UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional Ingeniería de Sistemas

Implementación de un modelo matemático para la planificación de un viaje personalizado basado en optimización multiobjetivo para los turistas en la región Puno

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:

Yuselenin Anquise Jihuaña

Asesor:

Mg. Abel Ángel Sullón Macalupú

Co-Asesor:

Dra. Sc. Nélida Gladys Maquera Sosa

Juliaca, julio de 2019

DECLARACIÓN JURADA
DE AUTORÍA DEL INFORME DE TESIS

Mg. Abel Ángel Sullón Macalupú, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: "IMPLEMENTACIÓN DE UN MODELO MATEMÁTICO PARA LA PLANIFICACIÓN DE UN VIAJE PERSONALIZADO BASADO EN OPTIMIZACIÓN MULTIOBJETIVO PARA LOS TURISTAS EN LA REGIÓN PUNO" constituye la memoria que presenta la bachiller Yuselenin Anquise Jihuaña para aspirar al título Profesional de Ingeniero de Sistemas ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca a los veinte y un días del mes de agosto del año dos mil diecinueve.

Mg. Abel Ángel Sullón Macalupú

042

ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquiani, a...dos.....día(s) del mes de...Julio.....del año 20.19. siendo las...8:40...horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Mg. Lennin Henry Centurión Julca, el secretario: Dr. Jorge Alejandro Sánchez Garcés y los demás miembros: Ing. Angel Rosendo Condori Zaquira y el asesor Mg. Abel Angel Sullon Macalupu con el propósito de administrar el acto académico de sustentación de la tesis titulada: Implementación de un modelo matemático para la planificación de un viaje personalizado basado en optimización multiobjetivo para los turistas en la region Puno de el(los)/la(las) bachiller/es: a) Yuselenin Anguise Tihuaña b) conducente a la obtención del título profesional de Ingeniero de Sistemas

(Nombre del Título Profesional)

con mención en.....

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): Yuselenin Anguise Tihuaña

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	15	B-	Bueno	Muy bueno

Candidato (b): .....

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente

Secretario

Asesor

Miembro

Miembro

Candidato/a (a)

Candidato/a (b)

DEDICATORIA

Dedico este trabajo a Dios porque es el todopoderoso que me dio la vida, que me cuidó hasta aquí, por no rendirse conmigo, y a mis padres por su gran apoyo, sus oraciones en todo el proceso de la tesis.

AGRADECIMIENTOS

Agradecer a Dios por darme la inteligencia, por darme las fuerzas para poder realizar este trabajo de investigación de fin de carrera. Recuerdo cuando Dios escuchó mi oración, cuando no sabía que hacer, al salir del colegio, y por su plan estuve en esta universidad con la bendición de ser parte de Beca 18.

Agradecer a mis padres por su apoyo, por sus oraciones, por su esfuerzo y sacrificio que realizan en el día a día para que yo y mis hermanos podamos aprovechar las oportunidades y salir adelante.

Agradecer de gran manera a la Dra. Gladys Maquera Sosa, por todo su apoyo, por sus consejos, por no rendirse conmigo, por compartir conocimiento y su experiencia.

Agradecer a mi asesor por su apoyo.

Y agradecer a Ing. Aymeth Quispe Marín, Germán Castro Vilchez, Rosbel Neftalí Ccana Yucra, Mirian Calcina Vilcapaza, Roger Ticona Nina, Ing. Oscar Mendoza Apaza, por su apoyo, por su tiempo y por sus palabras de aliento.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

ABREVIATURAS Y ACRÓNIMOS

- TSP: siglas en inglés Traveling Salesman Problem de Problema del Agente Viajero
- BTSP: siglas en inglés Bi-objetive Traveling Salesman Problem de Problema del Agente Viajero Bi-Objetivo
- MOTSP: siglas en inglés Multi objective Traveling Salesman Problem de Problema del Agente Viajero Multiobjetivo
- NSGA-II: siglas en inglés Non-dominated Sorting Genetic Algorithm II
- IA: siglas de Inteligencia Artificial
- MINCETUR: siglas de Ministerio de Comercio Exterior y Turismo
- PROMPERU: siglas de Comisión de Promoción del Perú para la Exportación y el Turismo
- PBI: siglas de Producto Bruto Interno

RESUMEN

Al observar una gran diferencia de preferencias de turistas al planificar su viaje en el cual el 74% de turistas que visitan al Perú organizan su viaje por cuenta propia comparado al 26% que organizan su viaje usando un paquete turístico prediseñado, se entiende que actualmente se tiene bastante información disponible en internet de los destinos, que la tarea de diseñar una ruta puede convertirse compleja por la cantidad de posibilidades a evaluar para un determinado número de atractivos y/o actividades, requiriendo tiempo y esfuerzo.

El problema de diseñar un viaje es estudiado en la literatura con el nombre del Problema del Agente Viajero el cual trata con un criterio, la minimización de un costo (que puede ser distancias, costo en dinero, tiempo, etc.) entre atractivos turísticos.

Al revisar los datos disponibles en la región de Puno se vio que aparte de las distancias también se podía recolectar datos de tiempos de duración, por lo tanto se vio la oportunidad implementar una solución a un modelo multiobjetivo considerando la distancia y el tiempo, una solución aproximada por medio de un algoritmo metaheurítico llamado NSGA-II.

Para lo cual se recolectaron 200 atractivos turísticos de un inventario publicado por Ministerio de Comercio Exterior y Turismo, datos de distancias y tiempos de duración en una matriz de 164x164 atractivos.

Seguidamente se implementó una solución aproximada para el modelo del Problema del Agente Viajero Multiobjetivo utilizando NSGA-II incluido en la herramienta de jMetal.

Finalmente se hizo una simulación del proceso de diseño de rutas, en la cual primero se selecciona los atractivos por medio de filtros, luego pasa por el algoritmo configurado que devuelve un conjunto de soluciones de la frontera de Pareto en la cual el usuario selecciona la ruta que desea. Finalmente puede visualizar la ruta seleccionada en un mapa.

Palabras clave: planificación de rutas turísticas, Problema del Agente Viajero Multiobjetivo, búsqueda informada, algoritmos genéticos, NSGA-II**.**

ABSTRACT

By observing a large difference in tourist preferences when planning your trip in which 74% of tourists visiting Peru organize their own trip compared to 26% who organize their trip using a pre-designed tour package, it is understood that currently It has enough information available on the internet of destinations, that the task of designing a route can become complex due to the amount of possibilities to be evaluated for a certain number of attractions and/or activities, requiring time and effort.

The problem of designing a trip is studied in the literature with the name of the Traveling Salesman Problem which deals with a criterion, the minimization of a cost (which can be distances, cost in money, time, etc.) between tourist attractions.

When reviewing the data available in the Puno region, it was seen that apart from the distances, time duration data could also be collected, therefore the opportunity was seen to implement a solution to a multiobjective model considering distance and time, a solution approximated by means of a metaheuritic algorithm called NSGA-II.

For which 200 tourist attractions were collected from an inventory published by the Ministry of Foreign Trade and Tourism, distance and duration data in a matrix of 164x164 attractions.

An approximate solution was then implemented for the Multiobjective Travel Agent Problem model using NSGA-II included in the jMetal tool.

Finally, a simulation of the route design process was made, in which first select the attractions through filters, then go through the configured algorithm that returns a set of Pareto border solutions in which the user selects the route that want. Finally you can view the selected route on a map.

Keywords: touristics route planning, Multiobjetive Traveling Salesman Problem, informed search, genetic algorithm, NSGA-II.

CAPÍTULO I. El problema

1.1. Descripción del problema

En el Perú, en el año 2017, turismo y viajes fue parte con el 4.6% en uno de los años con más crecimiento del Producto Bruto Interno (PBI), según el último reporte publicado por el Consejo Mundial de Viajes y Turismo (*Travel and tourism economic impact 2018 Perú*, 2018), generado por actividades económicas como el transporte de pasajeros, provisión de alimentos y bebidas, alojamiento, industria cultural, recreativa y deportiva, agencias de viajes, producción y comercio de artesanía, entre otros (*Medición económica del Turismo,* 2016).

En el 2018, el ingreso de divisas generado por el turismo receptivo mostró un incremento del 7,0% en comparación a lo reportado el año 2017. Creciendo por cuarto año consecutivo y consolidando al turismo como el tercer generador de divisas después de los sectores minero y agropecuario (*Reporte mensual de turismo de diciembre 2018,* 2018).

Según un reporte publicado por la Comisión de la Promoción de Perú para la Exportación y el Turismo (PROMPERU) "Perfil del Turista Extranjero 2017" se ha observado una gran diferencia en las preferencias del turista extranjero que visita el Perú para el año 2017 el 74% organiza su visita por cuenta propia y el 26% usa paquete turístico (*Perfil del Turista Extranjero*, 2017).

Un turista que desea visitar un destino sin depender de opciones pre-diseñadas lleva consigo la tarea de planificar por su propia cuenta su viaje, para ello en primer lugar una vez que tiene el destino busca información; en la actualidad se puede acceder a bastante información en la internet, en las que se pueden encontrar guías, en blogs, páginas de entidades del estado que brindan información de los destinos, los comentarios de otros respecto a lugar, actividad y hospedajes que desea visitar, en las que encontramos a Tripadvisor, poder elegir y reservar hospedajes en aplicaciones populares de Tripadvisor, Booking, Trivago y Airbnb, también herramientas como Google Maps en la cual se pueden visualizar formas de acceder al lugar, servicios cerca de un lugar específico, entre otra información gracias al crecimiento de la internet. Está tarea de buscar información puede convertirse tediosa por la gran cantidad de información disponible (Rodríguez Díaz, 2012).

En segundo lugar, realiza un plan que incluye una ruta de viaje dependiendo del detalle con el que desea planificar incluye por día las actividades que realizará, precios, información del lugar, reservas de hospedajes y transporte. La tarea principal de elegir una ruta puede convertirse en una tarea compleja por la cantidad de posibilidades, por lo tanto demande un alto costo en tiempo y esfuerzo para evaluar todas las alternativas.

Al buscar soluciones que devuelvan una solución confiable, encontramos que es posible modelar matemáticamente en la cual podemos tener una solución mejor, apoyándose del poder de las computadoras. Al intentar solucionar de forma exacta y rápida no es posible solucionar por la gran cantidad de variables (número de actividades a realizar en el destino) que se tiene en el momento de la planificación (Hernández & García, 2007).

El encontrar la mejor ruta posible reduciendo costos es uno de los problemas más estudiados en matemática computacional con nombre de Problema del Agente Viajero (TSP siglas en inglés) el cual es uno de los problemas que no se puede encontrar una solución con un coste computacional y tiempo razonable (Matai, Singh, & Mittal, 2010).

El problema se describe como "Dado un conjunto de ciudades y el costo del viaje (o distancia) entre cada par posible, el TSP, es encontrar la mejor manera posible de visitar todas las ciudades y regresar al punto de partida que minimice el costo del viaje (o la distancia de viaje)" (Matai et al., 2010).

El problema ha sido formulado en programación lineal en enteros de la siguiente forma por (Ahmia & Skoudarli, 2017):

Función objetivo:

$$min \sum_{\substack{i,j \\ i \neq j}} c_{ij} x_{ij}$$

Sujeto a:

$$\sum_{\substack{i \\ i \neq j}} x_{ij} = 1 \ \forall i \in N$$

$$\sum_{\substack{j \\ i \neq j}} x_{ij} = 1 \ \forall j \in N$$

$$u_i - u_j + n \times x_{ij} \leq n - 1 \ \forall i, j \in N - 1, i \neq j$$

Sea $N = \{1,2,3, \ldots n\}$ el conjunto de ciudades. Sea c_{ij} la distancia de i a la ciudad j. Sea x_{ij} igual a 1 si existe una ruta de i a la ciudad j y 0 en otro caso. La primera restricción garantiza que a cada ciudad llegue solo una ruta. La segunda restricción garantiza que de cada ciudad solo salga una ruta. La última restricción elimina los subtours, asegurándose de que solo exista una ruta que cubra todas las ciudades.

En la Figura *1* se puede observar una solución en la que el viajante inicia en la ciudad A y la ruta más corta es A-B-C-D-A, en la que sumado los costos es $7 + 5 + 9 + 8 = 29$, para evaluar todas las posibilidades y seleccionar cual es la menor existen $\frac{(n-1)!}{2}$ posibles combinaciones para el Problema de Agente Viajero, en la que el costo (que puede ser distancia, tiempo, costo, etc.) de ir de una ciudad a otra es igual a la inversa (el cual es denominado el Problema de Agente Viajero Simétrico).

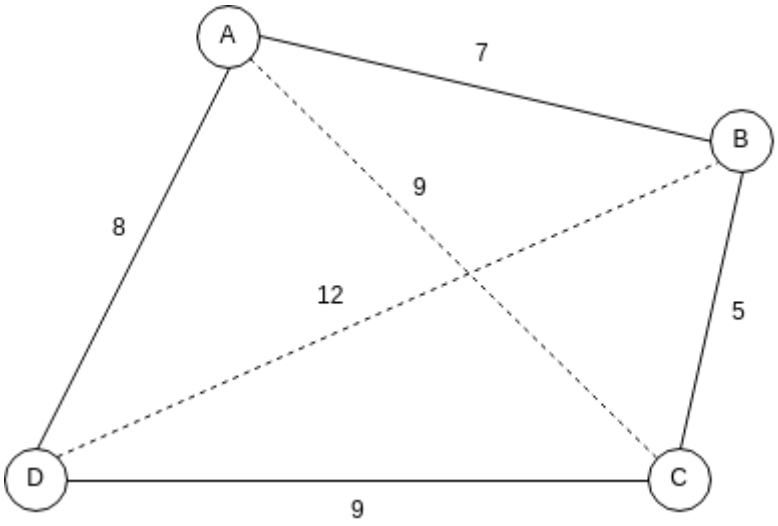


Figura 1. Una solución para el Problema del Agente Viajero Simétrico
Fuente: Elaboración propiawe

Al revisar los datos disponibles para ayudar al turista a diseñar su ruta de viaje se encontró el inventario de atractivos turísticos del MINCETUR, en la que contiene las ubicaciones, con estos datos es posible obtener las distancias no lineales (considerando las

distancias de los ejes viales) a través de las API´s de Google Maps, y es posible también obtener datos de duración de ir de un atractivo a otro, para aprovechar estos datos se considera una oportunidad el modelar una variante del Problema del Agente Viajero que considera más de una función objetivo e implementar para ayudar en la toma de decisiones al turista en la planificación de su viaje. Ya que existen diferentes formas de resolver el Problema del Agente Viajero, sean por métodos exactos o métodos heurísticos (métodos aproximados), en este trabajo de investigación se utilizará un algoritmo genético (método heurístico) y será explicado posteriormente.

1.2. Objetivos.

1.2.1. Objetivo general

Implementación de un modelo matemático para la planificación de un viaje personalizado basado en optimización multiobjetivo para los turistas en la región Puno.

1.2.2. Objetivos específicos

- Recolectar datos de ubicaciones, distancias y tiempos de duración de viaje de un atractivo turístico a otro.
- Aplicar el algoritmo NSGA-II para resolver el modelo de Problema del Agente Viajero Multiobjetivo.
- Implementar una interfaz donde se pueda simular la planificación de ruta de un viaje turistico.

1.3. Justificación de la investigación.

1.3.1. Social

El resultado de este trabajo de investigación apoyará directamente al turista en la toma de decisiones en el diseño de una ruta de viaje para realizar su viaje de forma más satisfactoria a la región de Puno, porque ayudará en la tarea de evaluar de entre varias posibilidades cual es la mejor ruta, en menos tiempo y sin mucho esfuerzo.

Apoyará a los ciudadanos, los emprendedores, el gobierno regional, el estado, de forma indirecta en crecimiento y desarrollo del país, como indicó la Ministra de Comercio Exterior y Turismo Magali Silva Velarde Álvarez “Estoy convencida de la importancia de

la actividad turística como elemento clave para el crecimiento y desarrollo del país, por los diversos efectos positivos en la generación de ingresos, empleo y mejora de la calidad de vida de la población.” en (*Memorial de Turismo Rural Comunitario en el Perú*, 2015).

1.3.2. Económico

Según el último reporte publicado del impacto en la economía de viajes y turismo para el Perú hasta el momento por el Consejo Mundial de Viajes y Turismo (*Travel and tourism economic impact 2018 Perú*, 2018), menciona que los viajes y el turismo es una actividad económica importante en la mayoría de los países del mundo, en el año 2017 la contribución directa de viajes y turismo al PIB fue de 3,8% del PBI. Esto refleja principalmente la actividad económica generada por industrias como hoteles, agencias de viajes, aerolíneas y otros servicios de transportes de pasajeros. También incluye, las actividades de las industrias de restaurantes y de ocio.

Se ha observado que Puno ha subido en la última década en pobreza de acuerdo a una clasificación en el cual Puno se ubicaba antes en el tercer grupo pasando al segundo grupo de departamentos con más pobreza y pobreza extrema según reporte de Informe Técnico publicado por el INEI (Instituto Nacional de Estadística e Informática): (*Evolución de la pobreza monetaria 2007 - 2018*, 2019; “Pobreza en el sur aumentó ligeramente | Sociedad - La República”, 2019), por ello es importante considerar que con esta investigación al ayudar en que tenga una alternativa que le puede brindar a que tenga una visita más satisfactoria para que pueda luego recomendar, de esa manera traer a más turistas. Ello ayudará en la reducción de la pobreza como lo menciona (“El turismo y la atenuación de la pobreza”, s/f) “El turismo, en muchos países en desarrollo y menos adelantados, es la opción de desarrollo económico más viable y sostenible y, en algunos de ellos, la principal fuente de entrada de divisas. Parte de estos ingresos revierte en diferentes grupos de la sociedad y, si el turismo se gestiona centrándose prioritariamente en la atenuación de la pobreza, puede beneficiar directamente a los grupos más pobres mediante el empleo de la población local en empresas turísticas, el suministro de bienes y servicios a los turistas, la gestión de pequeñas empresas y empresas comunitarias, etc., con el consecuente impacto positivo en la reducción de la pobreza.”

1.3.3. Ambiental

Directamente se considera que ayudará en que se reduzcan el uso de papeles en la guías, etc., al usar la aplicación web, considerado también por (*Plan Estratégico Nacional de Turismo 2025 (PENTUR)*, 2015) como una de las políticas de ambientales en el sector turismo promover la reducción del consumo de recursos, el reúso, el reciclaje y la eco eficiencia como estrategias de apoyo al control del deterioro ambiental.

