

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



MÉTODOS NUMÉRICOS Y OPTIMIZACIÓN

**Optimización Entera Mixta
Multiobjetivo: recomendación de
clases para *FitPass***

ROMÁN ALBERTO VÉLEZ JIMÉNEZ
ELÍAS MERCADO GONZÁLEZ
MARITRINI GARCÍA RUIZ

C.U. 165462
C.U. 150031
C.U. 151490

8 de diciembre de 2023

1. Introducción

En la práctica de la ciencia de datos, una tarea común es encontrar la decisión óptima para un conjunto de múltiples elecciones con restricciones. Por ejemplo, *en portfolio management* se busca encontrar la *mejor* combinación de acciones para invertir sujeto al presupuesto de un fondo de inversiones. Para este ejemplo, se desea encontrar qué activos i comprar y cuánto comprar x_i de cada uno, de tal manera que se maximice un función objetivo $f(\underline{x})$ sujeta a un presupuesto M y a un conjunto de restricciones A . Para este ejemplo, es posible resolverlo con varios métodos de optimización que se han visitado en clase: optimización por gradiente descendiente, métodos quasi-newton, etc. Sin embargo, ¿qué pasaría si las soluciones de las variables x_i fueran enteras? De igual forma, ¿qué pasaría si tuviéramos más de una función objetivo $f_i(\underline{x})$? Este problema se conoce como *Multiobjective Mix Integer Programming* (MOILP) y es un problema NP-Hard. El presente trabajo se enfoca en explicar qué es MOILP, cómo se resuelve y una aplicación real para un caso de negocio de una empresa de fitness.

2. Multiobjective Mix Integer Programming

■ Definición Formal

La programación mixta entera multi objetivo es un enfoque en la programación matemática en el que se optimizan múltiples funciones objetivo simultáneamente, teniendo en cuenta restricciones lineales y no lineales, donde algunas de las variables decisión tienen la restricción de tomar valores enteros. El objetivo de la programación mixta entera multi objetivo es encontrar soluciones que sean compromisos eficientes y equilibrados con base en los diferentes objetivos, considerando trade-offs entre diferentes metas y criterios [1].

■ Métodos de MOILP

Existen diferentes enfoques para abordar los problemas de programación mixta entera multi objetivo, los dos enfoques más comunes son el método de Pareto y las soluciones con peso, y la elección entre los métodos depende de los requisitos y la naturaleza del problema a resolver.

● Método Pareto

Este método se basa en la eficiencia Pareto, que decreta que una solución es eficiente si ya no es posible mejorar alguno de los objetivos sin empeorar al menos uno de los objetivos restantes. En este método se busca un conjunto de soluciones no dominadas de Pareto, siendo x una solución no dominada si no existe otra solución x' tal que $f_i(x') \leq f_i(x)$ para todo i [2].

● Soluciones con peso

Este método asigna un peso a cada objetivo para así tener la solución que optimicen una combinación ponderada de los objetivos, teniendo en cuenta la asignación de pesos. Este método permite plasmar las preferencias que se tienen para el problema de ciertos objetivos sobre otros, se ajustan los pesos para influir en la solución óptima para así reflejar las prioridades que sean relevantes en el contexto del problema. Sea $f_i(x)$ la i -ésima función objetivo y w_i un peso asociado a

esta función, un problema con peso busca minimizar $\sum_i w_i \cdot f_i(x)$, con w_i representando la importancia relativa de la función objetiva asociada a una función objetiva correspondiente [3].

- Solución Técnica (Cortes de Gomory)

Los cortes de Gomory son una técnica utilizada en programación lineal entera mixta (MILP) para mejorar la relajación lineal de un problema. La idea principal es agregar restricciones adicionales (cortes) al modelo para restringir la solución a enteros y así acercarse más a la solución óptima del problema entero [4].

Cuando se resuelve un problema MILP, se suele comenzar resolviendo la relajación lineal, que es el problema sin las restricciones de que las variables deben ser enteras. La solución de la relajación lineal proporciona una solución fraccional que luego se debe redondear o ajustar para satisfacer las restricciones enteras. Al introducir los cortes de Gomory se agregan restricciones adicionales que eliminan esa solución fraccional específica, estos cortes son generados de manera iterativa hasta que la solución de la relajación lineal se vuelve entera.

