Facultad de Matemática y Computación

Universidad de La Habana

Tesis de diploma de la especialidad

Ciencia de la Computación

Optimización de redes de agua en Parques Ecológicos Industriales *

Autora: Paula Rodríguez Pérez

Tutoras: Dr. C. Gemayqzel Bouza Allende
MSc. Yeneit Delgado Kios

La Habana, 17 de enero de 2024

*Proyecto Nacional de Ciencias Básicas: PN223H010-005 Desarrollo de modelos y métodos matemáticos para la toma de decisiones.

RESUMEN

En esta tesis estudiamos la optimización de redes de agua en Parques Ecológicos Industriales con teoría de juego. El modelo matemático analizado tiene un líder en el nivel superior y múltiples seguidores en el inferior. La entidad reguladora del parque es el líder y las empresas son los seguidores del problema. El objetivo de la entidad reguladora es minimizar el consumo de agua fresca, sabiendo que las empresas buscan minimizar su costo operativo. Se parte de la existencia de un punto de equilibrio de Nash entre los problemas de los seguidores en el nivel inferior y estos son sustituidos por sus condiciones Karush-Kuhn-Tucker. El problema resultante es un Programa Matemático con Restricciones de Complementariedad. Con la biblioteca Pyomo de Python se construye un modelo abstracto del Programa Matemático con Restricciones de Equilibrio, que se obtiene de la transformación del problema de dos niveles. Se analizan los resultados de tres casos de estudio. Para cada uno, se crea una instancia del modelo abstracto y se resuelve con tres algoritmos incluidos en bibliotecas de Python.

Palabras clave: Parques Ecológicos Industriales, Teoría de juego, Juegos de un líder y múltiples seguidores, Programas Matemáticos con Restricciones de Equilibrio, Python, Pyomo.

ABSTRACT

In this thesis we study the optimization of water networks in Eco-Industrial Parks with game theory. The mathematical model studied has a leader in the upper level and multiple followers in the lower level. The park regulator is the leader and the enterprises are the followers of the problem. The objective of the regulator is to minimize the consumption of fresh water, knowing that the enterprises seek to minimize their annualized operative cost. We start from the existence of a Nash equilibrium point among the followers at the lower level and these problems are replaced by their Karush-Kuhn-Tucker conditions. With the Python package Pyomo, an abstract model of the Mathematical Program with Equilibrium Constraints is built, which is obtained from the transformation of the bilevel problem. For each one, an instance of the abstract model is created and solved with three algorithms included in Python libraries.

Keywords: Eco-Industrial Parks, Game theory, Single-Leader-Multi-Followers Games, Mathematical Programs with Equilibrium Constraints, Python, Pyomo.

AGRADECIMIENTOS

A todos los profesores que han sido parte de mi camino universitario. A mi tutora Yeneit por toda su dedicación en este proceso. A mi tutora Gema por su guía y paciencia durante la realización de este trabajo y no permitirme abandonar el camino. A mi familia por su apoyo incondicional.

Índice general

1. INTRODUCCIÓN

La Organización de las Naciones Unidas (ONU) define la sostenibilidad como lo que permite "satisfacer las necesidades del presente sin comprometer la habilidad de las futuras generaciones de satisfacer sus necesidades propias", es el desarrollo que garantiza el equilibrio entre sus tres pilares: el crecimiento económico, el cuidado del medio ambiente y el bienestar social (ONU, 2022).

La industrialización ha contribuido al rápido agotamiento de recursos naturales, como los combustibles fósiles y el agua. El agua está en el epicentro del desarrollo sostenible y es fundamental para la vida, la producción de alimentos, la energía y el desarrollo socio-económico. También forma parte crucial de la adaptación al cambio climático siendo un decisivo vínculo entre la sociedad y el medio ambiente. De ahí la necesidad real de que las industrias logren mantener niveles adecuados de producción con un consumo mínimo de agua. Se estima que del consumo mundial del agua el 20 % corresponde a la industria, el desarrollo industrial suele provocar el uso de grandes volúmenes de agua dulce (Boix et al., 2010, 2011).

Según la Organización de las Naciones Unidas para el Desarrollo Industrial (ONUDI) un Parque Ecológico Industrial (PEI) es: "una comunidad de empresas manufactureras y de servicios ubicadas juntas en una propiedad común. Las empresas miembros buscan mejorar el desempeño ambiental, económico y social a través de la colaboración en la gestión de asuntos ambientales y de recursos". Con este tipo de simbiosis industrial se intercambian materias primas, energía, agua y subproductos de forma tal que se fomente el desarrollo sostenible (ONUDI,

2019, 2022). Una condición básica para que un PEI sea económicamente viable es demostrar que los beneficios de cada industria vinculada son mayores trabajando de forma integrada que cuando lo hacen como instalaciones independientes (Boix et al., 2012).

Para lograr la configuración óptima en el diseño de los PEI se propone el uso de la optimización, de forma tal que las funciones objetivo y las restricciones asociadas reflejen los compromisos de este tipo de parques.

Uno de los enfoques que han sido empleados para la modelación de este problema es la optimización multiobjetivo. A diferencia de estudios anteriores en los que solo se consideraba minimizar el costo total o global de las redes de agua, existen reportes que consideran más de una función objetivo de forma tal que se consideren tanto los objetivos ecológicos, económicos, como sociales de los PEI (Boix et al., 2015).

Existen estudios que diseñan redes de agua industriales a partir de un enfoque de optimización multiobjetivo. Estos consideran como objetivos minimizar el flujo de agua fresca en la entrada de la red y el flujo de agua en la entrada de las unidades de regeneración, mientras que el número de conexiones en la red lo consideran una restricción.

En el caso de Boix et al., 2012 se desarrolla una estrategia basada en el método de la ϵ-restricción donde se formula el problema como uno de programación lineal de enteros mixtos, teniendo en cuenta las funciones objetivos anteriores. Consideran el número de conexiones en la red una restricción de igualdad. Luego de obtener el frente de Pareto se aplica una herramienta de toma de decisiones multicriterio. Esta idea es extendida para el diseño de PEI, donde consideran los siguientes casos: PEI sin unidades de regeneración, PEI donde cada empresa tiene una unidad de regeneración y PEI donde tres empresas comparten unidad de regeneración. No hay reciclaje en el mismo proceso o unidad de regeneración. Concluyen que

las soluciones obtenidas por un frete Pareto son solo resultados teóricos, es necesario continuar con la investigación sobre el costo equivalente global y el número de conexiones, dado que no requiere de inicialización puede abordar problemas de gran escala. Las unidades de regeneración conllevan ganancias significativas, que aumentan en un PEI. La mejor solución consideran que es un PEI donde cada empresa tiene su propia unidad de regeneración, existe la misma ganancia para cada empresa y un número restringido de conexiones.

En Montastruc et al., 2013 se consideran las mismas funciones objetivo que en Boix et al., 2012 y se analiza la flexibilidad de la solución encontrada en este trabajo anterior, es decir, su capacidad de recibir variaciones imprevistas en el flujo de contaminantes. Se estudia el incremento/decremento máximo del flujo de contaminantes de forma que el PEI se mantenga factible, que respete el medio ambiente y sea rentable. En el artículo se concluye que son necesarios estudios de flexibilidad en las soluciones propuestas para comprobar su interés en la práctica.