Una de las formas de turismo dentro del grupo de turismo que pueden desarrollarse de forma sostenible, se tiene al Turismo Rural Comunitario que es plateado por (Agüera, 2013) como aplicación al turismo de los pilares del desarrollo sostenible, donde se encuentra el desarrollo socioeconómico y la conservación y protección hacia el medio ambiente, quien al hacer una revisión bibliográfica profunda concluye: "Así, el desarrollo del turismo comunitario en zonas desfavorecidas puede ayudar a generar recursos económicos en las poblaciones locales, contribuyendo así al desarrollo social y económico de la población y del destino. Además, esta forma de turismo puede generar una mayor concienciación de la comunidad local, fomentando aspectos como la conservación y respeto hacia la naturaleza y demás recursos existentes en el destino (cultura, patrimonio, etc.)."

Al mencionar a turismo sostenible también podemos como define la Organización Mundial de Turismo (OIM) como "un modelo de desarrollo económico concebido para mejorar la calidad de vida de la comunidad receptora, para facilitar al visitante una experiencia de alta calidad y mantener la calidad del medio ambiente, del que tanto la comunidad anfitriona como los visitantes dependen" citado en (Cabezas, 2006).

Según una publicación del Ministerio de Comercio Exterior y Turismo titulada "Consejos prácticos para el prestador turístico responsable" se tiene como uno de sus consejos: "Disponga de la información sobre las rutas y circuitos turísticos, conozca y recorra las rutas elegidas, las comunidades que visitara, su cultura y estilos de vida, previo a la visita ofrecida" ("Naciones Unidas declaró el 27 de setiembre como Año Internacional del Turismo Sostenible para el Desarrollo", 2017). Mencionado también, que es con el fin de generar un turismo con desarrollo sostenible el cual tiene como uno de sus pilares a la

parte ambiental, conservando y enriqueciendo el medio ambiente y luchando contra el cambio climático.

1.3.4. Ciencia

Se ha observado que hay escasez de investigaciones relacionadas con el tema de planificación de viajes personalizados en el Perú, de acuerdo al Repositorio Digital del CONCYTEC (Consejo Nacional de Ciencia y Tecnología e Innovación Tecnológica) llamado ALICIA (Acceso Libre a la Información Científica). Al publicar este trabajo de investigación se hará un aporte con el tema de rutas turísticas, solución al problema del Agente Viajero con Múltiples Objetivos usando la metaheurística de algoritmos genéticos, además de ser local por tratarse de la región de Puno también será un aporte a nivel país.

El problema que se aborda en este proyecto, denominado Problema del Agente Viajero Mutiobjetivo es una variante del Problema de Agente Viajero, se encuentra clasificado como un problema NP-Duro por su inherente complejidad por la teoría de la complejidad computacional, el cual estudia cómo crece el coste computacional principalmente en memoria y tiempo de resolver un determinado problema en relación a lo que crece el tamaño de dicho problema (de Cabezón Irigaray, 2017). Según el mismo autor estudiar este tipo de problemas es una cuestión muy importante en computación y con muchas aplicaciones.

Tiene su importancia el estudio del Problema del agente viajero porque ha llevado a mejorar los métodos de solución en muchas áreas de la optimización matemática (“The Problem”, 2015). En el misma publicación también se menciona que: “La simplicidad de la declaración del problema es engañosa: el TSP es uno de los problemas más estudiados en matemáticas computacionales y, sin embargo, no se conoce ningún método de solución eficaz para el caso general. De hecho, la resolución del TSP resolvería el problema P versus NP y obtendría un premio de 1000000 dólares del Clay Mathematics Institute.”

Como resultado de este trabajo de investigación, en la cual se pretende recolectar información del turista y recolectar información de los atractivos, está podrá ser usado para un posterior análisis para otros usos. Uno de los usos se considera, para una comprensión mejor del comportamiento de los turistas para así mejorar los servicios.

CAPÍTULO II. Revisión de literatura/Marco teórico

2.1. Revisión de la literatura

2.1.1. A New Evolutionary Multiobjective Model for Traveling Salesman Problem.

En este artículo hecha por (Chen et al., 2019) traducido al español como "Una nuevo modelo evolutivo multiobjetivo para el Problema del Agente Viajero" tiene el objetivo de: proponer un método mejorado para superar problemas convergencia prematura, diversidad insuficiente y la distribución no uniforme de soluciones, de los algoritmos genéticos al resolver el Problema de Agente Viajero con Objetivos Múltiples, basado en un nuevo modelo evolutivo computacional inspirado en Physarum (PCM).

Como métodos tiene, el presentar las formulaciones de los métodos basados en Algoritmos Genéticos para resolver el Problema del Agente Viajero Bi-objetivo (BTSP), luego proponer un nuevo algoritmo inspirado en Physarum y basado en el algoritmo NSGA-II, finalmente realizar algunos experimentos y discusiones para mostrar la efectividad del algoritmo propuesto.

Como resultados del trabajo presenta que el nuevo algoritmo denominado pNSGA-II (Physarum-Inspired Non-dominated Sorting Genetic Algorithm II), el cual tiene un mejor rendimiento al comparar con NSGA-II y otros algoritmos, al realizar experimentos con varias instancias BTSP.

2.1.2. Un enfoque bi-objetivo al problema del reparador

En esta tesis hecha por (Arellano Arriaga, 2014), tiene por objetivo de su trabajo estudiar un problema bi-objetivo de juntar el Problema del Agente Viajero y el Problema de Mínima Latencia no estudiado en la literatura y proponer de un método de solución basado en un algoritmo genético elitista.

Utilizó los siguientes métodos:

- Revisión bibliográfica donde se aborden problemas bi-objetivo que involucren alguno de los criterios considerados en el problema bajo estudio, así como el estudio de las metodologías aplicadas en problemas bi-objetivo.
- Formulación matemática del problema bajo estudio.

- Desarrollo e implementación computacional de una metodología de solución para el problema bajo estudio. Análisis de los resultados obtenidos en la experimentación computacional.

Teniendo como resultados: una propuesta de un modelo bi-objetivo lineal entero, en la que logró obtener para 20 clientes con un método exacto, no pudiendo más por más porque el tiempo computacional se incrementaba desmesuradamente. La implementación final del algoritmo NSGA-II fue capaz de brindar soluciones con buena aproximación a las soluciones exactas.

2.1.3. Sistema de ayuda al turista - Modelo para la planificación de un viaje personalizado.

En este trabajo de investigación realizada por (Rodríguez Díaz & Caballero Fernández, 2012), tuvo el objetivo de desarrollar una herramienta que proporciona a cada turista un itinerario lo más adecuado posible a sus necesidades, que incluye las distintas actividades que puede realizar en un horario establecido.

Utilizó el método multicriterio, que considera la conflictividad entre sus objetivos, y tiene en cuenta las preferencias de los turistas, así como las características del entorno. Los métodos para sus objetivos fueron:

- Formular un modelo matemático multiobjetivo (minimizar el coste de transporte, minimizar el coste de las actividades, maximizar la utilidad que reportan las actividades al turista, ajustar el tiempo dedicado a cada tipo de visita durante el tour).

- Resolver el modelo matemático utilizando metaheurística Búsqueda Tabú. Recolectar los datos (localización, distancias, precio, valoración, duración en cada actividad, horario de apertura y cierre, tiempos de desplazamiento de una actividad a otra) e incorporar en una base de datos.

- Desarrollar una interfaz (aplicación de escritorio) para recoger información del turista (preferencias) a través de formularios, y devolver la solución óptima.

Los resultados fueron: La construcción de una aplicación de escritorio de ayuda para el turista, y demostró su utilidad aplicando en un caso de Andalucía (España) donde incorporó en la base de datos un total de 2.154 atractivos turísticos (548 alojamientos, 296 de restantes y 1.206 visitas turísticas).

2.1.4. Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú.

En esta tesis publicada por (Mendoza Apaza, 2017), tuvo el objetivo de Diseñar e implementar un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú teniendo en cuenta las preferencias de los turistas. El trabajo realizado tomó el enfoque de inteligencia artificial.

Los métodos para sus objetivos fueron:

- Diseñar el algoritmo de heurística de construcción basada en el método el Vecino más Cercano.
- Diseñar un algoritmo de heurística de mejoría basada en el Procedimiento de 2 intercambios (2-opt).
- Implementar el algoritmo de metaheurística Búsqueda Tabú considerando como parámetros de entrada a la solución obtenida en la heurística de mejoría.
- Validar la solución del algoritmo implementado de Búsqueda Tabú con instancias artificiales encontrado en librerías de dominio público que contenga la mejor solución obtenida hasta la fecha y/o óptima, y datos reales del distrito de Juli (Región Puno) recolectados a través de cuestionarios en las que se consideraron datos de distancias, costos, tiempo, que se tiene al ir de un recurso turístico o emprendimiento a otro recurso.
- Desarrollar una aplicación web que integre el algoritmo de Búsqueda Tabú, usando la metodología de desarrollo Ágil Programación Extrema (XP), en la que se consideró también el uso de casos de uso y prototipos.

Como resultados validó el algoritmo implementado de Búsqueda Tabú con 5 instancias artificiales, con las que al comparar logró igualar al 60 % de las mejores soluciones

encontradas hasta la fecha en la literatura, y se validó también con la información de 13 emprendimientos y 7 recursos turísticos del distrito de Juli, en las que obtuvo un ciclo hamiltoniano proponiendo un circuito turístico a medida para el turista, que inicia y termina en un mismo lugar y minimiza el costo del recorrido. La aplicación web se publicó en un nuevo módulo (ruta inteligente) de la plataforma desarrollada en el proyecto patrocinado por el CONCYTEC y la UPeU "Plataforma digital inteligente y Big Data para el turismo rural comunitario en la Región Puno". La aplicación permite que las preferencias puedan ser editadas, eligiendo priorizar entre el costo, distancia y tiempo.

2.1.5. Single-objective and bi-objetive parallel heuristics for the travel planning problema.

En esta tesis publicada por (Beirigo, 2016), tuvo como objetivo de su trabajo: dado un conjunto de destinos, sus tiempos mínimos de residencia y datos de viaje realistas que cubren el transporte y los hoteles, su objetivo es determinar: El mejor, o cerca del mejor itinerario de viaje con respecto al costo global del viaje (en menos de 1 minuto), y el mejor o el cerca del mejor conjunto de Pareto de itinerarios de viaje con respecto al costo global del viaje y tiempo de espera (en menos de 1 minuto). Utiliza el método de una heurística de Búsqueda Local Iterada (ILS) para resolver solo un objetivo y el Algoritmo Genético de Clasificación No Nominado II (NSGA-II) para el enfoque bi-objetivo.

Los objetivos y los métodos fueron:

- Recopilar y almacenar datos de viajes reales relacionados con vuelos y alojamiento. Diseñar una red de viajes para manejar los datos recogidos.
- Diseñar una heurística para resolver rápidamente las versiones de un solo objetivo y de dos objetivos del problema de planificación de viajes.
- Desarrollar enfoques exactos para evaluar el rendimiento de ambos métodos heurísticos.

Como resultados tuvo para 285 casos de prueba de un solo objetivo, la heurística ILS pudo alcanzar soluciones en promedio menos del 4.1% divergentes de una implementación exacta, además de alcanzar la solución óptima en aproximadamente el 30% de los casos de prueba. A su vez, para 180 casos de prueba bi-objetivos, la implementación NSGA-II pudo

alcanzar una solución aproximada en promedio hasta un 8% divergente de una implementación exacta.

2.2. Marco teórico

2.2.1. Inteligencia Artificial

Definido por John McCarthy en 1956 como “la ciencia y la ingeniería de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”(“Artificial Intelligence Overview”, s/f). También se puede definir como la rama de la ciencia de la computación que se ocupa de la automatización del comportamiento inteligente (Luger, 2009).

La investigación en inteligencia artificial (IA) se ha centrado principalmente en los siguientes componentes de la inteligencia según (“Artificial Intelligence Overview”, s/f; Copeland, 2018):

- Aprendizaje
- Razonamiento
- Resolución de problemas
- Percepción
- Uso del lenguaje

La resolución de problemas, particularmente en inteligencia artificial, puede caracterizarse como una búsqueda sistemática a través de un rango de posibles acciones para alcanzar una meta o solución predefinida (Copeland, 2018).

2.2.1.1. Búsqueda en inteligencia artificial.

Se puede decir que casi todos los programas de inteligencia artificial están realizando algún tipo de solución de problemas, ya sea interpretando una escena visual, analizando una oración o planificando una secuencia de acciones de robots. La búsqueda es uno de los problemas centrales en los sistemas de resolución de problemas. Se vuelve así cada vez que el sistema, a través de la falta de conocimiento, se enfrenta a una elección de una serie

de alternativas, donde cada elección lleva a la necesidad de realizar más elecciones, y así sucesivamente hasta que se resuelva el problema (Thornton, 1992).

Cuando el número de posibilidades es pequeño, el programa puede realizar un análisis exhaustivo de todas ellas y luego elegir las mejores. En general, sin embargo, los métodos exhaustivos no serán posibles y se deberá tomar una decisión en cada punto de elección para examinar solo un número limitado de las alternativas más prometedoras. En el ajedrez, por ejemplo, hay demasiadas posibilidades para que un programa (o una persona) imagine cada movimiento posible, cada una de las posibles respuestas correspondientes, movimientos adicionales en respuesta y así sucesivamente a la conclusión anticipada del juego. Pero decidir lo que cuenta como "prometedor" a veces es muy difícil. Un buen jugador de ajedrez es bueno precisamente porque, entre otras habilidades, él o ella tienen buen ojo para encontrar formas plausibles de proceder y puede concentrarse en ellas. Si bien es fácil diseñar programas de resolución de problemas que sean buenos para realizar un seguimiento de las elecciones realizadas y las que aún no se han explorado, es difícil proporcionar un programa con el sentido común que pueda atravesar un abanico de posibilidades para concentrar su principal Análisis sobre el pequeño número de elecciones críticas (Thornton, 1992).

2.2.1.1.1. Búsqueda no informada.

También llamada búsqueda ciega, este tipo de búsquedas no tienen información adicional acerca de los estados más allá de la que proporciona la definición del problema. Esto significa que no utiliza ninguna información que lo ayude a alcanzar la meta, como la cercanía o la ubicación de la misma. Las estrategias o algoritmos, utilizando esta forma de búsqueda, ignoran a dónde van hasta que encuentran una meta y reportan el éxito. Es una clase de algoritmos de búsqueda de propósito general que operan de manera bruta (Russell & Norvig, 2004; "Search Algorithms in Artificial Intelligence", s/f; "Search techniques", 2018).

2.2.1.1.2. Búsqueda informada.

Este tipo de búsqueda utiliza la información del problema y el costo del estado actual al objetivo. Generalmente utiliza una función heurística que estima qué tan cerca está un estado del objetivo. Esta heurística no necesita ser perfecta. Esta función se utiliza para

estimar el costo de un estado al objetivo más cercano (“Search Algorithms in Artificial Intelligence”, s/f).

2.2.2. Heurísticas

“Se califica de heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación. Se usa el calificativo heurístico en contraposición a exacto (Melián, Belén. Perez, Jose A. et al)” (Riojas Cañari, 2005).

2.2.3. Metaheurísticas

La aparición de un nuevo tipo de método, llamado metaheurística se han desarrollado desde 1980 con un objetivo común: resolver problemas difíciles de optimización de la mejor manera posible (Collette & Siarry, 2003). Tienen en común, además, las siguientes características:

- Son, al menos en parte, estocásticas: este enfoque puede manejar la explosión combinatoria de posibilidades.
- Sus orígenes son combinatorios: tienen la ventaja, crucial en el caso continuo, de ser directos, lo que significa que no necesitan computar los derivados de la función objetivo.
- Están inspirados en analogías: con la física (recocido simulado, difusión simulada, etc.), con la biología (algoritmos genéticos, búsqueda de tabú, etc.) o con la etología (colonias de hormigas y métodos de enjambre de partículas, etc.).
- Pueden guiar, en una tarea particular, otro método de búsqueda especializado (por ejemplo, otro método heurístico o de exploración local). Comparten los mismos inconvenientes: dificultades para ajustar los parámetros del método y el alto tiempo de cálculo.

Se define como: “Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización

combinatoria, en los que los heurísticos clásicos no son efectivos. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos (Osman y Kelly,1995)" (Martí, s/f).

2.2.4. Algoritmos genéticos

Es una metaheuristica utilizado para resolver problemas de búsqueda y optimización, fue creado para encontrar una explicación de los modelos de comportamiento de la teoría de la evolución propuesto por Charles Darwin en su libro "El origen de las especies", mediante la optimización (Barrera Núñez, 2006).

Un algoritmo genético se basa en la idea de la selección natural de la teoría de la evolución, en la que utiliza la naturaleza para que los individuos más aptos de una población sobrevivan, debido a su capacidad para adaptarse más fácilmente a los cambios que se producen en su entorno. Estos cambios que se efectúan en los genes de un individuo y cuyos atributos son transmitidos a sus descendientes cuando éstos se reproducen sexualmente (Barrera Núñez, 2006).

Los algoritmos genéticos adoptan terminología usada en la teoría de evolución de natural y la biología genética. Un individuo de la población puede ser formado por uno o más cromosomas. Sabiendo que un cromosoma se encuentra en el centro (núcleo) de una célula, la cual contiene la cadena del ácido desoxirribonucleico (ADN) y los genes son pequeñas piezas del ADN que transportan información genética específica, como se puede observar en la Figura 2.