El método de corte de Gomory se basa en observar que, en la solución fraccional, al menos una variable toma un valor no entero. Se introduce una restricción adicional (corte) para forzar que esta variable (o variables) tome un valor entero, eliminando así la solución fraccional. Este proceso se repite iterativamente hasta que se obtiene una solución entera o se alcanza un criterio de terminación.

La formulación general del problema de programación lineal entera (PLE) es:

$$\begin{aligned} &\text{Maximizar } c^T x \\ &\text{sujeto a } Ax \leq b \\ &\quad x \geq 0 \\ &\quad x \in \mathbb{Z}^n \end{aligned}$$

La restricción de corte de Gomory, derivada de una solución fraccional x^* , es:

$$\sum_{i \in S} a_{ij} x_i \leq \lfloor x_j^* \rfloor$$

donde S es el conjunto de índices de las variables fraccionarias en la solución x^* , $\lfloor x_j^* \rfloor$ es la parte entera de x_j^* , y a_{ij} son los coeficientes de la matriz A .

- Solvers disponibles

En el contexto de MOILP, los solvers son herramientas computacionales que implementan algoritmos y técnicas específicas para abordar la complejidad de los problemas de optimización con múltiples objetivos y restricciones enteras. Estos solvers deben ser capaces de manejar la presencia de no linealidades, restricciones enteras, y la naturaleza multi objetivo del problema.

Algunos solvers populares para MOILP incluyen:

- Gurobi Optimization: Gurobi es un potente solver de optimización para programación lineal, entera mixta, cuadrática y convexa. Es conocido por su rendimiento eficiente y se utiliza en una variedad de aplicaciones en la industria y la investigación [5].
- IBM CPLEX: CPLEX es otro solver líder en optimización que aborda una amplia gama de problemas, incluidos problemas lineales, enteros mixtos, cuadráticos y convexos. Es utilizado en diversas industrias y sectores [6].
- COIN-OR (COIN-OR Optimization Suite): COIN-OR es un proyecto de código abierto que proporciona una colección de solvers y herramientas para problemas de optimización. Algunos solvers notables incluyen CBC (para problemas de enteros mixtos) y Clp (para programación lineal) [7].

3. Ejemplo: Fitpass

Fitpass es una empresa Mexicana que ofrece acceso a más de 1,500 estudios de fitness en todo el país, de los cuales ofrece 12 actividades A . Estas actividades $A_j \in A$ varían desde barre hasta yoga. La empresa ofrece dos planes de membresía que permiten a los usuarios asistir a un número limitado de clases K por estudio i al mes, lo único que varía entre uno y otro plan es el acceso a estudios *pro*. Cada estudio tiene un ranking r_i que va de 1 a 5, donde 5 es el mejor ranking, y este es calificado por los usuarios.

Un problema común que enfrenta un usuario de Fitpass es decidir a cuántas clases ir x_i por estudio al mes por estudio. Dado que el usuario sabe a cuántas M clases va a ir al mes, las actividades $A_j \in A$ que le gustan como también en las actividades que no le gustan y el *sentimiento* de distancia d_i que tiene que recorrer para llegar al estudio, lo que el usuario busca optimizar es minimizar la distancia que recorre, maximizar el ranking de los estudios a los que asiste y diversificar las actividades a las que asiste.

3.1. Datos del problema

En este caso, para obtener los datos de los distintos estudios afiliados a Fitpass se realizó *web scraping*, en donde se obtuvo la ubicación de cada estudio y las actividades que incluye. En total se obtuvo información de 863 estudios en la Zona Metropolitana del Valle de México. Asimismo, cada estudio tiene sus actividades. En total se encontraron 16 actividades como lo son barre, box, crossfit, ciclismo, danza, ems, funcional, gimnasio, entre otras.

Por otro lado, al no tener la información del ranking de cada estudio por parte de los usuarios, se simuló una distribución Beta multiplicada por 4 y sumándole 1 para tener un ranking artificial por estudio que se encuentre entre 0 y 5. Asimismo la idea es tener información por cada usuario. Es decir, su ubicación en coordenadas de longitud y latitud, sus actividades de preferencias, sus actividades de odio, el número máximo de clases que puede tomar por estudio, el número de clases que va a tomar en el mes y si tiene membresía *pro*. Si tiene membresía *pro*, se ponen como prioridad los estudios que son *pro*.

En resumen, se tienen dos fuentes de datos:

- La información del usuario: ubicación, actividades de preferencia, actividades de odio, tipo de membresía, número máximo de clases por estudio y número de clases a tomar en el mes.
- La información de cada estudio: Ubicación, actividades que incluye, tipo de estudio (pro o no pro).

