPARQL***. It is a query language for easy access to RDF stores. It is the query language recommended by W3C [142] to work with RDF graphs [143], then supporting queries and web data sources identified by URIs. Essentially, SPARQL is a graph-matching query language that can be used to extract knowledge from the model.

- ***SWRL***. The Semantic Web Rule Language provides the OWL-based ontologies with procedural knowledge, which compensates for some of the limitations of ontology inference, particularly in identifying semantic relationships between individuals [144]. SWRL uses the typical logic expression "$Antecedent \Rightarrow Consequent$" to represent semantic rules. Both antecedent (rule body) and consequent (rule head) can be conjunctions of one or more atoms written as "$atom_1 \wedge atom_2 \wedge \cdots \wedge atom_n$". Each atom is attached to one or more parameters represented by a question mark and a variable (e.g., $?x$). The most common uses of SWRL include transferring characteristics and inferring the existence of new individuals [145][2].

In this scenario, the concept of *Smart Data* emerges. It is defined as the result of the process of analysis performed to extract relevant information and knowledge from Big Data, including

[1] `https://www.w3.org/standards/semanticweb/`

[2] `https://www.w3.org/Submission/SWRL/`

Table 2.1: Basic OWL-DL semantic syntax used to formally define the proposed ontology

Descriptions	Abstract Syntax	DL Syntax
Operators	$intersection(C_1, C_2, \cdots, C_n)$	$C_1 \sqcap C_2 \sqcap \cdots \sqcap C_n$
	$union(C_1, C_2, \cdots, C_n)$	$C_1 \sqcup C_2 \sqcup \cdots \sqcap C_n$
Restrictions	for at least 1 value V from C	$\exists V.C$
	for all values V from C	$\forall V.C$
	R is Symmetric	$R \equiv R^-$
Class Axioms	$A\ partial(C_1, C_2, \cdots, C_n)$	$A \sqsubseteq C_1 \sqcap C_2 \sqcap \cdots \sqcap C_n$
	$A\ complete(C_1, C_2, \cdots, C_n)$	$A \equiv C_1 \sqcap C_2 \sqcap \cdots \sqcap C_n$

context information and using a standardized format. By context we mean all the relevant (meta) information to interpret the analysis results. This will lead to the enforceability of these results and thus facilitating their interpretation, the easy integration with other structured data, the integration of the Big Data analysis system with other systems, the interconnection (in a standardized way, at a lower cost and a higher accuracy and reliability) of third parties algorithms and services, etc.

In this thesis, we use semantic web as the technology that acts as the glue which binds each component of a workflow. Furthermore, we have defined an ontology called BIGOWL [18], where we describe the semantic model for designing Big Data workflow. In addition we have defined SWRL rules and SPARQL queries for accessing the information and identifying semantic relationships. These instances are stored in RDF triple format in a Stardog[3] repository, which is a commercial version of the Pellet OWL 2 reasoner [146], but enhanced with persistence capabilities, as well as the reasoning tasks.

[3] http://www.stardog.com/

Chapter 3

Methods and Materials

3.1 Intoduction

This Chapter is aimed at presents all the material carried out in this thesis as well as the methods that have been used for this goal. It is worth pointing out that in this thesis work we have presented not only new algorithms proposal, but also software tools. First, we present our Big Data optimization framewok (jMetalSP), its architecture of classes and design features. Second, it is presented the semantic model, BIGOWL ontology and RDF repository. Third, we describe the architecture of the execution platform used in this thesis, then it is described the experimental methodology that we have followed in this PhD Thesis and finally, a description of the software repositories where are hosted all the software developed.

3.2 jMetalSP

jMetalSP [14] is a framework for dynamic multi-objective Big Data optimization. It combines multi-objective methods with cluster computing execution capabilities to allow the solving of dynamic optimization problems from a number of external streaming data sources in Big Data contexts. In this section, we describe the design issues of jMetalSP, focusing mainly in its internal architecture, which has been changing over the time, with the aim of offering a comprehensive view of its main features. Among the covered features, we describe the main components that compose an application in jMetalSP, including dynamic problems, dynamic algorithms, streaming data sources and data consumers.

jMetalSP comes from the union of jMetal multi-objective optimization framework [3][147] and the Apache Spark cluster computing system [4]. jMetal is a widely used software in the field of multi-objective optimization and Spark is becoming one of the dominant technologies in Big Data, so using them in jMetalSP results in a framework that combines the features of the former (flexible and extensible architecture, lot of representative multi-objective metaheuristics and problems) and the latter (streaming processing, high level parallel model). Therefore, this approach allows to develop applications that can run on Hadoop [148], the *de facto* standard Big Data platform. jMetalSP is an open source that is hosted and maintained in GitHub.[1]

In jMetalSP, the simplification of the process of developing an application is essential. In particular, issues such as adding streaming data sources and connecting them to algorithms that

[1] jMetalSP project site: `https://github.com/jMetal/jMetalSP`

solve dynamic problems, must be done in a clean way. This is achieved by using an object-oriented architecture and an implementation based on solid software engineering principles.

3.2.1 Architecture of jMetalSP

jMetalSP is designed according to an object-oriented architecture, which is depicted in Figure 3.1. It is developed on top of jMetal, so all the components of this framework (algorithms, problems, encodings, operators, quality indicators, etc.) are available. jMetalSP is implemented in the Java programming language.

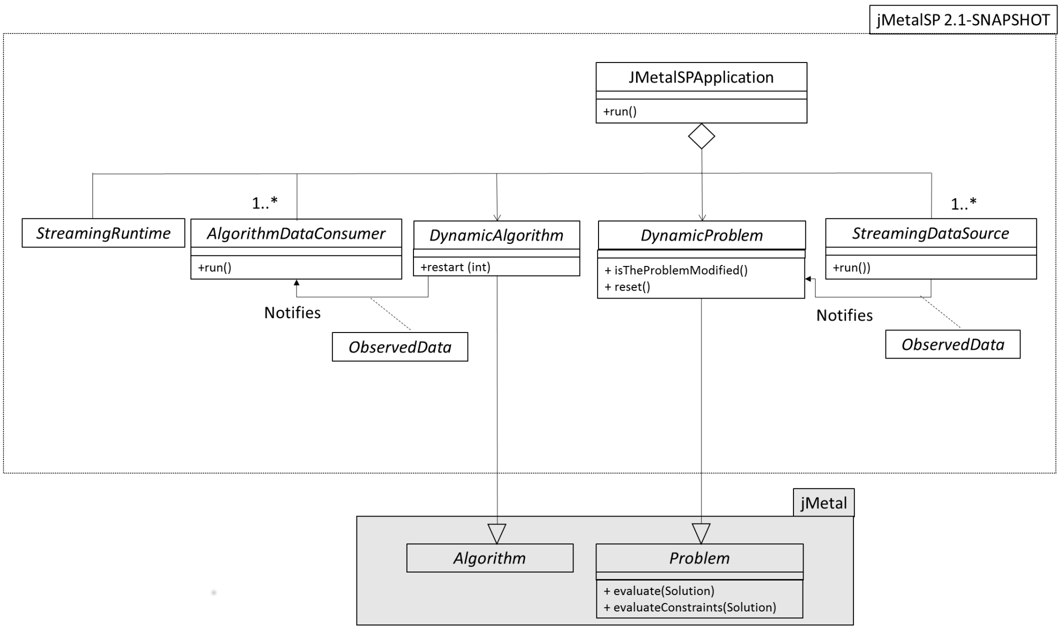


Figure 3.1: Overall class diagram of the jMetalSP architecture.

Code Snippet 3.1: *Observer* and *Observable* interfaces.

```
public interface Observer {
   void update(Observable<?> observable, Object data) ;
}

public interface Observable<Data> {
   void register(Observer observer) ;
   void unregister(Observer observer) ;

   void notifyObservers(Data data);
   int numberOfRegisteredObservers() ;
   void setChanged() ;
   boolean hasChanged() ;
   void clearChanged() ;
    String getName() ;
}
```

jMetalSP has been evolving over time and its current development version (2.1-SNAPSHOT) relies on the Observer pattern [149]. On the one hand, there are a number of *StreamingDataSource* elements (observables), each of them capable of receiving data continuously from external sources and analyze them, which can lead to updates in the *DynamicProblem* that is being optimized (observer). On the other hand, a *DynamicAlgorithm* (observable) is continuously optimizing the problem and generating results (e.g., Pareto front approximations) that, when are produced, are notified to a number of *AlgorithmDataConsumer* entities (observers). The interfaces that observers and observables have to implement are shown in Code Snippet 3.1.

The observable components produce instances of *ObservedData* subclasses, which are sent to the observers in the notification messages. They constitute key components because they determine which observers can be bound to an observable.

The *StreamingRuntime* class encapsulates the underlying streaming engine, which currently can be Spark or plain Java (based on threads).

Code Snippet 3.2: Template of a jMetalSP application.

```
public class JMetalSPApplicationTemplate {
   public static void main(String[] args) {
      JMetalSPApplication<
         ObservedDataFromStreamingSources,
         ObservedDataFromAlgorithm,
         DynamicProblem,
         DynamicAlgorithm,
         StreamingDataSource,
         AlgorithmDataConsumer> application ;

      application = new JMetalSPApplication<>();

    application
           .setStreamingRuntime(new SparkRuntime())
           .setProblem(new DynamicProblem())
           .setAlgorithm(new DynamicAlgorithm())
           .addStreamingDataSource(new StreamingDataSource1())
           .addStreamingDataSource(new StreamingDataSource2())
           .addAlgorithmDataConsumer(new DataConsumer1())
           .addAlgorithmDataConsumer(new DataConsumer2())
           .run();
   }
}
```

All the jMetaSP applications share the code template depicted in Code Snippet 3.2. We can observe how different data consumers and streaming data sources can be incorporated into an application. All the classes and interfaces on jMetalSP are generic, which means they are parametrized over types, then assuring that all the classes are compatible before the execution (during compilation time). The main architecture components are described as follows.

3.2.2 Dynamic Multi-Objective Problems

As explained in Chapter 2, Dynamic multi-objective optimization problems are characterized by the fact that their objectives or their search space can vary over time, which may affect their Pareto set, their Pareto front or both of them. In the context of jMetalSP, changes in the problems will be originated by the results of the processing and analysis of one or more streaming data sources.

As we can see in Figure 3.1, the *DynamicProblem* class inherits from jMetal's *Problem* class, so it contains two basic methods: *evaluate()* and *evaluateConstraints()*. Both methods receive a *Solution*; the first method evaluates it, and the second one determines the overall constraint violation degree.

Additionally, the *DynamicProblem* class has its own methods:

- *isTheProblemModified()*. Indicates whether the data problem has been modified or not.
- *reset()*. Resets the state of the problem to unmodified.

All these methods (including *evaluate()* and *evaluateConstraints()*) must be tagged as synchronized to ensure mutual exclusion between the clients of the problem, i.e., the algorithm and the streaming data sources.

As dynamic problems implement the *Observer* interface, they must define the *update()* method and they have to register themselves into the streaming data sources they want to observe.

3.2.3 Dynamic Multi-Objective Algorithms

A dynamic algorithm in jMetalSP is a conventional metaheuristic that should consider two main issues: first, the problem can change during the algorithm execution, so the state of the problem should be checked somehow and, in case of detecting a change, a re-starting procedure must be applied; second, when the stopping condition is reached, the algorithm, instead of just terminating, starts again. As can be seen in Figure 3.1, a dynamic algorithm has to implement a *restart()* method.

Code Snippet 3.3: *run()* method of class *AbstractEvolutionaryAlgorithm*.

```java
@Override public void run() {
   List<S> offspringPop;
   List<S> matingPop;
   population = createInitialPopulation();
   population = evaluatePopulation(population);
   initProgress();
   while (!isStoppingConditionReached()) {
        matingPop = selection(population);
        offspringPop = reproduction(matingPop);
        offspringPop = evaluatePop(offspringPop);
        population = replacement(population, offspringPop);
        updateProgress();
   }
}
```

jMetalSP is based on jMetal 5.5.2, which includes, among other features, algorithm templates. For example, there is an *AbstractEvolutionaryAlgorithm* class that contains the *run()* method shown in Code Snippet 3.3, which mimics closely the pseudo-code of a generic evolutionary algorithm (similar templates are available for particle swarm optimization and scatter search algorithms). An advantage of using this template is that those algorithms implementing it (most of evolutionary algorithms in jMetal use it) can be easily extended by overriding only some methods. This is particularly useful to develop dynamic versions of existing algorithms. In this case, at least the *isStoppingConditionReached()* method should be redefined, because instead of stopping, the algorithm should start again.

A dynamic problem is considered as an observable entity, so when a new Pareto front approximation is produced, it is notified to the registered *AlgorithmDataConsumer* observer objects. In our version of dynamic metaheuristics, the number of produced fronts are also provided to its observers (see Section 3.2.5).