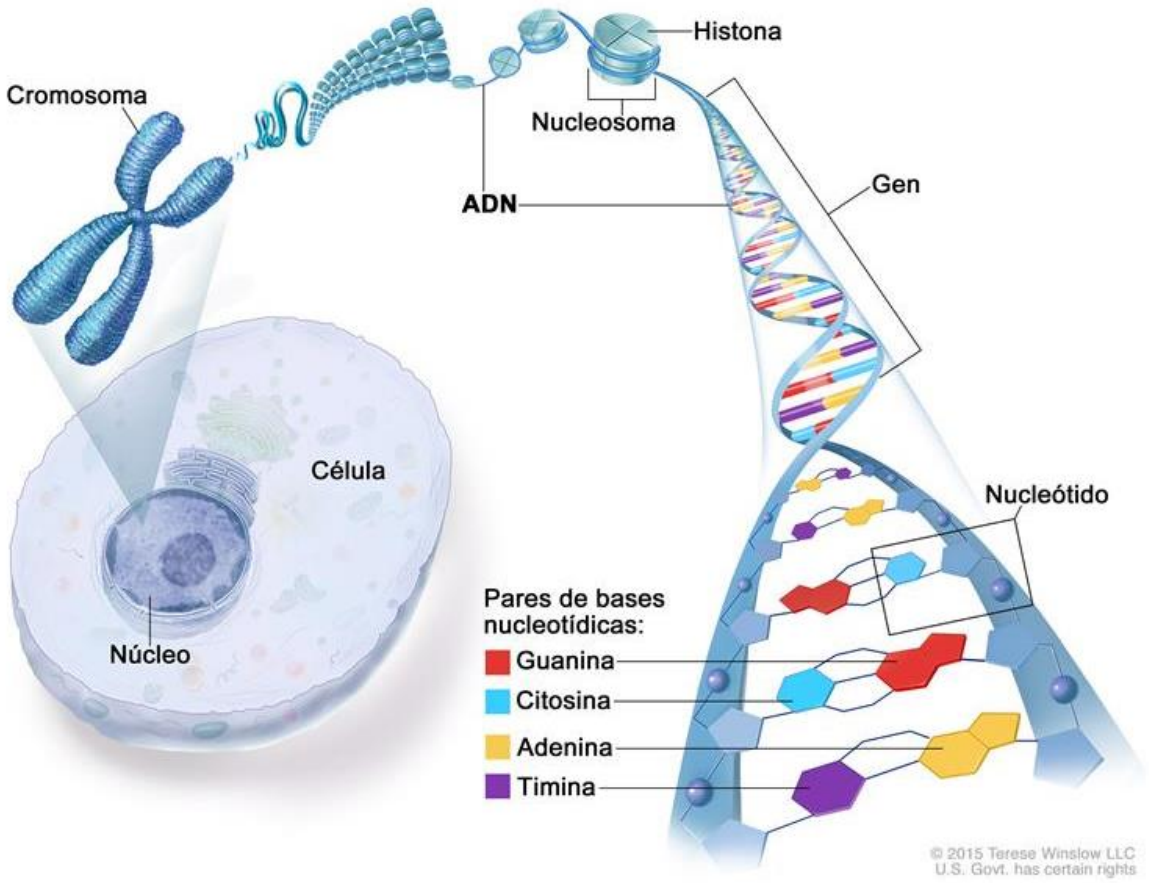


Figura 2. Relación entre célula, cromosoma y gen
Fuente: https://www.cancer.gov/espanol/publicaciones/diccionario/def/gen

Cuando este individuo es representado por un único cromosoma, se puede utilizar el término individuo o cromosoma indistintamente. Los cromosomas, en general son implementados en forma de vectores (secuencia de símbolos o algún tipo de cadena), donde cada componente del vector es conocido como gen. Los posibles valores de un determinado gen son denominados como alelos, entendido en genética que un ser vivo hereda dos alelos para cada gen uno del padre y el otro de la madre mediante la reproducción sexual (ver Figura 3), y están ubicadas en la misma posición dentro de los cromosomas homólogos, ver Figura 4.

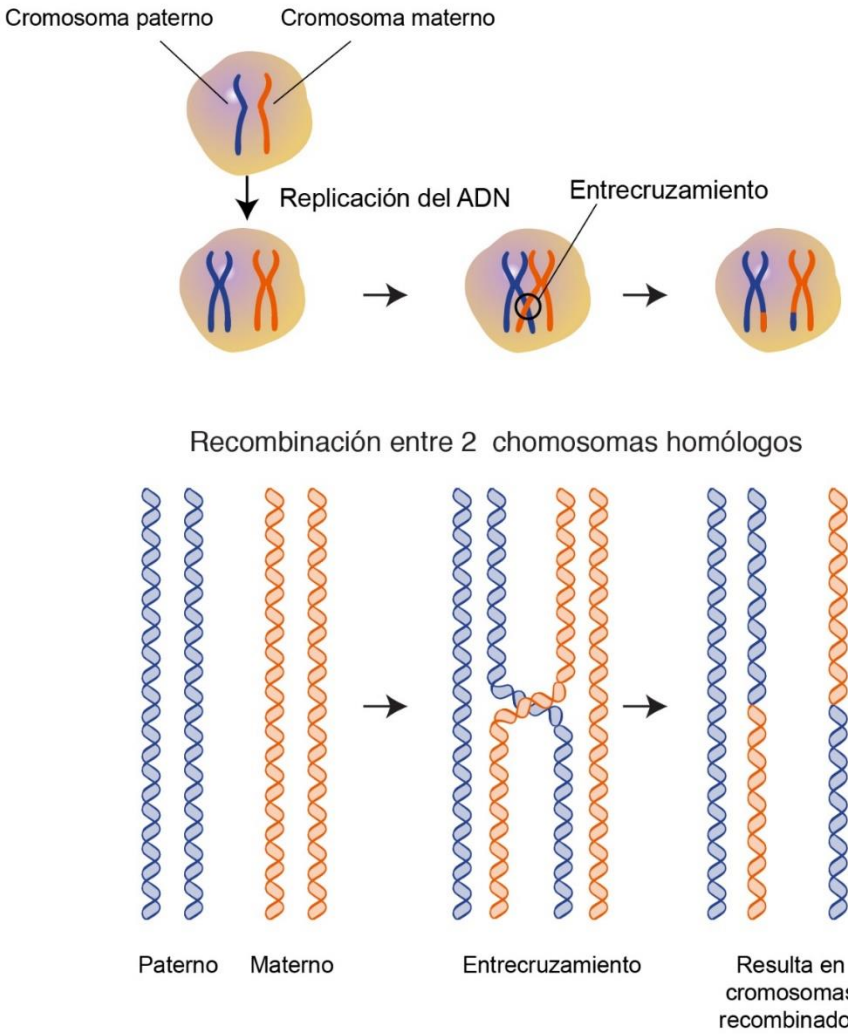


Figura 3. Recombinación en el proceso de la reproducción de los seres vivos.
Fuente: https://www.genome.gov/es/genetics-glossary/Recombinacion-homologa

Figura 4. Relación de locus y alelo.
Fuente: https://www.ck12.org/na/las-leyes-de-mendel-y-la-gen%C3%A9tica-1/lesson/Las-leyes-de-Mendel-y-la-gen%C3%A9tica/

Cada gen posee una ubicación fija en un cromosoma, denominado locus. El conjunto compuesto por cromosomas, genes y alelos se denomina genotipo y las características conferidas por esta forman un fenotipo, ver ***Tabla 1*** (Barboza, 2005).

Tabla 1.
Relación entre terminología de algoritmos genéticos con las biología, recuperado de (Barboza, 2005)

Biología	**Algoritmo genético**
Cromosoma	Individuo
Gen	Bit
Alelo	Valor de bit
Locus	Posición de un bit específico en un individuo
Genotipo	Individuo candidato a la solución – x
Fenotipo	Valor de función para un individuo dado – $f(x)$

Una definición para algoritmos genéticos se tiene como "un método de búsqueda computacional basado en mecanismos de evolución natural y genética. En un algoritmo genético, una población de posibles soluciones a un problema dado evoluciona de acuerdo con operadores probabilísticos concebidos a partir de conceptos biológicos, de modo que hay una tendencia que los individuos representan soluciones cada vez mejores a medida que avanza el proceso." (Costa et al.,2007) citado en (de Pinho, Montevechi, Marins, & de Carvalho Miranda, 2013).

El proceso de un algoritmo genético simple se tiene de la siguiente forma según (Chu, 1997).

Algoritmo 1. Algoritmo genético simple

```
generar una población inicial;
evaluar la aptitud de los individuos en la población;
repetir
    seleccionar los padres de la población;
    combinar padres para producir hijos;
    evaluar la aptitud de los niños;
    reemplazar parte o toda la población por los niños;
hasta que se cumpla el criterio de parada;
```

Un ejemplo en términos de un problema de optimización matemática sería hallar $(x_i, \dots, x_n)$ tales que $f(x_i, \dots, x_n)$ sea máximo. En un algoritmo genético, tras parametrizar el problema en una serie de variables, $(x_i, \dots, x_n)$ se codifican en un cromosoma (Merelo Guervós, s/f).

A continuación se presentará los principales componentes, conceptos y los fundamentos sobre como interfieren sus parámetros en el resultado final de los algoritmos genéticos canónicos o simples.

2.2.4.1. Codificación o representación genética de las soluciones del problema.

La codificación en algoritmos genéticos es un procedimiento que transfiere cada posible solución del problema de optimización a algún tipo de cadena como un cromosoma, para que pueda ser comprendida por las computadoras. Cuanto más adecuada al problema está la representación, mejor la calidad de los resultados obtenidos (de Pinho et al., 2013; Xu, 2015).

Existen numerosas formas de representar variables, como: binarios, enteros o números reales. La mayor parte del trabajo desarrollado utiliza codificación binaria, donde cada cromosoma es un vector compuesto de ceros y unos (Castro, 2001).

La representación binaria para un individuo es históricamente importante, ya que fue utilizado en los primeros trabajos pioneros, siendo fácil de utilizar y manipular, también simple para analizar teóricamente (Moreira Silva, 2005).

Una representación real presenta ventajas sobre una binaria respecto a la velocidad de procesamiento ya que constantemente realiza la conversión entre valores la función y binarios (Castro, 2001; de Pinho et al., 2013).

También, si un problema tiene parámetros continuos y desea trabajar con mayor precisión, probablemente terminará usando cromosomas largos para representar soluciones, que requieren una gran cantidad de memoria (Moreira Silva, 2005).

Para Soares (1997) citado por (de Pinho et al., 2013) menciona que la representación en enteros es la más indicada para problemas que involucran números enteros, como análisis combinatoria, arreglos y permutaciones.

2.2.4.2. Generación de la población inicial.

La población inicial se puede obtener de dos maneras. En el primero de estos individuos se generan aleatoriamente, mientras que en la segunda opción, la población se genera por

una heurística relacionada con las características específicas del problema (de Pinho et al., 2013).

Reeves (1995) citado por (de Pinho et al., 2013) menciona que al dirigir la población inicial con la ayuda de algunas heurísticas, el algoritmo genético puede alcanzar soluciones mejores y más rápidas en comparación con el proceso de generación aleatoria. Por otro lado, puede ocurrir un proceso de convergencia prematura a puntos máximos locales, lo que no es conveniente para resolver el problema. Es importante recordar que la población inicial debe cubrir la mayor parte del espacio de búsqueda de soluciones.

2.2.4.3. Función de evaluación.

Es una función de tipo particular que evalúa el mérito (valor de aptitud) de cada individuo. Se obtiene evaluando el cromosoma a través de la función a optimizar (Barboza, 2005; Xu, 2015).

La aptitud puede definirse como la cuantificación de la adaptación del individuo, es decir, el valor obtenido con la aplicación de este a la función de costo. Si el objetivo es maximizar, del valor de aptitud es directamente proporcional al valor de la función. Si el objetivo es la minimización, el valor de aptitud es inversamente proporcional al valor de la función (Barboza, 2005).

En problemas de optimización sin restricciones, el valor de aptitud de un individuo puede corresponder al valor de la función objetivo. En problemas de optimización con restricciones, el enfoque más común es utilizar la función de aptitud asociada con una función de penalización (Barboza, 2005).

Suele usarse también para analizar la convergencia y/o continuidad del proceso. Cuando el grado de adaptación es aceptable, el mejor individuo de esta generación probablemente será la solución deseada (Barboza, 2005).

2.2.4.4. Selección.

La selección en algoritmos genéticos es un mecanismo que ayuda a seleccionar los cromosomas parentales adecuados para generar la próxima generación (Xu, 2015). Se utiliza para dirigir el proceso a las mejores regiones del espacio de búsqueda. Su función es

seleccionar individuos de la población para la reproducción, dando preferencia a individuos más adaptados al medio ambiente (Barboza, 2005).

Según (Gestal, Rivero, Rabuñal, Dorado, & Pazos, 2010) “elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda”.

Entre las múltiples formas de selección comúnmente utilizadas, presentaremos algunas a continuación:

2.2.4.4.1. Ruleta simple o selección proporcional.

El método de la ruleta es el método de selección más simple y también el más utilizado. Los individuos de una generación se seleccionan para la próxima generación usando la ruleta, similar a la ruleta utilizada en los juegos de azar (ver Figura 5). (Moreira Silva, 2005)

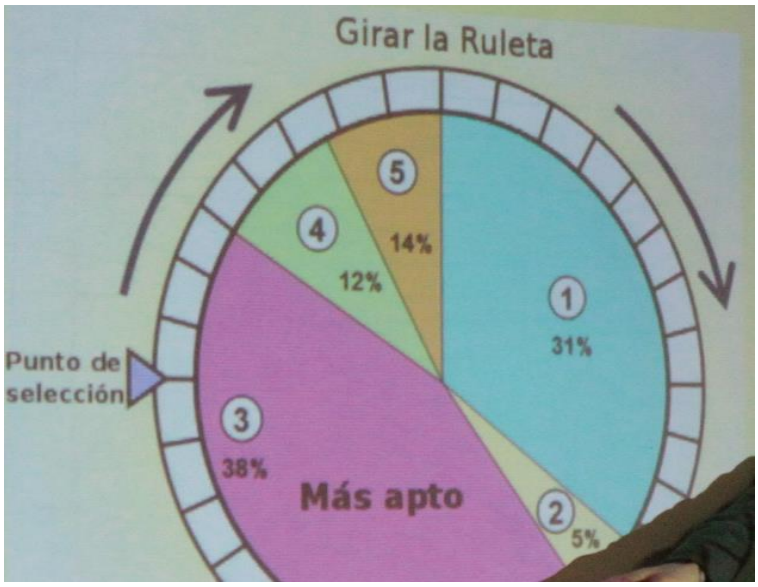


Figura 5. Selección por ruleta.
Fuente: https://www.frba.utn.edu.ar/jornadas-de-impulso-a-la-investigacion-en-el-departamento-de-ensenanza-de-ciencias-basicas/

En el método de la ruleta, cada individuo tiene su valor de aptitud física representado en proporción al valor total de las habilidades en la población. Por lo tanto, la rueda de la

ruleta se hace girar y los individuos seleccionados por el sorteo. Aquellos con la mayor participación serán más propensos a ser seleccionados. Cada vez que se hace girar la ruleta, se selecciona un nuevo individuo (de Pinho et al., 2013).

El método de la ruleta tiene la desventaja de tener una gran variación, lo que puede conducir a una gran cantidad de copias de un cromosoma con un alto valor de aptitud, lo que reduce la diversidad de la población. Esta falla puede conducir a una convergencia prematura del algoritmo a una solución local. Por otro lado, cuando la evolución avanza, hay un estancamiento del algoritmo, es decir, una baja presión de selección entre los individuos (Barboza, 2005).

2.2.4.4.2. Selección por torneo.

La selección de torneos es un método de selección ampliamente utilizado, principalmente debido a su eficiencia y simplicidad en su implementación. Este método preserva la diversidad de la población, ya que la elección se realiza comparando el valor de la función de evaluación entre los individuos que participan en el torneo (de Pinho et al., 2013).

En la selección de torneos hay un parámetro llamado tamaño del torneo (K). Este parámetro define cuántos individuos serán seleccionados al azar dentro de la población para competir. Una vez que se definen los competidores, se selecciona el que tiene la mejor aptitud. Sin embargo, cuanto mayor es el tamaño del torneo, mayor es la pérdida de diversidad (de Pinho et al., 2013).

2.2.4.4.3. Selección elitista

Este tipo de selección generalmente se combina con otros métodos de selección. Consiste en copiar o reproducir los mejores individuos de la población actual a la próxima generación, asegurando que estos cromosomas no se destruyan en los pasos de recombinación y mutación (Barboza, 2005).

La principal ventaja de este método es que garantiza la convergencia, es decir, si se descubre el óptimo global durante el proceso de búsqueda, el algoritmo genético debe converger a dicha solución (Castro, 2001).

A continuación se presentarán los operadores genéticos más comúnmente utilizados para manipular individuos seleccionados de una generación anterior para obtener nuevos individuos, básicamente son de dos tipos: recombinación o cruce y mutación (Barboza, 2005).

2.2.4.5. Cruzamiento o recombinación.

Se entiende como un operador aquel mecanismo que intercambia aleatoriamente las mismas cadenas de longitud en cada par de cromosomas seleccionados (Xu, 2015). Según (de Pinho et al., 2013) es uno de los operadores mas importantes.

Es responsable de la propagación de las características de los individuos más aptos (padres) mediante el intercambio de segmentos de información entre ellos, lo que dará lugar a nuevos individuos (Barboza, 2005).

Existen varios tipos de cruzamiento y estos difieren entre sí al elegir el locus del cromosoma que se intercambiará entre los cromosomas originales y la forma en que se efectúa este intercambio. Los tipos de recombinación o cruce descritos en la literatura varían según la codificación cromosómica. A continuación tratamos con solo un tipo de codificación según (Barboza, 2005):

Para una codificación binaria se tiene: cruzamiento en un punto, cruzamiento en dos puntos, en las cuales se eligen los puntos de corte al azar y desde estos puntos se intercambiará la información genética de los padres. Todo el material genético original entre los dos límites se intercambia y el resto permanece sin cambios, y el cruzamiento uniforme consiste en emparejar los dos cromosomas principales y cada locus cromosómico tiene un 50% de posibilidades de ser reemplazado, ver Figura 6.