Posteriormente se cruza la información para obtener una base de datos que contenga el id del estudio, las actividades que tiene, la distancia con respecto al usuario, el ranking de cada estudio, la preferencia del usuario (si el estudio tiene las actividades que le interesan al usuario se multiplica el ranking por dos, si el estudio tiene las actividades que el usuario odia se multiplica por cero y en otro caso se multiplica por uno), y si el estudio es *pro*.

Por último, todas las variables que se usan en el modelo se escalan por medio del escalamiento min-max para que se encuentren entre 0 y 1.

3.1.1. Sentimiento de distancia

Adicional al procesamiento de la información se consideraron tres tipos de sentimiento de distancias mediante una transformación exponencial. En dicho procesamiento se asumen tres posibles casos. Uno de ellos es que el sentimiento de distancia es totalmente lineal. Otro de ellos es cuando en un inicio no importa mucho la distancia pero posteriormente crece de forma exponencial, y el último es el inverso del anterior, en donde en un inicio el sentimiento de la distancia aumenta muy rápido pero después de cierta distancia el sentimiento ya disminuyendo cada vez mas lento. Dichos ejemplos se pueden representar en la Figura 1.

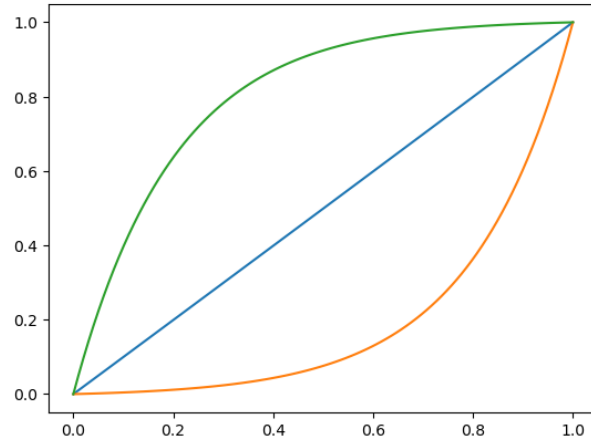


Figura 1: Formas de medir el sentimiento de distancia

El primer tipo de sentimiento de distancia podría ser un usuario que solo se puede mover a pie, por lo que, desde un inicio, las distancias largas representan un crecimiento en el sentimiento de distancia muy grande. El último tipo de sentimiento de distancia podría referirse a un usuario que tiene un automóvil, ya que, en un inicio, el aumento de las distancias podrían no representar un aumento considerable en el sentimiento de la misma.

3.2. Formulación del problema

Sea x_i el número de clases a las que asiste el usuario en el estudio i , y_i una variable binaria que indica si el usuario asiste a la actividad A_j , d_i el sentimiento de distancia que recorre el usuario para llegar al estudio, r_i el ranking del estudio y p_i una variable binaria que indica si el estudio i es un estudio *pro* o *no pro*.

Por sencillez de notación, \underline{x} es el vector de variables x_i , \underline{y} es el vector de variables y_i , \underline{d} es el vector de variables d_i y \underline{r} es el vector de variables r_i . De igual forma, se introduce el producto de Hadamard en vectores, $\underline{x} \circ \underline{y}$, que es el vector que resulta de multiplicar elemento a elemento los vectores \underline{x} y \underline{y} .

Mencionado lo anterior, el problema de optimización que busca resolver el usuario es el siguiente:

$$\begin{aligned} \min_{\underline{x}, \underline{y}} \quad & (\omega_1) \underline{x}^T \underline{d} - (\omega_2) \underline{x}^T \underline{p} \circ \underline{r} - (\omega_3) \mathbf{1}^T \underline{y} \quad \text{s.a.} \quad x_i \leq K, \\ & \sum_{i=1}^n x_i = M. \\ & y_i \leq \sum_{j \in A_i} x_j, \\ & \underline{x} \in \mathbf{Z}^n, \\ & \underline{y} \in \{0, 1\}^n. \end{aligned}$$

En donde M es el número de clases a las que el usuario va a asistir en el mes y K es el número máximo de clases que un usuario puede asistir a un estudio fijo. En este caso, se tomará a $K = 4$ ya que es una limitante real dentro de las políticas de *FitPass*.