3.2.4 Streaming Data Sources

The role of a streaming data source in jMetalSP is twofold: it must capture the new incoming data, which will be then analyzed. The results of this analysis may produce an instance of the *ObservedData* class to be notified to a registered observed (i.e, a dynamic problem). This is particularly interesting in the case of using Apache Spark, because its streaming features allow to make the analysis in parallel, taking advantage of Hadoop clusters.

The *StreamingDataSource* interface contains only a *run()* method. In the default plain Java implementation, a new thread is started and the *run()* method is invoked. An example is included in Code Snippet 3.4, which shows the code of a simple streaming data source that continuously produces the value of a counter. The observers are notified by the value of the counter after a delay (no analysis is carried out here).

Code Snippet 3.4: Example of a simple streaming counter data source (plain Java).

```
public class SimpleStreamingCounterDataSource {

   @Override
   public void run() {
      int counter = 0 ;
      while (true)
         Thread.sleep(DELAY);

         observedData.setChanged();
         observedData.notifyObservers(new SimpleObservedData(counter));
         counter ++ ;
      }
   }
}
```

In the case of using Spark, we assume that an external process is generating the counter values and writes them in files that are stored in a directory. A Spark class named *SimpleSparkStreamingCounterDataSource* that reads the files of that directory in a streaming fashion is included in Code Snippet 3.5. We can observe that the code is composed of two steps: assuming that each file contains a line with the generated value, the first sentence of the *run()* method reads all the lines in the files in the directory and transforms them into integer values; then, in the second step, the observers are notified. Compared with the former example, we can see here that there is not an implicit loop because the Spark streaming engine is executing theses sentences iteratively. The same engine takes care of reading only the new files stored in the directory since the last iteration. This two-step scheme is the same for all the streaming data sources supported by Spark (socket, directory, Kafka, etc.).

Code Snippet 3.5: Example of a simple streaming counter data source (Spark).

```
public class SimpleSparkStreamingCounterDataSource {

   @Override
   public void run() {
      JavaDStream<Integer> values = streamingContext
         .textFileStream(directoryName)
         .map(line -> Integer.parseInt(line)) ;

      values.foreachRDD(numbers -> {
         List<Integer> numberList = numbers.collect() ;
         for (Integer value : numberList) {
            updateData.setChanged();
            updateData.notifyObservers(new SimpleObservedData(value));
         }
      }) ;
   }
}
```

The issue to note here is that the processing of the *map()* function can be executed in parallel in a cluster, which would be advantageous if there are many lines to process and their analysis are complex procedures. In the case of complex data, their analysis would be carried out with Spark operations as this *map()* and many others, including filtering, sampling, etc.

3.2.5 Observed Data

This class is intended to represent the type of data the observable entities produce, so they determine which observers can register in a given observable.

The data produced by the streaming data sources can be very varied, while in the case of the algorithms, we provide a concrete class called *AlgorithmObservedData*, which is used to bound algorithms and data consumers. This class contains the Pareto front approximation obtained and the value of a generated fronts counter, but it can be easily extended to include more algorithm's related data (i.e., the computing time of the last execution, the value of a quality indicator, etc.).

3.2.6 Algorithm Data Consumers

A dynamic algorithm is supposed to run forever to produce at least Pareto front approximations periodically, so any component interested in getting those fronts cannot wait for the completion of the algorithm, as in the case of techniques solving static problems.

Algorithm data consumers register into algorithms to be notified from the latter when new information (i.e., an *AlgorithmObservedData*, as commented in the previous section) is generated. jMetalSP includes two consumer components: one that stores the fronts into a directory and another one that prints information about the fronts (number of generated fronts, number of solutions of the last front).

3.2.7 Streaming Runtime

The last component in the jMetalSP architecture is *StreamingRuntime*, which represents the underlying streaming system. Two classes implementing this interface are included:

- A default plain-Java based runtime (Spark is not required), which starts each streaming data source in a dedicated thread.

- An Spark-based runtime, which sets the parameters of Spark and initialize the so-called streaming context. The streaming model of Spark is based on micro-batches, so the runtime receives the batch interval (see the Spark streaming programming guide for further information[2]) as a parameter.

[2] http://spark.apache.org/docs/latest/streaming-programming-guide.html

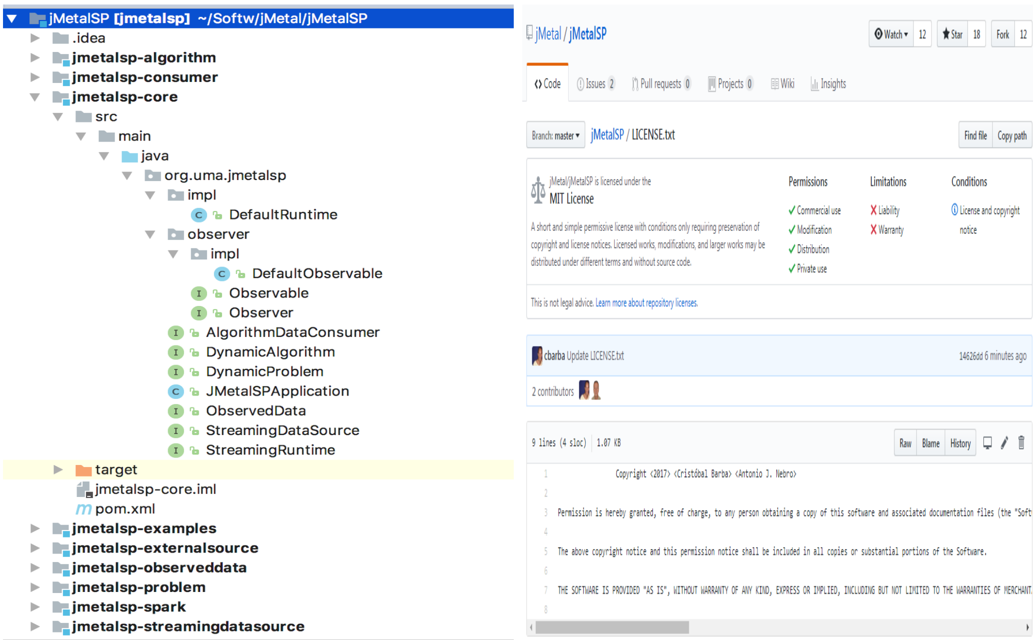
In former versions of jMetalSP only a Spark-based runtime were available, although we have now uncoupled it and generalized the streaming runtime into an interface (i.e., applying the dependence inversion principle). There are three main reasons to adopt this approach. First, Spark has other streaming implementation, called structured streaming, so we would like to easily change from the current one to the new one when available; second, some users could be interested in using jMetalSP for dynamic optimization but without Spark, so an only-Java version would be easier to them; finally, there are other streaming engines, such as Apache Flink[3], that could be incorporated to jMetalSP in a future release, although keeping the same architectural design.

3.2.8 Current Status and Implementation Details

jMetalSP is an active project that is in continuous development and involves external developers apart from authors. jMetalSP has MIT license, consequently anyone can contribute. In its current status it is fully usable, although it is sure that new features will be added in the near features and some changes in the architecture will be foreseeable from the experiences we gain when using it and from the feedback of interested users.

The version described in this thesis (2-1.SNAPSHOT) has been developed with the following tools: Java JDK 1.8.0_101, Spark 2.3.1, Maven 3.3.9, and jMetal 5.5.2.

The project is structured in nine Maven sub-modules as shown in Figure 3.2. This figure shows as well, the MIT license (permissions, limitations, and conditions) and some statistics about Github, like: contributors, issues, etc . The *jmetal-core* module contains the interfaces and classes of the architecture, the *Observer* and *Observable* interfaces, and two classes providing default implementations of the runtime and observable (plain Java) interfaces. More information about the jMetalSP internals can be found in the project GitHub page in https://github.com/jMetal/jMetalSP.



```
▼ jMetalSP [jmetalsp] ~/Softw/jMetal/jMetalSP
  ▶ .idea
  ▶ jmetalsp-algorithm
  ▶ jmetalsp-consumer
  ▼ jmetalsp-core
    ▼ src
      ▼ main
        ▼ java
          ▼ org.uma.jmetalsp
            ▼ impl
                C DefaultRuntime
            ▼ observer
              ▼ impl
                  C DefaultObservable
                I Observable
                I Observer
              I AlgorithmDataConsumer
              I DynamicAlgorithm
              I DynamicProblem
              C JMetalSPApplication
              I ObservedData
              I StreamingDataSource
              I StreamingRuntime
    ▶ target
      jmetalsp-core.iml
      pom.xml
  ▶ jmetalsp-examples
  ▶ jmetalsp-externalsource
  ▶ jmetalsp-observeddata
  ▶ jmetalsp-problem
  ▶ jmetalsp-spark
  ▶ jmetalsp-streamingdatasource
```

jMetal/jMetalSP is licensed under the
MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	ⓘ License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		

This is not legal advice. Learn more about repository licenses.

cbarba Update LICENSE.txt 14626dd 6 minutes ago

2 contributors

9 lines (4 sloc) | 1.07 KB

```
            Copyright <2017> <Cristóbal Barba> <Antonio J. Nebro>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Soft

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANT
```

Figure 3.2: Structure of the jMetalSP project. It shows its license.

[3] Apache Flink Web Site: https://flink.apache.org

3.3 Semantic Model Framework

Once we can use a framework for Big Data optimization. A key issue followed in this thesis is to capture all data. With the aim of reaching this purpose, we have designed a semantic model framework, which is composed of an ontology BIGOWL, reasoning rules SWRL, SPARQL queries, mapping in RDF triples and all of these are stored in a Stardog, [4] which is a knowledge graph platform.

3.3.1 BIGOWL

BIGOWL is an ontology to support knowledge management in Big Data analytics therefore, it is designed to cover a wide vocabulary of terms concerning Big Data analytics workflows, including their components and how they are connected, from data sources to the analytics visualization.

It is worth mentioning that it also takes into consideration aspects, such as: parameters, restrictions and formats, in order to set all their features. This ontology defines not only the taxonomic relationships between the different concepts, but also instances representing specific individuals to guide the users in the design of Big Data analytics workflows, consequently BIGOWL takes into consideration the context of all the elements that are defined in the workflow, from how to gather the data to how to visualize the results, that is to say, describes the data value chain of workflows and their components.

On the one hand, the use of semantics as contextual information will enhance the analytical power of the algorithms and ensure the reuse of single components in Big Data analytic workflows. On the other hand, the absence of context (semantic) in the data, as well as in algorithm design, would produce a series of drawbacks like: analytic algorithms that cannot properly capture the specific domain knowledge when tackling a Big Data task; analytic algorithms that are no reflective and that cannot auto-adapt; lack of reusability, i.e., applications requiring almost complete ad-hoc development; impedance mismatch, algorithms that cannot properly inject obtained knowledge to other components of the applications (visualizations, data sink, etc.); and finally managed data are difficult to integrate with other third party data (other datasets or Open Data).

BIGOWL has been defined to test our scientific hypothesis. In addition, the semantic annotation can provide the background for reasoning methods based on axiomatic and rules recommendations logic. Besides, these rule provide us the possibility of assessing the workflow through reasoning; or even better, give us new relations between components, which are inferred with the rules and consequently are compatible logically. This is a powerful feature, because the more workflow are defined more knowledge will have the reasoner for inferring new relations without the need for adding new rules, so the ontology becomes self-sufficient over time.

A research paper describing and testing BIGOWL is available in Chapter 4. Figure 3.3 gives a briefly summary of its main classes, generated with *Protégé* [5]. BIGOWL defines classes for *workflow*, *tasks*, *components*, *algorithms* and *problems*. The aforementioned classes can cover different workflows, such as optimization or data mining, because it has a thorough hierarchy of these type of algorithms.

As illustrated Figure 3.4, a workflow is composed by one or more tasks and these are indeed composed by one component. Furthermore, components may have zero or more inputs or/and outputs data, parameters and according to its type, could have an algorithm (optimization or data mining) or a problem or any other operation, such as: managing database, writing files, or a script.

