

Universidad de la República
Facultad de Ingeniería
Instituto de Computación
Departamento de Investigación Operativa

Proyecto de Grado para obtención del
título:
Ingeniero en Computación

"Herramienta para la
optimización de los usos del
suelo en predios de pequeños y
medianos productores rurales"

Autores: Federico Matonte y Martin Calcagno
Supervisor: Antonio Mauttone

Montevideo Uruguay
Octubre de 2021

Resumen

Una correcta planificación de la rotación de cultivos es un factor fundamental para el desarrollo de un modelo productivo sustentable. Sumado esto a la tendencia global a la concientización de las sociedades sobre el efecto medioambiental que la producción agrícola tiene sobre el planeta, se vuelve prioritaria la creación de herramientas y tecnologías que puedan generar planificaciones de rotaciones de cultivos que consigan un balance entre las necesidades productivas, medioambientales, y sanitarias.

En el marco de una línea de investigación interdisciplinaria se genera una de estas herramientas, una heurística Greedy Randomized Adaptive Search Procedures (GRASP), centrada en el contexto de la producción lechera. La heurística busca obtener soluciones de calidad para el problema de planificación de rotación de cultivos. Problema en el que dado un horizonte de planificación (generalmente en estaciones), una región de estudio (compuesta de parcelas pertenecientes a productores), y un conjunto de usos productivos (con estaciones iniciales de plantado, duración en estaciones, valores de productividad, valores de exportación de fósforo, y reglas de precedencia de plantado), se debe definir qué uso productivo se planta en cada parcela en cada estación. Además, esta planificación debe tomar en consideración restricciones que fuercen tanto mínimos de productividad, como diversidad de usos productivos en cada estación para cada productor, pudiendo combinar linealmente el cumplimiento de dichas restricciones con la exportación de fósforo de la planificación para obtener una función objetivo que permita comparar las soluciones.

Este proyecto aborda dos aspectos: (i) el modelo y algoritmo de optimización, y (ii) el desarrollo de un Sistema de Soporte a la Decisión (DSS por sus siglas en inglés).

Sobre el modelo y el algoritmo, se busca agregar un mecanismo que priorice los distintos criterios que dan calidad a una solución, junto a un mecanismo de actualización que permita alterar la priorización según las soluciones obtenidas hasta el momento. Realizando para el algoritmo un estudio de su comportamiento, que permita una mejor comprensión de su desempeño y sugerir valores para la configuración paramétrica según el tamaño de la instancia a resolver. Utilizando estos valores se realizan ejecuciones del algoritmo para instancias representativas del problema, para luego analizar los registros buscando visualizar el recorrido realizado por cada búsqueda en el espacio de soluciones, determinar el intercambio entre la exploración y la explotación, y comparar el desempeño con una variante del algoritmo.

Sobre el desarrollo del DSS, se busca que otorgue usabilidad a la herramienta y permita caracterizar visualmente las soluciones. Para esto se diseña e implementa una interfaz gráfica de usuario enfocada en los conceptos principales del problema: usos productivos a ser utilizados en regiones que junto a configuraciones paramétricas definen instancias del problema, las cuales al ser resueltas devuelven soluciones. Se agregan mecanismos de alta, baja, modificación, y persistencia para los conceptos, generando herramientas de visualización gráficas y geográficas para las soluciones, y un mecanismo de listas priorizadas que permita almacenar las mejores soluciones encontradas durante la búsqueda.

Palabras claves: Heurística, DSS, Problema de planificación de rotación de cultivos, GRASP, Planificación de uso de suelo.

Tabla de contenidos

Resumen	1
Tabla de contenidos	3
1 - Presentación	5
1.1 - Contexto, Antecedentes y Motivación	5
1.2 - Objetivos	7
1.2.1 - Estudio Computacional	7
1.2.2 - Desarrollo de un Sistema de Soporte a las Decisiones	7
1.3 - Resultados esperados	7
1.4 - Estructura del documento	7
2 - Marco Teórico	9
2.1- Planificación de Rotación de Cultivos	9
2.2- Sistemas de Soporte a las Decisiones (Decision Support Systems)	9
2.3 - Caso de estudio	10
2.3.1 - Componente Geográfico	11
2.4 - Definición del problema	11
2.4.1 - Conceptos	11
2.4.2 - Formulación implícita	14
3 - Solución Heurística	17
3.1 - Descripción de la solución Heurística	17
3.1.1 -GRASP	17
3.1.2 -Construcción de la solución inicial	19
3.1.3 -Búsqueda Local	19
3.1.4- First Improve	20
3.1.5- Función Objetivo	21
3.1.6 - Pesos y su actualización	22
3.1.7 - Detalles de la implementación	23
3.2 - Experimentos Computacionales	24
3.2.1 - Descripción del plan de experimentación	24
3.2.2 - Parámetros de actualización de pesos.	25
3.2.3 - Estudio del algoritmo	26
Experimento 1: Fósforo e incumplimiento de restricciones durante la búsqueda local	27
Restricciones unidas	27
Cantidad de usos distintos	32
Productividad	35
Experimento 2: Desempeño de la búsqueda local.	38
Experimento 3: Aumento de la exploración en la búsqueda local.	39
Experimento 4: Actualización alternativa de pesos.	40
3.2.4 - Conclusiones de los experimentos.	41

4 - Desarrollo del Sistema de Soporte a las Decisiones	43
4.1 - Requerimientos y Análisis	44
4.1.1 - Requerimientos no funcionales	44
4.1.2 - Requerimientos funcionales	44
4.1.3 - Modelo Conceptual	45
4.1.4 - Casos de uso	45
4.1.5 - Definición de Conceptos del Dominio	49
4.2 - Diseño	50
4.2.1 - Arquitectura	50
4.2.2 - Capa de presentación	51
Glosario de elementos de la interfaz	51
Pestaña de Inicio	53
Pestaña de usos	54
Ventana de uso ampliado	55
Pestaña regiones	56
Ventana de mapa de una región	57
Pestaña de configuraciones paramétricas	58
Ventana de configuración paramétrica	59
Pestaña de problemas	60
Ventana de problema	61
Ventana de selección de usos de un problema	62
Ventana de selección de región de un problema	63
Ventana de selección de configuración paramétrica de un problema	64
Pestaña de historial de resultados	65
Ventanas de errores	67
4.2.3 - Capa lógica	67
Diagrama de clases	68
Soluciones no dominadas	69
4.2.4 - Capa de persistencia	70
4.3 - Implementación	71
4.4 - Ejecución de un caso típico de uso	71
4.5 - Despliegue	72
4.6 - Testeo y puesta en producción	72
5 - Conclusiones y trabajos a futuro	75
5.1 - Conclusiones	75
5.2 - Trabajos a futuro.	76
Referencias	79
Anexo	81
Documentos adicionales	81

1 - Presentación

Se estima que un 46% de la superficie terrestre se encuentra destinada a la agricultura (Memmah et al., 2015). En cada uno de los modelos productivos que cubren dicha superficie, se debe decidir cómo balancear la necesidad productiva de asegurar una viabilidad económica, con la responsabilidad de asegurar una sustentabilidad medioambiental, volviendo a la correcta utilización del uso de suelos con fines agrícolas, un problema de vital importancia (Steffen et al., 2011).

Dicho problema ha sido abordado en varias ocasiones utilizando métodos cuantitativos para la planificación de la producción (Clarke, 1989; Klein et al., 2005; Dettlesen y Leck, 2007), permitiendo generar modelos que estimen los efectos de la actividad agrícola. Para luego aplicar sobre estos modelos, métodos de optimización que permitan hallar las planificaciones productivas óptimas según los intereses priorizados. Este enfoque tiene dos debilidades principales: la alta complejidad que los métodos cuantitativos pueden alcanzar (volviendo demasiado costoso, en tiempo y cómputo, obtener una solución óptima), y la barrera técnica que el lenguaje matemático implica para los posibles usuarios (técnicos agrónomos que suelen planificar y regular, o productores que suelen aprobar e implementar los planes).

Un enfoque alternativo que ha ganado popularidad en los últimos años es la utilización de Heurísticas para obtener soluciones sub óptimas pero de calidad considerable (Ridier et al., 2016; Li et al., 2015 y Rodrigues et al., 2011) y utilizar Sistemas de Soporte a la Decisión (DSS por sus siglas en inglés) para asistir a los interesados a optar por la solución de mayor conveniencia (Zhai et al., 2020 y Rupnik et al., 2019). Bajo estas dos premisas es que este trabajo intenta brindar la información necesaria para validar el comportamiento de una heurística y generar un prototipo de DSS que facilite su utilización y asista a seleccionar entre las soluciones reportadas.