En Ramos et al., 2015 se propone por primera vez usar la programación por metas para optimizar el intercambio de agua y se estudia para el caso académico de un PEI visto en Boix et al., 2012 donde hay tres empresas. Consideran como funciones objetivos los costos de cada empresa: consumo de agua fresca, tasa de flujo de agua regenerada, costo de conexiones internas y externas entre procesos y el costo capital de las unidades de regeneración. Hallan los valores mínimo y máximo de cada función objetivo y determinan el costo mínimo de cada entidad involucrada en el PEI. Tienen en cuenta tres escenarios distintos con respecto al número de unidades de regeneración.

Al concluir en Ramos et al., 2015 se propone integrar la programación por metas con un enfoque de teoría de juegos para obtener resultados más equilibrados bajo el concepto del equilibrio de Nash. En este caso la solución está dada por el conjunto de estrategias en las que cada jugador selecciona la suya óptima simultáneamente y

en dependencia de las óptimas elegidas por los otros jugadores. Desde la perspectiva del equilibrio de Nash ninguna empresa puede desviarse en busca de mejorar su rentabilidad y ser favorecida. El problema se vería como la intención de minimizar por parte de una entidad reguladora el impacto que tiene el PEI en el medio ambiente y por parte de las empresas sus costos anuales (Ramos et al., 2016).

Para la configuración de una PEI con empresas y una entidad reguladora en Ramos et al., 2016 se plantea que se podría modelar como un juego de múltiples líderes y seguidores (MLFG por sus siglas en inglés), donde las prioridades del PEI determinará el papel de los mismos.

Un estudio realizado por Lou et al., 2004 analiza el uso de la teoría de juegos donde se tiene un equilibrio de Nash. En los casos de estudio evaluados, se evidencia que las entidades alcanzan su equilibrio de Nash para la rentabilidad y el equilibrio de Nash para la sostenibilidad con diferentes conjuntos de estrategias. La ganancia de cada jugador puede ser su rendimiento económico o medioambiental.

Otras investigaciones realizadas por Chew et al., 2009 plantean cómo la teoría de juegos puede ser utilizada para la toma de decisiones a la hora de lograr un esquema de integración de agua entre plantas óptimo en un PEI. Se analizan tanto juegos cooperativos como no cooperativos.

Un modelo de optimización difuso de dos niveles con un juego de un líder y múltiples seguidores (SLMFG por sus siglas en inglés), donde el líder es la entidad reguladora y los seguidores las empresas, fue desarrollado por Aviso et al., 2010. Se comprueban los efectos de cobrar tarifas por la compra de agua fresca y el tratamiento del agua contaminada en la optimización de una red de intercambio de agua de las plantas de un PEI.

En Ramos et al., 2016 se formulan y resuelven los PEI como MLFG y se demuestran sus ventajas respecto a los enfoques de optimización multiobjetivo. Se estudian juegos de múltiples líderes y un seguidor (MLSFG por sus siglas en

inglés) y SLMFG, tanto para casos donde hay unidades de regeneración en el parque como en los que no. El objetivo de cada empresa es minimizar su costo operativo anual, mientras que el de una autoridad reguladora es minimizar el consumo de agua fresca dentro del PEI. Para resolver el problema de dos niveles, lo transforman en un Programa Matemático con Restricciones de Equilibrio (MPEC por sus siglas en inglés): para una red dada, plantean las condiciones de Karush-Kuhn-Tucker (KKT) de cada uno de los problemas del nivel inferior, y las añaden como restricciones del problema de la autoridad.

Investigaciones posteriores realizadas por Salas et al., 2020 analizan los modelos obtenidos por Ramos et al., 2016 usando SLMFG, pero considerando un modelo abstracto llamado *Blind-Input* donde se asume que cada empresa solo puede controlar su distribución de salida y por tanto son forzadas a aceptar lo que les sea enviado a través de la red de intercambio. Esto podría ser restrictivo, ya que la empresa puede verse obligada a recibir mucha agua contaminada y a su vez tener mayores gastos que operando de forma independiente. Es por ello que las empresas tienen un contrato con la entidad reguladora en el que se garantiza una mejora relativa de su costo operativo con respecto a operar fuera del PEI. Se prueba que bajo algunas estructuras lineales de las funciones de los costos de cada empresa el modelo *Blind-Input* se transforma de un SLMFG a un solo problema de optimización de enteros mixtos.

En un trabajo realizado por Aussel et al., 2022 se formula y resuelve el problema como un SLMFG, donde el líder estará en el nivel superior de la toma de decisiones y los seguidores en el nivel inferior escogerán un punto de equilibrio de Nash generalizado. Plantea que SLMFG es más realista para el diseño óptimo de un PEI donde se quiere minimizar el consumo de agua fresca y reducir el costo operativo de cada empresa en un escenario donde las empresas no tienen que compartir información entre sí. Se tiene un modelo *Control-Input* en el que cada empresa

controla su flujo de entrada (lo que le es enviado desde las otras empresas). Los objetivos de cada entidad se definen igual que en Ramos et al., 2016.

La teoría de juegos también ha sido empleada para modelar otras situaciones. En Ramos et al., 2018 se plantea minimizar el consumo de CO_2 en un PEI y el costo de las empresas el problema como SLMFG y MLSFG. En el caso de SLMFG para las empresas se quiere minimizar el costo de operación anual, dadas las emisiones mínimas de CO_2 causadas por el consumo de productos frescos en el PEI, determinado por la autoridad. En Abapour et al., 2020 se usa para sistemas energéticos.

El enfoque generalmente utilizado para resolver problemas de optimización de dos niveles, tales como los MLFG, es formular condiciones de optimalidad, ya sean necesarias o suficientes, transformándolo en un problema de optimización de un solo nivel (Dempe et al., 2015). Esto pudiera ser con la transformación KKT del problema del nivel inferior y con la transformación del valor óptimo.

Para el caso particular de los MPEC como SLMFG en la teoría de juegos hay dos reformulaciones del enfoque optimista (Aussel & Svensson, 2020). En ambos casos la reformulación está basada en el reemplazo del Problema de Nash Generalizado (GNEP por sus siglas en inglés) del nivel inferior por un problema relacionado. Los casos son los siguientes: reemplazar el problema del nivel inferior por el (cuasi) Problema de Desigualdad Variacional y reemplazar el inferior por la concatenación de las condiciones de KKT paramétricas de cada uno de los seguidores obteniendo un Programa Matemático con Restricciones de Complementariedad (MPCC por sus siglas en inglés).

Un problema de equilibrio con restricciones de equilibrio (EPEC por sus siglas en inglés) es un programa matemático en el que se encuentra un punto de equilibrio que resuelve simultáneamente diversos MPEC. Los EPEC son empleados cuando se tienen juegos no cooperativos, particularmente MLFG, donde cada líder resuel-

ve un juego de Stackelberg formulado como un MPEC. Los MPEC compartirán variables de decisión y restricciones de equilibrio (Su, 2004).

Los enfoques fundamentales para la solución de EPEC son métodos de diagonalización, como son los métodos no lineales que realizan un ciclo resolviendo en cada iteración uno de los MPEC hasta encontrar un punto de equilibrio. Algunos de estos métodos son Gauss-Seidel y Jacobi (Leyffer & Munson, 2005; Su, 2004).