[4] Stardog Web Site: https://www.stardog.com

[5] https://protege.stanford.edu/

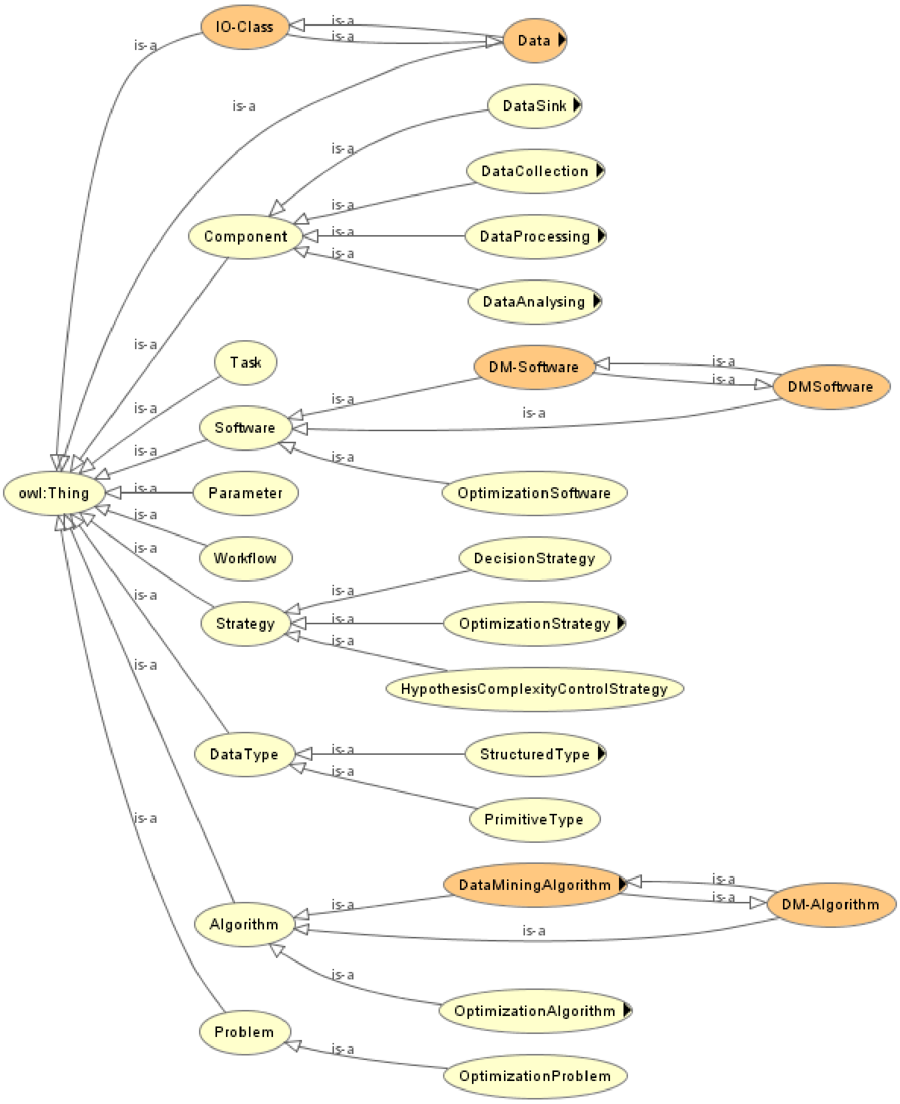


Figure 3.3: Main classes define in BIGOWL .

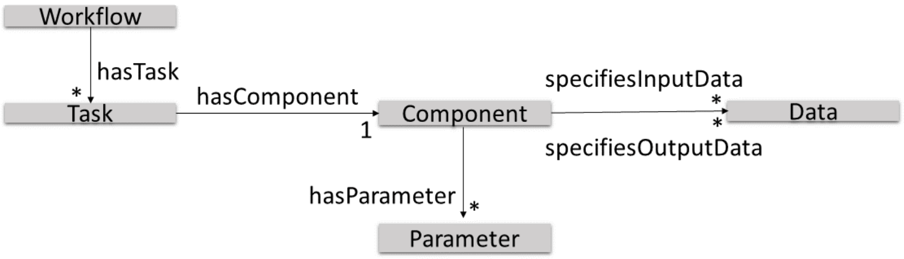


Figure 3.4: Core concepts in BIGOWL.

3.4 Architecture of the Execution Platform

For the purpose of carrying out all the tests in this thesis, we have designed a virtualization platform with 10 virtual machines and a physical one that acts as Master node.

We have conducted all the experiments in the aforementioned virtualization environment running on a private high-performance cluster computing platform. This installation comprises a number of IBM hosting racks for storage, units of virtualization, server compounds and backup services.

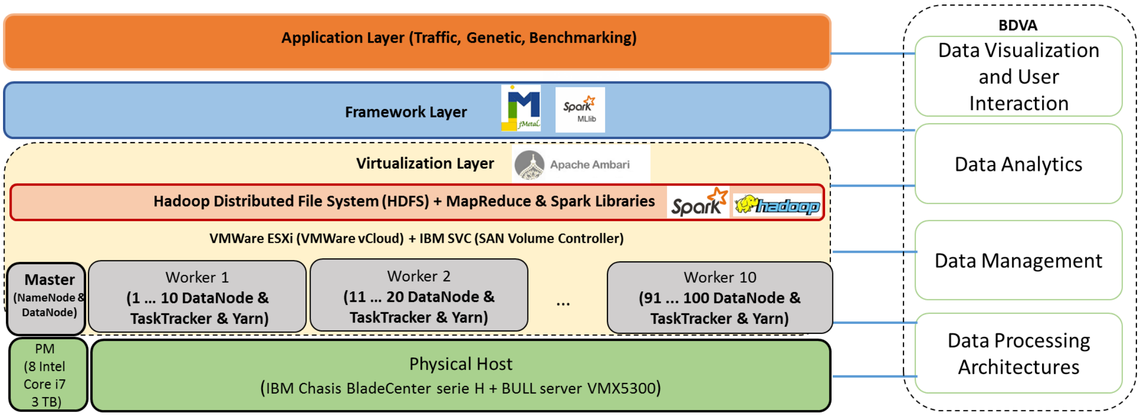

Application Layer (Traffic, Genetic, Benchmarking)
Framework Layer
jMetal
Spark MLlib
Virtualization Layer
Apache Ambari
Hadoop Distributed File System (HDFS) + MapReduce & Spark Libraries
Spark
hadoop
VMWare ESXi (VMWare vCloud) + IBM SVC (SAN Volume Controller)
Master (NameNode & DataNode)
Worker 1 (1 ... 10 DataNode & TaskTracker & Yarn)
Worker 2 (11 ... 20 DataNode & TaskTracker & Yarn)
...
Worker 10 (91 ... 100 DataNode & TaskTracker & Yarn)
PM (8 Intel Core i7 3 TB)
Physical Host (IBM Chasis BladeCenter serie H + BULL server VMX5300)
BDVA
Data Visualization and User Interaction
Data Analytics
Data Management
Data Processing Architectures

Figure 3.5: Computational environment for Big Data optimization used in this thesis aligned with BDVA priorities.

The physical platform comprises an IBM Chassis BladeCenter H type and BULL server with a high-performance VMX5300, unified CPU, memory and storage. It also contains a NovaScale Blade BL265 only for storage. The virtualization layer supports computing resources through VMWare ESXi (and VMWare vCloud), storage through IBM SVC (SAN Volume Controller), backup through Veem Backup and IBM Protectier (virtualization of tapes) and desktops are managed with VMWare Horizon and Virtual Cable UDS. The general characteristics of this virtualization installation are: a processor with 256 cores, 2.75 TB RAM, 84.41 TB storage space, 10 Gbps LAN network, 1 Gbps internet network, iSCSI 10 Gbps and FC 8 Gbps storage network.

Our platform is hosted in this computing environment, whose main components are illustrated in Figure 3.5. In summary, this platform is made up of 10 virtual machines (Worker 1 to Worker 10), each one with 10 cores, 10 GB RAM and 250 GB virtual storage (summing up 100 cores, 100 GBs of memory and 2.5 TB HD storage). These virtual machines are used as Worker nodes with the role of TaskTracker (Spark), DataNode (HDFS) and Yarn client (Hadoop) to perform results in parallel. The Master node is hosted in a different machine (Master) with 8 Intel Core i7 processors at 3.40 GHz, 32 GB RAM and 3 TB storage space. All these nodes are configured with a Linux CentOS 6.6 64-bit distribution.

Currently, the whole cluster is managed with Apache Ambari 2.6.2 and executes the Apache Hadoop version 2.7.0. This Hadoop distribution integrates Hadoop Distributed File System (HDFS), MapReduce framework libraries, and Apache Spark v2.3.1.

Since this thesis started until now we have been updating the software so, we have been using different versions of Spark, Hadoop and Ambari.

It is worth saying that jMetalSP framework is deployed on this infrastructure, providing optimization algorithms with Spark methods to manage the data sources in parallel, as well as to induce updates in the dynamic problems.

In case of data mining, we can use this platform as well, because MLlib[6] library runs on Spark. The application runner are always executed in the master machine and the Job Scheduling of Spark splits up and distributed the task to all the workers. This platform has been designed following the BDVA recommendations. Its main features are described in Chapter 2.

3.5 Experimental Methodology

Multi-objective optimization algorithms are non-deterministic techniques, hence different executions of the same algorithm over the same problem instances, but with different random seed, are likely to bring different results. This can lean some inconvenient when the algorithm are assessed or even when comparing different algorithms.

Although there are works that cope with the theoretical analysis of many heuristic methods and problems [150, 151], some of them are often highly complex. Consequently, they only can be compared using empirical data from multiple experimental runs.

The comparison between two multi-objective optimization algorithms are in terms of qualities of their results, that is to say, we compare their outcomes, either directly or by means of measures that summarize multiple executions of the optimizers (a value of 31 executions is a commonly adopted and accepted amount).

In this thesis, we focus on multi-objective problems. The outcome of that class of algorithm is a Pareto set. On the one hand, Pareto sets can be partially ordered according to Pareto-optimality. Consequently, some Pareto sets can be said to be better than others. On the other hand, Pareto sets are often incomparable in terms of Pareto-optimality. Hence, the analysis of multi-objective algorithms often requires more advanced techniques.

To sum up, due to the tests we carried out, we used different quality indicator since, in some cases we wanted to evaluate the convergence or the uniform diversity of a metaheuristics or in other cases, we focused on measuring the performance of parallel algorithms, in terms of computing effort, their scalability.

3.5.1 Convergence and Uniform Diversity

In multi-objective optimization, there exists different quality indicators for measuring these criteria. Convergence and diversity, in the literature has been proposed the followings: Generational Distance (I_{GD}) [152], Inverse Generational Distance (I_{IGD}), Hypervolume (I_{HV})[153], Epsilon($I^1_\epsilon+$) [154], Spread or $\triangle$ ($I_\triangle$) [88], and others. However, in this thesis we focus in I_{GD} and I_{HV}. Some of them are intended to measure only the convergence or diversity, and others take into account both criteria. Figure 3.6 depicts a classification of some indicators based on which aspect they measure.

- I_{GD}. The goal of this indicator is to measure how far the elements in the computed approximation from those in the optimal Pareto front [152] an it is defined as.

$$I_{GD} = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{3.1}$$

  where n is the number of solutions in the approximation and d_i is defined as, the Euclidean distance (measured in objective space) between each of these solutions and the nearest member in the optimal Pareto front. Fronts with lower values of I_{GD} are better. A value of $I_{GD} = 0$ indicates that all the generated elements are in the Pareto front.

[6] MLLib Web Site:https://spark.apache.org/mllib/

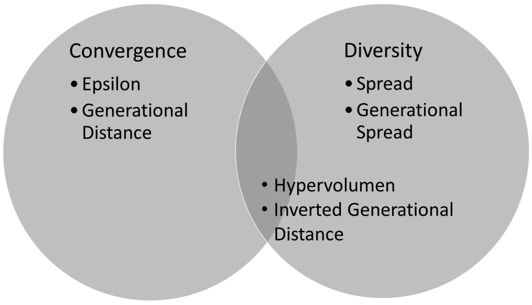


Figure 3.6: Classification of quality indicators.

- I_{HV}. This indicator calculates the volume, in the objective space, covered by members of a non-dominated set of solutions Q, for instance, the region enclosed into the discontinuous in Figure 3.7, $Q = \{A, B, C\}$. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube.

  Assuming a minimization problem involving d objectives the reference point $W \in \mathbb{R}^d$ can simply be found by constructing a vector of the same dimension as the number of objective functions, where each component is the worst value found for that objective. Thereafter, a union of all hypercubes is found and its I_{HV} is calculated:

$$I_{HV} = volume\left(\bigcup_{i=1}^{|Q|} v_i\right) \tag{3.2}$$

  Fronts with large values of I_{HV} are desirable. Sometimes, the I_{HV} takes a value equal to 0, meaning that the fronts obtained by the algorithms are outside the limits of the Pareto front.