Es importante mencionar que en la función que se busca optimizar, se escalan los componentes de la función de tal forma que el valor mínimo que puedan tomar sea -1 en el caso de la expresión que contenga a las distancias ya que se encuentra minimizando ese producto, o 1 como máximo en los otros dos casos donde se busca maximizar (preferencias y diversificación de actividades).

En el problema la suma de los pesos ω_i debe ser 1, es decir

$$\omega_1 + \omega_2 + \omega_3 = 1$$

3.3. Ponderaciones de ω_i

Como se puede ver en la formulación del problema, los pesos (ω_i) que se le asigna a la distancia (ω_1), las preferencias (ω_2) o la diversificación de las actividades (ω_3) son cruciales en los distintos resultados que se pueden obtener, por lo que se hace un análisis de los valores que puede tomar la función objetivo para distintos valores de las ω_i .

Para analizar los mínimos de la función objetivo para cada combinación de valores de ω_i , se simulieron los tres valores de ω_i por medio de una distribución Dirichlet con parámetros $\alpha = (1, 1, 1)$. Es importante mencionar que cuando se simulan valores Dirichlet de tres

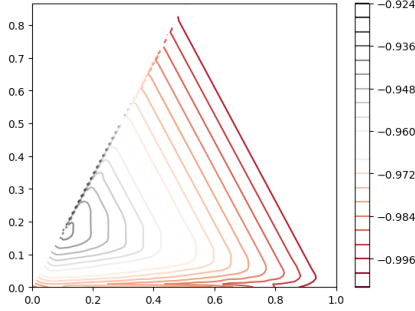


Figura 2: Curvas de nivel de la optimización para distintos valores de ω

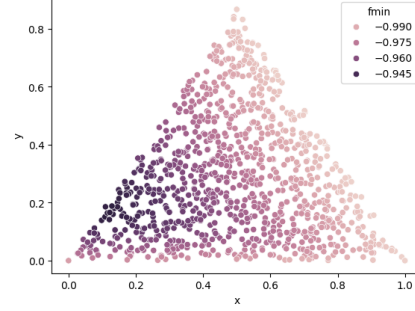


Figura 3: Resultados optimización para distintos valores de ω

dimensiones con parámetros $\alpha = (1, 1, 1)$, se simulan de forma uniforme los valores de cada entrada de tal forma que la suma de ellos siempre de 1, por lo que al final se tienen 1,000 valores de ω_1 , ω_2 y ω_3 distribuidos uniformemente con la restricción de que la suma de los 3 siempre tiene que dar uno.

Se realizó una transformación por medio de la matrix

$$A = \begin{pmatrix} 0 & \frac{1}{2} & 1 \\ 0 & \frac{\sqrt{3}}{2} & 0 \end{pmatrix}$$

de tal forma que al multiplicar ΩA^t , donde Ω es una matriz de 1,000 x 3 en donde cada renglón son los valores de ω_i a evaluar para un usuario específico, se pueda obtener una representación visual en un triángulo equilátero de todos los puntos de ω . En este caso, las combinaciones en donde haya un fuerte peso en el sentimiento de distancia se va a acercar a la esquina izquierda, en el caso que haya un mayor peso a la preferencia de los estudios, la solución se va a acercar a la esquina superior, y, por último en el caso de que se le asigne un peso importante a la diversificación, el resultado se acercaría a la esquina derecha del triángulo.

En las Figuras 2 y 3 se pueden visualizar los distintos valores que toma la función objetivo de un problema de optimización específico de un usuario para cada combinación de pesos que se le den a los distintos objetivos.

Como se mencionó anteriormente, en la 2 y 3 se pueden observar los valores que tomó la función objetivo para los distintos valores de ω_i a partir de un problema en específico. Se observan las curvas de nivel y se puede ver que los mínimos valores se alcanzan en las esquina derecha y superior del triángulo. En el caso de la esquina izquierda del triángulo, si llega a tomar valores pequeños pero dichos valores no son los mínimos. Se puede decir que el modelo obtiene sus valores mínimos cuando se le asigna completamente el peso a un factor de preferencia o diversificación de actividades o cuando se le quita por completo el peso al "sentimiento" de distancia (línea derecha del triángulo).