En Su, 2004 se propone un algoritmo de complementariedad no lineal secuencial (SNCP por sus siglas en inglés) para EPEC. Basándose en el esquema de regularización de Scholtes para MPEC se relajan las restricciones de complementariedad de cada MPEC y se perturban los coeficientes de la función objetivo simultáneamente, resolviendo el EPEC como una secuencia de problemas de complementariedad no lineal.

En Basilico et al., 2020 consideran que los seguidores encuentran un equilibrio de Nash una vez que el líder se compromete con una estrategia. Para casos optimistas proponen formulaciones matemáticas y algoritmos para encontrar el equilibrio y para los pesimistas una heurística *black-box*.

Liu y Jia, 2021 introducen la heurística Optimización del Enjambre de Partículas Inmunes (IPSO por sus siglas en inglés) para resolver SLMFG transformados en un problema no lineal con las condiciones KKT. Concluyen, dados los resultados numéricos, que IPSO es un método eficiente para encontrar puntos de equilibrio en SLMFG.

En Hori y Fukushima, 2019 se propone un algoritmo tipo Gauss-Seidel con una técnica de penalización para resolver EPEC asociado a MLFG y luego se sugiere un procedimiento de refinamiento para obtener soluciones más exactas. Primeramente tienen un método Gauss-Seidel con penalización (*Gauss-Seidel Penalty Method*). Con este, la solución computada puede no ser exacta, pero puede ofrecer información útil acerca de los conjuntos activos en las restricciones de complemen-

tariedad. Si los conjuntos activos son identificados correctamente, se puede refinar la solución encontrada con el primer algoritmo. Entonces desarrollan otro método Gauss-Seidel refinado (*Refined Gauss-Seidel Method*) que tiene como punto inicial la última solución dada por el primer método.

En Nie et al., 2021 usan el método Gauss-Seidel para resolver problemas de equilibrio de Nash generalizados de polinomios (GNEPP por sus siglas en inglés). Las condiciones KKT de optimalidad para cada jugador pueden ser usadas juntas con el método de tipo Newton semi-suavizado (*semismooth Newton-type method*).

Debido a la importancia económica y ecológica del diseño óptimo de PEI y la complejidad del modelo resultante, es de interés contar con métodos eficientes para su solución. En esta tesis se considerará la modelación de un PEI como un SLMFG y la transformación del problema de dos niveles mediante la sustitución del problema de los seguidores por sus condiciones KKT. Luego se emplearán distintos algoritmos incluidos en bibliotecas de Python para resolver los problemas no lineales resultantes.

La tesis está compuesta, luego del capítulo de la introducción, por un segundo capítulo donde se precisa la notación a emplear, se define formalmente un problema dos niveles con un líder y múltiples seguidores y el equilibrio de Nash, se explica la transformación a realizar al problema de dos niveles y se introduce la herramienta de Python que se va a usar. En un tercer capítulo se presenta y discute el modelo matemático enunciado en Ramos et al., 2016 y su expresión como un MPEC. Además, contiene detalles de la implementación del modelo y su solución. En un cuarto capítulo se analizan tres casos de estudio. Posteriormente se dan las conclusiones y recomendaciones del trabajo realizado.

2. PRELIMINARES

Uno de los enfoques empleados para la modelación de PEI es la teoría de juegos, ya sea juegos donde se tiene un líder y múltiples seguidores (SLMFG), varios líderes y un seguidor (MLSFG), o múltiples líderes y seguidores (MLMFG).

Si se tiene un PEI donde existe una entidad reguladora que provee el agua fresca y múltiples empresas que realizan intercambio de agua se puede modelar un problema de optimización de dos niveles con una estructura de SLMFG. En este caso la entidad reguladora sería el líder y las empresas los seguidores.

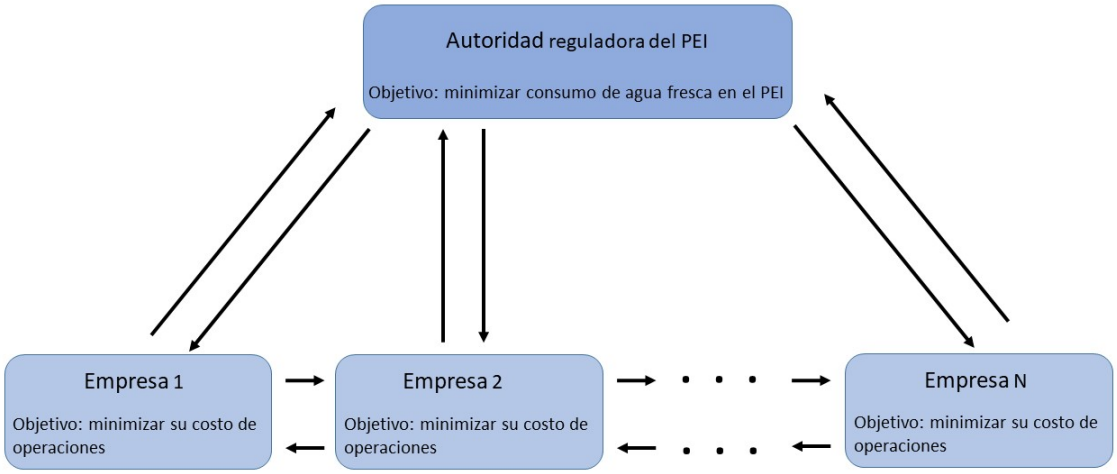

Fig. 2.1. Esquema de un PEI con una entidad reguladora y múltiples empresas

2.1. Problemas de un líder y múltiples seguidores

Sean n la cantidad de seguidores del problema, $F = \{1, \ldots, n\}$ conjunto de seguidores, w_j la variable de decisión para el problema del seguidor j ($j \in F$), w_{-j} las variables de decisión del resto de los seguidores, x la variable de decisión del líder, se tiene el siguiente SLMFG:

$$
\begin{array}{ll}
\underset{x,w}{\text{mín}} & f(x,w) \\
s.a & \left\{ \begin{array}{l}
x \geq 0 \\
g(x,w) \geq 0 \\
h(x,w) = 0 \\
\forall j \in F,\ w_j \text{ resuelve:} \\
\underset{w_j}{\text{mín}} \quad z_j(x, w_j, w_{-j}) \\
\quad s.a \left\{ \begin{array}{l}
m_j(x, w_j, w_{-j}) \geq 0 \\
p_j(x, w_j, w_{-j}) = 0 \\
w_j \geq 0, \quad \forall j \in F
\end{array} \right.
\end{array} \right.
\end{array}
$$

Problema SLMFG

El líder minimiza su función objetivo f con respecto a x, que a su vez tiene como restricciones a las funciones g y h y las soluciones de los problemas de los seguidores. Cada seguidor $j \in F$ quiere minimizar z_j bajo un conjunto de restricciones $m_j,\ p_j$.

Para la solución de este tipo de problema se quiere encontrar un punto de equilibrio de Nash entre los problemas de los seguidores que minimice el consumo de agua fresca que aporta la entidad reguladora en el PEI.

La demostración de la existencia de un punto de equilibrio en SLMFG se puede encontrar en Aussel y Svensson, 2020.