1.1 - Contexto, Antecedentes y Motivación

Este proyecto surge en el marco de una línea de investigación interdisciplinaria que nuclea Ingenieros, Biólogos y Agrónomos, pertenecientes a la Facultad de Ingeniería, el Centro Universitario Regional del Este (CURE), y el Instituto Nacional de Investigación Agropecuaria (INIA), llevada adelante por Antonio Mauttone, Lorena Rodríguez-Gallego, y Oscar Blumetto, que tiene un fuerte eje en la generación de herramientas para reducir el impacto medioambiental de los modelos productivos agrícolas. La trayectoria de dicho grupo puede resumirse en los siguientes hitos:

- Tesis de Doctorado en Ciencias Biológicas “Eutrofización de las lagunas costeras de Uruguay : impacto y optimización de los usos del suelo” por parte de la Dra. Rodríguez-Gallego en el 2011.
- Tesis de Maestría en Geociencias “Optimización de usos del suelo para prevenir floraciones nocivas de fitoplancton en la Laguna de Rocha, Uruguay” por parte de la Mg. Cabrera en el 2015.
- Proyecto Final de Licenciatura en Ciencias Biológicas “Modelos de optimización y multiatributo para la asignación de usos del suelo en la cuenca de la Laguna de Rocha” por parte de la Lic. Barletta en el 2017.

- Proyecto Final de Licenciatura en Computación. “Modelado del problema de Planificación de Rotación de Cultivos utilizando Programación Entera Mixta” por parte del Lic. Matonte en el 2019.
- Tesis de Maestría en Ciencias Agrarias “Modelación del uso de suelo en una cuenca lechera para minimizar la exportación de nutrientes hacia aguas superficiales.” por parte del Ing. Castagna en el 2019.
- Proyecto de Grado de Ingeniería de Producción. “Optimización de usos del suelo en el contexto de rotación de cultivos considerando aspectos productivos y medioambientales” por parte de los estudiantes de Ingeniería de Producción Agustín Araujo, Agustina Parrilla, Santiago Mancebo, y Valentina Moncalvo en el 2020.

Del 2018 a la fecha, varios de los trabajos del grupo de investigación se centran en el contexto de pequeños productores lecheros situados en la cuenca del río Santa Lucía, particularmente en el contexto de pequeños productores lecheros situados en la sub-cuenca del arroyo Tala, donde la actividad debe considerar tanto la sustentabilidad económica (de interés para los productores lecheros por ser uno de los métodos menos costosos de alimentar su ganado) como la sustentabilidad ambiental (por ser considerada prioritaria y estratégica por las entidades reguladoras). En particular, el río Santa Lucía provee de agua potable a más de la mitad de la población del país (F. Quintans et al., 2020) volviendo prioritario controlar la exportación de fósforo producida por la actividad agrícola, para evitar que la eutrofización (contaminación del agua producida por exceso de nutrientes) de la cuenca (Rodríguez-Gallego et al., 2012), y la floración asociada (aumento significativo de cianobacterias y algas eucariotas del fitoplancton) causen un aumento en la toxicidad del agua consumida (Haakonsson et al., 2017).

El proyecto del INIA “Co-innovación para la Sostenibilidad de la Producción Familiar Lechera en la zona sur del país” incluyó como objetivo implementar un modelo de optimización que encuentre soluciones al problema de rotación de cultivos intentando minimizar la exportación de fósforo al río sin perjudicar a los productores. En el contexto de dicho proyecto, el modelo de optimización se resolvió de forma aproximada mediante una heurística GRASP (Feo y Resende, 1995), quedando pendiente la comparación de las soluciones obtenidas con las óptimas, la determinación de parámetros del algoritmo que mejoran la calidad de las soluciones obtenidas, y la generalización de la herramienta para poder contemplar otras instancias del problema. Continuando con el trabajo realizado, este proyecto busca realizar un estudio del comportamiento del algoritmo, así como un ajuste de su implementación, la elaboración de una interfaz gráfica que facilite a los usuarios la definición de nuevas instancias del problema y la comparación de distintas soluciones generadas por el modelo de optimización.

1.2 - Objetivos

1.2.1 - Estudio Computacional

El estudio del algoritmo se centra en dos principales preguntas: Primero ¿Cuáles son los valores paramétricos que generan un buen desempeño de la heurística, en términos de optimalidad y factibilidad de las soluciones? Para permitir brindar al usuario final la información necesaria para conseguir mejores soluciones. Y segundo ¿Cómo se mueve el algoritmo en el espacio de soluciones durante su ejecución? Buscando visualizar y caracterizar el balance que se genera entre la exploración del espacio de soluciones, en busca de regiones pobladas por soluciones de calidad y la explotación de la experiencia acumulada durante la búsqueda (Blum y Roli, 2001), así como entre la optimalidad y la factibilidad de la solución.

1.2.2 - Desarrollo de un Sistema de Soporte a las Decisiones

El diseño e implementación de una interfaz gráfica, tiene como objetivo generar una herramienta que sea funcional al tipo de usuario técnico con un perfil agronómico que pueda generar nuevas instancias del problema y estudiar la viabilidad real de las soluciones encontradas por la herramienta. Agregando a la herramienta la capacidad de almacenar las soluciones evaluadas por el algoritmo durante la búsqueda que, sin ser necesariamente óptimas o factibles, puedan ser de interés para el usuario.

1.3 - Resultados esperados

Se espera obtener un mejor conocimiento del comportamiento del algoritmo GRASP, definiendo valores de la configuración paramétrica que permitan un mejor desempeño para el caso de estudio. Realizando un estudio computacional del algoritmo, que permita determinar qué modificaciones o ajustes al algoritmo, son necesarias para gestionar el compromiso entre optimalidad y factibilidad, y en términos generales la diversidad de las soluciones que se presentan al usuario.

También es de interés brindar al usuario una interfaz gráfica que permita agilizar y simplificar la creación de instancias del problema, permitiendo ampliar el rango de acción a otros problemas de producción agrícola. Otorgar un entorno con el cual ejecutar el algoritmo y presentar las soluciones encontradas de forma ordenada con representaciones visuales que faciliten su comprensión y comparación.

1.4 - Estructura del documento

Este documento cuenta con cinco secciones principales. El segundo capítulo, *Marco Teórico*, tiene como objetivo definir el problema de planificación de rotación de cultivos. El tercer capítulo, *Solución Heurística*, describe la herramienta heurística investigada y explica los distintos experimentos y modificaciones realizadas sobre ella, junto a sus resultados. El cuarto capítulo, *Desarrollo del Sistema de Soporte a las Decisiones*, aborda el proceso de ingeniería del software que genera la interfaz gráfica, y brinda el seguimiento de un uso típico de la misma. Finalmente el quinto capítulo, *Conclusiones y Trabajos futuros*, lista las conclusiones generadas durante el trabajo así como las oportunidades de mejoras que quedaron por fuera de su alcance.

2 - Marco Teórico

2.1- Planificación de Rotación de Cultivos

La práctica de Rotación de Cultivos implica el alternar en un suelo, plantaciones de diversas variedades y con diferentes necesidades nutricionales durante distintos ciclos productivos. Sus objetivos suelen ser: asegurar cuotas productivas, reducir costos económicos, disminuir el efecto nocivo de la producción sobre el suelo, y/o detener la reproducción de enfermedades o plagas que afecten a una sola variedad de cultivos (Ridier et al., 2016; Clarke, 1989; Li et al., 2015; Rodrigues et al., 2011).

La Planificación de Rotación de Cultivos busca ir un paso más allá, generando secuencias de rotaciones de cultivos más extensas que no solo busquen cumplir con los objetivos, sino también cuenten desde el inicio con una lista ordenada de plantaciones a cultivar, permitiendo estimar los resultados positivos y negativos del proceso productivo (Ridier et al., 2016). El problema de Planificación de Rotación de Cultivos es la búsqueda de dichas secuencias de plantaciones a cultivar.

Este proyecto aborda en particular el problema con un enfoque de optimización, definiendo un objetivo medioambiental junto con consideraciones productivas y sanitarias. Buscando que las planificaciones sugeridas minimicen la exportación de fósforo, restrinjan la cantidad de usos distintos a utilizar por cada productor, y marquen cuotas de productividad para cada estación.

2.2- Sistemas de Soporte a las Decisiones (Decision Support Systems)

Dada la ausencia de una definición universalmente aceptada para los Sistema de Soporte a las Decisiones, se señalan algunas de las características principales relevadas al momento de diseñar el DSS presentado en este trabajo.