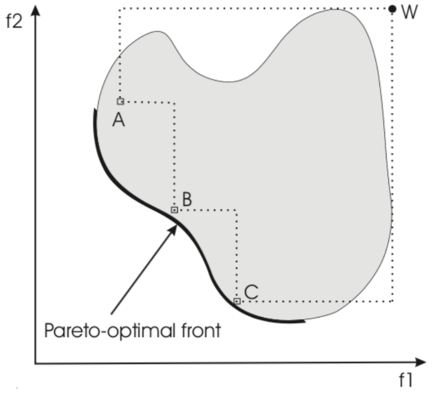


Figure 3.7: The hypervolume enclosed by the non-dominated solutions.

3.5.2 Speedup and Efficiency

One of the most widely used indicators for measuring the performance of a parallel algorithm is the *Speedup* (S_N). The standard formula of the speedup is represented in Equation 3.3 and calculates the ratio of T_1 over T_N, where T_1 is the running time of the analyzed algorithm in 1 processor and T_N is the running time of the parallelized algorithm on N processors (cores) [7].

$$S_N = \frac{T_1}{T_N} \tag{3.3}$$

$$E_N = \frac{S_N}{N} \times 100 \tag{3.4}$$

A related measure is the *Efficiency* of a parallel algorithm, which is calculated with the formula of Equation 3.4. An algorithm scales linearly (ideal) when it reaches a speedup $S_N = N$ and hence, the parallel efficiency is $E_N = 100\%$. In the execution of an algorithm with linear speedup, doubling the number of processors means doubling the speed.

3.5.3 Computational Effort

To measure the parallel computing performance of jMetalSP framework and the Big Data platform 3.5, we have executed the algorithms, with the same setup but, using different numbers of cores and workers in order to assess the scalability of the proposals. With this methodology, we can compare running time of the sequential algorithm (usually we estimate this value) with the parallel versions. In this thesis, we have configured the cluster as it is shown in Table 3.1. We always start the tests with 100 cores since we can assess first of all whether the proposed algorithm has good performance or not. Next steps are conducted by decrementing the number of cores, following two strategies: the first one decreases the number of cores per worker and no change in the number of workers are performed; the second one, decreases the number of worker and keep the number of cores per worker. The first strategy can show the overhead of communications, we still have 10 workers consequently this increase in the runtime has not change. However, in the second setting, we reduce the number of workers and consequently the overhead time of communications, but the workers have less cores so their running times are higher.

Table 3.1: Different configurations of cluster setting to explore or test

N^o Total Cores	N^o Cores per Worker	N^o Workers
100	10	10
80	10	8
	8	10
50	10	5
	5	10
20	10	2
	2	10
10	10	1
	1	10
1	1	1

[7] We use either the term processor or core to refer the same processing unit.

3.6 Statistical Validation Procedure

Metaheuristics are stochastic based algorithms with different random components in their operations. Opposite to deterministic procedures, for which, just a single execution is required, when working with metaheuristics, performing a series of independent runs for each algorithm's configuration is a mandatory task in order to obtain a distribution of results. In this case, it is possible to compute a global indicator (median, mean, standard deviation, etc.) from the resulted distribution to measure the performance of the studied algorithm. Nevertheless, using one single global indicator to directly compare metaheuristics should lead empirical analyses to biased conclusions. Therefore, the correct practice is to compare the distributions of results by means of statistical tests, which are indispensable tools to validate and to provide confidence to our empirical analysis.

The standard procedure, recommended by the scientific community [155, 156], for the statistical comparison of metaheuristics lies in the use of parametric or non-parametric tests.

To analyze the results we have used non-parametric [155] tests. These tests use the mean ranking of each algorithm. We have applied them since several times the functions might not follow the conditions of normality and homoskedasticity to apply parametric tests with security [156]. In particular, we have considered the application of the *Friedman's or Iman Davenport's tests* in order to check whether statistical differences exist or not.

3.7 Software Repository

During thesis process, we have been developing a number of software release deliverables. In order to manage the source code, we have been using two software repositories, Github[8] for public projects and Bitbucket for private projects[9]. When we finish one private project, we put it on Github, hence making the code available for scientific community. In addition, they are able to make contributions as happen in jMetalSP. Table 3.2 shows a series of links where are hosted the software created in this thesis.

Table 3.2: Official software repositories generated in the scope of this thesis with MIT License.

Deliverable	Software repository	Type repository
jMetalSP [14]	https://github.com/jMetal/jMetalSP	Github
	https://bitbucket.org/ajnebro/jmetalsp/	Bitbucket
BIGOWL [18]	https://github.com/KhaosResearch/BIGOWL	Github
	https://bitbucket.org/khaosresearchgroup/bigowl	Bitbucket
ADM-PSO [20]	https://github.com/KhaosResearch/admpso	Github
	https://bitbucket.org/ajnebro/jmetalsp	Bitbucket
jMetal [3]	https://github.com/jMetal/jMetal	Github
	https://bitbucket.org/ajnebro/jmetal	Bitbucket
jMetalPy [23]	https://github.com/jMetal/jMetalPy	Github

Most papers in this thesis have been developed in jMetalSP, so all the dynamic algorithms and problems are included in those repositories, namely, InDM2 [17], FDA problems [36] and SMPSO/RP [21].

[8] https://www.github.com

[9] https://www.bitbucket.org

Chapter 4

Published Work

We have published a series of research studies based on Big Data analytic, particularly focusing on Big Data optimization. Specifically, 3 articles have been published in journals indexed in the Journal of Citation Report (JCR) from the Institute of Scientific Information and other one paper in Emerging Source Citation index in Web of Science, Thomson Reuters. In addition, 11 articles have been published in conferences. 7 of them have been published in international conferences and 3 of them are indexed in GGS conferences rating as Class 2 (CORE A). The remain 3 are published in national conferences.

4.1 List with Research Contributions

We have three JCR articles, two of them are related to dynamic multi-objective algorithms, and the other one is about semantic in Big Data analysis. **Articles published in journal indexed in JCR:**

1. C. Barba-González, J. García-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes. “jMetalSP: a framework for dynamic multi-objective Big Data optimization”. *Applied Soft Computing* 69 (2017), pp. 737–748
   Impact Factor: 3,541. Q1 (21/133) in the category of Computer Science and Artificial Intelligence.

2. A. J. Nebro, A. B. Ruiz, C. Barba-González, J. García-Nieto, M. Luque, and J. F. Aldana-Montes. “InDM2: Interactive Dynamic Multi-Objective Decision Making Using Evolutionary Algorithms”. *Swarm and Evolutionary Computation* 40 (2018), pp. 184–195
   Impact Factor: 3,893. Q1 (19/133) in the category of Computer Science and Artificial Intelligence.

3. C. Barba-González, J. García-Nieto, M. d. M. Rodan-García, I. Navas-Delagado, and J. F. Aldana-Montes. “BIGOWL: Knowledge Centered Big Data Analytics”. *Expert Systems with Applications* 115, (2018), pp. 543–556
   Impact Factor: 3,928. Q1 (19/133) in the category of Computer Science and Artificial Intelligence.

Articles published in journal indexed in Emerging Source Citation index:

1. C. Barba-González, J. García-Nieto, I. N. Delgado, and J. F. A. Montes. “A fine grain sentiment analysis with semantics in Tweets”. *IJIMAI* 3.6 (2016), pp. 22–28.

Articles published in international conferences:

1. J. A. Cordero, A. J. Nebro, C. Barba-González, J. J. Durillo, J. García-Nieto, I. Navas-Delgado, and J. F. Aldana-Montes. "Dynamic multi-objective optimization with jMetal and Spark: a case study". *LNCS of International Workshop on Machine Learning, Optimization and Big Data (MOD 2016).* Springer. 2016, pp. 106–117.

2. C. Barba-Gonzaléz, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. "Multi-objective big data optimization with jMetal and spark". *LNCS of International Conference on Evolutionary Multi-Criterion Optimization (EMO'17), GGS class 2 (CORE A).* Springer. 2017, pp. 16–30.

3. A. J. Nebro, C. Barba-González, J. García-Nieto, J. A. Cordero, and J. F. A. Montes. "Design and architecture of the jMetalSP framework". *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO'17).* ACM. 2017, pp. 1239–1246.

4. C. Barba-González, V. Ojalehto, J. García-Nieto, A. J. Nebro, K. Miettinen, and J. F. Aldana-Montes. "Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods". *Proceeding of 15th International conference on parallel problem solving from nature (PPSN'18), GGS A class 2(CORE A).* Springer, 2018, pp. 274–285.

5. A. J. Nebro, J. J. Durillo, J. García-Nieto, C. Barba-González, J. Del Ser, C. A. Coello Coello, A. Benítez-Hidalgo, and J. F. Aldana-Montes. "Extending the Speed-constrained Multi-Objective PSO (SMPSO) With Reference Point Based Preference Articulation". *Proceeding of 15th International conference on parallel problem solving from nature (PPSN'18), GGS A-, CORE A.* Springer, 2018, pp. 298–310.

6. C. Barba-González, J. García-Nieto, A. J. Nebro, A. Benítez-Hidalgo, and J. F. Aldana-Montes. "Scalable Inference of Gene Regulatory Networks with the Spark Distributed Computing Platform". *Springer Series of 12th International Symposium on Intelligent Distributed Computing (IDC'18).* 2018.

7. A. Benítez-Hidalgo, A. J. Nebro, J. J. Durilllo, J. García-Nieto, E. Camacho-López, C. Barba-González, B, and J. F. Aldana-Montes. "About Designing an Observer Pattern-Based Architecture for a Multi-ObjectiveMetaheuristic Optimization Framework". *Springer Series of 12th International Symposium on Intelligent Distributed Computing (IDC'18).* 2018.

Articles published in national conferences:

- C. Barba-González, A. J. Nebro, J. García-Nieto, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes. "Un Framework para Big Data Optimization Basado en jMetal y Spark" (2016), pp. 159–168.

- S. Huratdo-Requena, C. Barba-González, M. Rybiński, F. J. Barón-López, J. Wärnberg, I. Navas-Delgado, and J. F. Aldana-Montes. "Análisis de datos de acelerometría para la detección de tipos de actividades". *Jornadas de Ingeniería del Software y Bases de Datos (In press).* 2018.

- C. Barba-González, J. García-Nieto, A. B. Ruiz, A. J. Nebro, M. Luque, and J. F. Aldana-Montes. "Algoritmo Evolutivo Multi-Objetivo para la Toma de Decisiones Interactiva en Optimización Dinámica" (2018).

4.2 Summary of the Articles that Support This Thesis

This section summarizes the articles that support this thesis. All these papers are related to Big Data optimization. In the first article, we have performed a study of the influence of accessing data stored in the Hadoop File System (HDFS) in each evaluation step of a metaheuristic. In the second published article, we have described the framework for dealing with Big Data optimization multi-objective problems (jMetalSP) in a real-world problem of traffic scheduling with streaming Big Data. In the third article, we have presented the semantic model for Big Data Analytic. This semantic model lets describe the different components engaged in Big Data Analytic. In this work we have presented: ontology (BIGOWL), RDF repository, reasoning rules and SPARQL queries. In the fourth article, we have presented the mechanism for assessing interactive multi-objective algorithms. This mechanism acts as Automatic Decision Maker based on PSO metaheuristic and it is able to assess the interactive algorithm created in our Big Data optimization framework. Finally, in the fifth study, we have applied in parallel multi-objective metaheuristics that optimize two objectives, guiding the algorithm to search the best inference of Gene Regulatory Networks. Therefore we cover contributions in Big Data optimization on benchmarking and real-world problems.

4.2.1 Multi-objective Big Data Optimization with jMetal and Spark

Reference: [15] C. Barba-Gonzaléz, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. "Multi-objective big data optimization with jMetal and spark". *LNCS of International Conference on Evolutionary Multi-Criterion Optimization (EMO'17), GGS class 2 (CORE A).* Springer. 2017, pp. 16–30
DOI: https://doi.org/10.1007/978-3-319-54157-0_2

In [15] we introduced an early version of our framework and the platform. We evaluated two scenarios: first, the used of Spark as an engine to evaluate the solutions of a metaheuristic in parallel and, second the study of the influence of accessing a massive amount of data in each evaluation of a metaheuristic algorithm. Instead of focusing on a particular optimization problem, we have defined a generic scenario, in which a benchmark problem is modified to artificially increase its computing time and to read data from the Hadoop file system (HDFS). Henceforth, this study had two different steps. In the first one, we tuned the computing time in order to assess the performance of Spark when executing the evaluation step of a metaheuristic. In the second one, we measured the size of the read data so as to assess the capacity of the platform.