2.2. Equilibrio de Nash

Sean los problemas de optimización:

$$\begin{array}{ll} \min\limits_{w_1} & z_1(x, w_1, w_{-1}) \\ s.a\,\{ & w_1 \in K_1(w_{-1}) \end{array} \quad \cdots \quad \begin{array}{ll} \min\limits_{w_n} & z_n(x, w_n, w_{-n}) \\ s.a\,\{ & w_n \in K_n(w_{-n}) \end{array}$$

se dice que $\bar{w} = (\bar{w}_1, \ldots, \bar{w}_n)$ es un equilibrio de Nash si y solo si $\forall j \in F$ tal que $\bar{w}_j \in K(\bar{w}_{-j})$ se cumple que: $z_j(x, \bar{w}_j, \bar{w}_{-j}) \leq z_j(x, w_j, \bar{w}_{-j}),\ \forall w_j \in K_j(w_{-j})$.

2.3. Transformación de los problemas de dos niveles

Los problemas de dos niveles pueden ser reformulados en un problema de un solo nivel al reemplazar el problema del nivel inferior por las condiciones KKT de este en las restricciones del primer nivel.

Para el caso de los SLMFG donde se tienen múltiples problemas de optimización en el nivel inferior estos se sustituyen por la concatenación de las condiciones KKT de cada uno de los problemas de optimización, obteniendo un MPEC (Aussel & Svensson, 2020).

$$
\begin{array}{ll}
\underset{\substack{x,w_j \\ \mu_j,\lambda_j}}{\text{mín}} & f(x,w_j) \\
s.a & \begin{cases}
x \geq 0 \\
g(x,w_j) \geq 0 \\
\nabla_{w_j} z_j(x,w_j,w_{-j}) + \nabla_{w_j} m_j(x,w_j,w_{-j})\mu_j + \nabla_{p_j} m_j(x,w_j,w_{-j})\lambda_j = 0, \quad \forall j \in F \\
m_j(x,w_j,w_{-j}) \geq 0, \quad \forall j \in F \\
p_j(x,w_j,w_{-j}) = 0, \quad \forall j \in F \\
m_j(x,w_j,w_{-j})\mu_j = 0, \quad \forall j \in F \\
w_j \geq 0, \quad \forall j \in F \\
\mu_j \geq 0, \quad \forall j \in F
\end{cases}
\end{array}
$$

MPEC resultante

2.4. Modelación en Python

Para la implementación de modelos matemáticos en Python existen herramientas como Pyomo (*Python Optimization Modeling Objects*) que permite formular y resolver problemas de optimización complejos definiendo de forma intuitiva las variables, funciones objetivos, restricciones y parámetros que los conforman.

Pueden ser implementados problemas abstractos, a partir de los cuales se crean instancias de problemas específicos. Para el caso de los MPEC brinda herramientas para la formulación de restricciones de complementariedad.

Pyomo posee interfaces que permiten resolver y analizar los modelos empleando *solvers* externos como es el caso de Ipopt (*Interior Point Optimizer*) para problemas de programación no lineal.

La documentación de Pyomo se puede ver en «Pyomo Documentation», 2023 y Bynum et al., 2021.

3. MODELACIÓN MATEMÁTICA E IMPLEMENTACIÓN

En Ramos et al., 2016 se presentan distintos modelos para el diseño de redes de agua de un PEI. A continuación se analizará el modelo planteado como SLMFG donde hay una entidad reguladora, múltiples empresas y no se emplean unidades de regeneración.

3.1. Optimización de redes de agua en un PEI

EP	conjunto de empresas
P	conjunto de procesos
AWH	horas de operación anual del EIP
$M_{ep,p}$	carga contaminante del proceso p de la empresa ep
$Fp_{ep,p,ep',p'}$	flujo de agua entre el proceso p de la empresa ep y el proceso p' de la empresa ep' ($\langle ep, p\rangle \neq \langle ep', p'\rangle$)
$Cmax^{in}_{ep,p}$	máxima concentración de contaminante permitida para la entrada del proceso p de la empresa ep
$Cmax^{out}_{ep,p}$	máxima concentración de contaminante permitida para la salida del proceso p de la empresa ep
$Fw_{ep,p}$	flujo de entrada de agua fresca al proceso p de la empresa ep
α	precio del agua fresca
β	costo de descarga del agua contaminada
δ	costo del bombeo de agua contaminada de un proceso a otro

$$C_{ep}^{tot}(Fp_{ep}, Fp_{-ep}, Fw) = AWH \left[\alpha \sum_{p \in P} Fw_{ep,p} + \beta \sum_{p \in P} \left(Fw_{ep,p} + \sum_{ep' \in EP} \sum_{p' \in P} Fp_{ep',p',ep,p} - \sum_{ep' \in EP} \sum_{p' \in P} Fp_{ep,p,ep',p'} \right) + \delta \sum_{p \in P} \sum_{\substack{p' \in P \\ p \neq p'}} Fp_{ep,p,ep,p'} + \frac{\delta}{2} \sum_{\substack{ep' \in EP \\ ep \neq ep'}} \sum_{p' \in P} \sum_{p \in P} (Fp_{ep,p,ep',p'} + Fp_{ep',p',ep,p}) \right] \tag{3.1}$$

Cada una de las empresas $ep \in EP$ involucradas en el PEI tiene como objetivo minimizar su costo operativo anual dado por 3.1 y determinará cuánta agua envía a otros procesos en las Fp_{ep} de la forma $Fp_{ep,p,ep',p'}$ con $ep' \in EP,\ p, p' \in P,\ \langle ep, p \rangle \neq \langle ep', p' \rangle$.

El costo operativo anual planteado por 3.1 se desglosa de la siguiente manera:

$\alpha \sum_{p \in P} Fw_{ep,p}$ cantidad que debe pagar al estado por el agua fresca.

$\beta \sum_{p \in P} \left(Fw_{ep,p} + \sum_{ep' \in EP} \sum_{p' \in P} Fp_{ep',p',ep,p} - \sum_{ep' \in EP} \sum_{p' \in P} Fp_{ep,p,ep',p'} \right)$ costo de descarga del agua que desecha.

$\delta \sum_{p \in P} \sum_{\substack{p' \in P \\ p \neq p'}} Fp_{ep,p,ep,p'}$ costo del bombeo de agua de un proceso a otro de sus procesos.

$\frac{\delta}{2} \sum_{\substack{ep' \in EP \\ ep \neq ep'}} \sum_{p' \in P} \sum_{p \in P} (Fp_{ep,p,ep',p'} + Fp_{ep',p',ep,p})$ costo del bombeo de agua de un proceso a los de otra empresa.

Además el intercambio de agua entre los distintos procesos va a estar sujeto a las restricciones 3.2, 3.3, 3.4 y 3.5 descritas a continuación.

$g_{1,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)$:

$$\sum_{\substack{ep'\in EP \\ \langle ep,p\rangle\neq\langle ep',p'\rangle}} \sum_{p'\in P} Cmax^{out}_{ep',p'} Fp_{ep',p',ep,p} \leq Cmax^{in}_{ep,p} \left(Fw_{ep,p} + \sum_{\substack{ep'\in EP \\ \langle ep,p\rangle\neq\langle ep',p'\rangle}} \sum_{p'\in P} Fp_{ep',p',ep,p} \right) \ \forall p \in P \tag{3.2}$$

En 3.2 se plantea que la concentración máxima de contaminante permitida en la entrada del proceso $p \in P$ de la empresa $ep \in EP$ es menor igual que la concentración máxima permitida a la salida de los procesos que le abastecen.