Un DSS consta de un sistema informático que brinda apoyo a un proceso de toma de decisión sobre un problema (Sheng y Zhang, 2009), en este caso, qué planificación de rotación de cultivos se debe seguir. Debe centralizar datos relevantes al problema, generando información a partir de ellos y desplegarla de forma adecuada (Taechatanasat and Armstrong, 2014). Brindando una interfaz conveniente al usuario objetivo, evitando que la herramienta complejice la tarea de decisión (Terribile et al., 2015).

Es deseable, para aumentar su utilidad, que no se centre en resolver instancias puntuales del problema (en este caso, los productores de la cuenca del Río Santa Lucía), sino que se pueda trabajar con el problema genérico, sobre el cual es posible definir instancias que resolver. Y que cuente con herramientas visuales (tablas, gráficas, y mapas) que faciliten una mayor comprensión de la instancia del problema a trabajar y las soluciones propuestas.

Se mencionan ejemplos de DSS orientados a la actividad agrícola: Agrícolas AgriSupport II system (B. Recio et al., 2003), AgroDSS (Rupnik et al., 2019), y LANDMARK (M. Debeljakm et al., 2019).

2.3 - Caso de estudio

En la cuenca del río Santa Lucía (que provee agua a más de la mitad de la población del país), se encuentran casi el 50% de los tambos lecheros del país (INALE, 2014). En particular, la subcuenca del arroyo Tala (retratada en la figura 1), cuenta con 41 productores (con los que el grupo de investigación trabaja) que poseen 792 parcelas independientes con una superficie total de 114 km². Estos productores necesitan alimentar su ganado, y por su tamaño reducido y estructura de costos, la manera más ventajosa de hacerlo es mediante el cultivo de pasturas (llamadas usos en este proyecto) en sus parcelas. Cada una de estas parcelas es trabajada con una periodicidad estacional (siendo posible cambiar los usos productivos solo al cambiar la estación), pudiendo plantarse un solo uso productivo a la vez, sin poder cambiar el uso hasta completar su ciclo productivo.

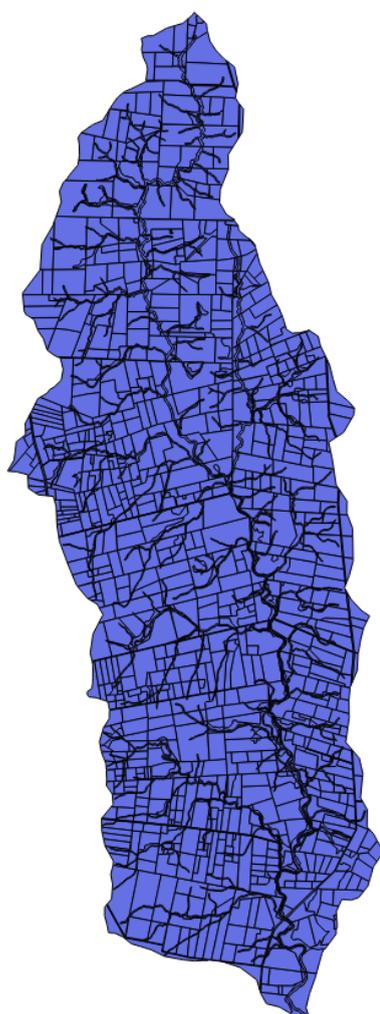


Figura 1 - Cuenca del Río Santa Lucía, subcuenca arroyo Tala

Los usos tienen un ciclo productivo, durante el cual en cada estación tendrán una productividad (cantidad de Materia Seca medida en Kg por hectárea), y exportarán fósforo al suelo (medido en kg por hectárea). Y una vez completado su ciclo productivo restringirán qué otros usos pueden ser plantados a continuación.

2.3.1 - Componente Geográfico

Dado que las parcelas poseen un atributo geoespacial, siendo normal que se trabaje con ellas a través de programas de Sistemas de Información Geográfica (SIG), se decide priorizar la compatibilidad de los archivos ESRI Shape (.shp, .shx y .dbf), por ser una forma estándar de intercambio de información geográfica. Marcando estos archivos como medio de importación de las regiones al sistema y definiendo un estándar sobre los datos necesarios para que el DSS pueda trabajar con ellos.

Sobre las salidas, cabe mencionar que hasta el momento los usuarios del sistema utilizan las planificaciones de rotación de cultivo en programas SIG, generando a partir de ellos: mapas, análisis, y reportes, lo que condiciona las decisiones que se deban tomar al momento de desarrollar el DSS.

2.4 - Definición del problema

Previo a poder enmarcar el problema de Planificación de Rotación de Cultivos en un modelo de optimización es necesario especificar los principales conceptos que incluye. Pudiendo, a partir de estos describir el problema y generar una formulación que formalice el mismo. Optando por una formulación implícita debido a: la necesidad de contar con una solución operativa al problema de optimización en un tiempo acotado (por la dependencia de otras actividades del proyecto) y el desconocimiento de la dificultad de lograr una formulación exacta.

2.4.1 - Conceptos

- **Usos Productivos (usos):** Un uso es un tipo de plantación a ser utilizada en una parcela de un productor. Sobre este uso es necesario saber la duración en estaciones, de su ciclo productivo. Para cada periodo de este ciclo se debe conocer la producción de materia seca (medida en Kg por hectárea) y exportación de fósforo (medido en Kg por hectárea) que el uso puede generar. Finalmente es necesario saber qué otros usos pueden ser plantados en la misma parcela luego de terminar el ciclo productivo para lograr disminuir posibles riesgos sanitarios (reproducción de pestes y plagas). A modo de ejemplo la tabla 1 detalla los usos productivos manejados en el caso de estudio.
- **Región:** Una región es un conjunto de parcelas pertenecientes a productores. Mostrándose en la figura 2 la región Cuenca del Río Santa Lucía, subcuenca arroyo Tala, separada entre los productores que la integran.
- **Productor:** Dueño de parcelas en las que se desean plantar usos con el objetivo de satisfacer su necesidad de producción de materia seca, intentando exportar el mínimo fósforo posible. A su vez, con la idea de controlar riesgos, es de interés de los productores asegurar cierta diversidad entre los usos plantados en cada estación. Sobre los productores solo es necesario conocer las parcelas que poseen, sus necesidades de materia seca en los distintos periodos del ciclo productivo, y cuán diversa debe ser su lista de usos a plantar.

N°	USO	Estación plantación	Próximos posibles	Coeficiente de Productividad (kg MS/ha)				COEFICIENTE EXPORTACIÓN FÓSFORO (kg P/ha)			
				Otoño	Invierno	Primavera	Verano	Otoño	Invierno	Primavera	Verano
1	Alfalfa (1° año)	Otoño	2 al 11 y 14	0	0	3025	2475	0,513	0,513	0,513	0,513
	Alfalfa (2° año)			3000	1200	4200	3600	0,513	0,513	0,513	0,513
	Alfalfa (3° año)			1600	800	3200	2400	0,513	0,513	0,513	0,513
	Alfalfa (4° año)			700	350	3500	2450	0,513	0,513	0,513	0,513
2	FE+TB+L (1° año)	Otoño	1, 3 al 11 y 14	0	450	3150	900	0,345	0,345	0,345	0,345
	FE+TB+L (2° año)			2500	1500	4000	2000	0,345	0,345	0,345	0,345
	FE+TB+L (3° año)			1400	1120	3500	980	0,345	0,345	0,345	0,345
	FE+TB+L (4° año)			700	650	3000	650	0,345	0,345	0,345	0,345
3	TR+Cebadilla (1° año)	Otoño	1, 2, 4 al 11 y 14	0	2000	4400	1600	0,43	0,43	0,43	0,43
	TR+Cebadilla (2° año)			2800	2000	4200	1000	0,43	0,43	0,43	0,43
4	TR+TB+Raigrás (1° año)	Otoño	1, 2, 3, 5 al 11 y 14	1161	2212	3768	1483	0,43	0,43	0,43	0,43
	TR+TB+Raigrás (2° año)			2176	1780	3495	1483	0,43	0,43	0,43	0,43
5	Lotus Puro (1° año)	Otoño	1 al 4, 6 al 11 y 1	558	1075	2127	962	0,473	0,473	0,473	0,473
	Lotus Puro (2° año)			1318	1107	2826	1387	0,473	0,473	0,473	0,473
	Lotus Puro (3° año)			1134	828	2340	971	0,473	0,473	0,473	0,473
6	Raigrás+TB+L (1° año)	Otoño	1 al 5, 7 al 11 y 14	775	1811	3159	816	0,345	0,345	0,345	0,345
	Raigrás+TB+L (2° año)			1840	1749	3342	1365	0,345	0,345	0,345	0,345
	Raigrás+TB+L (3° año)			1318	1287	2523	999	0,345	0,345	0,345	0,345
7	Achicoria (1° año)	Otoño	1 al 6, 8 al 11 y 14	65	232	2653	2849	0,3	0,3	0,3	0,3
	Achicoria (2° año)			1572	2342	3193	NE	0,3	0,3	0,3	NE
8	Moha (anual)	Primavera	1 al 7, 10, 11, 14	NE	NE	0	5000	NE	NE	0,393	0,393
9	Sorgo Forrajero (anual)	Primavera	1, 2, 5, 7, 8, 9 y 12, 14	3568	NE	0	9687	0,548	NE	0,548	0,548
10	Avena pastoreo (anual)	Otoño	1, 2, 5, 7, 8, 9 y 12, 14	1625	3250	1625	NE	0,52	0,52	0,52	NE
11	Avena+Raigrás Temp. (anual)	Otoño	8, 9 y 12, 14	2083	2211	2735	NE	0,52	0,52	0,52	NE
12	Maiz (anual)	Primavera	1 al 7, 10, 11, 14	NE	NE	0	12882	NE	NE	0,56	0,56
13	Campo Natural	No se planta	1 al 14	675	425	1039	932	0,168	0,168	0,168	0,168
14	Rastrojo	No se planta	1 al 12, 14	0	0	0	0	0,244	0,244	0,244	0,244