4.2.2 jMetalSP: A framework for Dynamic Multi-Objective Big Data Optimization

Reference: [14] C. Barba-González, J. García-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes. "jMetalSP: a framework for dynamic multi-objective Big Data optimization". *Applied Soft Computing* 69 (2017), pp. 737–748
DOI: https://doi.org/10.1016/j.asoc.2017.05.004

In [14] we introduced jMetalSP, the framework based on jMetal for optimizing Big Data problems. jMetalSP is intended to solve dynamic optimization problems with dynamic algorithms by analyzing multiple streaming data sources. jMetalSP has dynamic multi-objective algorithms and or interactive algorithms (some of those algorithms are both). As we presented above this framework combines the software of jMetal with Apache Spark with the goal of managing parallel computing, streaming data sources and huge amount of data. jMetalSP is able to allow the dynamic algorithms to detect changes on the problem and to react according to them (for example, by applying a re-start strategy). jMealSP was published online[1] first to be freely used by the scientific community.

4.2.3 BIGOWL: Knowledge Centered Big Data Analytics

Reference: [18] C. Barba-González, J. García-Nieto, M. d. M. Rodan-García, I. Navas-Delagado, and J. F. Aldana-Montes. "BIGOWL: Knowledge Centered Big Data Analytics". *Expert Systems with Applications* 115, (2018), pp. 543–556
DOI: https://doi.org/10.1016/j.eswa.2018.08.026

In this study we presented BIGOWL that was designed and implemented for the representation and consolidation of knowledge in Big Data analytics. It considered a large and complemented set of concepts, attributes and relationships that have been taken from Big Data ecosystem. The semantic model captured all the needed semantics to guide the smart design of Big Data analytics workflows and to enhance their performance. The semantic model was evaluated in the context of two realistic use cases: real-time routing calculation in urban traffic and classical classification with decision trees. With this two use cases we assessed the ontology with Big Data optimization problem and Data mining analysis.

4.2.4 Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods

Reference: [20] C. Barba-González, V. Ojalehto, J. García-Nieto, A. J. Nebro, K. Miettinen, and J. F. Aldana-Montes. "Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods". *Proceeding of 15th International conference on parallel problem solving from nature (PPSN'18), GGS A class 2(CORE A).* Springer, 2018, pp. 274–285
DOI: https://doi.org/10.1007/978-3-319-99253-2_22

This work has been recently presented in the 15th International Conference on Parallel Problem solving for Nature (PPSN 2018), celebrated in Coimbra, Portugal in September of 2018. It was derived from the idea of assessing interactive algorithms. ADM has been developed for evaluating reference point based interactive methods. It was able to calculate reference points based on information about solutions derived so far. The decision maker through reference points indicated desirable objective function values to interactively direct the solution process. Nevertheless, when analyzing the performance of these methods, a critical issue is how to systematically involve decision makers. In this study, we presented an artificial decision maker, which reuses the dynamics of particle swarm optimization for guiding the generation of consecutive reference points, hence, replacing the decision maker in preference articulation. We used the artificial decision maker to compare interactive methods. We demonstrated the artificial decision maker using the DTLZ benchmark problems with 3, 5 and 7 objectives to compare R-NSGA-II and WASF-GA as interactive methods. The experimental results showed that the proposed artificial decision maker was useful and efficient if we compared with others ADM. In addition, our version obtained similar results in less number of iterations. It offered an intuitive and flexible mechanism to capture the current context when testing interactive methods for decision making.

4.2.5 Scalable Inference of Gene Regulatory Networks with the Spark Distributed Computing Platform

Reference: [22] C. Barba-González, J. García-Nieto, A. J. Nebro, A. Benítez-Hidalgo, and J. F. Aldana-Montes. "Scalable Inference of Gene Regulatory Networks with the Spark Distributed

[1] https://github.com/jMetal/jMetalSP/

Computing Platform". *Springer Series of 12th International Symposium on Intelligent Distributed Computing (IDC'18).* 2018
DOI: https://doi.org/10.1007/978-3-319-99626-4_6

This work was presented in the 12th International Symposium on Intelligent Distributed Computing (IDC 2018), celebrated in Bilbao, Spain in October of 2018. In this work we dealt with the inference of a Gene Regulatory Networks (GRN) problem, that is a complex optimization problem that involve processing S-System models, which include large amount of gene expression data from hundreds (even thousands) of genes in multiple timeseries. So we can use the parallelism of Spark in the evaluation of the solutions of the metaheuristic. In this approach, we have used MOCell algorithm with Spark for evaluating each solution of the population in parallel.

The tests carried out showed that the proposed approach was able to obtain actual reductions in computing time from several days (10) to hours (8.3) when facing complex inference of large scale GRNs.

In addition, a noteworthy trade-off between performance and resource requirement was obtained with a configuration of the parallel MOCell in the range of 20 and 50 cores, since it finalized all the jobs launch in less than half a day and is 50% cost efficiently.

We did experiments on realistic benchmarking data of DREAM3 in order to assess the proposal and that showed that MOCell was competitive and often outperforms state-of-the-art GRN inference procedures.

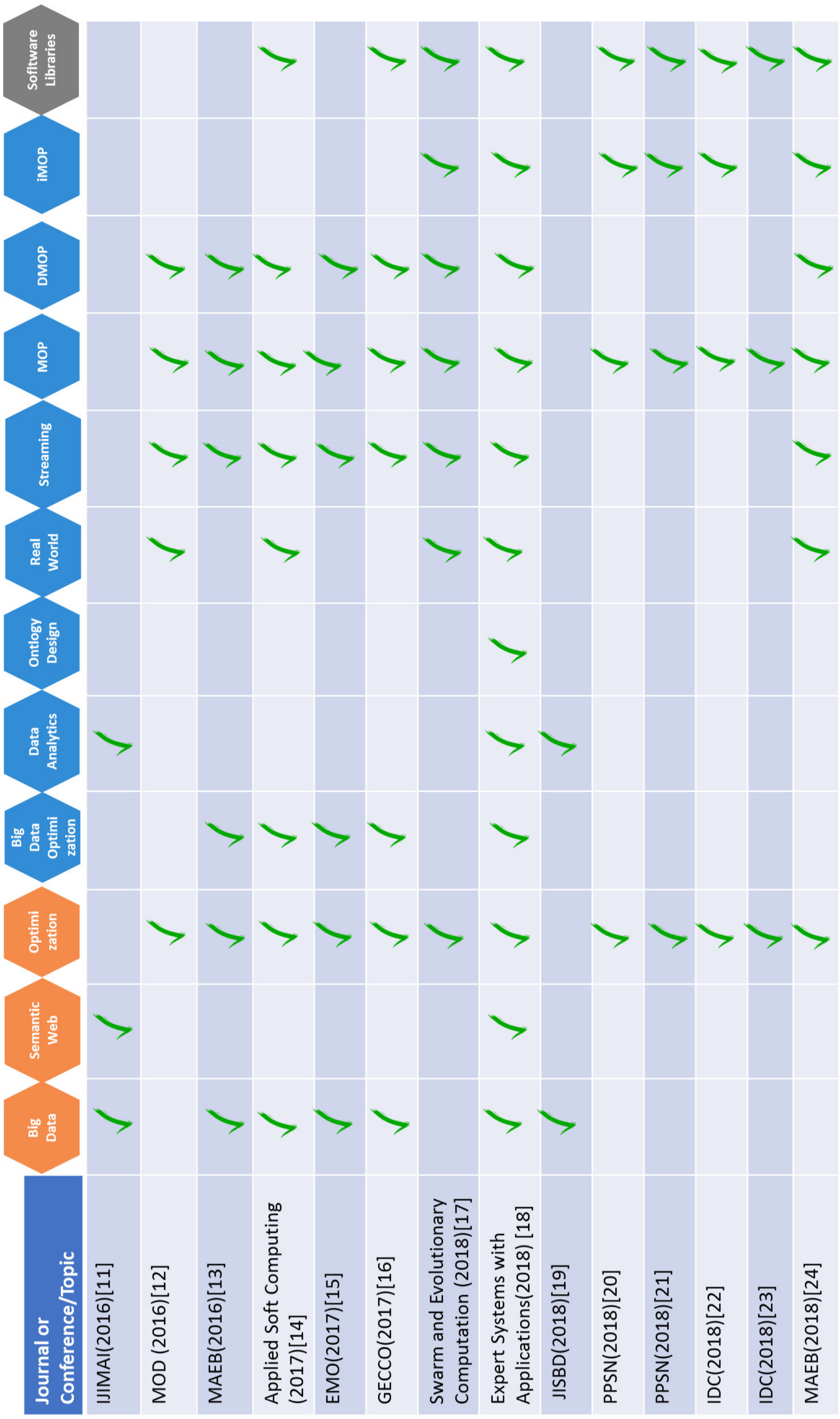
4.3 Summary of Other Publications Related to this Thesis

This section briefly comments the other articles that do not support this thesis but have been produced in the scope of it. One of them was published in the *Swarm and Evolutionary Computation* journal. The other five were published in international conferences and the other three were published in national conferences.

In [17] we presented InDM2, which combines dynamic multi-objective optimization, multiple criteria decision making and interactivity. A key feature in InDM2 was the mechanism to visualize, in optimization time, the approximations of the region of interest that are being generated throughout the solution process, together with the reference point driving these approximations. InDM2 was able to incorporate any reference point based evolutionary algorithms so as to handle the preference information interactively. Another important InDM2's feature was that allows the incorporation of strategies for reacting to changes in both the problem and the reference point. As InDM2 was developed in jMetalSP, it is free available. The performance of InDM2 was validated with three FDA benchmark DMOPs and with a real-world problem, consisting of a dynamic version of a bi-objective Traveling Salesman Problem (TSP) based on real traffic data provided by the New York City Department of Traffic.

The following articles [12, 16] were presented in different congresses and both are related to jMetalSP. The first one, *Dynamic Multi-Objective Optimization With jMetal and Spark: a Case Study* [12] was presented in the International Workshop on Machine Learning, Optimization, and Big Data (MOD 2016) and it was a first approximation of combining metaheuristics with Spark [16]. The second one, *Design and architecture of the jMetaISP framework* was presented in the conference Genetic and Evolutionary Computation Conference (GECCO 2017), an this work was about a re-design of jMetalSP. We did a new architecture of classes and new methods in order to make jMetlaSP more independent of Spark.

In the work, *Extending the Speed-constrained Multi-Objective PSO (SMPSO) With Reference Point Based Preference Articulation* [21], we presented in the conference 15th International Conference on Parallel Problem solving for Nature (PPSN 2018) a new version of the metaheuristic SMPSO. SMPSO/RP, incorporated interactive reference point preference articulation. In addition,



Journal or Conference/Topic	Big Data	Semantic Web	Optimi zation	Big Data Optimi zation	Data Analytics	Ontlogy Design	Real World	Streaming	MOP	DMOP	iMOP	Sofltware Libraries
IJIMAI(2016)[11]	✓	✓			✓							
MOD (2016)[12]			✓				✓	✓	✓	✓		
MAEB(2016)[13]	✓		✓	✓				✓	✓	✓		
Applied Soft Computing (2017)[14]	✓		✓	✓			✓	✓	✓	✓		✓
EMO(2017)[15]	✓		✓	✓				✓	✓	✓		
GECCO(2017)[16]	✓		✓	✓				✓	✓	✓		✓
Swarm and Evolutionary Computation (2018)[17]			✓				✓	✓	✓	✓	✓	✓
Expert Systems with Applications(2018) [18]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
JISBD(2018)[19]	✓				✓							
PPSN(2018)[20]			✓						✓		✓	✓
PPSN(2018)[21]			✓						✓		✓	✓
IDC(2018)[22]			✓						✓		✓	✓
IDC(2018)[23]			✓						✓			✓
MAEB(2018)[24]			✓				✓	✓	✓	✓	✓	✓

Figure 4.1: Research contributions in this thesis.

this algorithm proposal is able to deal with one or more DM preferences or regions of interest. SMPSO/RP has the ability to interactively change DM preferences by means of changing the desired reference points. Besides, included ability of parallel evaluations of particles. Finally, SMPSO included GUI for visualizing the computed front evolution for problems with two and three objectives.

A recent work,*About Designing an Observer Pattern-Based Architecture for a Multi-Objective Metaheuristic Optimization Framework* [23], it was presented in the 12th International Symposium on Intelligent Distributed Computing (IDC 2018).We have described in this paper a study about the design of an observer pattern-based architecture for a framework for multi-objective metaheuristics. In this architecture, all the algorithm components were classified into three categories of components: observable, observer/observable, and observer. By taking a multi-objective evolutionary algorithm (NSGA-II) as a case of study, we have shown how it can be implemented using the proposed scheme.

Our analysis indicates that the resulting implementation were very flexible, allowing to develop new variants of an algorithm by simply adding or replacing components. The computing times of two algorithms using the proposed architecture indicated a minimal time overhead regarding the original monolithic-based implementations.