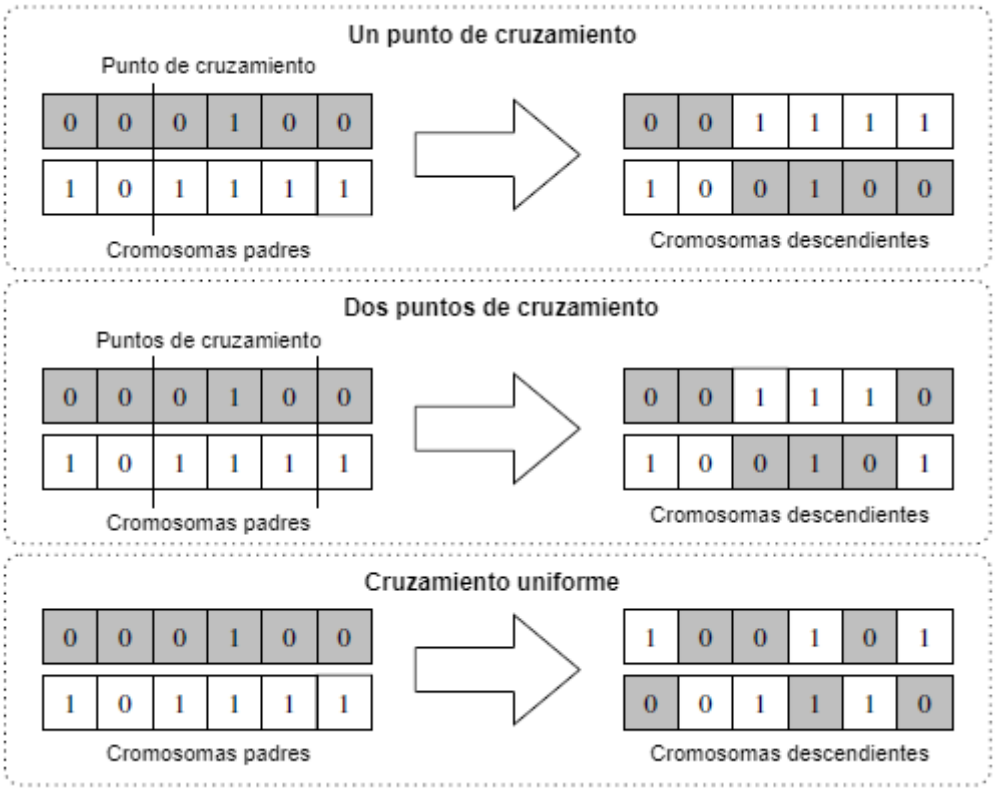


Figura 6. Métodos de cruce de un punto, dos puntos y uniformes.
Fuente: http://providing.blogspot.com/2015/06/genetic-algorithms-crossover.html

Según Jong (1975) citado en (Moujahid, Inza, & Larrañaga, s/f), "el comportamiento del operador de cruce basado en múltiples puntos, concluyendo que el cruce basado en dos puntos, representaba una mejora mientras que añadir más puntos de cruce no beneficiaba el comportamiento del algoritmo. La ventaja de tener más de un punto de cruce radica en que el espacio de búsqueda puede ser explorado más fácilmente, siendo la principal desventaja el hecho de aumentar la probabilidad de ruptura de buenos esquemas."

2.2.4.6. Mutación.

La mutación es un mecanismo que altera uno o más valores genéticos en un cromosoma de su estado inicial a cualquier otro estado al azar (Xu, 2015). Sin embargo, la mutación reintroduce la diversidad genética en la población y ayuda al algoritmo a escapar de los máximos o mínimos locales (de Pinho et al., 2013).

(Moujahid et al., s/f) menciona que el operador de mutación va ganando en importancia a medida que la población de individuos va convergiendo, y que algunos autores han

obtenido mejores resultados experimentales modificando la probabilidad de mutación a medida que aumenta el número de iteraciones.

Entre los operadores de mutación de codificación binaria, el más utilizado es la mutación binaria simple. En este tipo de mutación, se dibujan las posiciones del individuo y se invierten los genes correspondientes, es decir, si el valor del gen es 1, entonces este valor se cambia a 0 y viceversa, ver Figura 7 (de Pinho et al., 2013).

Figura 7. Ejemplo de mutación binaria simple.
Fuente: (de Pinho et al., 2013)

2.2.4.7. Reemplazo.

Mecanismo que se aplica cuando se hayan generado los descendientes, los descendientes reemplazarán a los miembros de la población existente para mantener constante el tamaño de la población, y el ciclo reproductivo se reiniciará. Actualmente se utilizan dos técnicas para mantener la población: el generacional y el estado estacionario (Chu, 1997).

- Algoritmo genético generacional: toda la población (de padres) es sustituida por nuevos individuos (de hijos) generados por el proceso de selección y la aplicación de los operadores genéticos. Cabe señalar que no hay coexistencia, por lo que se pierden buenos individuos en el proceso. Por esta razón, especialmente en problemas de optimización, un procedimiento utilizado con frecuencia es el elitismo, es decir, se preservan los mejores individuos de una generación (Castro, 2001).

- Algoritmo genético estado estacionario: en este otro extremo del algoritmo, solo se crea un individuo a la vez, y después de su evaluación se insertará en la población en lugar de algún otro elemento, por ejemplo, el peor de todos. Si es peor que todo lo existente, entonces nada cambia y se produce una nueva

creación individual. Para facilitar la comparación del individuo generado con los individuos ya existentes en la población, se utiliza un orden dentro de la población, por lo tanto, el individuo generado se compara solo con el último individuo del pedido, si es superior a él, asumirá posición correspondiente en el ordenamiento, siendo eliminado este último por selección natural (Castro, 2001).

2.2.4.8. Criterio de parada.

Según Hicks (2006) citado por (de Pinho et al., 2013), los criterios de parada son enfoques que delimitan el número de búsquedas: después de que se complete un cierto número de generaciones, período de tiempo máximo o cuando una población no mejora en generaciones sucesivas (convergencia de la población).

2.2.4.9. Parámetros de los algoritmos genéticos.

El rendimiento de un Algoritmo Genético está fuertemente influenciado por la definición de los parámetros que se utilizarán, por lo que es importante analizar cómo algunos parámetros influyen en el comportamiento de los algoritmos genéticos, para que puedan establecerse de acuerdo con las necesidades del problema y recursos disponibles (Moreira Silva, 2005).

2.2.4.9.1. Tamaño de población.

El tamaño de la población afecta el rendimiento general y la eficiencia de los algoritmos genéticos. Con una población pequeña, el rendimiento puede disminuir ya que esto proporciona una pequeña cobertura del espacio de búsqueda de problemas. Una población grande generalmente proporciona una cobertura representativa del dominio del problema y evita convergencias prematuras a soluciones locales en lugar de globales. Sin embargo, trabajar con grandes poblaciones requiere más recursos computacionales o el algoritmo para funcionar durante un período de tiempo mucho más largo (Moreira Silva, 2005).

En cuanto a cuál es el mejor tamaño de la población se tiene por algunos autores como (Castro, 2001) relacionar el tamaño de la población con el tamaño del cromosoma, es decir, cuanto mayor sea el cromosoma, mayor será el tamaño de la población para una diversidad razonable.

Y Alander(1992) citado por (de Pinho et al., 2013) basándose en evidencia empírica sugiere que un tamaño de población comprendida entre l (tamaño del cromosoma) y $2l$ es suficiente para atacar con exito los problemas por el considerados.

2.2.4.9.2. Número de generaciones.

(Barboza, 2005) indica que el número de generaciones depende de la complejidad del problema de optimización y debe determinarse experimentalmente.

En cuanto al número ideal se tiene según Goldberg (1989) citado por (Barboza, 2005) considera la desviación estándar de los valores de aptitud de los individuos de la población como uno de los parámetros más utilizados. Por lo tanto, hay una comparación del rendimiento de la generación actual con la generación anterior y si la desviación estándar es igual o menor que la establecida como una aproximación aceptable, el proceso de búsqueda finaliza. En tales casos, el número de generaciones no se proporciona al comienzo del proceso y solo se conoce cuando se cumple el criterio de detención.

2.2.4.9.3. Tasa de cruzamiento.

Este parámetro indica a qué velocidad o probabilidad se producirán cruces entre individuos seleccionados de la población. Para hacerlo, generamos un número aleatorio en el intervalo $[0,1]$ y lo comparamos con la tasa. Si el número es menor que la tasa, se realiza el cruce. Se tiene que en general, la tasa de cruce oscila entre 0,5 y 0,95 (Barboza, 2005; Castro, 2001).

(Barboza, 2005) indica que se puede observar experimentalmente que cuanto mayor sea esta tasa, más rápido se introducirán nuevos individuos en la población. Pero si es demasiado alto, la mayoría de la población será reemplazada y puede ocurrir la pérdida de estructuras de buena forma física. Con un valor demasiado bajo, el algoritmo puede volverse muy lento.

2.2.4.9.4. Tasa de mutación.

Esta tasa determina si los genes cromosómicos en la población seleccionada mutarán o no. Para la toma de decisiones, se genera un número aleatorio en el rango $[0,1]$ para cada uno de los genes de todos los cromosomas y se compara con la tasa de mutación. Si este número es menor que la tasa, el gen se modificará de acuerdo con el operador elegido. Con

respecto a los valores de la tasa de mutación, se puede observar que una baja tasa de mutación evita que una posición determinada se estanque en un valor y le permite alcanzar cualquier punto en el espacio de búsqueda. A un ritmo muy alto, la búsqueda se vuelve esencialmente aleatoria. Los valores encontrados en la literatura para la tasa de mutación generalmente están entre 0.001 y 0.05 (Barboza, 2005).

De Jong (1975) citado por (Moujahid et al., s/f) recomienda la utilización de una tasa de mutación inversamente proporcional al tamaño del cromosoma l^{-1}.

2.2.5. Non-dominated Sorting Genetic Algorithm II (NSGA-II).

El algoritmo NSGA-II es uno de los algoritmos multiobjetivos utilizados popularmente, que intentan encontrar múltiples soluciones óptimas de Pareto en un problema de optimización de objetivos múltiples y tiene las siguientes tres características (Deb, 2011):

- Utiliza un principio elitista.
- Utiliza un mecanismo explícito de preservación de la diversidad.
- Enfatiza soluciones no dominadas.

El proceso del algoritmo NSGA-II en términos generales se describe de la siguiente manera, para cualquier generación t, la población de descendientes (por ejemplo, Q_t) se crea primero utilizando la población parental (por ejemplo, P_t) y los operadores genéticos habituales. A partir de entonces, las dos poblaciones se combinan juntos para formar una nueva población (por ejemplo, R_t) de tamaño $2N$. Entonces, la población R_t es clasificado en diferentes clases(o rangos) de no dominación. A partir de entonces, la nueva población se llena de puntos de diferentes frentes de no dominación, uno a la vez. El relleno comienza con el primer frente de no dominación (de la clase uno) y continúa con los puntos del segundo frente de no dominación, y así sucesivamente. Dado que el tamaño total de la población de R_t es $2N$, no todos los frentes pueden acomodarse en N ranuras disponibles para la nueva población. Se eliminan todos los frentes que no se pudieron acomodar. Cuando se está considerando el último frente permitido, puede haber más puntos en el frente que las ranuras restantes en la nueva población. Este escenario se ilustra en la Figura 8. En lugar de descartar arbitrariamente a algunos miembros del último frente,

se eligen los puntos que harán que la diversidad de los puntos seleccionados sea la más alta (Deb, 2011).

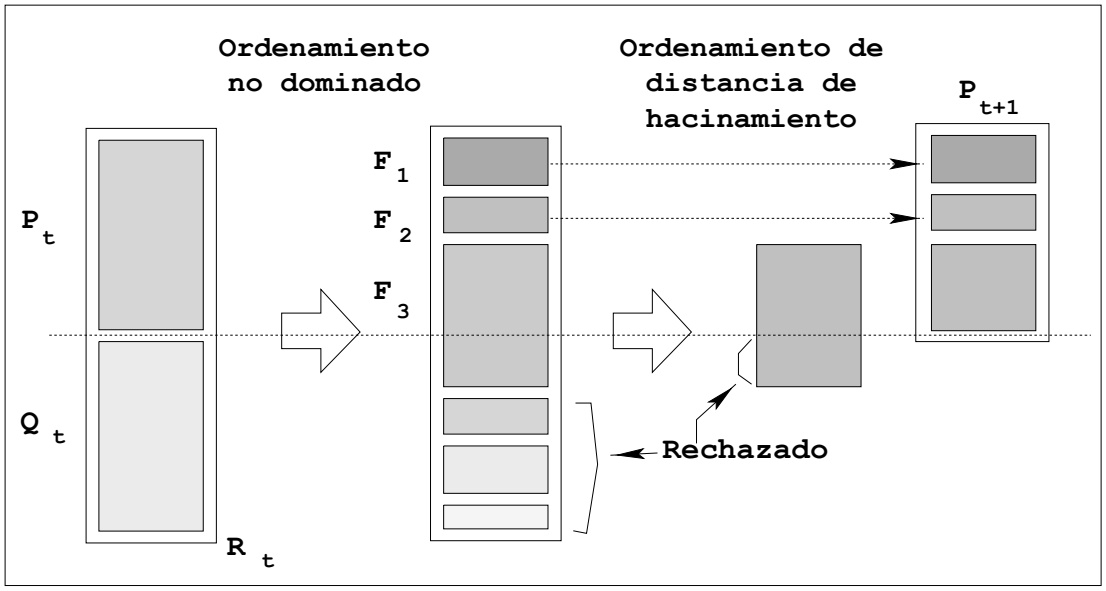


Figura 8. Esquema del procedimiento NSGA-II.
Fuente: (Deb, 2011)

2.2.5.1. Estimación de clases o frentes de no dominancia.

Las soluciones no dominadas se ordenan según dos principios: primero clasificación por rango de soluciones no dominadas, y segundo un contador de dominación, que asigna a una solución individual el número de soluciones que domina a este individuo. Las soluciones no dominadas obtenidas de toda la población se agrupan como un conjunto no dominado para el primer rango donde un contador de dominación se establece en cero. A partir de entonces, los contadores de dominación de otras soluciones que excluyen el conjunto del primer rango se reducen en uno. Las soluciones que tienen el contador igual a cero se agrupan en el segundo rango. El procedimiento se repite hasta que se obtienen diferentes rangos de soluciones no dominadas. Una ilustración de diferentes rangos de soluciones no dominadas para un problema de minimización se muestra en la Figura 9.

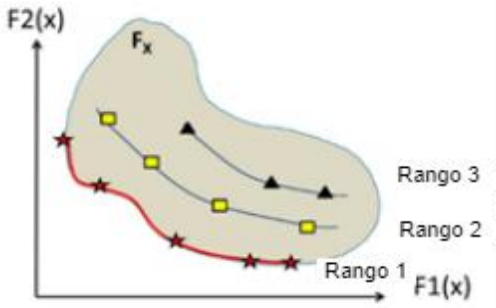


Figura 9. Frente de Pareto con rango ordenado para minimizar dos objetivos.
Fuente: (Petvipusit, 2015)

2.2.5.2. Estimación de densidad o distancia de hacinamiento.

Para obtener una estimación de la densidad de soluciones que rodean una solución particular en la población, calculamos la distancia promedio de dos puntos a cada lado de este punto a lo largo de cada uno de los objetivos. Esta cantidad sirve como una estimación del perímetro del cuboide formado usando los vecinos más cercanos como los vértices (llame a esto la distancia de hacinamiento), ver en la *Figura 10* .

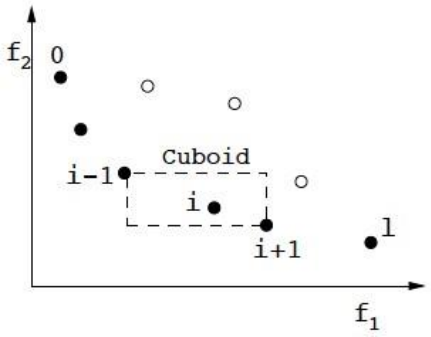


Figura 10. El cálculo de la distancia de hacinamiento. Los puntos marcados en los círculos rellenos son soluciones del mismo frente no dominado.
Fuente: (Deb, 2011)

El cálculo de la distancia de hacinamiento requiere clasificar la población de acuerdo con cada valor de función objetivo en orden de magnitud ascendente. A partir de entonces, para cada función objetivo, a las soluciones de límites (soluciones con valores de función más pequeños y más grandes) se les asigna un valor de distancia infinita. A todas las demás soluciones intermedias se les asigna un valor de distancia igual a la diferencia normalizada absoluta en los valores de función de dos soluciones adyacentes. Este cálculo continúa con otras funciones objetivas. El valor global de distancia de hacinamiento se calcula como la suma de los valores de distancia individuales correspondientes a cada

objetivo. Cada función objetivo se normaliza antes de calcular la distancia de hacinamiento. El algoritmo que se muestra en la parte inferior de la página describe el procedimiento de cálculo de distancia de hacinamiento de todas las soluciones en un conjunto no dominado.

CAPÍTULO III. Materiales y métodos

3.1. Descripción del lugar de ejecución

El proyecto ha sido implementado en la región Puno. El lugar es elegido porque ésta investigación es parte del proyecto "Plataforma digital inteligente y Big Data para el turismo rural comunitario en la Región Puno", proyecto financiado por el CONCYTEC-FONDECYT en el marco de la convocatoria "Proyecto Investigación Básica 2015" [número de contrato 184-2015].

3.2. Materiales e insumos.

A continuación se describen los materiales de software usados más importantes en el desarrollo del proyecto.

3.2.1. JMetal.

JMetal una herramienta para resolver algoritmos metaheurísticos en el lenguaje programación de Java, y es un marco basado en Java orientado a objetos para la optimización de objetivos múltiples con metaheurísticas. ("jMetal 5 Web site", s/f) Proporciona un buen número de algoritmos evolutivos para problemas de objetivo único y de objetivos múltiples, y también proporciona algunos problemas. Tiene la capacidad particular de ejecutar algoritmos paralelos. Proporciona la capacidad de mezclar codificaciones reales y binarias, y proporciona indicadores de calidad (Oliver, 2014).

JMetal trabaja con cuatro interfaces las cuales se pueden personalizar. En la Figura 11 Se muestra la funcionalidad típica proporcionada por jMetal: un algoritmo resuelve un problema al manipular un conjunto de posibles objetos de solución mediante el uso de varios operadores. La interfaz de la solución representa a los individuos en los algoritmos evolutivos y las partículas en el caso de los algoritmos de optimización de enjambres de partículas. Un problema puede crear nuevas soluciones y evaluarlas ("GitHub - jMetal/jMetalDocumentation: jMetal user manual", s/f).