$g_{2,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)$:

$$Fp_{ep,p,ep',p'} \geq 0 \quad \forall ep' \in EP,\ p, p' \in P,\ \langle ep, p\rangle \neq \langle ep', p'\rangle \tag{3.3}$$

En 3.3 se establece que el flujo de agua entre procesos es no negativo.

$h_{1,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)$:

$$M_{ep,p} + \sum_{\substack{ep'\in EP \\ \langle ep,p\rangle\neq\langle ep',p'\rangle}} \sum_{p'\in P} Cmax^{out}_{ep',p'} Fp_{ep',p',ep,p} = Cmax^{out}_{ep,p} \left(Fw_{ep,p} + \sum_{\substack{ep'\in EP \\ \langle ep,p\rangle\neq\langle ep',p'\rangle}} \sum_{p'\in P} Fp_{ep',p',ep,p} \right) \ \forall p \in P \tag{3.4}$$

La restricción 3.4 determina que la carga contaminante que aporta un proceso $p' \in P$ de la empresa $ep' \in EP$ más las concentraciones máximas de contaminantes que puede tener el agua que recibe de otros procesos debe ser igual a la concentración máxima de contaminante a la salida del proceso de toda el agua recibida por este, sea fresca o enviada por otro proceso.

$g_{3,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)$:

$$Fw_{ep,p} + \sum_{\substack{ep'\in EP \\ \langle ep,p\rangle\neq\langle ep',p'\rangle}} \sum_{p'\in P} Fp_{ep',p',ep,p} \geq \sum_{ep'\in EP} \sum_{p'\in P} Fp_{ep,p,ep',p'} \ \forall p \in P,\ \langle ep, p\rangle \neq \langle ep', p'\rangle \tag{3.5}$$

En 3.5 se plantea que la cantidad total de agua que recibe el proceso $p \in P$ de la empresa $ep \in EP$ es mayor igual a la cantidad de agua que envía a otros procesos.

El objetivo del estado está dado por la minimización de la cantidad de agua fresca que le abastece a las empresas (3.6) y se establece que la cantidad que le aporta a cada proceso de estas debe ser una cantidad no negativa (3.7).

$$\min_{\substack{Fw \\ Fp}} \sum_{ep \in EP} \sum_{p \in P} Fw_{ep,p} \tag{3.6}$$

$$s.a \begin{cases} Fw_{ep,p} \geq 0 \quad \forall ep \in EP,\ p \in P & (3.7) \\ Fp_{ep} \text{ resuelve } \forall ep \in EP \\ \quad \min\limits_{Fp_{ep}} C^{tot}_{ep}(Fp_{ep}, Fp_{-ep}, Fw) \\ \quad\quad s.a \begin{cases} \text{Ecuación 3.2} \\ \text{Ecuación 3.3} \\ \text{Ecuación 3.4} \\ \text{Ecuación 3.5} \end{cases} \end{cases} \text{Problema de la empresa } ep$$

3.2. Transformación del problema de dos niveles en un MPEC

El problema de optimización de cada empresa $ep \in EP$ será sustituido por las condiciones KKT del mismo.

Sean las variables de decisión de la empresa $ep \in EP$ de la forma $Fp_{ep,p,ep',p'}$ donde $ep' \in EP,\ p, p' \in P,\ \langle ep, p\rangle \neq \langle ep', p'\rangle$, se obtiene que los gradientes de la función objetivo y las restricciones tomarán los siguientes valores:

$$\nabla_{Fp_{ep}} C^{tot}_{ep}(Fp_{ep}, Fp_{-ep}, Fw) = \begin{cases} -\beta + \dfrac{\delta}{2} & \text{para } \ Fp_{ep,p,ep',p'},\ \langle ep, p\rangle \neq \langle ep', p'\rangle \\ \delta & \text{para } \ Fp_{ep,p,ep,p'},\ p \neq p' \end{cases}$$

$$\nabla_{Fp_{ep}} g_{1,ep,p}(Fp_{ep}, Fp_{-ep}, Fw) = \begin{cases} Cmax^{out}_{ep,p'} - Cmax^{in}_{ep,p} & \text{para } \ Fp_{ep,p',ep,p},\ p \neq p' \\ 0 & \text{en otro caso} \end{cases}$$

$\nabla_{Fp_{ep}} g_{2,ep,p}(Fp_{ep}, Fp_{-ep}, Fw) = -1$ con respecto a $Fp_{ep,p,ep',p'},\ \langle ep, p\rangle \neq \langle ep', p'\rangle$ cuando $g_{2,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)$ es $Fp_{ep,p,ep',p'} \geq 0$

$$\nabla_{Fp_{ep}} g_{3,ep,p}(Fp_{ep}, Fp_{-ep}, Fw) = \begin{cases} 1 & \text{para } \ Fp_{ep,p,ep',p'},\ p \neq p' \\ -1 & \text{para } \ Fp_{ep,p',ep,p},\ p \neq p' \\ 0 & \text{en otro caso} \end{cases}$$

$$\nabla_{Fp_{ep}} h_{ep,p}(Fp_{ep}, Fp_{-ep}, Fw) := \begin{cases} Cmax^{out}_{ep,p'} - Cmax^{out}_{ep,p} & \text{para } \ Fp_{ep,p',ep,p},\ p \neq p' \\ 0 & \text{en otro caso} \end{cases}$$

Entonces el sistema KKT para cada empresa $ep \in EP$ queda de la forma:

$$KKT_{ep} \begin{cases} \nabla_{Fp_{ep}} C^{tot}_{ep}(Fp_{ep}, Fp_{-ep}, Fw) + \sum\limits_{\substack{i=1,2,3 \\ p \in P}} \mu_{i,ep,p} \nabla_{Fp_{ep}} g_{i,ep,p}(Fp_{ep}, Fp_{-ep}, Fw) \\ \qquad + \sum\limits_{p \in P} \lambda_p \nabla_{Fp_{ep}} h_{ep,p}(Fp_{ep}, Fp_{-ep}, Fw) = 0, \quad \forall Fp_{ep,p,ep',p'},\ \langle ep, p\rangle \neq \langle ep', p'\rangle \\ g_{i,ep,p}(Fp_{ep}, Fp_{-ep}, Fw) \leq 0, \quad \forall i = \{1,2,3\},\ p \in P \\ h_{ep,p}(Fp_{ep}, Fp_{-ep}, Fw) = 0, \quad \forall p \in P \\ g_{i,ep,p}(Fp_{ep}, Fp_{-ep}, Fw)\ \mu_{i,ep,p} = 0, \quad \forall i = 1,2,3,\ p \in P \\ \mu_{i,ep,p} \geq 0 \quad \forall i = \{1,2,3\},\ p \in P \end{cases}$$

con $i = \{1, 2, 3\}$ el índice de las restricciones de desigualdad.