A complementary paper was presented in Emerging Citation index, *A Fine Grain Sentiment Analysis with Semantics in Tweets* [11], which provides a service for assessing opinions on Twitter through classifying the sentiment of the tweets as positive or negative. We offered a combination of Big Data tools (under the Apache Hadoop framework) and sentiment analysis using RDF graphs supporting the study of the tweet's lexicon. This work was empirically validated using a sporting event, the 2014 Phillips 66 Big 12 Men's Basketball Championship. The experimental results showed a clear correlation between the predicted sentiments with specific events during the championship. This was our first work where we combined both Web Semantic and Big Data analysis.

Finally, with the aim of disseminating our results also in the scope of national research community, we have presented in national congresses our articles. The first one [13], in the *XI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2016)*, we presented a spanish version of the jMetalSP work. The second one [19], in *XXIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2018)*, we presented a work related to analyze, using the platform of this thesis for doing an activity classification using realistic data from wearable sensors in the context of cardiovascular data, we used Spark and its library MLlib. The workflow was designed using BIGOWL. Finally [24], in the *XIII Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2018)*, we presented a spanish version of InDM2 algorithm.

Figure 4.1 illustrates the list of all the research contributions developed during this thesis. In the table rows we see the articles published and in the columns their corresponding topics covering contributions in this thesis.

Chapter 5

Conclusions and Future Work

This chapter exposes the final ideas of this dissertation. Section 5.1 contains the conclusions obtained from all the experiments and discussions conducted in supporting publications. Then, in Section 5.2, we explain the future lines of research that we plan to explore from the latter works.

5.1 Conclusions

This thesis proposes and analyzes several new dynamic multi-objective metaheuristics to tackle Big Data optimization problems, based on extending multi-objective techniques and on identifying drawbacks and possible improvement opportunities, like interactivity or streaming process in existing dynamic multi-objective optimization algorithms.

This thesis provides software and conceptual frameworks for the implementation and experimentation (jMetalSP), as well as, for the semantic annotation (BIGOWL) of Big Data analytics. The proposed semantic model is materialized by means of an RDF repository, programmatic querying and reasoning functions.

Summarizing, the main contributions of this thesis can be enumerated as follows:

1. ***Design and analysis of dynamic multi-objective algorithm***. Taking as starting point NSGA-II algorithm, we have designed and implemented its dynamic version in our framework jMetalSP. Dynamic NSGA-II is used in our first works in order to assess the dynamic metaheuristics with streaming data in our cluster. Our goal is to asses the performance of running a metaheuristic based in our parallel scheme on three scenarios: parallel computation, parallel data access, and a combination of both. We have carried out experiments to measure the performance of the proposed parallel infrastructure in an environment based on virtual machines in a local cluster composed of up to 100 cores. The results lead us to get interesting conclusions about computational effort and to propose guidelines when facing Big Data optimization problems. First, our approach is able to obtain actual reductions in computing time from more than a week to just half a day, when addressing complex and time consuming optimization tasks. This performance has been obtained in the scope of an in-house (virtualized) computational environment with limited resources. Second, in those experiments where we only focus on data management without spending extra computational effort on evaluating the solutions, the overall performance of the parallel model is usually impaired when using a large number of nodes (more than 50), because of the network overhead and the increasing data transfer between nodes. Finally, a noteworthy trade-off between performance and resource requirements is obtained with a configuration of the cluster in the

range of 50 to 100 cores. We have designed and developed the dynamic version of NSGA-II, R-NSGA-II, NSGA-III, MOCell, SMPSO and WASF-GA.

2. ***Design and analysis of interactive multi-objective algorithms***. We propose in this thesis two interactive multi-objective algorithms. On one hand, we present for the first time in the state of the art, a dynamic multi-objective interactive algorithm, called InDM2, *this is an interactive multi-objective optimization metaheuristic for solving dynamic multi-objective optimization problems.* This proposal enables the DM to interactively change the region of interest that (s)he desires to approximate by giving and updating a reference point containing her/his preferences. InDM2 incorporates a reference point-based evolutionary algorithm as a base optimizer, currently including the WASF-GA and R-NSGA-II algorithms, which indeed allows to change the reference points during the optimization process. To assist the DM, the approximations obtained by the algorithm are shown in a graphical window. A key component of InDM2 is that its internal design allows to specify different restarting strategies to be applied when changes in the reference point and/or in the problem configuration are detected, making it more versatile. We analyzed InDM2 for solving DMOPs (benchmark FDA family problems) and a dynamic version of the combinatorial bi-objective optimization Traveling Salesman Problem, built using real-world streaming traffic data from New York City. InDM2 is able to react and adapt when the problem and the reference points change. Furthermore, it can handle preferences interactively, which has been able to generate approximation adjusted to the given preferences (i.e., the region of interest), in real-time, while the problem also changes at the same time. On the other hand, we introduced SMPSO/RP, this is an extension of the SMPSO metaheuristic incorporating a preference articulation mechanism based on indicating reference points. Our approach allows changing the reference points interactively and evaluating particles of the swarm in parallel. We compared SMPSO/RP against other interactive metaheuristics like gSMS-EMOA, gNSGA-II and WASF-GA. The results showed that SMPSO/RP achieves the best overall performance when indicating both achievable and unachievable reference points. We have also measured the time reductions that have been achieved when running the algorithm in a multi-core processor platform.

3. ***Design and analysis of artificial decision maker for testing interactive metaheuristics***. We introduce an artificial decision maker for preference articulation in the form of reference points guided by PSO. This approach, called ADM-PSO, enables comparing interactive reference point based EMOs without involving human DMs. The proposed approach was evaluated on the DTLZ benchmark problems with many-objectives and using R-NSGA-II and WASF-GA as interactive reference point based methods to be compared. The experimental results showed that ADM-PSO is useful and efficient in comparison with the previous artificial decision maker. It offers a bio-inspired and flexible mechanism to capture the current context of an ADM in interactive solution processes.

4. ***jMetalSP***. We have designed and implemented a software framework for developing and analyzing dynamic and interactive multi-objective metaheuristics. jMetalSP can manage streaming process, so it is able to deal with Big Data optimization. All the metaheuristics used in this thesis have been developed using jMetalSP. This way, any researcher can easily reproduce the results presented here. Furthermore, as jMetalSP is based on jMetal, it has all the algorithm, operators, type of solutions, etc, inherited from it. We have designed a methodology to easily cover any static metaheuristic from jMetal to dynamic one for jMetalSP. For all these reasons, we consider jMetalSP as an important contribution of this thesis.

5. ***Design and analysis of ontology model for Big Data optimization***. We propose an ontological approach, called BIGOWL to provide conceptual framework for the annotation

of Big Data analytics. To show the advantage of using BIGOWL, two case studies have been developed, which consist in: first, real-world streaming traffic data processing for route optimization in urban environment; and second, academic data mining classification on local/on-cloud platforms. The experience on these cases revealed that BIGOWL approach is useful when integrating knowledge domain concerning a specific analytic problem. Consequently, the integrated knowledge is used for guiding the design of Big Data analytic workflows, by recommending next components to be linked, and supporting on final validation.

6. ***Design a methodology for the annotating of data analytic workflows***. In this PhD thesis, we capture all the semantics needed to guide the smart design of Big Data analytic workflows and to enhance their performance. Therefore, following the BIGOWL ontology, it is possible to annotate analytic algorithms, datasets, problems, and workflows in Big Data context. Thus describing the data value chain of all of them. The semantic annotation and querying procedure involves all key concepts in analytic process like: algorithms, technological/platform features, and attributes of problem domain of knowledge; and automatic querying by means of SPARQL sentences. In addition, we designed a series of SWRL rules for reasoning new information that is not explicitly expressed in the knowledge base. The new information will indicate, when applicable and among others, whether an analytic workflow is correctly composed, or not.

Summarizing, during this thesis work we have made a wide number of contributions to the field of Big Data optimization in several ways. From the algorithmic point of view, new techniques have been developed and further analyzed attending to different issues. From the point of view of applications, we have tackled several engineering problems belonging to different areas, showing the utility of our proposals for addressing problems that could arise in academy and industry. And from the semantic point of view, we have developed an ontology that deals with the annotation of analytic workflows.

5.2 Future Work

As future lines of research in general, we plan to continue this proposal and analysis of Big Data optimization algorithms. In particular, one of the topics that we find of particular interest is the study of dynamic and interactive multi-objective optimization algorithms for Big Data environments. Our point to carry out such type of study that this kind of metaheuristics can tackle real world problem, since they are able to detect changes in the problem and copes with them. Interactivity helps to the analyst to add his/her preferred information and make easier the search on the solution space of the problem. Nowadays, some studies pointing in this direction are appearing [25, 26], where the idea is to combine traditional well-know algorithm with interactive operators in order to add interactivity to them.

We would also like to continue working in the design a new artificial decision makers. It is conceptually intuitive and straightforward, although it opens a promising line of future research. The future work that we want to address as a continuation of this thesis can be summarized as follows:

- We want to add semantic in dynamic and interactive multi-objective algorithms for solving Big Data optimization problems. Through semantic we can add the problem domain in interactive algorithm, via reference points, and improving the searching solution process.

- We plan to explore the possibilities of using different metaheuristics like DE, CMA-ES and GA for the generation of reference points instead of PSO in new ADMs.

- Other future research is to test parameter tuning in PSO (and other metaheuristics), e.g., φ_1 and φ_2, to control the influence of the current reference point (global best) and/or local history of particles, hence to induce the ADM's behavior in terms of intensification/diversification mechanisms.

- We want to carry out further comparisons of multiple state-of-the art iEMOs, to test their performances in a controlled and computationally fair execution framework.

- We also plan to apply the algorithms proposed in this thesis for solving new challenging real-world instance problems. On concrete, we are interested in applying them for solving more complex problems related to life sciences, such as: bioinformatics and agri-food.

- Taking as starting point the article *Building and Using an Ontology of Preference-Based Multiobjective Evolutionary Algorithms* [27], we want to improve BIGOWL with preference-based multi-objective algorithms, thus BIGOWL will be able to support MCDM algorithms.

- In the same way that we have annotated jMetalSP with BIGOWL we plan to annotate Machine Learning libraries like: Spark MLlib, BIGML, Weka and so on.

- We want to apply the idea of using semantic for designing workflows in designing algorithms, that is to say, we can use the semantic for annotating the components of optimization algorithms (selector operators crossover, mutation and so on) and through reasoner rules, generate new designs of algorithms. Currently, some studies addressing the generation of automatic designed of algorithms are appearing, like [28, 29, 30, 31, 32, 33].

List of Tables

List of Figures

Bibliography

[1] M. Chen, S. Mao, and Y. Liu. “Big data: A survey”. *Mobile networks and applications* 19.2 (2014), pp. 171–209.

[2] G.-H. Kim, S. Trimi, and J.-H. Chung. “Big-data applications in the government sector”. *Communications of the ACM* 57.3 (2014), pp. 78–85.

[3] J. J. Durillo and A. J. Nebro. “jMetal: A Java framework for multi-objective optimization”. *Advances in Engineering Software* 42.10 (2011), pp. 760–771.

[4] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. “Spark: Cluster Computing with Working Sets”. *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*. HotCloud’10. USENIX Association, 2010, pp. 10–10.

[5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. “The hadoop distributed file system”. *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee. 2010, pp. 1–10.

[6] V. Ojalehto, D. Podkopaev, and K. Miettinen. “Towards automatic testing of reference point based interactive methods”. *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 483–492.

[7] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen. “Action research”. *Communications of the ACM* 42.1 (1999), pp. 94–97.

[8] C. B. Seaman. “Qualitative methods in empirical studies of software engineering”. *IEEE Transactions on software engineering* 25.4 (1999), pp. 557–572.

[9] M. Bunge. “La Investigación Científica”. *Ariel S.A.* (1976).

[10] P. Reason and H. Bradbury. *Handbook of action research: Participative inquiry and practice*. Sage, 2001.

[11] C. Barba-González, J. García-Nieto, I. N. Delgado, and J. F. A. Montes. “A fine grain sentiment analysis with semantics in Tweets”. *IJIMAI* 3.6 (2016), pp. 22–28.

[12] J. A. Cordero, A. J. Nebro, C. Barba-González, J. J. Durillo, J. García-Nieto, I. Navas-Delgado, and J. F. Aldana-Montes. “Dynamic multi-objective optimization with jMetal and Spark: a case study”. *LNCS of International Workshop on Machine Learning, Optimization and Big Data (MOD 2016)*. Springer. 2016, pp. 106–117.

[13] C. Barba-González, A. J. Nebro, J. García-Nieto, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes. “Un Framework para Big Data Optimization Basado en jMetal y Spark” (2016), pp. 159–168.

[14] C. Barba-González, J. García-Nieto, A. J. Nebro, J. A. Cordero, J. J. Durillo, I. Navas-Delgado, and J. F. Aldana-Montes. “jMetalSP: a framework for dynamic multi-objective Big Data optimization”. *Applied Soft Computing* 69 (2017), pp. 737–748.