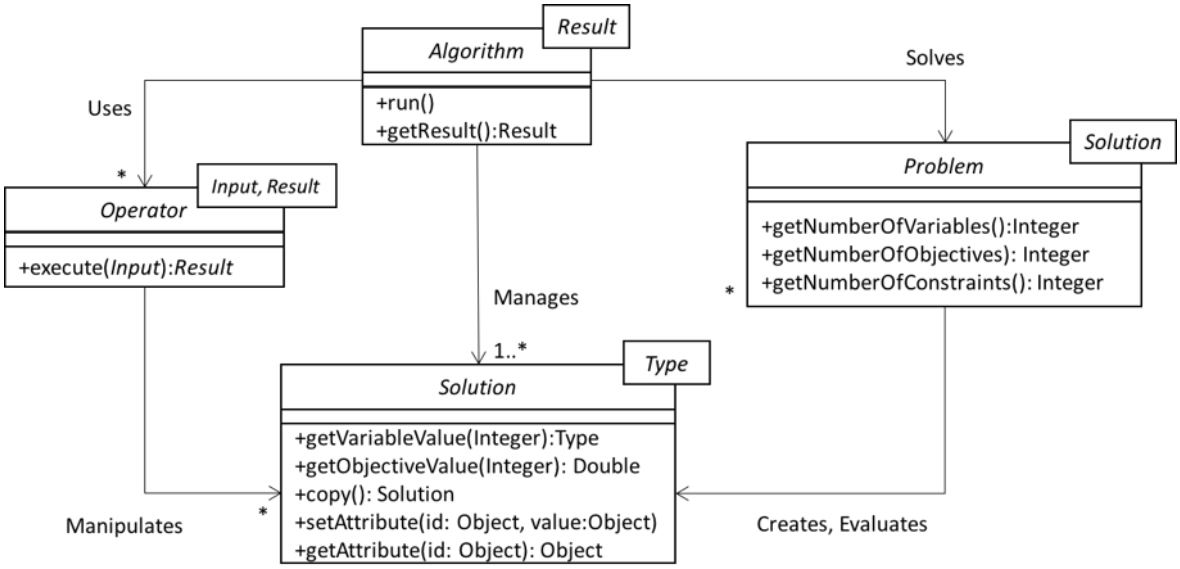


Figura 11. La arquitectura en interfaces de jMetal 5 en UML.
Fuente: https://github.com/jMetal/jMetalDocumentation/blob/master/architecture.md

Esta herramienta facilita que se puedan implementar con facilidad las variantes de los algoritmos, usando a herencia, por ejemplo el algoritmos NSGA-II herada de tres clases (ver *Figura 12*), y hay una implementación de una variante.

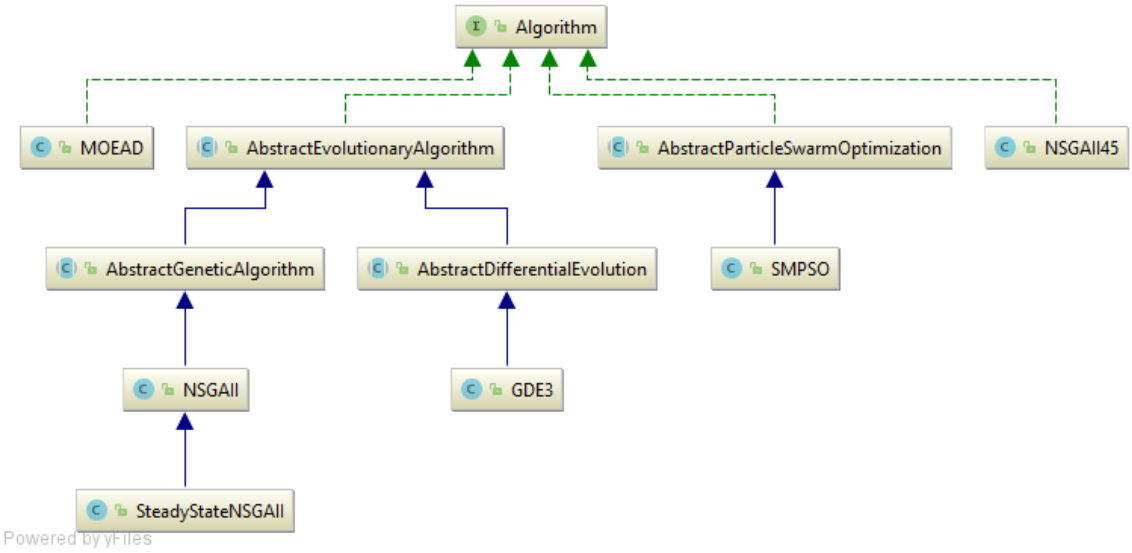


Figura 12. Jerarquía de clases que heredan de un algoritmo NSGA-II en jMetal 5.
Fuente: https://github.com/jMetal/jMetalDocumentation/blob/master/architecture.md

3.2.2. Google Maps API

Google Maps Platform es un conjunto de API's y SDK's del servidor de aplicaciones de mapas web que pertenecen a Alphabet Inc., que se administran desde la consola de Google

Cloud Platform. Para utilizar la plataforma Google Maps, necesita una cuenta de facturación, un proyecto, una o más API o SDK habilitados y una clave de API restringida para usar con aplicaciones y solicitudes.

3.2.2.1. API Distance Matrix.

La API de matriz de distancia es un servicio que proporciona la distancia y el tiempo de viaje para una matriz de orígenes y destinos. La API devuelve información basada en la ruta recomendada entre los puntos de inicio y final, según lo calcula la API de Google Maps, y consta de filas que contienen valores de duración y distancia para cada par ("Developer Guide | Distance Matrix API | Google Developers", 2019).

3.2.2.2. Directions API.

La API de Direcciones es un servicio que calcula las direcciones entre ubicaciones. Puede buscar direcciones para varios modos de transporte, incluidos tránsito, manejo, caminar o andar en bicicleta ("Get Started | Directions API | Google Developers", 2019).

Esta API se usó para mostrar el resultado del algoritmo en un mapa donde se pueda visualizar la ruta de viaje a realizar en un automóvil.

3.3. Tipo de Investigación

El presente trabajo de investigación se adecua como una investigación propositiva por lo siguiente:

Teoría: Optimización multiobjetivo.

Hecho: Diseñar viajes personalizados.

Solución: Implementar la solución para un modelo de optimización multiobjetivo.

3.4. Desarrollo de la investigación

A continuación se describen los métodos para los objetivos.

3.4.1. Recolectar datos de ubicaciones, distancias y tiempos de duración de viaje de un atractivo turístico a otro.

Para la obtención de datos de recursos turísticos de la Región Puno se eligió una fuente oficial que es el inventario que realiza el MINCETUR publicado en ("Mapa de ubicación

de recursos turísticos y emprendimientos de turismo rural comunitario", s/f), de la cual se extrajeron las ubicaciones de 200 recursos turísticos. Se utilizó el método de manual, ver **Anexo A** para ver todo el proceso.

Luego para obtener los datos de distancias y duraciones a partir de las ubicaciones obtenidas de los atractivos turísticos, se usó un cliente desarrollado en Python para consumir las API´s de Google Maps Distance Matrix.

En la Figura 13 se puede observar que parámetros se enviaron en la consulta a cliente de Google Maps API, también cabe mencionar que es posible enviar más datos en los parámetros.

```python
import json
from datetime import datetime

import googlemaps
import requests
from contantes import API_KEY

def get_distances_and_durations_with_googlemapsclient(lat_x, lon_x, lat_y, lon_y):
    gmaps = googlemaps.Client(key=API_KEY)

    # Geocoding an address
    now = datetime.now()
    directions_result = gmaps.distance_matrix(
        origins=(lat_x, lon_x),
        destinations=(lat_y, lon_y),
        mode='driving',
        language='es-419',
        # avoid=None,
        # units=None,
        # departure_time=None,
        # arrival_time=None,
        transit_mode='bus',
        # transit_routing_preference=None,
        # traffic_model=None,
        # region=None,
    )
    if not directions_result.get('status')=="OK":
        print(directions_result)
        return None
    distance = None
    duration = None
    if directions_result.get('rows', None):
        for j in directions_result['rows'][0]['elements']:
            if j.get('distance', None):
                distance = j['distance']['value']
            if j.get('duration', None):
                duration = j['duration']['value']

    return {
        'distance': distance,  # m.
        'duration': duration  # min.
        # 'duration_in_traffic':
    }
```

Figura 13. Código fuente de la función que utiliza el cliente Google Maps API.

Parte de este paso también fue el preparar los datos extraídos para ingresar al algoritmo, de la extracción de datos de distancias y duraciones se obtuvieron dos matrices de 164x164, siendo seleccionados 164 atractivos turísticos de 200, en el proceso se perdieron algunos datos por algunos motivos como falla en la conexión de internet del investigador (siendo posible mejorar en la implementación en la aplicación real como un trabajo a futuro) y algunos porque en si Google Maps devuelve que no hay vías de transporte para automóvil para algunas ubicaciones de atractivos consultadas.

3.4.2. Aplicar el algoritmo NSGA-II para resolver el modelo de Problema del Agente Viajero Multiobjetivo.

Para obtener la solución al modelo se eligió un método aproximado para obtener la solución en un tiempo razonable, ya que problemas de optimización combinatoria como es el caso del Problema del Agente Viajero Multiobjetivo crecen exponencialmente en tiempo con relación al tamaño del problema. Según (Nebro & Durillo, 2009) los algoritmos más conocidos en el campo de optimización multiobjetivo son los algoritmos genéticos, específicamente NSGA-II y SPEA2. Para problema abordado en este proyecto se eligió la implementación desarrollada en jMetal del algoritmo NSGA-II.

3.4.2.1. NSGA-II

Se descargó la librería de jMetal, el cual contiene algoritmos multiobjetivo y algunos ejemplos entre la cuales se encontró al Problema del Agente Viajero Multiobjetivo (MOTSP) y su solución con el algoritmo NSGA-II. Parte de los componentes implementados son propios de la herramienta de jMetal, se describirán a continuación para una mejor comprensión.

3.4.2.1.1. Codificación o representación genética de las soluciones del problema.

Teniendo en cuenta que cuanto más adecuada al problema está la codificación, mejor la calidad de los resultados obtenidos. Para codificar las soluciones del problema en cuestión de este trabajo de investigación, Problema del Agente Viajero Multiojetivo, se usa generalmente la misma codificación del Problema del Agente Viajero (con un solo objetivo).

La codificación usada generalmente para el Problema del Agente Viajero es en forma de permutaciones de números enteros, cabe mencionar que esta representación tiene ventajas

comparado a la codificación binaria tradicional de los algoritmos genéticos (ver 2.2.4.1) (Keresztury, 2017).

Ya que una solución para el Problema del Agente Viajero es una ruta en la que se tiene el orden en que se van a visitar las n ciudades, en un ejemplo donde $n = 5$, puede ser representado en una lista:

(1 5 2 4 3)

Se tiene que el viajero parte de la ciudad 1 siguiendo la secuencia de 1-5-2-4-3 en la que finalmente retorna a la ciudad de origen.

Las ventajas de la representación del camino incluyen la simplicidad de la evaluación de la aptitud y la utilidad de la representación final. La aptitud de un individuo determinado (un recorrido TSP) es fácil de evaluar, ya que se puede calcular sumando los costos de cada par de nodos adyacentes. La representación final es útil, ya que genera directamente la lista de las ciudades en el orden en que deben ser visitadas (Keresztury, 2017).

3.4.2.1.2. Generación de la población inicial.

Se consideró el método de generación de la población aleatoria para tratar de cubrir todo el espacio de búsqueda, ya que es el que se usa generalmente.

A continuación se muestra el pseudocódigo del algoritmo NSGA-II implementado en la framework de jMetal:

Algoritmo 3. Creación de la población inicial

```
Inicio createInitialPopulation()
    crear variable vacía pupulation de tipo lista
    para contador1 ← 0 hasta problem.populationSize incremento 1 hacer
        crear variable vacía randomSequence de tipo lista
        para contador2 ← 0 hasta problem.getPermutationLength 1 hacer
            agrega contador2 a randomSequence
        fin_para contador2
        shuffle(randomSequence) # cambia el orden de forma aleatoria
        para contador3 ← 0 hasta problem.getNumberOfVariables 1 hacer
            setVariableValue(contador3,randomSequence.get(contador3))
        fin_para contador2
        agrega randomSequence a pupulation
    fin_para contador1
    retorna population
```

Fin

Se puede ver en un ejemplo como: si se tiene el parámetro de número de individuos igual al 100, para cada individuo se genera una lista de números de 1 hasta 100 (número de individuos), luego esta se cambia de orden aleatoriamente. De tal forma se tiene una población de individuos creados aleatoriamente que cumpla las restricciones.

Se tiene entendido que en la implementación del problema en jMetal no se considera como un problema con restricciones explícitamente, ya que el Problema del Agente Viajero en si tiene restricciones, se vio que están adecuadas en el proceso del algoritmo. Se observa que al generar la población de individuos se crea con lista de todas las ciudades y durante el proceso del algoritmo solo se cambia de orden entre estos, sin agregar o quitar alguna ciudad con los operadores de cruzamiento y mutación.

3.4.2.1.3. Función de evaluación.

En la función de evaluación se obtendrá directamente de evaluar las funciones objetivo del Problema del Agente Viajero Multiobjetivo, ya que en este trabajo de investigación se consideró dos objetivos las cuales son:

- Minimizar la distancia de viaje en un automóvil de visitar los atractivos seleccionados.
- Minimizar la duración de viaje en un automóvil de visitar los atractivos seleccionados.

En jMetal la implementación de la función de evaluación implementada retorna dos valores reales para cada individuo, esta porque se tiene dos objetivos. Esta función se implementa en el problema.

3.4.2.1.4. Selección.

Este mecanismo de seleccionar quienes serán los padres, con la finalidad de dirigir el proceso a mejores regiones de espacio de búsqueda, es propio del algoritmo NSGA-II, en la que tiene un operador de selección por torneo binario con un comparador de no dominancia y hacinamiento.

En la implementación es posible cambiar del operador de selección en jMetal (ver en la Figura 14 los operadores disponibles de selección) y usar normalmente el comparador de NSGA-II. El que usaremos para este proyecto será BinaryTournamentSelection traducido como selección binaria por torneo, se aplica para devolver la mejor solución entre dos elegidos al azar de una lista de soluciones.

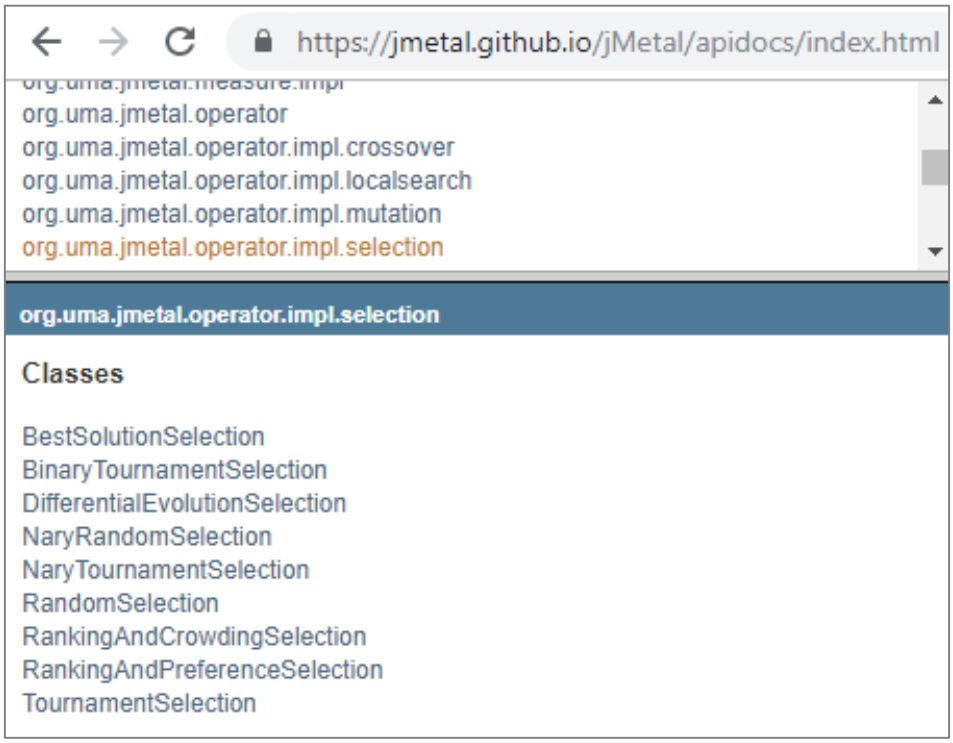


Figura 14. Operadores de selección disponibles en el framework jMetal.
Fuente: https://jmetal.github.io/jMetal/apidocs/index.html

Se ha observado que en la implementación de jMetal de NSGA-II para elegir cada padre manda toda la población actual al operador de selección hasta seleccionar los padres del tamaño de población, por cada ciclo selecciona dos individuos aleatoriamente y elige el un individuo i de acuerdo a dos atributos:

- Un rango de no dominancia r_i, según el frente de Pareto.
- Una distancia local de hacinamiento d_i, que es una medida del espacio de búsqueda alrededor de la solución i que no está ocupado por ninguna otra solución de la población.

La selección retorna la solución ganadora i comparado con j basándose en dos criterios fundamentales.

- Si el individuo i tiene tiene mejor rango que j, es decir, si $r_i < r_j$.
- Si los individuos i y j tiene el mismo rango, i tiene mejor distancia de hacinamiento que el individuo j, es decir, si $r_i = r_j$ y $d_i > d_j$. Y si $d_i = d_j$entonces se elige aletoriamente.

La primera condición asegura que la solución escogida se encuentra en un frente no dominado mejor. La segunda condición resuelve el conflicto en caso de que ambas soluciones se encuentren en el mismo frente, decidiéndose por aquella solución que tenga mejor distancia de hacinamiento, asegurando la diversidad (Palma Méndez & Morales, 2008).

Cabe mencionar que este proceso del algoritmo NSGA-II descarta algunos individuos y genera duplicidad de otros individuos, con el fin de mantener constante el tamaño de la población.

3.4.2.1.5. Cruzamiento.

No existe un operador de cruzamiento específico para el algoritmo NSGA-II, esta se elige de la forma tradicional dependiendo a la codificación de la soluciones del problema. Para el caso de este proyecto ya que la codificación que se eligió es permutación de enteros, se revisó los operadores de cruce que tiene disponible el framework jMetal (ver Figura 15). El criterio de selección que se consideró fue elegir uno de los operadores de cruzamiento más utilizados para problemas de tipo de ruta, según (Keresztury, 2017) es PMX (Partially-Mapped Crossover).