Tabla 1 - Usos Productivos

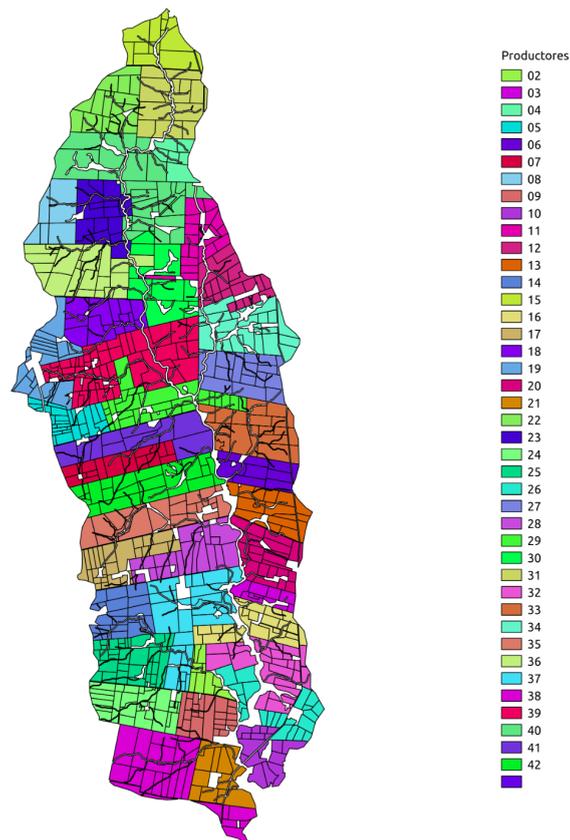


Figura 2 - Cuenca del Río Santa Lucía, subcuenca arroyo Tala

- **Parcela:** Es una unidad delimitada de tierra perteneciente a un productor, la cual puede ser trabajada de forma autónoma, posibilitando plantar en ella un uso para producir materia seca a costo de exportar fósforo. Sobre las parcelas debe conocerse su ubicación, área, topología, y uso plantado antes de comenzar la plantación. En este trabajo el concepto de parcelas es referenciado también como píxel.
- **Estación:** Período de tiempo en la que se divide un año (otoño, invierno, primavera, y verano). En este proyecto se utiliza este término como medida de tiempo de plantación de usos, en los que los usos y los productores deben cumplir restricciones.
- **Productividad:** Es la producción que tiene un productor en una determinada parcela (de determinada superficie), en un cierto período de tiempo (medido en estaciones).
- **Problema de Planificación de Rotación de Cultivo:** Se entiende una planificación de rotación de cultivo como la definición de qué uso debe ser plantado en cada parcela de cada productor de una región para cada periodo del horizonte de planificación. El problema de planificación de rotación de cultivos abordado en este trabajo consta no solo de poder producir planificaciones viables (aquellas que cumplen con las precedencias definidas entre usos), sino también de lograr encontrar entre ellas cuáles son las que reducen el impacto ecológico de la actividad productiva, pero sin descuidar la necesidad de diversidad entre los usos, ni la de producción de materia seca. Formalizando esta definición, el objetivo del problema es encontrar soluciones que minimicen

la exportación de fósforo, pero cumplan restricciones respecto a la cantidad de usos distintos que utiliza cada productor y la productividad promedio que obtienen en cada estación. La figura 3 permite apreciar una de las principales dificultades del problema de optimización combinatoria, que es compatibilizar las restricciones definidas en términos temporales (dimensión horizontal) con las definidas en términos espaciales (dimensión vertical).

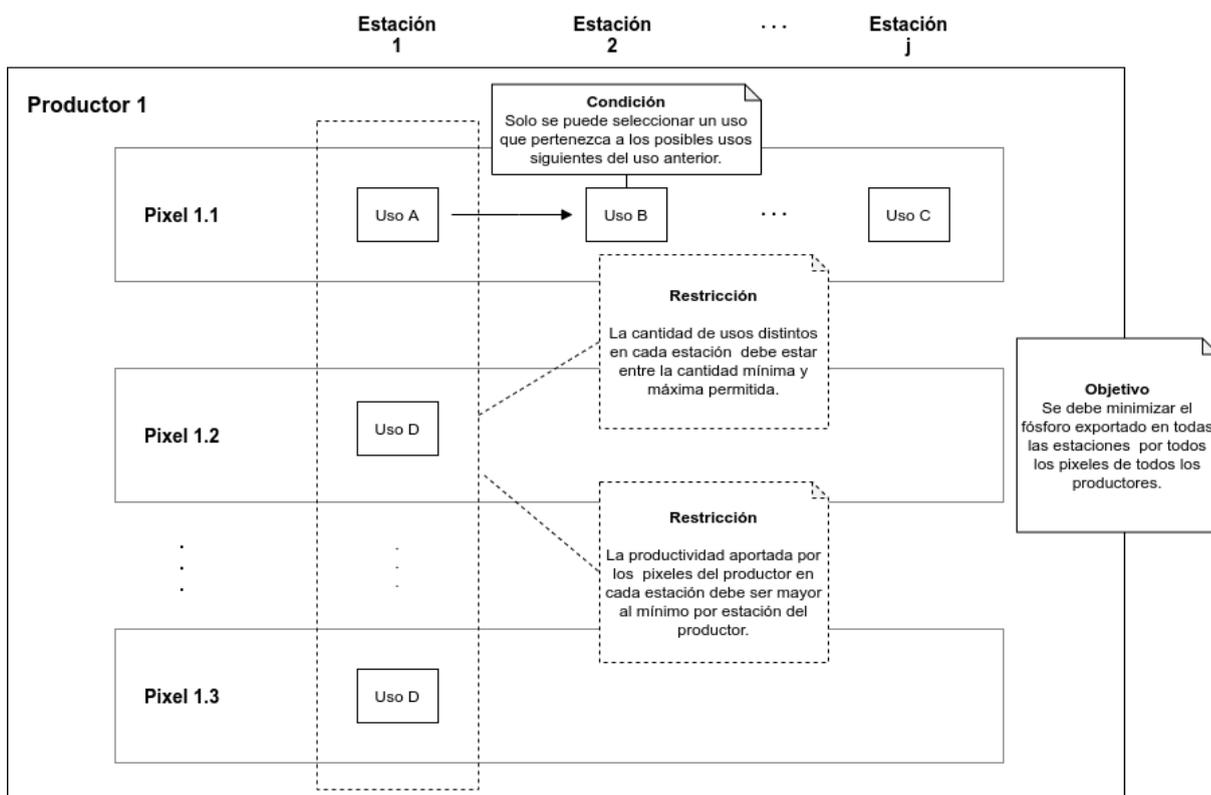


Figura 3 - Condiciones, restricciones y objetivos de una Planificación de Rotación de Cultivo

2.4.2 - Formulación implícita

Se modela el problema como una instancia del problema de optimización de programación entera en el que las variables de decisión indican si un uso es utilizado en una parcela durante una estación. Pudiendo, a partir de estas variables de decisión definir de forma explícita la función objetivo a minimizar como la sumatoria del producto de la variable de decisión por el producto de la superficie de cada parcela con el coeficiente de exportación de cada uso.

Se explicita el cumplimiento de la precedencia entre usos (obligando que en la estación siguiente a terminar el ciclo productivo de un uso en una parcela, solo se pueda plantar usos que pertenezcan al conjunto de usos siguientes del primero), y las restricciones de productividad (exigiendo estación por estación que cada productor supere su mínima productividad requerida).