[15] C. Barba-Gonzaléz, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Multi-objective big data optimization with jMetal and spark”. *LNCS of International Conference on Evolutionary Multi-Criterion Optimization (EMO’17), GGS class 2 (CORE A).* Springer. 2017, pp. 16–30.

[16] A. J. Nebro, C. Barba-González, J. García-Nieto, J. A. Cordero, and J. F. A. Montes. “Design and architecture of the jMetalSP framework”. *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO’17).* ACM. 2017, pp. 1239–1246.

[17] A. J. Nebro, A. B. Ruiz, C. Barba-González, J. García-Nieto, M. Luque, and J. F. Aldana-Montes. “InDM2: Interactive Dynamic Multi-Objective Decision Making Using Evolutionary Algorithms”. *Swarm and Evolutionary Computation* 40 (2018), pp. 184–195.

[18] C. Barba-González, J. García-Nieto, M. d. M. Rodan-García, I. Navas-Delagado, and J. F. Aldana-Montes. “BIGOWL: Knowledge Centered Big Data Analytics”. *Expert Systems with Applications* 115, (2018), pp. 543–556.

[19] S. Huratdo-Requena, C. Barba-González, M. Rybiński, F. J. Barón-López, J. Wärnberg, I. Navas-Delgado, and J. F. Aldana-Montes. “Análisis de datos de acelerometría para la detección de tipos de actividades”. *Jornadas de Ingeniería del Software y Bases de Datos (In press).* 2018.

[20] C. Barba-González, V. Ojalehto, J. García-Nieto, A. J. Nebro, K. Miettinen, and J. F. Aldana-Montes. “Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods”. *Proceeding of 15th International conference on parallel problem solving from nature (PPSN’18), GGS A class 2(CORE A).* Springer, 2018, pp. 274–285.

[21] A. J. Nebro, J. J. Durillo, J. García-Nieto, C. Barba-González, J. Del Ser, C. A. Coello Coello, A. Benítez-Hidalgo, and J. F. Aldana-Montes. “Extending the Speed-constrained Multi-Objective PSO (SMPSO) With Reference Point Based Preference Articulation”. *Proceeding of 15th International conference on parallel problem solving from nature (PPSN’18), GGS A-, CORE A.* Springer, 2018, pp. 298–310.

[22] C. Barba-González, J. García-Nieto, A. J. Nebro, A. Benítez-Hidalgo, and J. F. Aldana-Montes. “Scalable Inference of Gene Regulatory Networks with the Spark Distributed Computing Platform”. *Springer Series of 12th International Symposium on Intelligent Distributed Computing (IDC’18).* 2018.

[23] A. Benítez-Hidalgo, A. J. Nebro, J. J. Durilllo, J. García-Nieto, E. Camacho-López, C. Barba-González, B, and J. F. Aldana-Montes. “About Designing an Observer Pattern-Based Architecture for a Multi-ObjectiveMetaheuristic Optimization Framework”. *Springer Series of 12th International Symposium on Intelligent Distributed Computing (IDC’18).* 2018.

[24] C. Barba-González, J. García-Nieto, A. B. Ruiz, A. J. Nebro, M. Luque, and J. F. Aldana-Montes. “Algoritmo Evolutivo Multi-Objetivo para la Toma de Decisiones Interactiva en Optimización Dinámica” (2018).

[25] K. Deb and J Sundar. “Reference point based multi-objective optimization using evolutionary algorithms”. *Proceedings of the 8th annual conference on Genetic and evolutionary computation.* ACM. 2006, pp. 635–642.

[26] L. B. Said, S. Bechikh, and K. Ghédira. “The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making”. *IEEE Transactions on Evolutionary Computation* 14.5 (2010), pp. 801–818.

[27] L. Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, and M. Emmerich. “Building and using an ontology of preference-based multiobjective evolutionary algorithms”. *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2017, pp. 406–421.

[28] T Stützle and M López-Ibánez. “Automated Design of Metaheuristic Algorithms” (2018).

[29] L. Bezerra, M López-Ibánez, and T Stützle. “Automatically Designing State-of-the-Art Multi-and Many-Objective Evolutionary Algorithms” (2018).

[30] M. López-Ibánez, M.-E. Kessaci, and T. Stützle. “Automatic Design of Hybrid Metaheuristics from Algorithmic Components” (2017).

[31] A. R. KhudaBukhsh, L. Xu, H. H. Hoos, and K. Leyton-Brown. “SATenstein: Automatically building local search SAT solvers from components”. *Artificial Intelligence* 232 (2016), pp. 20–42.

[32] M. López-Ibáñez and T. Stützle. “Automated offline design of algorithms”. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2017, pp. 1038–1065.

[33] L. Bezerra, M López-Ibánez, and T Stützle. “Automatic Configuration of Multi-objective Optimizers and Multi-objective Configuration” (2017).

[34] B. D. V. Association. “European Big Data Value Strategic Research and Innovation Agenda”. *Big Data value association*. 2017. URL: `http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf`.

[35] T. Becker, E. Curry, A. Jentzsch, and W. Palmetshofer. “New Horizons for a Data-Driven Economy: Roadmaps and Action Plans for Technology, Businesses, Policy, and Society”. *New Horizons for a Data-Driven Economy*. Springer, 2016, pp. 277–291.

[36] M. Farina, K. Deb, and P. Amato. “Dynamic multiobjective optimization problems: test cases, approximations, and applications”. *IEEE Transactions on Evolutionary Computation* 8.5 (2004), pp. 425–442.

[37] M. López-Ibáñez and J. Knowles. “Machine decision makers as a laboratory for interactive EMO”. *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2015, pp. 295–309.

[38] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, eds. *Multiobjective optimization: Interactive and evolutionary approaches*. Springer, 2008.

[39] V. Belton and T. Stewart. *Multiple criteria decision analysis: an integrated approach*. Springer Science & Business Media, 2002.

[40] S. Greco, J Figueira, and M Ehrgott. “Multiple criteria decision analysis”. *Springer’s International series* (2005).

[41] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang. “A review of hybrid evolutionary multiple criteria decision making methods”. *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE. 2014, pp. 1147–1154.

[42] K. Belhajjame, S. M. Embury, N. W. Paton, R. Stevens, and C. A. Goble. “Automatic annotation of web services based on workflow definitions”. *ACM Transactions on the Web (TWEB)* 2.2 (2008), p. 11.

[43] Z. H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams. “Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives [Discussion Forum]”. *IEEE Computational Intelligence Magazine* 9.4 (2014), pp. 62–74. ISSN: 1556-603X.

[44] S. John Walker. *Big data: A revolution that will transform how we live, work, and think.* 2014.

[45] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo. “Next-generation big data analytics: State of the art, challenges, and future research topics”. *IEEE Transactions on Industrial Informatics* 13.4 (2017), pp. 1891–1899.

[46] D. Laney. “3D data management: Controlling data volume, velocity and variety”. *META Group Research Note* 6.70 (2001).

[47] J. Zakir, T. Seymour, and K. Berg. “BIG DATA ANALYTICS.” *Issues in Information Systems* 16.2 (2015).

[48] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, M. Ali, W. Kamaleldin, M. Alam, M. Shiraz, and A. Gani. “Big data: survey, technologies, opportunities, and challenges”. *The Scientific World Journal* 2014 (2014).

[49] P. Mell and T. Grance. “The NIST definition of cloud computing” (2011).

[50] D. Agrawal, S. Das, and A. El Abbadi. “Big data and cloud computing: new wine or just new bottles?” *Proceedings of the VLDB Endowment* 3.1-2 (2010), pp. 1647–1648.

[51] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé. “Vision and challenges for realising the Internet of Things”. *Cluster of European Research Projects on the Internet of Things, European Commision* 3.3 (2010), pp. 34–36.

[52] L. Atzori, A. Iera, and G. Morabito. “The internet of things: A survey”. *Computer networks* 54.15 (2010), pp. 2787–2805.

[53] A. Moniruzzaman and S. A. Hossain. “NoSQL database: New era of databases for Big Data analytics-classification, characteristics and comparison”. *arXiv preprint arXiv:1307.0191* (2013).

[54] J. Han, E Haihong, G. Le, and J. Du. “Survey on NoSQL database”. *Pervasive computing and applications (ICPCA), 2011 6th international conference on.* IEEE. 2011, pp. 363–366.

[55] J. L. Carlson. *Redis in action.* Manning Publications Co., 2013.

[56] R. Cattell. “Scalable SQL and NoSQL data stores”. *ACM Sigmod Record* 39.4 (2011), pp. 12–27.

[57] N. Leavitt. “Will NoSQL databases live up to their promise?” *Computer* 43.2 (2010).

[58] L. George. *HBase: the definitive guide: random access to your planet-size data.* " O’Reilly Media, Inc.", 2011.

[59] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, and A. Rasin. “HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads”. *Proceedings of the VLDB Endowment* 2.1 (2009), pp. 922–933.

[60] E. Dede, B. Sendir, P. Kuzlu, J. Hartog, and M. Govindaraju. “An evaluation of Cassandra for Hadoop”. *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on.* IEEE. 2013, pp. 494–501.

[61] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. “Bigtable: A distributed storage system for structured data”. *ACM Transactions on Computer Systems (TOCS)* 26.2 (2008), p. 4.

[62] K. Banker. *MongoDB in action.* Manning Publications Co., 2011.

[63] F. Provost and T. Fawcett. “Data science and its relationship to big data and data-driven decision making”. *Big data* 1.1 (2013), pp. 51–59.

[64] S. Ghemawat, H. Gobioff, and S.-T. Leung. *The Google file system.* Vol. 37. 5. ACM, 2003.

[65] S. Owen and S. Owen. "Mahout in action" (2012).

[66] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, and S. Owen. "Mllib: Machine learning in apache spark". *The Journal of Machine Learning Research* 17.1 (2016), pp. 1235–1241.

[67] W. Huang, L. Meng, D. Zhang, and W. Zhang. "In-memory parallel processing of massive remotely sensed data using an Apache Spark on Hadoop Yarn model". *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10.1 (2017), pp. 3–19.

[68] A. Arasu and H. Garcia-Molina. "Extracting structured data from web pages". *Proceedings of the 2003 ACM SIGMOD international conference on Management of data.* ACM. 2003, pp. 337–348.

[69] P. Buneman. "Semistructured data". *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* ACM. 1997, pp. 117–121.

[70] H. Baars and H.-G. Kemper. "Management support with structured and unstructured data—an integrated business intelligence framework". *Information Systems Management* 25.2 (2008), pp. 132–148.

[71] B. Brown, M. Chui, and J. Manyika. "Are you ready for the era of 'Big Data'". *McKinsey Quarterly* 4.1 (2011), pp. 24–35.

[72] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. "Advances in knowledge discovery and data mining" (1996).

[73] S. Cheng, Y. Shi, Q. Qin, and R. Bai. "Swarm intelligence in Big Data analytics". *International Conference on Intelligent Data Engineering and Automated Learning.* Springer. 2013, pp. 417–426.

[74] S. Abdul-Rahman, A. A. Bakar, and Z. A. Mohamed-Hussein. "Optimizing Big Data in Bioinformatics with Swarm Algorithms". *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on.* 2013, pp. 1091–1095.

[75] K. Govindarajan, T. S. Somasundaram, V. S. Kumar, and Kinshuk. "Continuous Clustering in Big Data Learning Analytics". *Technology for Education (T4E), 2013 IEEE Fifth International Conference on.* 2013, pp. 61–64.

[76] B. K. Tannahill and M. Jamshidi. "System of Systems and Big Data analytics, Bridging the gap". *Computers and Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, pp. 2 –15. ISSN: 0045-7906.

[77] A. Cabanas-Abascal, E. García-Machicado, L. Prieto-González, and A. de Amescua Seco. "An Item based Geo-Recommender System Inspired by Artificial Immune Algorithms". *j-jucs* 19.13 (2013), pp. 2013–2033.

[78] W.-P. Lee, Y.-T. Hsiao, and W.-C. Hwang. "Designing a parallel evolutionary algorithm for inferring gene networks on the cloud computing environment". *BMC Systems Biology* 8.1 (2014), pp. 1–19. ISSN: 1752-0509.

[79] W. Sun, N. Zhang, H. Wang, W. Yin, and T. Qiu. "PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing". *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on.* 2013, pp. 482–486.

[80] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, 1996.

[81] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[82] K. Deb and K. Miettinen. “Nadir point estimation using evolutionary approaches: better accuracy and computational speed through focused search”. *Multiple criteria decision making for sustainable energy and transportation systems.* Springer, 2010, pp. 339–354.

[83] K. Miettinen. *Nonlinear Multiobjective Optimization, volume 12 of International Series in Operations Research and Management Science.* 1999.