Sustituyendo el problema de cada empresa por su sistema KKT se tiene el siguiente MPEC:

$$\min_{\substack{Fw,Fp \\ \mu,\lambda}} \sum_{ep \in EP} \sum_{p \in P} Fw_{ep,p}$$
$$s.a \begin{cases} Fw_{ep,p} \geq 0 \quad \forall ep \in EP,\ \forall p \in P \\ KKT_{ep}\ \forall ep \in EP \end{cases}$$

3.3. Implementación del modelo

Para la implementación del PEI descrito anteriormente en Python fue usada la herramienta Pyomo. A continuación se describen los módulos desarrollados para ello.

En el módulo `eip_model.py` se tiene un modelo abstracto de tipo `AbstractModel` de `pyomo.environ` para crear las distintas instancias de los casos de experimentación.

```
model = AbstractModel(name=model_name)
```

Se declaran cuatro tipos de variables: F_w, F_p, μ, λ. Se usó la clase `Var` para representar las variables de decisión del modelo original y los multiplicadores asociados a las restricciones en la expresión lagrangiana del problema de cada empresa.

```
model.Fw = Var(model.EP_P, within=NonNegativeReals, rule=fw_rule)
model.Fp = Var(model.EP_P_EP_P, within=NonNegativeReals, rule=
fp_rule)


model.mu = Var(model.EP_P, model.C, within=NonNegativeReals, rule
=mu_rule)
model.mu_2 = Var(model.EP_P_EP_P, within=NonNegativeReals, rule=
mu_2_rule)
model.lmbd = Var(model.EP_P, within=Reals, rule=lmbd_rule)
```

Además se tienen componentes de tipo `Param` para representar los parámetros de entrada en la construcción de un modelo ($\alpha, \beta, \delta, C^{out}_{max}, C^{in}_{max}$ y M) y `Set` para indicar los índices que pueden tomar las variables.

```
model.alpha = Param(mutable=True)
model.beta = Param(mutable=True)
model.delta = Param(mutable=True)

model.Cmax_out = Param(model.EP_P, mutable=True)
model.Cmax_in = Param(model.EP_P, mutable=True)
model.M = Param(model.EP_P, mutable=True)

model.EP_P = Set(dimen=2, ordered=Set.SortedOrder)
model.EP_P_EP_P = Set(dimen=4, ordered=Set.SortedOrder)
```

En el caso de la función objetivo de la entidad reguladora se crea a partir de la clase `Objective`.

```
model.upper_level_objective = Objective(sense=minimize,
    rule=upper_level_objective_rule)
```

Las restricciones se declaran usando la clase `Constraint` y fueron indexadas en los índices de las empresas y procesos para determinar a qué problema del modelo inicial o variable está asociada.

```
model.upper_level_constraint = Constraint(
    model.EP_P, rule=upper_level_constraint_rule)

model.constraint_1 = Constraint(model.EP_P, rule=
lower_level_constraint_1_rule)

model.constraint_2 = Constraint(model.EP_P_EP_P, rule=
lower_level_constraint_2_rule)

model.constraint_3 = Constraint(model.EP_P, rule=
lower_level_constraint_3_rule)

model.constraint_4 = Constraint(model.EP_P, rule=
lower_level_constraint_4_rule)
```

```
model.mu_constraint = Constraint(model.EP_P, model.C, rule=
mu_constraint_rule)

model.mu_2_constraint = Constraint(model.EP_P_EP_P, rule=
mu_2_constraint_rule)
```

Con respecto a cada Fp se calculan las derivadas parciales de la función objetivo y las restricciones del problema del que es variable de decisión y se construye la expresión lagrangiana del sistema KKT. Luego esta es añadida como una restricción.

```
model.lagrangian = Constraint(model.EP_P_EP_P, rule=
lagrangian_expr)
```

Las restricciones de complementariedad que se incluyen con el sistema KKT de cada empresa fueron declaradas haciendo uso de la clase `Complementarity` de `pyomo.mpec`.

```
model.complementarity = Complementarity(
    model.EP_P, model.C, rule=complementarity_rule)

model.complementarity_2 = Complementarity(
    model.EP_P_EP_P, rule=complementarity_2_rule)
```

Todas las reglas para la construcción de las expresiones que definen las restricciones y funciones del problema se encuentran en el módulo `rules.py`.

Fue creada una base de datos `database.db` para guardar los parámetros de entrada y los resultados obtenidos del análisis de los distintos modelos.

Para generar las instancias de un modelo abstracto de Pyomo es necesario crear un diccionario que contenga los parámetros iniciales del problema. Se tiene una clase `Data` en `database/utils_db.py` encargada del manejo de la base de datos. En `data` se almacena el diccionario con los parámetros de entrada de los modelos

abstractos.

Para las distintas soluciones obtenidas se guardará:

- el *solver* empleado,
- los valores de las variables Fw y Fp obtenidos,
- el tiempo total que tomó la construcción del modelo y alcanzar un resultado,
- la condición de parada del algoritmo: si fue por razones desconocidas, si no hay solución factible y si se encontró el óptimo.

Se analizaron los resultados de emplear:

- El *solver* `ipopt` con la variante de considerar el problema con restricciones de complementariedad en su forma estándar. Esto se conoce como la transformación `mpec.standard_form`.
- El *meta-solver* `mpec_nlp`: esta variante primero escribe las restricciones de forma estándar y luego las relaja. Esto quiere decir que si se tiene una restricción del tipo

$$0 \leq a(x) \perp b(x) \geq 0 \tag{3.8}$$

  la misma se sustituye por

$$u = a(x),\ v = b(x)$$

$$0 \leq u \perp v \geq 0$$

- El *solver* `mpec_minlp` aplica la transformación `mpec.simple_disjunction` que transforma el MPEC en un programa disyuntivo y luego a este le aplica la

transformación *Big M*, resultando de un programa de enteros mixtos no lineal (MINLP por sus siglas en inglés). Por ejemplo en el caso de las restricciones descritas en 3.8, se sustituyen por

$$-a(x) \leq M\delta$$

$$-b(x) \leq M(1-\delta)$$

$$\delta \in \{0,1\}$$

donde M es una constante los suficientemente grande que logra acotar superiormente a las funciones $-a(x)$ y $-b(x)$.

4. EXPERIMENTACIÓN

En este capítulo se muestran tres ejemplos teóricos para comparar los algoritmos mencionados en el capítulo anterior. En cada sección se muestra un ejemplo.

Para todos los casos se consideraron los mismos valores que en Ramos et al., 2016 para los parámetros referentes a los costos de las operaciones de compra, descarga y bombeo de agua para cada empresa en el PEI.

Como punto inicial se considera que $Fp_{ep,p,ep',p'} = 0 \ \forall ep, ep' \in EP, \ p, p' \in P, \ \langle ep, p\rangle \neq \langle ep', p'\rangle$ y $Fw_{ep,p} = \dfrac{M_{ep,p}}{Cmax^{out}_{ep,p}} \ \forall ep \in EP, \ p \in P$, es decir no hay intercambio entre procesos y la entidad reguladora debe aportar toda el agua a la empresa. Esta es una solución factible, sin embargo no constituye una solución óptima.

$Parámetro$	$Valor(\$/tonelada)$
α	0.13
β	0.22
δ	$2e-2$

Tab. 4.1. Costos de operaciones

4.1. Caso de estudio 1

En este ejemplo se considera una red de agua para una empresa con cinco procesos. Si bien no es un PEI, este ejemplo sirve para ilustrar el intercambio

interno de una empresa. No sería un SLMFG, sino un problema de dos niveles ya que se considera un solo seguidor.