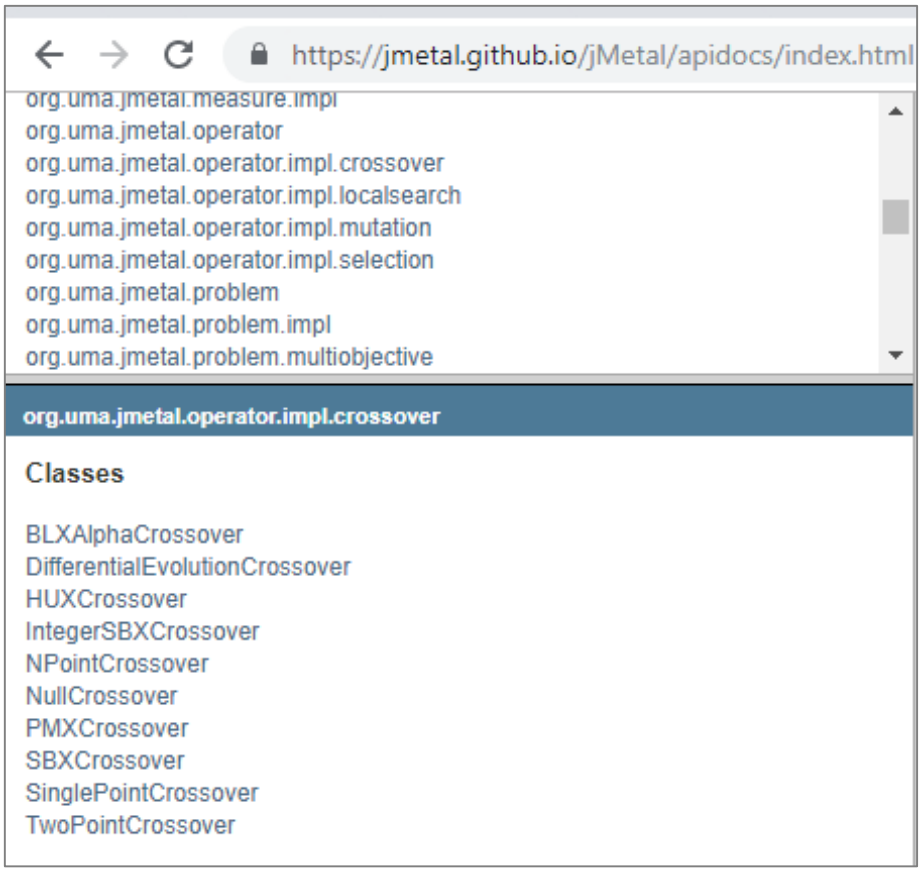


Figura 15. Operadores de cruzamiento disponibles en el framework jMetal.
Fuente: https://jmetal.github.io/jMetal/apidocs/index.html

Según (Hussain, Muhammad, & Sajid, 2019) se tiene la siguiente explicación del operador de cruzamiento PMX: Después de elegir dos puntos de corte aleatorios en los padres para construir descendientes, la porción entre los puntos de corte se mapea entre sí y se intercambia la información restante. Considere, por ejemplo, los recorridos de los dos padres con un punto de corte aleatorio entre los bits 3 y 4 y otro punto de corte entre los bits 6 y 7 (los dos puntos marcados con "|"):

$$P_1 = (9\ 4\ 5 \mid 2\ 8\ 1 \mid 6\ 7\ 3)$$

Y

$$P_2 = (3\ 6\ 1 \mid 9\ 7\ 8 \mid 2\ 4\ 5)$$

Las secciones de mapeo se encuentran entre los puntos de corte. En este ejemplo, los mapeos son $2 \leftarrow\rightarrow 9$, $8 \leftarrow\rightarrow 7$ y $1 \leftarrow\rightarrow 8$. Ahora se copian dos secciones de mapeo entre sí para hacer descendencia como:

$$O_1 = (\times \times \times \mid 9\ 7\ 8 \mid \times \times \times)$$

Y

$$O_2 = (\times \times \times \mid 2\ 8\ 1 \mid \times \times \times)$$

Entonces podemos llenar más bits (de los padres originales), para aquellos que no tienen conflicto como:

$$O_1 = (\times\ 4\ 5 \mid 9\ 7\ 8 \mid 6\ \times\ 3)$$

Y

$$O_2 = (3\ 6\ \times \mid 2\ 8\ 1 \mid \times\ 4\ 5)$$

Por lo tanto, la primera × en el primer descendiente es 9 que proviene del primer padre pero 9 ya está en esta en el descendiente, por lo que verificamos el mapeo 2 ← → 9, entonces 2 ocupará primero ×. De manera similar, la segunda × en la primer descendiente es 7, que proviene del primer padre pero 7 existe en este descendiente, verifique también el mapeo 8 ← → 7 y vea nuevamente 8 existe en este descendiente, nuevamente verifique el mapeo 8 ← → 1 para que 1 ocupe segundo ×. Por lo tanto, el descendendiente 1 es:

$$O_1 = (2\ 4\ 5 \mid 9\ 7\ 8 \mid 6\ 1\ 3)$$

Y de forma análoga, también completamos la segunda descendencia:

$$O_2 = (3\ 6\ 7 \mid 2\ 8\ 1 \mid 9\ 4\ 5)$$

3.4.2.1.6. Mutación.

De igual forma que el cruzamiento no existe un operador de mutación específico para el algoritmo NSGA-II, esta se elige de la forma tradicional dependiendo a la codificación de la soluciones del problema.

Ya que se tiene la codificación del problema en forma de permutación de enteros implementado es preferible usar un operador que trabaje con permutaciones, teniendo en cuenta que solo debe trabajar con los genes que el individuo ya tiene, y no agregar o quitar generando duplicidad de genes en el primer caso que no ayuda al resultado final ya que no cumplirá con la restricciones específicas que tiene el Problema del Agente Viajero. Ya que

solo se dispone de un operador de mutación en la herramienta de jMetal (ver Figura 16), se trabajará con PermutationSwapMutation (traducido como mutación por intercambio para permutacion). Este operador de mutación trabaja seleccionando dos genes al azar e intercambiando sus posiciones.

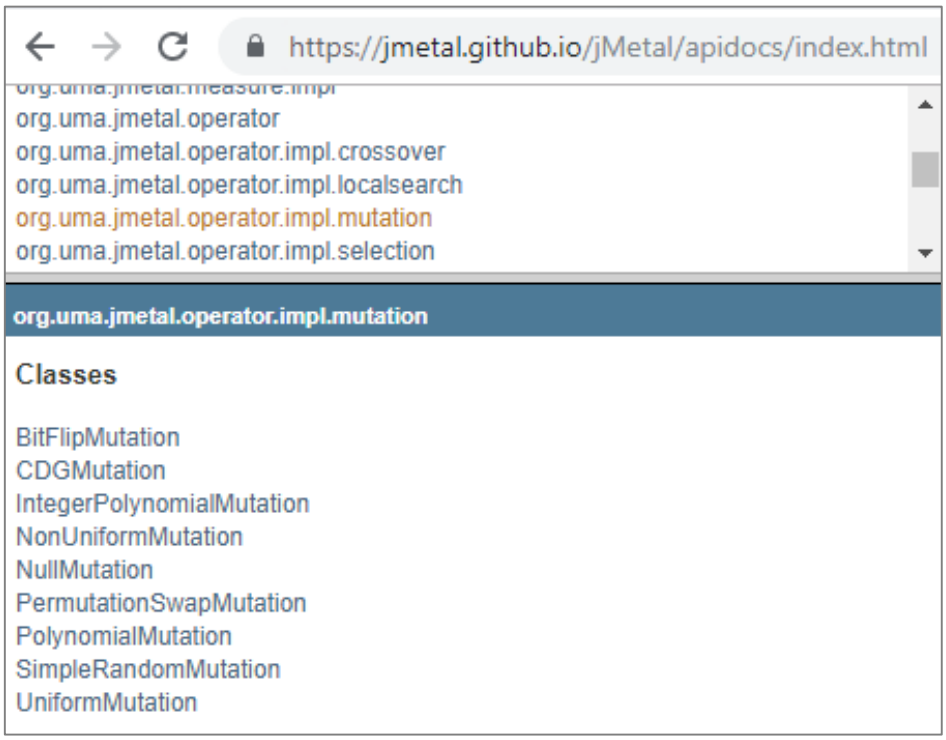


Figura 16. Operadores de mutación disponibles en el framework de jMetal
Fuente: https://jmetal.github.io/jMetal/apidocs/index.html

3.4.2.1.7. Reemplazo.

En esta parte específicamente el algoritmo NSGA-II trabaja con el modelo generacional con elitismo, donde junta los padres con los descendientes y selecciona usando la función de selección por ranking de no dominancia y hacinamiento, según el esquema mostrado en la *Figura 8*. Luego parte de proceso es verificar el criterio de parada y si cumple comenzar nuevamente el ciclo.

3.4.2.1.8. Criterio de parada.

El criterio de parada disponible en jMetal es de número máximo de iteraciones (generaciones), pero también tiene un ejemplo usando tiempo. Para este proyecto de investigación se usará el primero, ya que es el más usado.

3.4.3. Implementar una interfaz donde se pueda simular la planificación de ruta de un viaje turístico.

La idea inicial de implementación de fue crear una aplicación real para la planificación de rutas para un turista que desee viajar a Puno. Está idea de implementación se adaptó a aplicación de un algoritmo, a estudiarlo, y hacer un ajuste de parámetros adecuado a los datos disponibles, ya que este proceso fue necesario para tener un buen resultado. Sin embargo se tiene una presentación simulando la aplicación final, que incluye un mapa, está fue implementada en Jupyter Notebook.

Se usó Jupyter Notebook para como una herramienta para visualizar mejor la codificación por partes durante el proceso de limpieza de datos, simular la selección de atractivos de parte del turista y finalmente mostrar el resultado de la ruta en el mapa.

Como primer paso se tiene que una vez cargado los datos a Jupyter permiten elegir los atractivos por diferentes filtros, como categorías, sub-categorías, provincias, tipo de actividad que se puede realizar, entre otros, parte de este proceso una función guarda en dos archivos csv, el primero con una matriz asimétrica de distancias y la segunda de igual forma con una matriz asimétrica de tiempos de duración, de los atractivos seleccionados. (Ver Figura 17, Figura 18 y Figura 19)

```
codes=df_cleaned['codigo'].to_list()
with open('datasets/selectData-puno.json') as train_file:
    data_features = json.load(train_file)['features']
    data_properties = [i['properties'] for i in data_features]
    df_temp = pd.DataFrame.from_dict(data_properties)
    print("All data:",df_temp.shape)
    df = df_temp[df_temp['codigo'].isin(codes)]
    print("Filtered data:",df.shape)
df.head()
```

```
All data: (200, 25)
Filtered data: (164, 25)
```

	agrup	atrac_acti	atrac_acti_tipo	atrac_categ	atrac_stipo	atrac_tipo	categoria	codigo	desdpto	desprov	...	imagen	lstActiGeo
0	1	[7, 99]	[3, 45, 49]	2.0	65.0	17	2. MANIFESTACIONES CULTURALES	282	Puno	San Roman	...	cultural.png	[{'atrac_acti': 7, 'atrac_acti_descrip': 'Folc...
2	1	[7, 99]	[23, 42, 45, 49]	2.0	67.0	17	2. MANIFESTACIONES CULTURALES	315	Puno	San Roman	...	cultural.png	[{'atrac_acti': 7, 'atrac_acti_descrip': 'Folc...
4	1	[7, 99]	[19, 3, 41, 45]	2.0	65.0	17	2. MANIFESTACIONES CULTURALES	409	Puno	Puno	...	cultural.png	[{'atrac_acti': 7, 'atrac_acti_descrip': 'Folc...
6	1	[99]	[1, 19, 45]	2.0	90.0	20	2. MANIFESTACIONES CULTURALES	411	Puno	Puno	...	cultural.png	[{'atrac_acti': 99, 'atrac_acti_descrip': 'Otr...

Figura 17. Código fuente de cargado de datos seleccionados de atractivos turísticos.

```
In [5]: df.subtipo_categoria.value_counts()

Out[5]: Iglesias (Templo, Catedral, etc.)                                          39
        Edificaciones (Templos, fortalezas, plazas, cementerios...)                29
        Aguas Termales                                                             15
        Laguna                                                                      6
        Playas                                                                      5
        LAGO TITICACA                                                               5
        Pueblos Históricos                                                          5
        Bosque de Piedra                                                            5
        Cerro                                                                       5
        Pintura Rupestre                                                            4
        Bosques                                                                     4
        Museo y Otros (Pinacoteca)                                                  4
        Miradores                                                                   3
        Petroglifos (Grabados en piedra)                                            3
        Fiestas Patronales                                                          2
        Capilla                                                                     2
         Otros (monumento, pileta, etc)                                             2
        Edificación (fortaleza, escuela, cuartel, colegio ...)                      2
        Edificaciones (casas, balcones, cuartos, ventanas, patios, murallas, puentes...)  2
        Islas                                                                       1
        Reservas Nacionales                                                         1
        Piscigranjas                                                                1
        Cataratas/Cascadas                                                          1
        Ceramica                                                                    1
                                                                                    1
        Campo de Batalla                                                            1
        Grutas, cavernas y cuevas                                                   1
        Biblioteca                                                                  1
        Zonas Reservadas                                                            1
        Agricultura                                                                 1
        Música y Danza                                                              1
```

Figura 18. Resultado de cantidad de atractivos por sub tipo categoría.

```
def create_files(df_selected):
    # selecciona solo la parte las actividades que se selccionaron
    df_prepared = df_cleaned.loc[df_selected.index,df_selected.index]
    df_prepared = transform_matrix_simetric_to_asimetric(df_prepared)
    # obtiene loc codigos de la primera columna
    global selected_codes
    selected_codes = df_prepared[df_prepared.columns[0]].apply(lambda row: row[0] if row else None).tolist()
    df_prepared_distances=df_prepared.applymap(get_distancevaluefrom_row)
    df_prepared_durations=df_prepared.applymap(get_durationvaluefrom_row)
    df_prepared_distances.to_csv("/home/yuselenin/Descargas/jMetal-master/selected_distances.csv",header=False,inde
    df_prepared_durations.to_csv("/home/yuselenin/Descargas/jMetal-master/selected_durations.csv",header=False,inde
    #os.system("""/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/yuselenin/Descargas/idea-IU-191.
create_files(df[df['subtipo_categoria']=='Aguas Termales'])
```

Figura 19. Guardado de archivos de atractivos seleccionados para generar la ruta.

Luego como segundo paso este proceso cabe mencionar que aún es necesario ejecutar manualmente el algoritmo, el cual está configurado para que tome los archivos generados en el paso anterior. El algoritmo de NSGA-II se ejecuta en Java con toda su configuración dada en el proceso de ajuste de parámetros y todo los demás en Python. (Ver *Figura 20* y *Figura 21*)

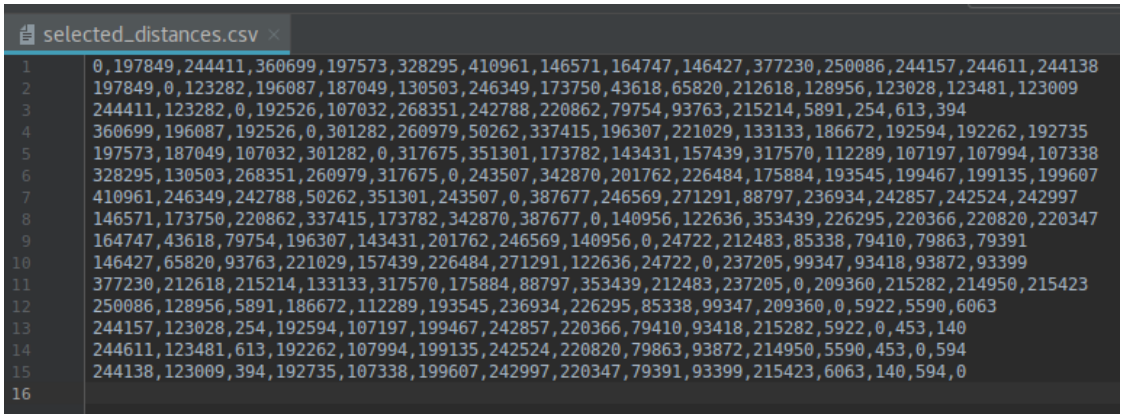


Figura 20. Archivo con matriz de distancias en metros.

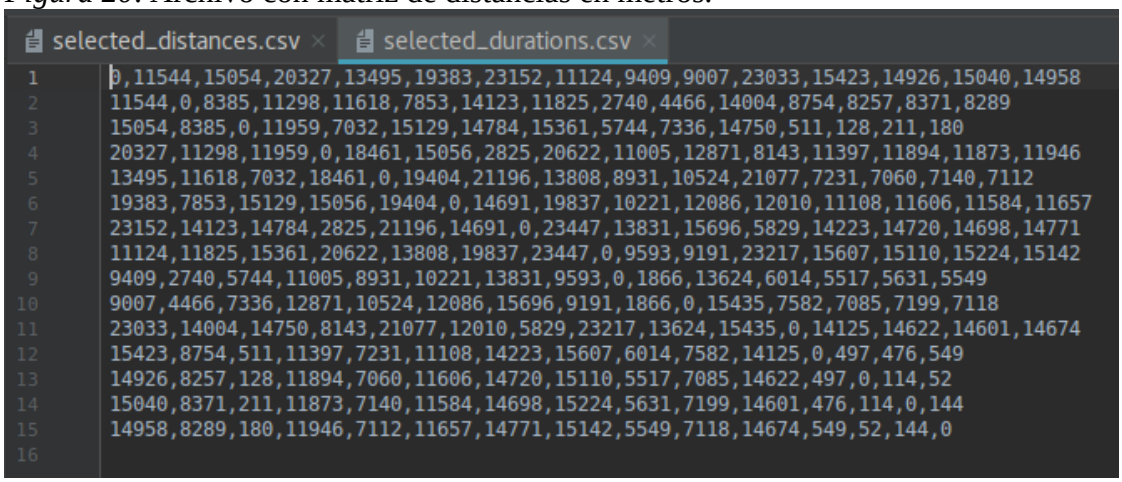


Figura 21. Archivo con matriz de duraciones en minutos.

Se usó la siguiente configuración de parámetros para el algoritmo NSGA-II, ver *Tabla 2*.

Tabla 2.
Configuración de parámetros del algoritmo genético.

Descripción del parámetro	**Valor**
Representación	Permutación
Tamaño de población	Tamaño del cromosoma
Método de selección	Selección por torneo binario con un comparador de no dominancia y hacinamiento.
Método de cruzamiento	PMX
Probabilidad de cruzamiento	0.9 (90%)
Método de mutación	PermutationSwap
Probabilidad de mutación	0.1 (10%)
N° máximo de generaciones	10000
Modelo de remplazamiento	Generacional con elitismo

Como tercer paso es necesario ejecutar en Jupyter nuevamente para ver los resultados del algoritmo, se tiene que seleccionar una de las soluciones de la frontera de Pareto. Una vez seleccionado se tiene puede visualizar en un mapa (Ver Figura 22).