Los resultados obtenidos por este análisis de los pesos para un problema en específico

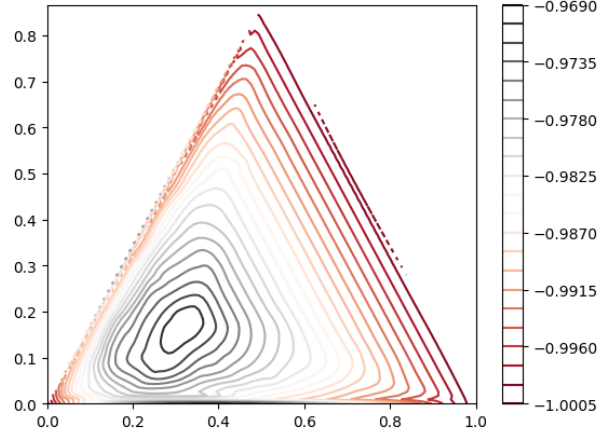


Figura 4: Curvas de nivel de la optimización para distintos valores de ω

podría interpretarse como que el usuario vive cerca de varios estudios que no están bien calificados, que no tienen sus actividades preferidas y en los cuales probablemente no haya mucha diversificación de actividades, por lo que el darle más peso a la distancia provoca que las distintas optimizaciones no lleguen a valores tan chicos como en el caso que en el que se elimina el peso del sentimiento de distancia.

En la Figura 4 se muestra el mismo análisis de pesos para otro usuario (problema de optimización distinto). En este caso se puede ver que el darle mayor peso a las distancias no están penalizando tanto el modelo, por lo que se podría asumir que los estudios cercanos no tienen tan malas preferencia y están mas diversificados.

3.4. Ejemplos de soluciones

Para una primera solución se va a suponer que el Usuario 1 se encuentra localizado en las coordenadas $(-99.1826, 19.3889)$, por lo que se encuentra en la colonia San Pedro de los Pinos en la alcaldía Benito Juárez de la CDMX. Este usuario es *pro* y tiene como preferencia el barre, yoga, ciclismo, pilates y el gimnasio y odia las actividades como crossfit y funcional. Asimismo, solo puede ir a 4 clases por estudio al mes (restricción de *FitPass*) y en total asistirá a 23 clases en el mes. En total se ajustaron cuatro modelos para este usuario en específico. En el primer modelo se le dio todo el peso a la distancia, en el segundo modelo se le dio todo el peso a las preferencias, en el tercer modelo se le dio todo el peso a la diversificación de actividades, y, por último, en el último modelo se usaron los valores de ω_i que obtuvieron los más grandes valores en las minimizaciones. (Punta de la pirámide de la Figura 2.



Figura 5: Resultados Modelo 1

3.4.1. Modelo donde el sentimiento de distancia tiene todo el peso: $\omega = (1, 0, 0)$

En la Figura 5 se muestran en el mapa los resultados del primer modelo donde se le otorga todo el peso a la distancia. En resumen se escogen los estudios de más cercanos sin tomar en cuenta las preferencias de los usuarios o la diversidad de actividades a las que se pueden asistir.

Asimismo, en el Cuadro 1 se muestran los resultados de los estudios sugeridos para este usuario.

gym name	gym times	activities	distance	preference score
Aka Fitness Gym	3	box, cycling	0.7	0.6
Cathe's Place Pilates	4	pilates	0.4	0.6
Juanca Yoga	4	yoga	0.16	0.6
Leyendas del Box Revolución	4	box	0.34	0.47
Ser Yoga	4	wellness, yoga	0.65	0.6
SmartBeat	4	running, yoga	0.19	0.6

Cuadro 1: Resultados Modelo 1

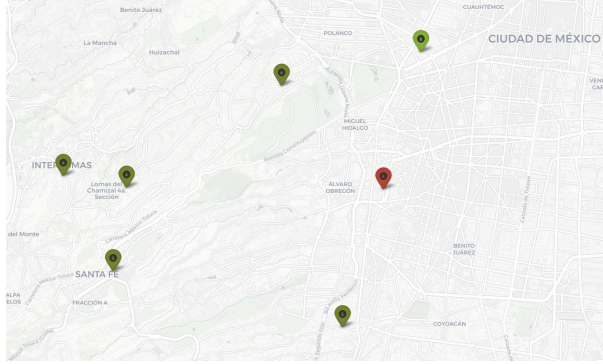


Figura 6: Resultados Modelo 2

3.4.2. Modelo donde las preferencias del estudio tienen todo el peso: $\omega = (0, 1, 0)$

En la Figura 6 se muestran en el mapa los resultados del segundo modelo donde se le otorga todo el peso a las preferencias del estudio. En resumen se escogen los estudios con mayor preferencia de los usuarios sin tomar en cuenta las distancias con respecto al usuario (hay unos muy lejos) o la diversidad de actividades a las que se pueden asistir. Este último punto es muy notorio ya que solo se recomiendan estudios con actividades funcional y pilates.