Se deja implícita la restricción de diversidad de usos por requerir de un modelado de mayor complejidad. De la misma forma la explicitación de la restricción de la primera estación de un uso así como el cumplimiento del ciclo productivo quedan de momento postergados, utilizando esta formulación. Estas condiciones se verificarán y asegurarán de forma algorítmica en la resolución heurística.

Conjuntos

P	Conjunto de productores.
J	Conjunto de parcelas.
$J_p \subset J$	Conjunto de parcelas que pertenecen al productor $p \in P$.
K	Conjunto de usos de Suelos.
D	Horizonte de planificación, definido como la repetición de las estaciones (otoño, invierno, primavera, y verano), tantos años como se quiera planificar.
$K_{kd} \subset K$	Conjunto de usos permitidos para la estación $d + 1$ dado que el uso $k \in K$ es usado en la estación $d \in [1, \dots, D - 1]$, para cualquier parcela.

Parámetros

$a_j > 0$	Área de la parcela $j \in J$, expresado en [ha].
$f_{kd} \geq 0$	Exportación de fósforo para el uso $k \in K$, en la estación $d \in D$, expresado en [kg/ha].
$g_{kd} \geq 0$	Productividad para el uso $k \in K$, en la estación $d \in D$, expresado en [kg/ha].
MIN_PROD	Productividad mínima para cualquier productor en cualquier estación, expresado en [kg/ha] de materia seca.
MIN_USOS [MAX_USOS]	Mínima (Máxima) cantidad de usos distintos para cualquier productor en cualquier estación.

Variables de Decisión

$x_{jkd} \in [0, 1]$	Igual a 1 si el uso $k \in K$ es utilizado en la parcela $j \in J$ en la estación $d \in D$, 0 en caso contrario.
$y_{jd} \in K$	Uso utilizado en la parcela $j \in J$ en la estación $d \in D$. Derivado de x_{jkd} .
$z_{pd} \in N$	Cantidad de usos distintos utilizados en la estación $d \in D$ por el productor $p \in P$. Derivado de x_{jkd} .

Restricciones

$y_{j(d+1)} \in K \quad y_{jd} \forall j \in J, d \in [1, \dots, D - 1]$	(1) Los usos respetan las reglas de precedencia.
$\sum_{j \in J, k \in K, d \in D} a_{jkd} x_{jkd} \geq MIN_PROD \forall p \in P, d \in D$	(2) Todo productor cumple con la productividad mínima en toda estación.
$MAX_USOS \geq z_{pd} \geq MIN_USOS \forall p \in P, d \in D$	(3) La cantidad de usos distintos por productor para cualquier estación se encuentra entre los límites superior e inferior.

Función Objetivo

$\sum_{j \in J, k \in K, d \in D} a_{jkd} x_{jkd}$	(4) Minimizar el fósforo total exportado.
--	---

Debido a las definiciones implícitas de las variables y_{id} y z_{fd} , no es posible expresar la formulación en un formato estándar de programación matemática, lo que evita conocer la naturaleza del problema (lineal, no lineal) o identificar un método exacto para la resolución de esta formulación. Se debe notar que tampoco representa explícitamente los comienzos de plantaciones ni las duraciones de cada uso.

Finalmente, es importante mencionar que ninguna de las restricciones vincula de forma alguna a las planificaciones de distintos productores. Esto permite separar cualquier instancia con varios productores en varias instancias de un solo productor, pudiendo resolver por separado las planificaciones individuales.

3 - Solución Heurística

3.1 - Descripción de la solución Heurística

El problema de planificación de rotaciones de cultivos puede ser conceptualizado como encontrar una solución suficientemente buena en el espacio de todas las posibles planificaciones (cumplan o no las restricciones, y sean o no factibles). Este espacio crece exponencialmente según la complejidad de la instancia a resolver, lo que vuelve inviable el análisis de cada solución y compromete el uso de una formulación exacta a la posesión de licencias de software especializado para programación matemática, que cuenten con la inteligencia necesaria para reducir el esfuerzo de la búsqueda. Habiendo descartado estas opciones, se decide utilizar una metaheurística de trayectoria, que trabaja con una sola solución a la vez (Blum y Roli, 2001), optando por un procedimiento de búsqueda voraz, aleatorio y adaptativo, GRASP por sus siglas en inglés Greedy Randomized Adaptive Search Procedures (detallado en el pseudocódigo 1) basado en la creación de una solución inicial seguida de búsqueda local (Feo y Resende,1995).

```
Procedimiento GRASP()  
  cargar_instancia()  
  Mientras condicion_de_parada = False  
    solucion <- crear_solucion()  
    solucion <- Local_Search(solucion)  
    mejor_solucion <- actualizar_solucion(solucion, mejor_solucion)  
  fin Mientras  
  devolver mejor_solucion  
fin GRASP
```

Pseudocódigo 1 -Traducción del GRASP genérico propuesto por Feo y Resende.

3.1.1 -GRASP

Greedy Randomized Adaptive Search Procedures (Feo y Resende,1995) es una Metaheurística normalmente aplicada en los problemas de Optimización Combinatoria. A nivel básico, consta de varias iteraciones de la generación aleatoria de una solución original mediante un algoritmo voraz (representada por la función `crear_solucion_aleatoria()` en el pseudocódigo 2), seguida de una iteración de mejoras mediante una búsqueda local que parte la solución original, pero puede terminar en una muy diferente. Dada la generación aleatoria de la solución original, cada exploración de la búsqueda local tiene un nuevo punto de partida.

Se identifican dos propiedades a cumplir por la solución, el cumplimiento de las tres restricciones (respetar la secuencia de cultivos durante toda la planificación, asegurar productividad mínima por estación para cada productor, y asegurar la diversidad de usos por estación para cada productor), y la reducción de la exportación de fósforo. Dada la naturaleza del problema, es obligatorio respetar la secuencia de cultivos, implementando para esto un control directo sobre la selección de usos que asegure siempre su cumplimiento. Por el contrario, dada la dificultad de encontrar soluciones factibles según la restricciones de productividad y cantidad de usos distintos por

estación, junto a la mayor tolerancia por parte de los productores a aceptar este tipo de incumplimientos, se decide aceptar las soluciones que las incumplan, pero intentando reducir esta cantidad de incumplimientos. Habiendo asegurado el cumplimiento de la primera restricción, se definen tres criterios que dan calidad a una solución: baja exportación de fósforo, poca cantidad de incumplimiento de productividad, y poca cantidad de incumplimiento de cantidad de usos distintos. Criterios bajo los cuales se pueden aceptar soluciones no factibles, dando la posibilidad de que la optimalidad compita con la factibilidad.

Buscando balancear de forma automática los tres criterios, en este proyecto se agrega a la versión original del algoritmo (desarrollado en un proyecto anterior) un conjunto de pesos dinámicos que altere la importancia de cada uno de ellos (*pesos_GRASP* presentes en el pseudocódigo 2, y *pesos_LS* presentes en el pseudocódigo 4). Estos pesos alteran el valor relativo que se da a cada criterio en la función objetivo, aumentando la importancia de los que no se hayan mejorado en las últimas evaluaciones, y reduciendo la importancia de los que sí se han mejorado. Este conjunto de pesos se compone de: un subconjunto enfocado a balancear los criterios a nivel de las iteraciones de generación aleatoria de soluciones (*pesos_GRASP*), y otro subconjunto a nivel de la búsqueda local (*pesos_LS*), cada uno de ellos afecta a toda la búsqueda, pero solo se actualiza en su propio nivel.

```

Procedimiento GRASP (max_cantidad):
  base <- crear_solucion_aleatoria()
  mejor_fosforo <- base
  mejor_cant_incumplimientos_productividad <- base
  mejor_cant_incumplimientos_usos <- base
  pesos_GRASP <- inicializar_pesos()
  mejor_solucion <- local_search(base)
  Para cada i_soluciones entre 0 y max_cantidad, i_soluciones++
    nueva_solucion <- crear_solucion()
    nueva_solucion <- local_search(nueva_solucion)
    mejore <-
      (nueva_solucion.evaluarFuncionObjetivo <
       mejor_solucion.evaluarFuncionObjetivo)
    Si mejore
      Si GRASP_actualizar_con_mejora
        pesos_GRASP <- actualizar_pesos
          (mejor_solucion, nueva_solucion, pesos_GRASP)
      Fin Si
      mejor_solucion <- nueva_solucion
    Fin Si
    Si !mejore
      Si GRASP_actualizar_Sin_mejora
        pesos_GRASP <- actualizar_pesos
          (mejor_solucion, nueva_solucion, pesos_GRASP)
      Fin Si
    Fin Si
  Fin Para
  devolver mejor_solucion
Fin Procedimiento GRASP

```

Pseudocódigo 2 -GRASP Modificado

Se presenta en el pseudocódigo 2 del algoritmo, resaltando las sentencias agregadas al algoritmo original dándoles formato de negrita. Del mismo se pueden destacar: la función `crear_solucion` que cumple el papel del generador aleatorio voraz de soluciones, la función `local_search` donde se realiza la búsqueda local desde una solución actual, y la función `actualizar_pesos` que balancea los criterios de búsqueda. Estas funciones serán detalladas a continuación.