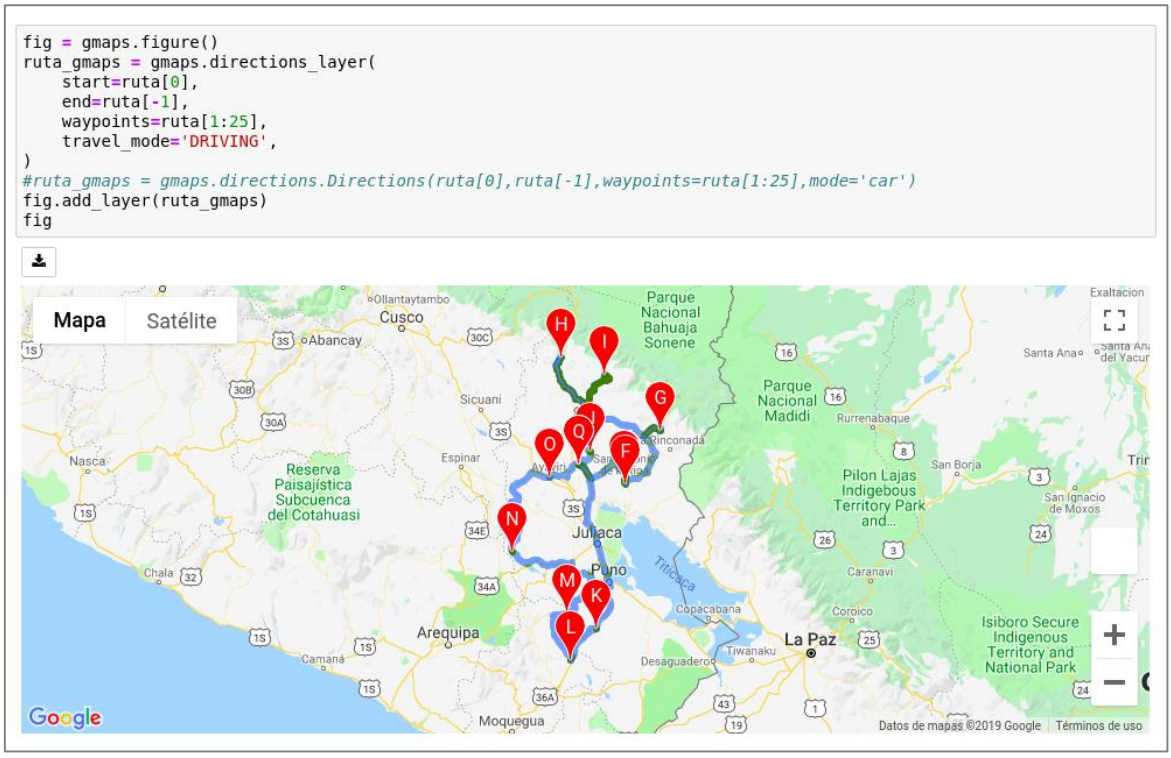


Figura 22. Resultado final de ruta en un mapa.

CAPÍTULO IV. Resultados y Discusión

4.1. Resultados

Como trabajo realizado para alcanzar el primer objetivo se logró recolectar datos de ubicaciones de 200 atractivos turísticos ubicados en la región de Puno a través de un proceso manual del inventario realizado por el Ministerio de Comercio Exterior y Turismo del Perú. También se logró recolectar datos disponibles de distancias (en metros) y tiempos de duración (en minutos) de viajar de un atractivo a otro, de 164 atractivos, para este proceso se usó un cliente en Python para consumir API's de Google Maps Distance Matrix, perdiendo algunos datos en el proceso y otros porque Google no reconoce vías de transporte que pueda conectar entre algunas ubicaciones de atractivos dadas.

Para alcanzar el segundo objetivo se logró aplicar el algoritmo Non-dominated Sorting Genetic Algorithm II en el lenguaje de programación de Java, usando una herramienta denominada jMetal que contiene algoritmos metaheúristicos para resolver problemas de optimización multiobjetivo. Se usó para resolver el modelo del Problema del Agente Viajero Multiobjetivo, el cual se consideró por su semejanza con planificación de rutas de viaje. Parte de este paso también fue estudiar cómo funciona los componentes del algoritmo y los operadores disponibles en jMetal.

Finalmente, para el último objetivo se logró implementar una interfaz en un entorno de trabajo interactivo que permite desarrollar código en Python denominado Jupyter Notebook para simular la planificación de ruta de un viaje turístico. En este pasó consta de un proceso en el cual el usuario interactúa con implementación: en primer lugar se ejecuta la carga los datos, luego elige por varios filtros que atractivos desea visitar y guarda en archivos csv, seguidamente ejecuta el algoritmo NSGA-II en usando jMetal, a continuación, elige una ruta del conjunto de soluciones denominada frontera de Pareto y finalmente ejecuta la sección de mostrar la ruta en el mapa consumiendo la API de Google Maps Directions (ver resultado en Figura 23).

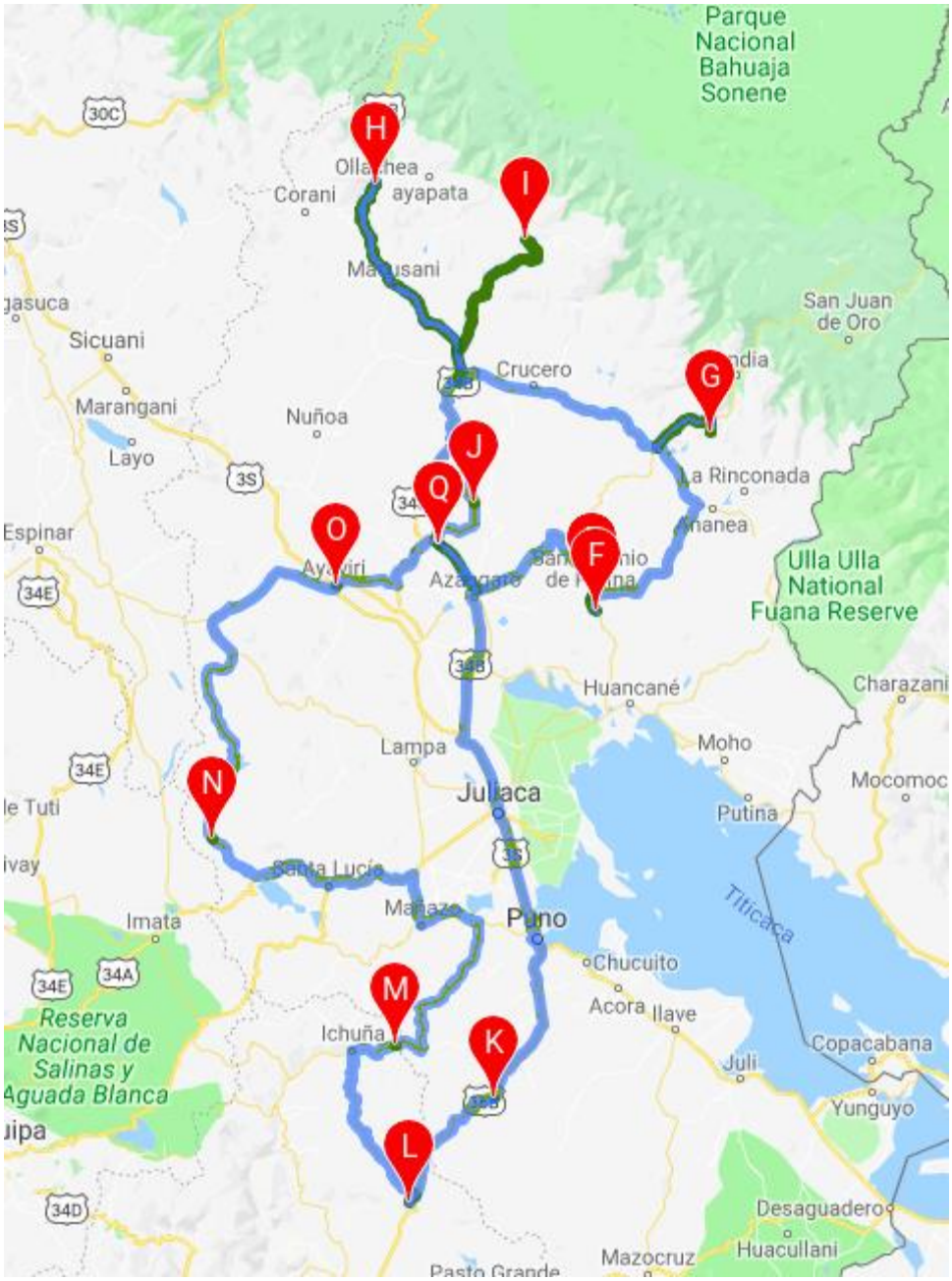


Figura 23. Resultado de la ruta en un mapa.

4.2. Discusión

4.2.1. Recolectar datos de ubicaciones, distancias y tiempos de duración de viaje de un atractivo turístico a otro.

Para el primer objetivo se consideró la mejor fuente de datos de atractivos turísticos disponibles, y se eligió hábilmente el método manual de extracción también pudiendo recurrir método de scraping para obtener más datos.

Para los datos de distancias y tiempos de duración que no se pudieron obtener es posible mejorar, ya que el objetivo de este trabajo no es realmente poner en producción con estos datos por esto no fue un proceso exigido, ya que es posible corregir el primer error de perdida de datos en el proceso.

Se observó que se pueden obtener más datos de Google Maps como precios de hoteles, restaurantes, y varios tipos de filtros de actividades, entre otros, que permiten buscar cerca de una ubicación, estas si se consiguen implementar podría ser de gran ayuda para incluir en la planificación del viaje.

También se observó que se pueden enviar más parámetros en la obtención de datos de Google Maps API Distance Matrix, como el tráfico, hora de partida, modos de transporte (como ir a pie, bicicleta, bus, etc.), estos datos podrían útiles, pero se requeriría hacer peticiones cada vez por todos los atractivos que se seleccionaron lo cual no se haría en la idea que se tiene actualmente solo se haría una vez. Es importante considerar también que la consulta de los datos de Google Maps tiene sus costos, pero también tiene una versión gratuita con limitaciones.

4.2.2. Aplicar el algoritmo NSGA-II para resolver el modelo de Problema del Agente Viajero Multiobjetivo.

Parte de este objetivo no está completa ya que sería mejor si solo estuviera implementado en un solo lenguaje de programación, siendo la idea inicial implementar la versión de jMetal en Python llamado jMetalPy, pero se encontró que su documentación estaba incompleta, y tenía pocos ejemplos, pero es posible reconsiderar la versión de Python nuevamente ya que se tiene conocimiento de cómo funciona la versión en Java.

4.2.3. Implementar una interfaz donde se pueda simular la planificación de ruta de un viaje turístico.

En esta parte se tiene una versión de prueba que está casi lista para ser implementada en una aplicación como fue la idea inicial de este proyecto de investigación, se tendría trabajar en algunos detalles como pasar el algoritmo de NSGA-II que está en Java a Python.

CAPÍTULO V. Conclusiones y recomendaciones

5.1. Conclusiones

Al observar que para un turista que desea planificar su propio viaje sin depender de opciones prediseñadas es complicada la tarea de evaluar todas las posibilidades para tener la ruta óptima de los atractivos que desea visitar, al recurrir por ayuda a las computadoras tampoco es posible tener soluciones exactas para una cantidad de atractivos considerable ya que el tiempo computacional crece exponencialmente al aumentar el tamaño del problema.

Se encontró que el buscar la ruta óptima es un problema estudiado con el término de Problema del Agente Viajero, al buscar datos de atractivos y distancias, se observó que se podían obtener más datos como el tiempo de duración de ir de un atractivo a otro, entonces con la finalidad de aprovechar estos datos y viendo que en la vida real se tiene más problemas donde pueden haber más de un objetivo a optimizar, se vio como una oportunidad implementar un modelo multiobjetivo para la planificación de rutas de viajes turísticos.

Para desarrollar esta idea primero se recolectaron datos de 200 atractivos turísticos, 164 datos de distancias y tiempos de duración de viaje en un automóvil de ir de un atractivo a otro. Se aplicó el algoritmo NSGA-II para resolver el modelo de Problema del Agente Viajero Multiobjetivo. Se implementó una interfaz en Jupyter Notebook donde se pueda simular la planificación de ruta de un viaje turístico.

5.2. Recomendaciones

- El resultado que se tiene se puede agregar algoritmos de recomendación con el fin de seleccionar los atractivos para luego realizar una ruta.

- Actualmente no se cuenta con una puntuación de relevancia a los atractivos turísticos obtenidos del inventario de MINCETUR lo cual seria de gran ayuda para una buena recomendación o filtro de los lugares que estan en buenas condiciones. Es posible obtener estos datos como lo hizo (Rodríguez Díaz, 2012) obteniendo estos datos de publicaciones de revistas, reportes, paginas web (como TripAdvisor), etc.

REFERENCIAS

Agüera, O. (2013). *El turismo comunitario como herramienta para el desarrollo sostenible de destinos subdesarrollados*. https://doi.org/10.5209/rev_NOMA.2013.v38.42908

Ahmia, I., & Skoudarli, A. (2017). *The Traveling Salesman Problem: An Overview of Applications, Formulations, Exact and Approximate Algorithms* (Vol. 02). Recuperado de http://www.laromad.usthb.xn--dzetmathmatiquesdeladcision-iocn

Arellano Arriaga, N. A. (2014). *Un enfoque biobjetivo al problema del reparador* (Universidad Autónoma de Nuevo León). Recuperado de http://pisis.fime.uanl.mx/ftp/pubs/thesis/msc/2014-nancy_arellano/2014-tesis-nancy_arellano.pdf

Artificial Intelligence Overview. (s/f). Recuperado el 24 de octubre de 2018, de https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm

Barboza, A. O. (2005). *Simulação e técnicas da computação evolucionária aplicadas a problemas de programação linear inteira mista*. Universidade Tecnológica Federal do Paraná.

Barrera Núñez, R. I. (2006). *Aplicación de los algoritmos genéticos en la optimización de funciones reales* (Universidad Nacional Abierta Matemática Centro Local Metropolitano). Recuperado de http://biblo.una.edu.ve/docu.7/bases/marc/texto/t7935.pdf

Beirigo, B. A. (2016). *Single-objective and bi-objetive parallel heuristics for the travel planning problem* (Universidade Federal de Viçosa). Recuperado de http://www.locus.ufv.br/handle/123456789/9489

Cabezas, J. L. (2006). La cultura y el turismo como medios de desarrollo socioeconómico. Recuperado el 22 de julio de 2019, de https://www.oei.es/historico/cultura/culturamhmejia.htm

Castro, R. E. de. (2001). *Otimização de Estruturas com Multi-objetivos Via Algoritmos Genéticos de Pareto*. Universidade Federal do Rio de Janeiro.

Chen, X., Liu, Y., Li, X., Wang, Z., Wang, S., & Gao, C. (2019). A New Evolutionary Multiobjective Model for Traveling Salesman Problem. *IEEE Access*, *7*, 66964–66979. https://doi.org/10.1109/ACCESS.2019.2917838

Chu, P. C. H. (1997). *A Genetic Algorithm Approach for Combinatorial Optimisation Problems*. University of London, London.

Collette, Y., & Siarry, P. (2003). *Multiobjective Optimization Principles and Case Studies* (2a ed.). https://doi.org/10.1007/978-3-642-07283-3

Copeland, B. J. (2018). Artificial Intelligence. Recuperado el 24 de octubre de 2018, de Encyclopædia Britannica website: https://www.britannica.com/technology/artificial-intelligence

de Cabezón Irigaray, E. S. (2017). *¿Qué es eso del problema P versus NP? - YouTube*. Recuperado de https://www.youtube.com/watch?v=UR2oDYZ-Sao

de Pinho, A. F., Montevechi, J. A. B., Marins, F. A. S., & de Carvalho Miranda, R. (2013). Algoritmos Genéticos: Fundamentos e Aplicações. En H. S. Lopes, L. C. de Abreu Rodrigues, & M. T. A. Steiner (Eds.), *Meta-Heurísticas em Pesquisa Operacional* (1a ed., pp. 21–32). https://doi.org/10.7436/2013.mhpo.02

Deb, K. (2011). *Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction*. Recuperado de http://www.iitk.ac.in/kangal/deb.htm

Developer Guide | Distance Matrix API | Google Developers. (2019). Recuperado el 25 de julio de 2019, de https://developers.google.com/maps/documentation/distance-matrix/intro

El turismo y la atenuación de la pobreza. (s/f). Recuperado el 19 de septiembre de 2018, de http://step.unwto.org/es/content/el-turismo-y-la-atenuacion-de-la-pobreza

Evolución de la pobreza monetaria 2007 - 2018. (2019). Recuperado de https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1646/libro.pdf

Gestal, M., Rivero, D., Rabuñal, J. R., Dorado, J., & Pazos, A. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética*. Recuperado de http://www.galeon.com/dantethedestroyer/algoritmos.pdf

Get Started | Directions API | Google Developers. (2019). Recuperado el 25 de julio de 2019, de https://developers.google.com/maps/documentation/directions/start

GitHub - jMetal/jMetalDocumentation: jMetal user manual. (s/f). Recuperado el 23 de julio de 2019, de https://github.com/jMetal/jMetalDocumentation

Hernández, J. G., & García, M. J. (2007). Investigación de operaciones y turismo. *Revista de Matemática: Teoría y Aplicaciones 2007 14(1): 221–238*, *14*(1), 1409–2433.

Recuperado de https://www.redalyc.org/pdf/453/45326939011.pdf

Hussain, A., Muhammad, Y. S., & Sajid, M. N. (2019). A Simulated Study of Genetic Algorithm with a New Crossover Operator using Traveling Salesman Problem. *Punjab University Journal of Mathematics*, *51*(5), 61–77. Recuperado de http://pu.edu.pk/images/journal/maths/PDF/Paper-5_51_5_2019.pdf

jMetal 5 Web site. (s/f). Recuperado el 23 de julio de 2019, de http://jmetal.github.io/jMetal/

Keresztury, B. (2017). *Genetic algorithms and the Traveling Salesman Problem* (Eötvös Loránd University). Recuperado de https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2017/keresztury_bence.pdf

Luger, G. F. (2009). *Artificial Intelligence Structures and Strategies for Complex Problem Solving* (6a ed.). Mexíco: Pearson Education.

Mapa de ubicación de recursos turísticos y emprendimientos de turismo rural comunitario. (s/f). Recuperado el 23 de julio de 2019, de http://sigmincetur.mincetur.gob.pe/turismo/

Martí, R. (s/f). *Procedimientos Metaheurísticos en Optimización Combinatoria*.