Asimismo, en el Cuadro 2 se muestran los resultados de los estudios sugeridos para este usuario.

gym name	gym times	activities	distance	preference score
P&B Studio Interlomas	4	functional, pilates	10	1
Pilatestudio360 Lomas	4	functional, pilates	4.6	1
Soulcore Altavista	4	functional, pilates	4.5	1
Soulcore Bosques	4	functional, pilates	8.2	1
Soulcore Santa Fe cerrado	4	functional, pilates	9	1
Urban Camp - Pilates Reformer	3	functional, pilates	4.5	1

Cuadro 2: Resultados Modelo 2

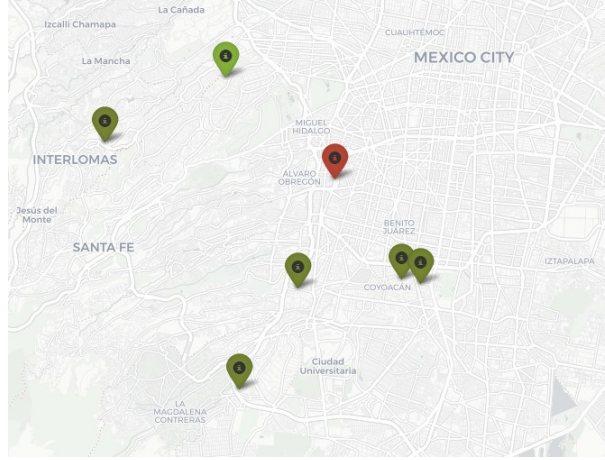


Figura 7: Resultados Modelo 3

3.4.3. Modelo donde la diversidad de actividades tiene todo el peso: $\omega = (0, 0, 1)$

En la Figura 7 se muestran en el mapa los resultados del tercer modelo donde se le otorga todo el peso a la diversidad de actividades. En resumen se escogen los estudios con mayor diversidad de actividades sin tomar en cuenta las distancias con respecto al usuario (hay unos muy lejos) o las preferencias. Se ve claramente en la Figura 7 que hay estudios lejos, pero por otro lado se ve en el Cuadro 3 que los estudios tienen una gran variedad de actividades. Asimismo hay estudios que tienen 0.6 en preferencia, por lo que hay estudios con bajas preferencias.

Asimismo, en el Cuadro 3 se muestran los resultados de los estudios sugeridos para este usuario.

gym name	gym times	activities	distance	preference score
AcademiaPrimeraPosiciónCoyoacan	4	barre, pilates, yoga, dance	4.7	0.6
Barre Flows	3	barre, pilates, yoga, dance	5.9	0.6
El T3mplo Altavista Outdoors	4	functional, box, hiit, mma, running, sports...	4.5	1
Momkikids	4	functional, box, hiit, barre, running, sports...	5.4	0.6
UFC Gym Artz Pedregalo	4	functional, box, hiit, barre, running, gym...	9	1
Urban Athlete / Balan-Cer	4	ems, functional, wellness, yoga	9.2	1

Cuadro 3: Resultados Modelo 3

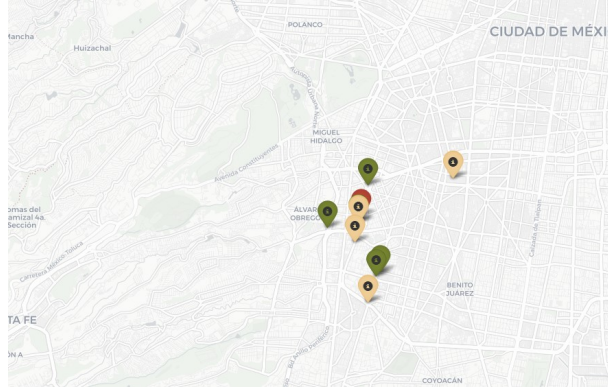


Figura 8: Resultados Modelo 4

3.4.4. Modelo variado: $\omega = (0.79, 0.19, 0.2)$

En la Figura 8 se muestran en el mapa los resultados del cuarto modelo donde los pesos se asignaron de acuerdo al punto máximo que se ve en la Figura 2. Si bien es el peor modelo para este usuario, es un buen ejemplo del comportamiento del mismo para cuando los pesos no están totalmente inclinados a cierto valor.