3.1.2 -Construcción de la solución inicial

Para que la heurística pueda explorar el espacio de soluciones de forma eficiente, es fundamental contar con un mecanismo de creación de soluciones iniciales, que sea capaz de generar soluciones diversas cargando una planificación consistente (que respete las precedencias entre usos) para cada pixel, detallando el proceso en el pseudocódigo 3. Para generar esta planificación, en caso de ser necesario, debe completar el ciclo productivo del uso heredado (aquel plantado en la estación previa al inicio de la planificación), para luego sortear uno de sus usos siguientes y agregarlo a la planificación, repitiendo esto hasta completar el horizonte de planificación.

```
Procedimiento crear_solucion()
  solucion <- solucion_vacia
  Para cada pixel
    solucion.cargar_pixel(pixel)
  Fin Para
Fin Procedimiento crear_solucion()

Procedimiento cargar_pixel(pixel)
  completar_ciclo_heredado(pixel)
  uso <- pixel.uso_heredado
  i_estacion <- pixel.duracion_uso_original
  uso_a_cargar <- uso.sortear_uso_siguiente()
  Mientras i_estacion < problema.cantidad_estaciones
    pixel.cargar_uso_desde(uso_a_cargar, i_estacion)
    Aumentar (iestacion, uso_a_cargar.duracion)
    uso_a_cargar <- uso_a_cargar.sortear_uso_siguiente()
  Fin Mientras
Fin Procedimiento cargar_pixel(pixel)
```

Pseudocódigo 3 - Creación de solución Inicial

3.1.3 -Búsqueda Local

Partiendo de una solución chequea sus soluciones vecinas (aquellas que solo varían en la planificación de un píxel), buscando la primera solución con un mejor valor objetivo. Este proceso se repite al menos una `max_cantidad_FI` veces (para asegurar cierta exploración del espacio de soluciones), pero deja de repetirse al fallar en encontrar una mejor solución `strikes_FI` cantidad de veces (por considerar que se logró explotar lo suficiente la región del espacio de soluciones en que se encuentra). Se opta por elegir la primera solución mejor encontrada (first improve), en vez de explorar toda la vecindad (best improve), debido a la naturaleza cambiante que los pesos aportan a la función objetivo.

El conjunto de pesos del GRASP (**pesos_GRASP**) sigue afectando la función objetivo, y se suma un segundo conjunto de pesos (**pesos_LS**) para alterar la importancia de los criterios en la función objetivo que puede ser modificado a este nivel.

```

Procedimiento local_search(solucion)
  strikes <-0
  solucion_original<-copiar(solucion)
pesos_LS <-inicializar_pesos
  Para i_soluciones entre 0 y max_cantidad_FI,
    cortando si llego a strikes> strikes_FI
    asegurando al menos i_solucion>min_cantidad_FI
  solucion <-first_improve(solucion)
  empate <- (solucion_original.evaluarFuncionObjetivo =
    solucion.evaluarFuncionObjetivo)
  mejore <- (solucion_original.evaluarFuncionObjetivo >
    solucion.evaluarFuncionObjetivo)

  Si empate
    Aumentar(strikes,1)
  SiNo
    Si mejore
      Si LS_actualizar_con_mejora
        pesos_LS <- actualizar_pesos (solucion_original,solucion,
          pesos_LS)

      Fin Si
      solucion_original<-copiar(solucion)
    SiNo
      Si LS_actualizar_sin_mejora
        pesos_LS <- actualizar_pesos (solucion_original, solucion,
          pesos_LS)

      Fin Si
      Aumentar(strikes,1)
    Fin Si
  Fin Si
  Fin Para
  devolver solucion
Fin Procedimiento local_search

```

Pseudocódigo 4 -Búsqueda local modificada

3.1.4- First Improve

Dada una solución, busca entre las soluciones cercanas una solución que tenga mejor valor objetivo. En este caso se define que una solución es cercana a otra cuando sus planificaciones son iguales salvo por la planificación de un píxel. Por lo que la búsqueda copia la solución actual, para luego elegir un píxel al azar, al que cambiar su planificación, para luego comparar sus funciones objetivo, conservando la mejor de ambas. El cambio de la planificación del píxel debe mantener el cumplimiento del ciclo productivo, quitar los usos anteriormente planificados, y sortear nuevos usos para suplantarlos. Los conjuntos de pesos siguen afectando a la función objetivo, pero los mismos no pueden ser modificados a este nivel.

```

Procedimiento first_improve(solucion)
  respaldo_solucion <- copia(solucion)
  Para intentos entre 0 y cantidad_intentos_FI
    pixel_random <- sortear_pixel()
    solucion<-cambiar_pixel(solucion, pixel_random)
    actualizar_mejores_soluciones(solucion)
    Si (solucion.evaluar_funcion_objetivo <
        respaldo_solucion.evaluar_funcion_objetivo)
      devuelvo solucion
    SiNo
      solucion<-respaldo_solucion
    Fin Si
  Fin Para
  devuelvo respaldo_solucion
Fin Procedimiento first_improve

Procedimiento cambiar_pixel(solucion, pixel_random)
  uso <- pixel_random.uso_heredado
  i_estacion <- pixel_random.duracion_uso_original
  uso_a_cargar= uso.sortear_uso_siguiete()
  Mientras i_estacion < Problema.cantidad_estaciones
    fin_uso_a_cargar <- i_estacion + uso_a_cargar.duracion
    pixel.quitar_usos_desde_hasta(i_estacion, fin_uso_a_cargar)
    pixel.cargar_uso_desde(uso_a_cargar,i_estacion)
    i_estacion <- fin_uso_a_cargar
    uso_a_cargar <- uso_a_cargar.sortear_uso_siguiete()
  Fin Mientras
Fin Procedimiento cambiar_pixel

```

Pseudocódigo 5 - Primera Mejora y Cambiar pixel

3.1.5- Función Objetivo

Dada la existencia de tres criterios para categorizar la bondad de una solución (la exportación de fósforo a minimizar como criterio principal, y la reducción de incumplimientos de productividad y cantidad de usos distintos como criterios secundarios), es necesario definir una función objetivo que las centralice y permita compararlas.

Para esto se debe calcular:

Fósforo exportado: La normalización de exportación total de fósforo producida por la planificación, calculada al sumar lo que exporta cada uso plantado en cada parcela en cada estación, y dividiendo ese valor por la máxima exportación posible (cantidad de estaciones por área total de la región por valor máximo de exportación de los usos).

$$1) \sum_{j \in J, k \in K, d \in D} a_j f_{kd} x_{jkd} / (|D| * MAX_FOSOFORO * \sum_{j \in J} a_j)$$

Incumplimientos de productividad: La normalización de la cantidad de incumplimientos de la restricción de productividad incurridos durante la planificación, calculada al contar la cantidad de estaciones que cada productor no pudo cumplir con su mínima productividad dividido la máxima cantidad de incumplimientos (horizonte de planificación multiplicado por la cantidad de productores).

$$2) |(p, d) \in P * D / \sum_{j \in J, k \in K} a_j g_{kd} x_{jkd} \leq MIN_PROD / (|D| * |P|)$$

Incumplimientos de cantidad de usos: La normalización de la cantidad de incumplimiento de la restricción de cantidad de usos distintos incurridos durante la planificación. Calculada al contar la cantidad de estaciones que cada productor plantó una cantidad de usos distintos menor a su mínima cantidad requerida, o mayor a su máxima cantidad requerida, dividido por la máxima cantidad de incumplimientos (horizonte de planificación multiplicado por la cantidad de productores).

$$3) (|z_{pd}/z_{pd} > MAX_USOS, p \in P, d \in D| + |z_{pd}/z_{pd} < MIN_USOS, p \in P, d \in D|) / (|D| * |P|)$$