Matai, R., Singh, S., & Mittal, M. L. (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches. En D. Davendra (Ed.), *Traveling Salesman Problem*. https://doi.org/10.5772/12909

Medición económica del Turismo. (2016). Recuperado de https://www.mincetur.gob.pe/wp-content/uploads/documentos/turismo/publicaciones/MEDICION_ECONOMICA_TURISMO_ALTA.pdf

Memorial de Turismo Rural Comunitario en el Perú. (2015). Recuperado de http://media.peru.info/koha/Memorial-Turismo-Rural-Comunitario-en-el-Peru.pdf

Mendoza Apaza, O. D. (2017). *Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú*.

Merelo Guervós, J. J. (s/f). Informática evolutiva: Algoritmos genéticos. Recuperado el 29 de julio de 2019, de http://geneura.ugr.es/~jmerelo/ie/ags.htm

Moreira Silva, A. J. (2005). *Implementação de um Algoritmo Genético utilizando o Modelo de Ilhas*. Universidade Federal do Rio de Janeiro.

Moujahid, A., Inza, I., & Larrañaga, P. (s/f). *Algoritmos genéticos*. Recuperado de

http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t2geneticos.pdf

Naciones Unidas declaró el 27 de setiembre como Año Internacional del Turismo Sostenible para el Desarrollo. (2017). Recuperado el 22 de julio de 2019, de https://sinia.minam.gob.pe/novedades/naciones-unidas-declaro-27-setiembre-ano-internacional-turismo

Nebro, A. J., & Durillo, J. J. (2009). On the Effect of Applying a Steady-State Selection Scheme in the Multi-Objective Genetic Algorithm NSGA-II. En R. Chiong (Ed.), *Nature-Inspired Algorithms for Optimisation* (pp. 435–456). https://doi.org/10.1007/978-3-642-00267-0_16

Oliver, J. M. (2014). *Multi-Objective Optimisation Methods Applied to Complex Engineering Systems*. Cranfield University.

Palma Méndez, J. T., & Morales, R. M. (2008). Inteligencia artificial. Técnicas, métodos y aplicaciones. En *Mc Graw*.

Perfil del Turista Extranjero. (2017). Recuperado de https://www.promperu.gob.pe/TurismoIN/sitio/VisorDocumentos?titulo=Perfil del Turista Extranjero 2017&url=~/Uploads/perfiles_extranjeros/40/PTEConsolidado2017.pdf&nombObjeto=PerfTuristaExt&back=/TurismoIN/sitio/PerfTuristaExt&issuuid=

Petvipusit, R. (2015). *Decision making and efficient surrogate-assisted optimisation techniques under uncertainty Application to CO 2 sequestration*. Imperial College London.

Plan Estratégico Nacional de Turismo 2025 (PENTUR). (2015). Recuperado de https://www.mincetur.gob.pe/wp-content/uploads/documentos/turismo/documentos/PENTUR/PENTUR_Final_JULIO2016.pdf

Pobreza en el sur aumentó ligeramente | Sociedad - La República. (2019). Recuperado el 19 de julio de 2019, de https://larepublica.pe/sociedad/1453164-pobreza-sur-aumento-ligeramente/

Reporte mensual de turismo de diciembre 2018. (2018). Recuperado de https://www.mincetur.gob.pe/wp-content/uploads/documentos/turismo/estadisticas/ReporteTurismoMensual/RMT_Diciembre_2018.pdf

Riojas Cañari, A. C. (2005). *Búsqueda Tabú: conceptos, algoritmo y aplicación al problema de las N-reinas*. Recuperado de http://sisbib.unmsm.edu.pe/bibvirtual/monografias/Basic/riojas_ca/contenido.htm

Rodríguez Díaz, B. (2012). *Planificando un viaje personalizado: Aplicación de un enfoque Multicriterio* (Universidad de Málaga). Recuperado de http://cuadernos.uma.es/pdfs/papeles57.pdf

Rodríguez Díaz, B., & Caballero Fernández, R. (2012). Sistema de ayuda al turista: Modelo para la planificación de un viaje personalizado. *Estudios y Perspectivas en Turismo*, *21*(1), 108–125.

Russell, S., & Norvig, P. (2004). *Inteligencia Artificial* (2a ed.). Recuperado de www.pearsoneducacion.com

Search Algorithms in Artificial Intelligence. (s/f). Recuperado el 25 de octubre de 2018, de Hacker Noon website: https://hackernoon.com/search-algorithms-in-artificial-intelligence-8d32c12f6bea

Search techniques. (2018). Recuperado el 25 de octubre de 2018, de Wikiversity website: https://en.wikiversity.org/wiki/Search_techniques

The Problem. (2015). Recuperado el 22 de julio de 2019, de http://www.math.uwaterloo.ca/tsp/problem/index.html

Thornton, C. J. (1992). *Artificial Intelligence Though Search*. https://doi.org/10.1007/978-94-011-2836-0

Travel and tourism economic impact 2018 Perú. (2018). Recuperado de https://www.wttc.org/economic-impact/country-analysis/country-reports/

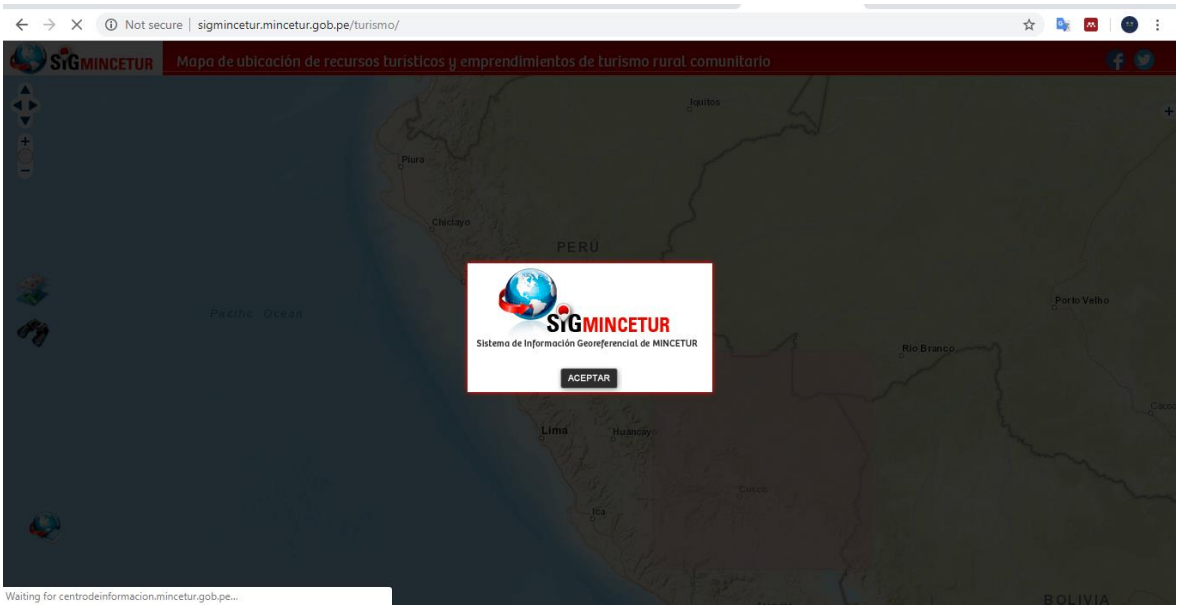
Xu, H. (2015). *Solver Tuning with Genetic Algorithms* (University of Dundee). Recuperado de https://discovery.dundee.ac.uk/en/studentTheses/eee8f5d9-ede2-4b87-af77-8142b2d0209c

ANEXOS

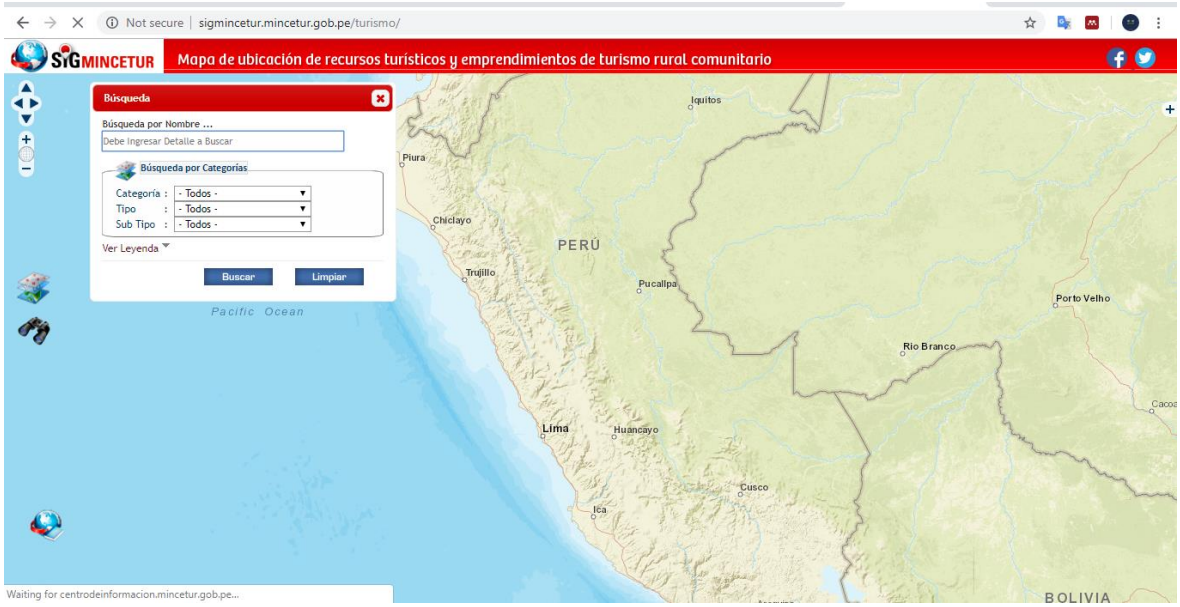
Anexo A. Obtención de datos de atractivos turísticos de inventario de MINCETUR.

El sitio que se eligió es denominado Sistema de Información Geo referencial de MINCETUR, el cual es una guía de ubicación de atractivos turísticos y turismo rural. A continuación se muestran el proceso de extracción de ubicaciones de loa atractivos para el modelo abordado en este proyecto.

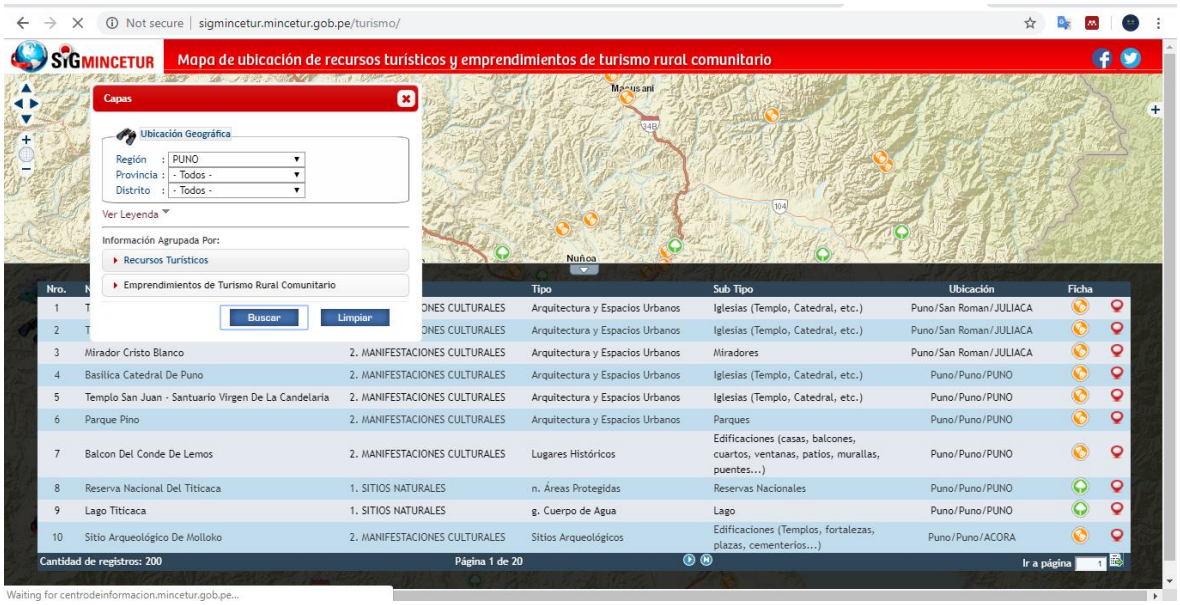
Al ingresar a http://sigmincetur.mincetur.gob.pe no encontramos con la siguiente pantalla.



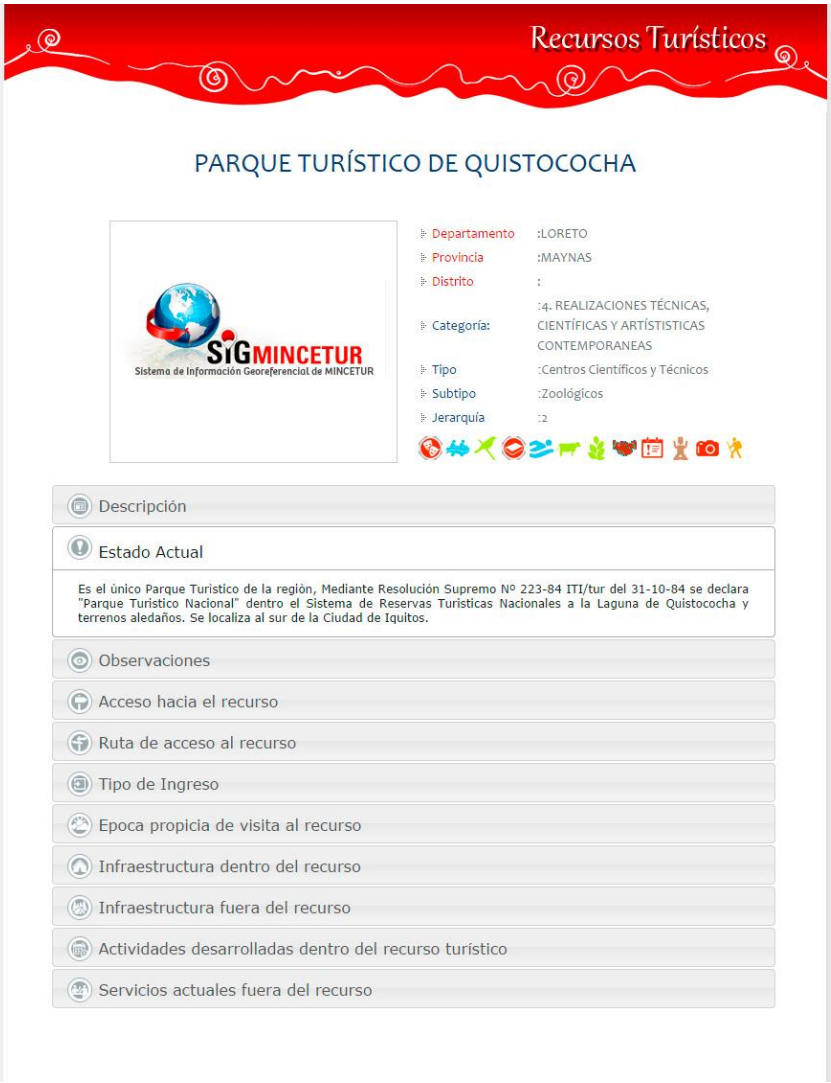
Luego de presionar aceptar tenemos la siguiente pantalla.



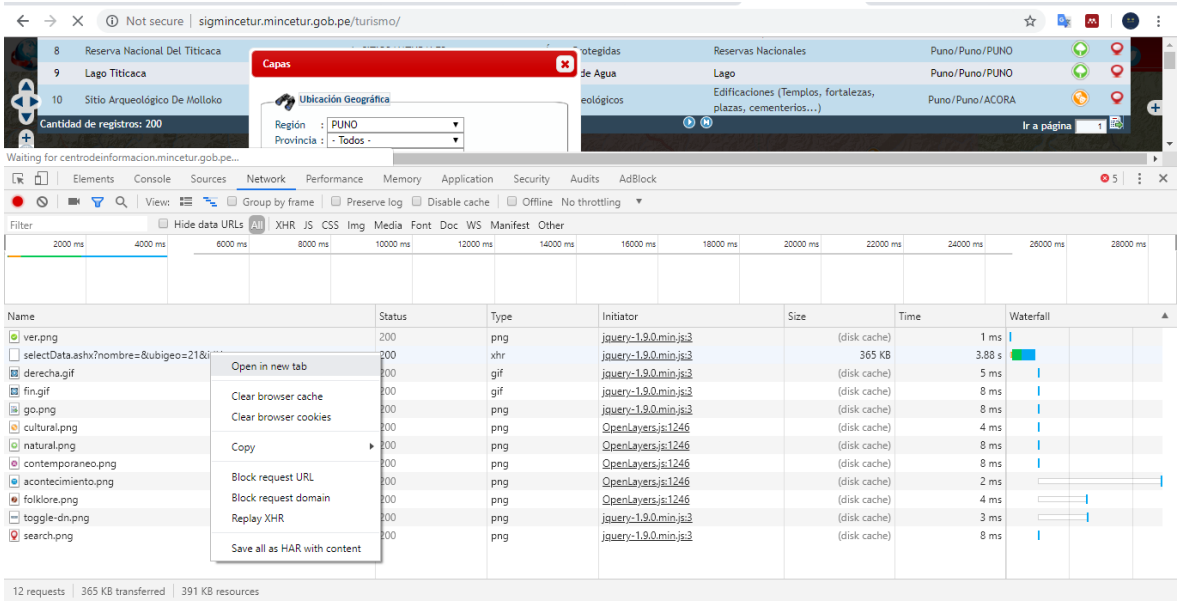
Ya que necesitamos datos de Puno cambiamos de búsqueda, luego seleccionamos el departamento de Puno y presionamos buscar y tenemos el siguiente resultado, donde indica que se tiene 200 atractivos turísticos.



Para ver de forma detallada podemos ver haciendo click en ver ficha, en el cual se tiene datos específicos para el recurso o atractivo turístico o emprendimiento seleccionado, como se puede ver en la siguiente imagen.



Para extraer datos de esta página se antes de recorrer al scraping se hizo una revisión de cómo se pueden obtener estos datos en su código fuente, en inspeccionar elemento del navegador como se muestra en la siguiente página.



Una vez encontrado la fuente, es posible expórtalo en un archivo como se muestra en la siguiente imagen, el archivo contiene datos de ubicación de los 200 atractivos, y algunos datos de la ficha como las categorías.