Asimismo, en el Cuadro 4 se muestran los resultados de los estudios sugeridos para este usuario.

gym name	gym times	activities	distance	preference score
Aka Fitness Gym	1	box, cycling, dance, functional, gym...	0.7	0.6
Dragonfly Pole Del Valle	1	barre, crossfit, dance, pilates, sports,	2.6	1
Equilibre et Relax	4	ems, wellness, yoga	1.5	1
Fever Club Del Valle Carracci	4	cycling, functional, sports, yoga	1.6	1
Juanca Yoga	3	yoga	0.16	0.6
SmartBeat	1	ems, running, wellness	90.19	0.6
Sport City Eureka	4	gym	0.92	1
Sports World Manacar	1	gym, pool	2.25	1
SSports World Patriotismo	4	gym	0.82	1

Cuadro 4: Resultados Modelo 4

Se puede notar que los resultados son variados. Si bien se inclinan más a tener gimnasios cerca, no se escogen los más cercanos ya que también entran otras variables como las preferencias de las actividades o, en menor medida, la diversificación de actividades.

3.4.5. Sensibilidad del cambio en el sentimiento de distancia

Se inicializa un nuevo usuario que vive en la colonia Nápoles. Dicho con preferencias en el ciclismo, gimnasio, danza y deportes, y odio en pilates, hiit, yoga y box. Asimismo decide ir a 26 clases en todo el mes. Se ajustan tres modelos con los tres tipos de sentimiento de distancia que se mencionaban en la sección de Datos. Para este caso se usaron los pesos $\omega = (0.65, 0.1, 0.25)$, donde se le da más peso al sentimiento de distancia, después a la diversificación de actividades y por último a las preferencias de los estudios.



Figura 9: Sentimiento de distancia *Medio*



Figura 10: Sentimiento de distancia *Bajo*



Figura 11: Sentimiento de distancia *Alto*

Los resultados obtenidos se muestran en las figuras 9, 10 y 11. Se puede ver como el sentimiento de distancia si arroja valores distintos para los mismos pesos. Por un lado, los estudios sugeridos están relativamente cercanos, por otro lado se recomiendan estudios mucho más lejanos y en el último caso los estudios se encuentran muy cerca.

4. Conclusiones

Se cumplió el objetivo del trabajo ya que se creó y se implementó un modelo en el que se optimizan tres objetivos distintos. En primer lugar se busca minimizar el sentimiento de distancia, en segundo lugar se busca maximizar las preferencias de los estudios que se sugieren y en tercer lugar se busca maximizar la diversidad de actividades. Este objetivo se cumplió con las restricciones mismas del negocio de *FitPass* como, por ejemplo, que no se puede ir más de 4 veces a un mismo estudio, se debe escoger un número de clase a las que se quieren ir y, sobre todo, que la solución es entera, ya que al final el resultado son el número de veces que se sugiere ir a un estudio en específico.

Por otro lado, es muy relevante saber que existen múltiples hiper parámetros como es el caso de los pesos que se le da a cada objetivo. Se pudo hacer un análisis de los pesos para dos problemas de optimización. Dicho análisis, por medio de una visualización en dos dimensiones, arroja resultados distintos para cada problema de optimización. Claramente es una muestra de que los pesos no funcionan de la misma manera para todos los usuarios.

Finalmente es importante mencionar la tecnología fue fundamental para la resolución de este tipo de problemas.

Referencias

- [1] L. A. Wolsey, *Integer Programming*. Wiley-Interscience, 2nd ed., 1998.
- [2] K. Deb, *Multiobjective Optimization Using the Pareto Differential Evolution Approach*. Springer, 2005.
- [3] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Springer, 2004.
- [4] R. E. Gomory, “Outline of an algorithm for integer solutions to linear programs,” *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–278, 1958.
- [5] *Gurobi Optimization*. Sitio web oficial: <https://www.gurobi.com/>.
- [6] *IBM ILOG CPLEX Optimization Studio*. Sitio web oficial: <https://www.ibm.com/analytics/cplex-optimizer>.
- [7] *COIN-OR Optimization Suite*. Sitio web oficial: <https://www.coin-or.org/>.