Estas normalizaciones serán ponderadas por sus pesos, para obtener los valores que cada criterio aporta a la función objetivo de la solución.

```

Procedimiento evaluar_Funcion_Objetivo (solucion)
  valor_fosforo <-
    pesos_GRASP.fosforo_exportado *
    pesos_LS.fosforo *
    solucion.fosforo
  valor_incumplimiento_productividad <-
    pesos_GRASP.incumplimientos_productividad *
    pesos_LS.incumplimientos_productividad *
    solucion.incumplimientos_productividad
  valor_incumplimientos_usos <-
    pesos_GRASP.incumplimientos_usos *
    pesos_LS.incumplimientos_usos *
    solucion.incumplimientos_usos
  devolver valor_fosforo + valor_incumplimiento_productividad +
  valor_incumplimientos_usos
Fin Procedimiento evaluar_Funcion_Objetivo

```

Pseudocódigo 6 - Evaluar función objetivo modificada

3.1.6 - Pesos y su actualización

Es de interés para el algoritmo que durante la comparación de las soluciones se altere la importancia de los criterios que afectan a la función objetivo (exportación de fósforo, incumplimiento de productividad, e incumplimiento de usos distintos), permitiéndole al algoritmo escapar a óptimos locales y mejorando su capacidad de exploración del espacio de soluciones. Para esto se utilizarán dos conjuntos de pesos, el primero se actualizará a nivel del GRASP, y el segundo a nivel de la búsqueda local. Esta separación busca tener una perspectiva a corto plazo, y otra a largo plazo, buscando qué criterios son sencillos de mejorar, también intentando controlar que esto no se haga a costo de empeorar los demás criterios. El mecanismo de actualización es el mismo para ambos conjuntos de pesos, y si bien varía entre el manejo del peso del fósforo y los pesos de los dos incumplimientos, ambos pueden ser resumidos en disminuir la importancia de un criterio cuando ha mejorado y aumentarla cuando no.

La principal diferencia entre los pesos de los fósforos y de los incumplimientos radica, en que el mínimo valor de los incumplimientos es conocido (siendo alcanzado cuando la solución es factible y valiendo cero), por lo que puede reiniciarse sus pesos cuando se llega a dicho valor. En cambio la mínima exportación de fósforo no es conocida, pero se opta por reiniciar su peso cuando se consigue una mejora.

Se define como 1 el valor inicial de los pesos (PESOS_BASE), al que también retornan al reiniciarlos, representando que todos los criterios tienen la misma importancia inicial al momento de buscar mejorar. Cada actualización puede aumentar/disminuir este valor multiplicado/dividiéndolo por 1.1, permitiendo variar entre 0.01 (MIN_PESO) y 100 (MAX_PESO). Esta variación coincide con el comportamiento de la función $f(x)=1,1^x$, donde x toma valores positivos para representar la cantidad de comparaciones consecutivas sin mejora, y valores negativos para representar la cantidad de comparaciones consecutivas con mejora. Se define el coeficiente 1.1 para permitir que los valores de los pesos alcancen sus cotas al darse 49 comparaciones seguidas con el mismo resultado.

```

Procedimiento inicializar_pesos()
  return (peso_fosforo = 1, peso_inc_prod = 1, peso_inc_cant_usos=1)
Fin Procedimiento inicializar_pesos()

Procedimiento actualizar_pesos (mejor_solucion, nueva_solucion, pesos)
  Si mejor_solucion.fosforo < nueva_solucion.fosforo:
    pesos.fosforo <- minimo(pesos.fosforo *1.1, MAX_PESO)
  SiNo Si mejor_solucion.fosforo > nueva_solucion.fosforo:
    pesos.fosforo <- PESOS_BASE
  Si (mejor_solucion.inc_prod < nueva_solucion.inc_prod):
    pesos.inc_prod <- minimo(pesos.inc_prod * 1.1, MAX_PESO)
  SiNo (mejor_solucion.inc_prod > nueva_solucion.inc_prod):
    pesos.inc_prod <- maximo(pesos.inc_prod / 1.1, MIN_PESO)
  SiNo Si mejor_solucion.inc_prod = 0:
    pesos.inc_prod <- PESOS_BASE
  Si mejor_solucion.inc_cant_usos < nueva_solucion.inc_cant_usos:
    pesos.inc_cant_usos <- minimo(pesos.inc_cant_usos * 1.1, MAX_PESO)
  SiNo (mejor_solucion.inc_cant_usos > nueva_solucion.inc_cant_usos):
    pesos.inc_cant_usos <- maximo(pesos.inc_cant_usos / 1.1, MIN_PESO)
  SiNo Si mejor_solucion.inc_cant_usos = 0:
    pesos.inc_cant_usos <- PESOS_BASE
Fin Procedimiento actualizar_pesos

```

Pseudocódigo 7 - Inicialización y actualización de pesos

3.1.7 - Detalles de la implementación

Dada la implementación del algoritmo legado, la nueva versión mantuvo Java 8¹ como lenguaje, utilizando la Clase Random² para proveer de un generador de números pseudo aleatorios a las funciones crear_solucion_aleatoria(), sortear_uso_siguiete() y sortear_pixel(), permitiendo reproducir experimentos al setear el valor de la semilla que los generó.

¹ <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

² <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

3.2 - Experimentos Computacionales

Buscando una mejor comprensión del algoritmo que permita un mejor desempeño, se diseña un plan de experimentación centrado en el registro y análisis de las soluciones evaluadas durante la ejecución del algoritmo.

3.2.1 - Descripción del plan de experimentación

Se divide el plan de experimentación en dos partes.

La primera tiene como objetivo brindar a los futuros usuarios de la herramienta valores por defecto, según el tamaño de la instancia a resolver (cantidad de píxeles del productor), para los siguientes parámetros:

- **GRASP_actualizar_sin_mejora:** Define si debe actualizarse **pesos_GRASP** cuando durante una ejecución del **GRASP** se comparan dos soluciones, y la solución nueva no tiene mejor valor objetivo que la solución actual.
- **GRASP_actualizar_con_mejora:** Define si debe actualizarse **pesos_GRASP** cuando durante una ejecución del **GRASP** se comparan dos soluciones, y la solución nueva tiene mejor valor objetivo que la solución actual.
- **LS_actualizar_sin_mejora:** Define si debe actualizarse **pesos_LS** cuando durante una ejecución de un **local_search** se comparan dos soluciones, y la solución nueva no tiene mejor valor objetivo que la solución actual.
- **LS_actualizar_con_mejora:** Define si debe actualizarse **pesos_LS** cuando durante una ejecución de un **local_search** se comparan dos soluciones, y la solución nueva tiene mejor valor objetivo que la solución actual.

Para obtener estos valores, se separa a los productores en tres categorías: (pequeños, medianos y grandes), seleccionando para cada categoría, un productor representativo. Se realizaron para cada uno de estos productores 10 búsquedas de duración moderada (10.000 iteraciones del algoritmo GRASP, **max_cantidad** en pseudocódigo 2) con cada una de las 16 posibles configuraciones de actualización de pesos (listadas en la tabla 2). A partir de los desempeños de estas búsquedas se sugieren las configuraciones que obtuvieron menor exportación de fósforo promedio entre sus soluciones.

Parámetros	Número de Configuración															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GRASP_actualizar_sin_mejora	false	true	false	true	false	true	false	true	false	true	false	true	false	true	false	true
GRASP_actualizar_con_mejora	false	false	true	true	false	false	true	true	false	false	true	true	false	false	true	true
LS_actualizar_sin_mejora	false	false	false	false	true	true	true	true	false	false	false	false	true	true	true	true
LS_actualizar_con_mejora	false	false	false	false	false	false	false	false	true	true						

Tabla 2 - Posibles configuraciones paramétricas

La segunda parte del plan de experimentación consiste en estudiar el comportamiento del algoritmo intentando retratar cómo se recorre el espacio de soluciones, realizando distintos experimentos para observar su comportamiento. El estudio comienza utilizando los valores sugeridos en la parte anterior, para realizar búsquedas de duración moderada en un conjunto de cinco productores (los productores de ejemplo, y dos nuevos productores pequeños), y analizando las soluciones evaluadas durante la búsqueda para discernir un patrón de comportamiento común a las búsquedas. Centrándose luego en los tres productores pequeños para evaluar el comportamiento del local search, evaluando el desempeño de búsquedas de duración breve (1000 iteraciones del algoritmo GRASP, `max_cantidad` en pseudocódigo 2), cuando se aumenta el parámetro 'cantidad de comparaciones sin mejora' (que detiene el local search) forzando a evaluar más soluciones, sin obtener una mejora significativa. Finalmente se compara el desempeño del algoritmo, nuevamente en búsquedas de duración breve, con una variación del mismo con un mecanismo distinto de actualización utilizado en una búsqueda tabú para resolver problemas de ruteo de vehículos (Gendreau, et al., 1994).

3.2.2 - Parámetros de actualización de pesos.

Dada la arquitectura del algoritmo, se decide realizar un estudio paramétrico que sugiera qué valores deben tomar los parámetros de actualización de los conjuntos de pesos.