Empresa	*Proceso*	$C^{in}_{max}(ppm)$	$C^{out}_{max}(ppm)$	$M_{ep,p}(g/h)$
	1	0	100	500
	2	50	80	500
1	3	50	100	1000
	4	80	800	2000
	5	400	800	1000

Tab. 4.2. Caso de estudio 1.

Solver	Función objetivo	Condición de parada	Tiempo(s)
`mpec_nlp`	20.556	Desconocida	0.804
`ipopt` `mpec.standard_form`	20.556	Óptima	0.538
`mpec_minlp`	20.556	Óptima	0.045

Tab. 4.3. Resultados para el caso de estudio 1.

Se analizaron los resultados obtenidos para las variables Fw y Fp. Para los *solver* `ipopt` con transformación `mpec.standard_form`, el punto alcanzado es óptimo, y se analizaron los valores que tomó cada restricción. En el caso de las restricciones de igualdad, se cumple que su módulo es menor o igual que 10^{-4} y para las desigualdades 10^{-5}. Como se observa en los dos últimos casos la función objetivo alcanza su mínimo para el mismo valor.

Nótese que usando `mpec_nlp` la función objetivo alcanza su mínimo en el mismo valor que en los casos anteriores y las restricciones se cumplen para las mismas condiciones, sin embargo el *solver* no reporta la condición de parada del algoritmo,

si bien es una solución adecuada. Esto sucede porque en el primer caso el algoritmo que se utiliza consiste en resolver el problema de optimización no lineal en el cual se comprueba si en todas las posibles combinaciones de índices activos la condición de optimalidad se cumple. Debe acontecer que como en una de estas combinaciones este conjunto es vacío y no lo puede verificar, la condición de parada no se cumple.

Con respecto al tiempo de creación del modelo su resolución, el que menos tardó fue el *solver* `mpec_minlp`.

4.2. Caso de estudio 2

En este ejemplo se tienen dos empresas, la primera con tres procesos y la segunda con dos.

Empresa	*Proceso*	$C^{in}_{max}(ppm)$	$C^{out}_{max}(ppm)$	$M_{ep,p}(g/h)$
1	1	0	100	2000
	2	50	80	2000
	3	50	100	5000
2	1	0	100	2000
	2	50	80	2000

Tab. 4.4. Caso de estudio 2.

Solver	Función objetivo	Condición de parada	Tiempo(s)
`mpec_nlp`	140.13	Infactible	0.466
`ipopt mpec.standard_form`	130.0	Óptima	0.337
`mpec_minlp`	130.0	Óptima	0.022

Tab. 4.5. Resultados para el caso de estudio 2.

Se analizaron los resultados obtenidos para las variables Fw y Fp. Para los *solver* `ipopt` con transformación `mpec.standard_form`, el punto alcanzado es óptimo, y se analizaron los valores que tomó cada restricción. En el primer caso las restricciones de igualdad, se cumple que su módulo es menor o igual que 10^{-8} y para las desigualdades del segundo 10^{-12}. Ambos *solvers* alcanzan el mínimo para el mismo valor.

El `mpec_nlp` devuelve que la solución es infactible, como se observa en 4.5 el mínimo difiere al valor óptimo alcanzado en los otros casos, sin embargo se comprobaron las restricciones y estas se cumplían.

Con respecto al tiempo de creación del modelo y su resolución, el *solver* que menos tardó fue `mpec_minlp`.

4.3. Caso de estudio 3

Aquí se presenta el caso de estudio visto en Ramos et al., 2016, donde hay tres empresas en el PEI y cada una de ellas tiene cinco procesos.

Empresa	*Proceso*	$C^{in}_{max}(ppm)$	$C^{out}_{max}(ppm)$	$M_{ep,p}(g/h)$
1	1	0	100	2000
	2	50	80	2000
	3	50	100	5000
	4	80	800	30000
	5	400	800	4000
2	1	0	100	2000
	2	50	80	2000
	3	80	400	5000
	4	100	800	30000
	5	400	1000	4000
3	1	0	100	2000
	2	25	50	2000
	3	25	125	5000
	4	50	800	30000
	5	100	150	15000

Tab. 4.6. Caso de estudio 3.

Solver	Función objetivo	Condición de parada	Tiempo(s)
`mpec_nlp`	496.464	Infactible	2.11
`ipopt mpec.standard_form`	314.355	Óptima	4.981
`mpec_minlp`	314.355	Óptima	0.64

Tab. 4.7. Resultados para el caso de estudio 3.

Se analizaron los resultados obtenidos para las variables Fw y Fp. Para los

solver `ipopt` con transformación `mpec.standard_form`, el punto alcanzado es óptimo, y se analizaron los valores que tomó cada restricción. En el primer caso las restricciones de igualdad y de desigualdad, cumplen que su módulo es menor o igual que 10^{-4} y para el segundo las desigualdades tienen un módulo menor igual que 10^{-11} y la igualdades 10^{-7}. Ambos *solvers* alcanzan el mínimo para el mismo valor.

Al emplear `mpec_nlp`, este reporta que la solución es infactible. Las restricciones fueron verificadas y se cumplen, sin embargo como se muestra en 4.7 el valor mínimo se alcanza en un punto distinto al óptimo alcanzado con los otros *solvers*.

Con respecto al tiempo de creación del modelo su resolución, el que menos tardó fue el *solver* `mpec_minlp`.

5. CONCLUSIONES Y RECOMENDACIONES

En este trabajo estudiamos la aplicación de la teoría de juegos en la optimización de redes de agua en PEI. En este caso consideramos como líder una entidad reguladora responsable de distribuir agua fresca a las empresas, las cuales constituyen los seguidores. El objetivo de la entidad reguladora es minimizar el consumo de agua fresca, sabiendo que las empresas buscan minimizar su costo operativo.

El modelo de dos niveles estudiado se transforma al sustituir los problemas de los seguidores por la concatenación de sus condiciones KKT. El problema resultante es de un solo nivel y tendrá como variables de decisión, no solo las líder, sino las de todos los seguidores.

Nótese que la cantidad de problemas en el segundo nivel será igual a la cantidad de empresas que se consideren en el PEI, y cada una de estas tiene por cada proceso un conjunto de restricciones. Esto implica que es necesario aplicar la transformación a distintos problemas para conformar el MPEC, problema cuyas restricciones son difíciles de tratar desde el punto de vista numérico.

Para realizar la trasformación del modelo SLMFG y conformar el MPEC fue usado la biblioteca Pyomo de Python. A partir de la clase `AbstractModel()` se creó un modelo abstracto, que posibilita crear distintos casos de estudio en dependencia de los parámetros y dimensiones dadas. Un resultado de esta tesis es la forma en que se estructuraron las variables. La misma permitió cálculos más racionales al evaluar las funciones que definen la función objetivo y las restricciones del modelo. Además, es posible aplicar otros algoritmos a la solución de instancias del modelo

abstracto creado.

Al implementar el modelo se tomó la función objetivo de las empresas y las restricciones asociadas y se construyó el sistema KKT, añadiéndolo como una restricción del problema de la entidad reguladora.