[84] K. Miettinen, J. Hakanen, and D. Podkopaev. “Interactive Nonlinear Multiobjective Optimization Methods”. *Multiple Criteria Decision Analysis: State of the Art Surveys.* Ed. by S. Greco, M. Ehrgott, and J. Figueira. Springer, 2016, pp. 931–980.

[85] T. J. Stewart. “Goal programming and cognitive biases in decision-making”. *Journal of the Operational Research Society* 56.10 (2005), pp. 1166–1175.

[86] J. L. Cohon and D. H. Marks. “A Review and Evaluation of Multiobjective Programming Techniques”. *Water Resources Research* 11.2 (1975), pp. 208 –220.

[87] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Second. Genetic and Evolutionary Computation Series. Springer, 2007.

[88] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* New York, NY, USA: John Wiley & Sons, 2001.

[89] V. Pareto. *Cours D’Economie Politique.* Vol. I and II. Lausanne: F. Rouge, 1896.

[90] P. C. Fishburn. “Exceptional paper—Lexicographic orders, utilities and decision rules: A survey”. *Management science* 20.11 (1974), pp. 1442–1471.

[91] A. Charnes and W. W. Cooper. “Goal programming and multiple objective optimizations: Part 1”. *European Journal of Operational Research* 1.1 (1977), pp. 39–54.

[92] A. Charnes, W. W. Cooper, and R. O. Ferguson. “Optimal estimation of executive compensation by linear programming”. *Management science* 1.2 (1955), pp. 138–151.

[93] R Benayoun, J De Montgolfier, J. Tergny, and O Laritchev. “Linear programming with multiple objective functions: Step method (STEM)”. *Mathematical programming* 1.1 (1971), pp. 366–375.

[94] A. P. Wierzbicki. “The use of reference objectives in multiobjective optimization”. *Multiple criteria decision making theory and application.* Springer, 1980, pp. 468–486.

[95] H. Nakayama and Y. Sawaragi. “Satisficing trade-off method for multiobjective programming”. *Interactive decision analysis.* Springer, 1984, pp. 113–122.

[96] K Miettinen and M. Mäkelä. “Interactive bundle-based method for nondifferentiable multiobjeective optimization: NIMBUS”. *Optimization* 34.3 (1995), pp. 231–246.

[97] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius. “An interactive evolutionary multiobjective optimization method based on progressively approximated value functions”. *IEEE Transactions on Evolutionary Computation* 14.5 (2010), pp. 723–739.

[98] J. W. Fowler, E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius. “Interactive evolutionary multi-objective optimization for quasi-concave preference functions”. *European Journal of Operational Research* 206.2 (2010), pp. 417–425.

[99] S. Phelps and M. Köksalan. “An interactive evolutionary metaheuristic for multiobjective combinatorial optimization”. *Management Science* 49.12 (2003), pp. 1726–1738.

[100] M. Koksalan and I. Karahan. “An interactive territory defining evolutionary algorithm: iTDEA”. *IEEE Transactions on Evolutionary Computation* 14.5 (2010), pp. 702–722.

[101] J. Branke, T. Kaußler, C. Smidt, and H. Schmeck. “A multi-population approach to dynamic optimization problems”. *Evolutionary Design and Manufacture.* Springer, 2000, pp. 299–307.

[102] R. Eberhart and J. Kennedy. “A new optimizer using particle swarm theory”. *Micro Machine and Human Science, 1995. MHS’95., Proceedings of the Sixth International Symposium on.* IEEE. 1995, pp. 39–43.

[103] C. Blum and A. Roli. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”. *ACM Computing Surveys* 35.3 (2003), pp. 268–308.

[104] G. Luque. “Resolución de Problemas Combinatorios con Aplicación Real en Sistemas Distribuidos”. PhD thesis. University of Málaga, 2006.

[105] J. F. Chicano. “Metaheurísticas e Ingeniería del Software”. PhD thesis. University of Málaga, 2007.

[106] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams. “Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]”. *IEEE Computational Intelligence Magazine* 9.4 (2014), pp. 62–74.

[107] R. Bellman. “Dynamic programming treatment of the travelling salesman problem”. *Journal of the ACM (JACM)* 9.1 (1962), pp. 61–63.

[108] Z. Bingul. “Adaptive genetic algorithms applied to dynamic multiobjective problems”. *Applied Soft Computing* 7.3 (2007), pp. 791–799.

[109] S.-U. Guan, Q. Chen, and W. Mo. “Evolving dynamic multi-objective optimization problems with objective replacement”. *Artificial Intelligence Review* 23.3 (2005), pp. 267–293.

[110] F. Szidarovszky and L. Duckstein. “Dynamic multiobjective optimization: a framework with application to regional water and mining management”. *European Journal of Operational Research* 24.2 (1986), pp. 305–317.

[111] K. Trojanowski and S. T. Wierzchoń. “Immune-based algorithms for dynamic optimization”. *Information Sciences* 179.10 (2009), pp. 1495–1515.

[112] T. T. Nguyen, S. Yang, and J. Branke. “Evolutionary dynamic optimization: A survey of the state of the art”. *Swarm and Evolutionary Computation* 6 (2012), pp. 1–24.

[113] M. Mavrovouniotis, C. Li, and S. Yang. “A survey of swarm intelligence for dynamic optimization: Algorithms and applications”. *Swarm and Evolutionary Computation* 33 (2017), pp. 1–17.

[114] J. Branke and H. Schmeck. “Designing evolutionary algorithms for dynamic optimization problems”. *Advances in evolutionary computing.* Springer, 2003, pp. 239–262.

[115] C. Cruz, J. R. González, and D. A. Pelta. “Optimization in dynamic environments: a survey on problems, methods and measures”. *Soft Computing* 15.7 (2011), pp. 1427–1448.

[116] T. Blackwell and J. Branke. “Multiswarms, exclusion, and anti-convergence in dynamic environments”. *IEEE transactions on evolutionary computation* 10.4 (2006), pp. 459–472.

[117] R. Tinós and S. Yang. “Evolutionary programming with q-Gaussian mutation for dynamic optimization problems”. *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on.* IEEE. 2008, pp. 1823–1830.

[118] H. A. Abbass, K. Sastry, and D. E. Goldberg. “Oiling the Wheels of Change: The Role of Adaptive Automatic Problem Decomposition in Non–Stationary Environments”. *arXiv preprint cs/0502021* (2005).

[119] H. G. Cobb. *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments.* Tech. rep. NAVAL RESEARCH LAB WASHINGTON DC, 1990.

[120] S. Yang, H. Cheng, and F. Wang. "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.1 (2010), pp. 52–63.

[121] D. Pelta, C. Cruz, and J. L. Verdegay. "Simple control rules in a cooperative system for dynamic optimisation problems". *International Journal of General Systems* 38.7 (2009), pp. 701–717.

[122] A. Karaman, Ş. Uyar, and G. Eryiğit. "The memory indexing evolutionary algorithm for dynamic environments". *Workshops on Applications of Evolutionary Computation.* Springer. 2005, pp. 563–573.

[123] I. Moser and T. Hendtlass. "A simple and efficient multi-component algorithm for solving dynamic function optimisation problems". *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.* IEEE. 2007, pp. 252–259.

[124] C. Li and S. Yang. "Fast multi-swarm optimization for dynamic optimization problems". *Natural Computation, 2008. ICNC'08. Fourth International Conference on.* Vol. 7. IEEE. 2008, pp. 624–628.

[125] P. Novoa, D. A. Pelta, C. Cruz, and I. G. del Amo. "Controlling particle trajectories in a multi-swarm approach for dynamic optimization problems". *International Work-Conference on the Interplay Between Natural and Artificial Computation.* Springer. 2009, pp. 285–294.

[126] A. Jaszkiewicz and R. Słowiński. "The 'Light Beam Search'approach–an overview of methodology applications". *European Journal of Operational Research* 113.2 (1999), pp. 300–314.

[127] P. Korhonen and J. Laakso. "Solving generalized goal programming problems using a visual interactive approach". *European Journal of Operational Research* 26.3 (1986), pp. 355–363.

[128] Ö. Özpeynirci, S. Özpeynirci, and A. Kaya. "An interactive approach for multiple criteria selection problem". *Computers & Operations Research* 78 (2017), pp. 154–162.

[129] B. Lokman, M. Köksalan, P. J. Korhonen, and J. Wallenius. "An interactive approximation algorithm for multi-objective integer programs". *Computers & Operations Research* 96 (2018), pp. 80–90.

[130] M. M. Dalini and A. Noura. "Interactive Algorithm for Obtain the Most Preferred Solution in DEA". *International Journal of Applied Mathematics and Statistics*^{TM} 56.5 (2017), pp. 177–184.

[131] A. Nebro, J. Durillo, J. García-Nieto, C. Coello Coello, F. Luna, and E. Alba. "SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization". *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009.* IEEE Press, 2009, pp. 66–73.

[132] E. E. Seidy. "A New Particle Swarm Optimization Based Stock Market Prediction Technique". *International Journal of Advanced Computer Science and Applications* 7.2 (2016).

[133] J. Nenortaite and R. Butleris. "Application of particle swarm optimization algorithm to decision making model incorporating cluster analysis". *Proceedings of the 2008 Conference on Human System Interactions.* IEEE, 2008, pp. 88–93.

[134] T. Berners-Lee, J. Hendler, and O. Lassila. "The semantic web". *Scientific american* 284.5 (2001), pp. 34–43.

[135] N. Shadbolt, T. Berners-Lee, and W. Hall. "The semantic web revisited". *IEEE intelligent systems* 21.3 (2006), pp. 96–101.

[136] N. F. Noy and D. L. McGuinness. *Ontology development 101: A guide to creating your first ontology.* 2001.

[137] N. Guarino. “Formal ontology and information systems”. *Proceedings of FOIS.* Vol. 98. 1998. 1998, pp. 81–97.

[138] B. McBride. “The resource description framework (RDF) and its vocabulary description language RDFS”. *Handbook on ontologies.* Springer, 2004, pp. 51–65.

[139] S. Staab and R. Studer. *Handbook on ontologies.* Springer Science & Business Media, 2013.

[140] T. R. Gruber. “A translation approach to portable ontology specifications”. *Knowledge acquisition* 5.2 (1993), pp. 199–220.

[141] D. L. McGuinness and F. Van Harmelen. “OWL web ontology language overview”. *W3C recommendation* 10.10 (2004), p. 2004.

[142] S. Harris, A. Seaborne, and E. Prud’hommeaux. “SPARQL 1.1 query language”. *W3C recommendation* 21.10 (2013).

[143] E. Prud and A. Seaborne. “SPARQL query language for RDF”. *W3C recommendation* (2006).

[144] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov. “OWL rules: A proposal and prototype implementation”. *Web Semantics: Science, Services and Agents on the World Wide Web* 3.1 (2005), pp. 23 –40. ISSN: 1570-8268.

[145] B. N. Grosof and T. C. Poon. “SweetDeal: Representing Agent Contracts with Exceptions Using Semantic Web Rules, Ontologies, and Process Descriptions”. *Int. Journal of Electronic Commerce* 8.4 (2004), pp. 61–97.

[146] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. “Pellet: A practical OWL-DL reasoner”. *Web Semantics: Science, Services and Agents on the WWW* 5.2 (2007), pp. 51 –53.

[147] A. Nebro, J. J. Durillo, and M. Vergne. “Redesigning the jMetal Multi-Objective Optimization Framework”. *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation.* GECCO Companion ’15. ACM, 2015, pp. 1093–1100. ISBN: 978-1-4503-3488-4.

[148] T. White. *Hadoop: The Definitive Guide.* 1st. O’Reilly Media, Inc., 2009. ISBN: 0596521979, 9780596521974.

[149] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN: 0-201-63361-2.

[150] R. L. Graham. “Bounds on multiprocessor timing anomalies”. *SIAM J. Appl. Math.* 17 (1969), pp. 263–269.

[151] R. M. Karp. “Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane”. *Mathematics of operations research* 2.3 (1977), pp. 209–224.

[152] D. A. Van Veldhuizen and G. B. Lamont. *Multiobjective evolutionary algorithm research: A history and analysis.* Tech. rep. Citeseer, 1998.

[153] E. Zitzler and L. Thiele. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”. *IEEE transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271.

[154] J. D. Knowles, L. Thiele, and E. Zitzler. “A tutorial on the performance assessment of stochastic multiobjective optimizers”. *TIK-Report* 214 (2006).

[155] D. J. Sheskin. *Handbook of parametric and nonparametric statistical procedures.* crc Press, 2003.

[156] S. García, D. Molina, M. Lozano, and F. Herrera. “A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the CEC’2005 special session on real parameter optimization”. *Journal of Heuristics* 15.6 (2009), p. 617.