Luego de la construcción del MPEC, se emplearon distintos *solvers* para resolver los casos estudiados. Los *solvers* `ipopt` con la transformación `mpec.standard_form` para las restricciones de complementariedad y `mpec_minlp` brindaron soluciones óptimas en la totalidad de los casos. De estos el que brindó mejores resultados en cuánto al tiempo de ejecución fue `mpec_minlp`.

El *solver* `mpec_nlp` reportó problemas en los tres casos. En el primero, la solución obtenida no fue dada como óptima, pero su evaluación de la función objetivo coincidió con el mínimo calculado por los otros *solvers*. En los otros dos declaró que el problema era infactible. A pesar de que se comprobó que la solución calculada cumplía con las restricciones, comparando con los resultados de los otros *solvers* se puede decir que no es óptima.

Como trabajos futuros se propone continuar con la experimentación numérica. Es de interés comprobar hasta cuántas empresas y procesos pueden incluirse en modelo, de forma tal que los *solvers* utilizados puedan manejarlos. Nótese que la cantidad de variables es cuadrática respecto a la cantidad de procesos.

Para las bibliotecas de Python que resuelven problemas de optimización es necesario darles como entrada el modelo matemático. En el caso del problema que nos ocupa, el mismo es difícil de escribir debido a la necesidad de racionalizar la organización y establecer los coeficientes que las acompañan de forma que no sea costoso su acceso. Este resultado puede aplicarse cuando se utilicen otros algoritmos.

BIBLIOGRAFíA

Abapour, S., Nazari-Heris, M., Mohammadi-Ivatloo, B., & Tarafdar Hagh, M. (2020). Game theory approaches for the solution of power system problems: A comprehensive review. *Archives of Computational Methods in Engineering*, *27*, 81-103.

Aussel, D., Cao, V. K., & Salas, D. (2022). Optimal design of exchange water networks with control inputs in Eco-Industrial Parks. *arXiv preprint arXiv:2204.09863*.

Aussel, D., & Svensson, A. (2020). A short state of the art on multi-leader-follower games. *Bilevel optimization: Advances and next challenges*, 53-76.

Aviso, K. B., Tan, R. R., Culaba, A. B., & Cruz Jr, J. B. (2010). Bi-level fuzzy optimization approach for water exchange in eco-industrial parks. *Process Safety and Environmental Protection*, *88*(1), 31-40.

Basilico, N., Coniglio, S., Gatti, N., & Marchesi, A. (2020). Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games. *EURO Journal on Computational Optimization*, *8*(1), 3-31.

Boix, M., Montastruc, L., Azzaro-Pantel, C., & Domenech, S. (2015). Optimization methods applied to the design of eco-industrial parks: a literature review. *Journal of Cleaner Production*, *87*, 303-317.

Boix, M., Montastruc, L., Pibouleau, L., Azzaro-Pantel, C., & Domenech, S. (2010). Multiobjective optimization of industrial water networks with con-

taminants (S. Pierucci & G. B. Ferraris, Eds.). *20th European Symposium on Computer Aided Process Engineering*.

Boix, M., Montastruc, L., Pibouleau, L., Azzaro-Pantel, C., & Domenech, S. (2011). A multiobjective optimization framework for multicontaminant industrial water network design. *Journal of Environmental Management*, *92*(7), 1802-1808.

Boix, M., Montastruc, L., Pibouleau, L., Azzaro-Pantel, C., & Domenech, S. (2012). Industrial water management by multiobjective optmization: from individual to collective solution through eco-industrial parks. *Journal o Cleaner Production*, *22*, 85-97.

Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., Woodruff, D. L., et al. (2021). *Pyomo-optimization modeling in python* (Vol. 67). Springer.

Chew, I., Tan, R., Foo, D., & Chiu, A. (2009). Game theory approach to the analysis of inter-plant water integration in an eco-industrial park. *Journal of Cleaner Production*.

Dempe, S., Kalashnikov, V., Pérez-Valdés, G., & Kalashnykova, N. (2015). *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*.

Hori, A., & Fukushima, M. (2019). Gauss–Seidel method for multi-leader–follower games. *Journal of Optimization Theory and Applications*, *180*(2), 651-670.

Leyffer, S., & Munson, T. (2005). Solving multi-leader-follower games [EPECs should be solved by a single nonlinear optimization problem simultaneously for all leaders. Our approach exploits the insight gained from applying nonlinear solvers to MPECs. In particular, we derive an NCP formulation of the EPEC based on the equivalence between the KKT conditions of the MPEC and strong stationarity. This NCP formulation is analyzed further

and we derive equivalent MPEC and nonlinear programming (NLP) formulations. We also introduce the notion of price consistency and show that it leads to a restricted square NCP that can be solved by applying standard NCP solvers.]. *Preprint ANL/MCS-P1243-0405*, *4*(04).

Liu, L.-P., & Jia, W.-S. (2021). An Intelligent Algorithm for Solving the Efficient Nash Equilibrium of a Single-Leader Multi-Follower Game. *Mathematics*, *9*(5), 454.

Lou, H., Singh, K., & Huang, Y. (2004). A game theory approach for emergy analysis of industrial ecosystem under uncertainty. *Clean Technologies and Environmental Policy*.

Montastruc, L., Boix, M., Pibouleau, L., Azzaro-Pantel, C., & Domenech, S. (2013). On the flexibility of an eco-industrial park (EIP) for managing industrial water. *Journal of Cleaner Production*, *43*, 1-11.

Nie, J., Tang, X., & Xu, L. (2021). The Gauss–Seidel method for generalized Nash equilibrium problems of polynomials. *Computational Optimization and Applications*, *78*, 529-557.

ONU. (2022). *Sostenibilidad*. https://www.un.org/es/impacto-acad%C3%A9mico/sostenibilidad

ONUDI. (2019). *Manual para la caja de herramientas de la ONUDI sobre Parques Eco-Industriales*.

ONUDI. (2022). *Eco-Industrial Parks*. https://www.unido.org/our-focus-safeguarding-environment-resource-efficient-and-low-carbon-industrial-production/eco-industrial-parks

Pyomo Documentation. (2023). https://pyomo.readthedocs.io/_/downloads/en/stable/pdf/

Ramos, M., Boix, M., Aussel, D., Montastruc, L., & Domenech, S. (2016). Water integration in eco-industrial parks using a multi-leader follower approach. *Computers and Chemical Engineering*, *87*, 190-207.

Ramos, M., Boix, M., Aussel, D., Montastruc, L., Vilamajo, P., & Domenech, S. (2015). Water Exchanges in Eco-industrial Parks through Multiobjective Optimization and Game Theory. *Computer Aided Chemical Engineering*, *37*, 1997-2002.

Ramos, M., Rocafull, M., Boix, M., Aussel, D., Montastruc, L., & Domenech, S. (2018). Utility network optimization in eco-industrial parks by a multi-leader follower game methodology. *Computers and Chemical Engineering*, *112*, 132-153.

Salas, D., Van, K. C., Aussel, D., & Montastruc, L. (2020). Optimal design of exchange networks with blind inputs and its application to Eco-Industrial parks. *Computer and Chemical Engineering*.

Su, C.-L. (2004). A sequential NCP algorithm for solving equilibrium problems with equilibrium constraints [Sequential NCP Algorithm para resolver EPEC]. *Manuscript, Department of Management Science and Engineering, Stanford University, Stanford, CA*.