Las actualizaciones se pueden producir cuando se comparan dos soluciones, esto puede suceder tanto en el nivel del GRASP, como en la búsqueda local, generando 16 comportamientos distintos que pueden codificarse en 4 variables de decisión binarias: actualizar cuando no hay mejora a nivel GRASP, actualizar cuando hay mejora a nivel GRASP, actualizar cuando no hay mejora a nivel de la búsqueda local, y actualizar cuando hay mejora a nivel búsqueda local.

Reconociendo que el grupo de productores posee una importante diversidad en cuanto a la cantidad de parcelas (atributo que aumenta la complejidad del problema), se agrupan los productores según esta cantidad, separándolos en tres categorías. Se definen las categorías productores pequeños (de 1 a 15 parcelas), productores medianos (16 a 30 parcelas), y productores grandes (más de 30 parcelas), seleccionando para cada una de ellas un representante (productores 5, 26, y 28).

Para cada pareja representante-configuración, se corren 10 ejecuciones independientes del algoritmo con 10.000 iteraciones del GRASP (`max_cantidad` en el pseudocódigo 2), identificando en la tabla 3 cuales producen los mejores (mejor solución, y mejor promedio de soluciones), y peores resultados según la exportación de fósforo (definida como función objetivo de la formulación implícita en la ecuación 4).

	Configuración Mejor Promedio	Configuración Mejor Mínimo	Configuración Peor Solución	Mejor Promedio (kg P/ha)	Mejor Mínimo (kg P/ha)	Peor Solución (kg P/ha)
Instancias Pequeñas	11	5	16	983,58	935,07	1.038,92
Instancias Medias	11	15	12	1.547,05	1.476,22	1.669,58
Instancias Grandes	15	14	10	1.613,88	1.523,23	1.661,62

Tabla 3 - Mejores resultados según fósforo.

Seleccionando las configuraciones con mejor promedio para ser sugeridas a los usuarios según el tamaño de las instancias que quieran resolver.

3.2.3 - Estudio del algoritmo

Buscando comprender el desempeño del algoritmo, se agrega al mismo un sistema de registro que en cada comparación de soluciones reporte los siguientes valores:

- **NumGRASP**: que almacena en qué iteración del GRASP se obtuvo la solución.
- **NumLS**: que almacena en qué iteración de la búsqueda local se obtuvo la solución.
- **strikes**: que almacena la cantidad de soluciones sin mejoras en la búsqueda local actual.
- **Fósforo**: exportación de fósforo en kg por hectárea.
- **Fósforo Escalado**: exportación de fósforo dividido por el máximo fósforo posible (máxima exportación de fósforo por cantidad de estaciones de la planificación).
- **Peso LS Fósforo**: peso dado al fósforo en las comparaciones de la búsqueda local.
- **Peso GRASP Fósforo**: peso dado al fósforo en las comparaciones del GRASP.
- **CantIncUsos**: cantidad de incumplimientos de la cantidad de uso distintos por estación por productor.
- **CantIncUsos_Escalado**: cantidad de incumplimientos de la cantidad de usos dividido por la cantidad máxima de incumplimientos posibles (cantidad de productores por cantidad de estaciones).
- **Peso LS CantIncUsos**: peso dado a los incumplimientos de cantidad de usos en las comparaciones de la búsqueda local.
- **Peso GRASP CantIncUsos**: peso dado a los incumplimientos de cantidad de usos distintos en las comparaciones del GRASP.
- **CantIncProd**: cantidad de incumplimientos de productividad por productor por estación.
- **CantIncProd Escalado**: cantidad de incumplimientos de productividad dividida por la cantidad máxima de incumplimientos posible (cantidad de productores por cantidad de estaciones).
- **Peso LS CantIncProd**: peso dado a los incumplimientos de productividad en las comparaciones de la búsqueda local.
- **Peso GRASP CantIncProd**: peso dado a los incumplimientos de productividad en las comparaciones del GRASP.
- **Función Objetivo Sin Pesos**: la suma fósforo escalado + CantIncUsos escalado + CantUsosProd escalado.
- **Función Objetivo Con Pesos**: la función anterior pero con cada término ponderado por sus respectivos pesos.

Se realiza una ejecución del algoritmo, con 1000 iteraciones del GRASP (`max_cantidad` en el pseudocódigo 2), para 3 productores pequeños (1, 2, y 5), un productor mediano (26), y un productor grande (28).

El sistema de registro se implementa en Java 8, para su análisis se utiliza Python 3.6.9³, con las bibliotecas, Pandas⁴ y Matplotlib⁵.

Experimento 1: Fósforo e incumplimiento de restricciones durante la búsqueda local

Se compara durante cada búsqueda, el desempeño del algoritmo para encontrar soluciones que reduzcan tanto el fósforo, como la cantidad de incumplimientos. Para esto se grafican los registros obtenidos en una serie de tiempo entre las soluciones encontradas (x), contra el fósforo exportado (y), donde el color del punto indica la cantidad de incumplimientos de la solución, y las distintas soluciones de cada ejecución independiente del GRASP están unidas por una línea punteada. A modo de comparación, se marca como línea punteada horizontal dorada los valores de fósforo de la solución óptima obtenida al modelar el problema utilizando programación entera mixta (Matonte, 2019), y para los productores 5, 26, y 28, como una línea punteada horizontal plateada, los valores de fósforo de la mejor solución encontrada por el algoritmo durante el estudio de Parámetros de actualización de pesos.

Restricciones unidas

En las figuras 4 a 8, se realiza una primera revisión sumando la cantidad de incumplimiento de ambas restricciones, para visualizar si existen mejoras en la calidad de las soluciones encontradas a medida que avanzan las búsquedas locales.

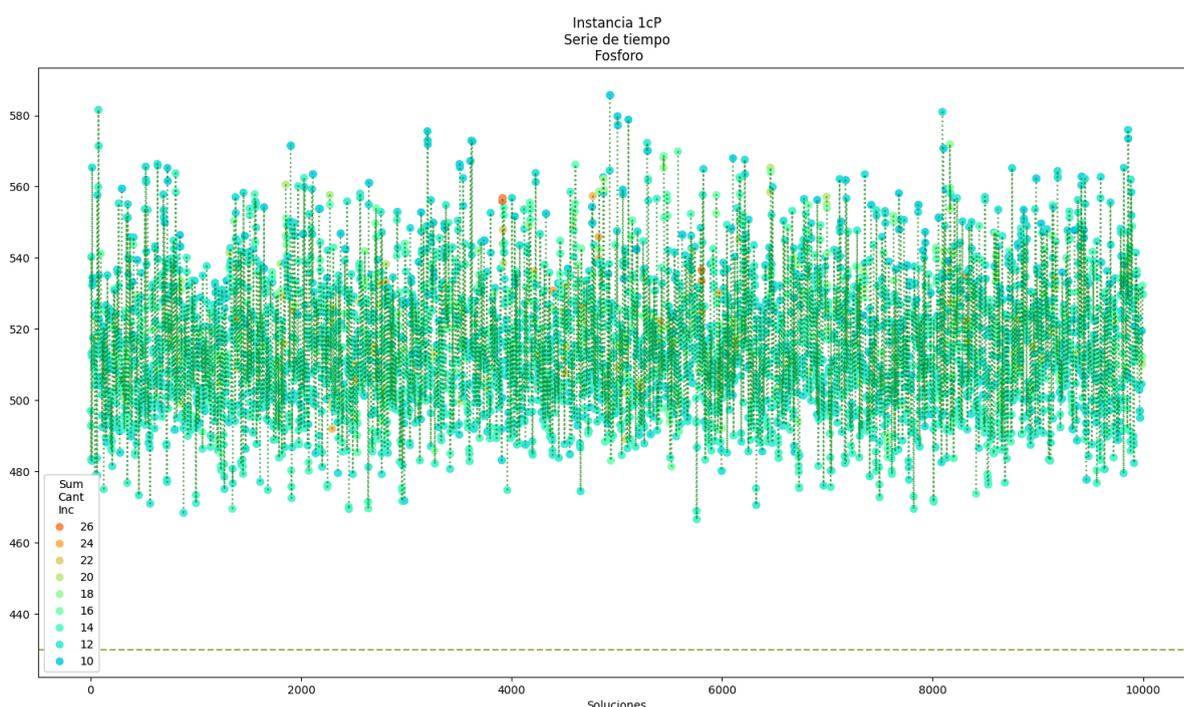


Figura 4 - Series de tiempo, fósforo contra Incumplimientos, productor 1

³ <https://www.python.org/downloads/release/python-369/>

⁴ <https://pandas.pydata.org/>

⁵ <https://matplotlib.org/>

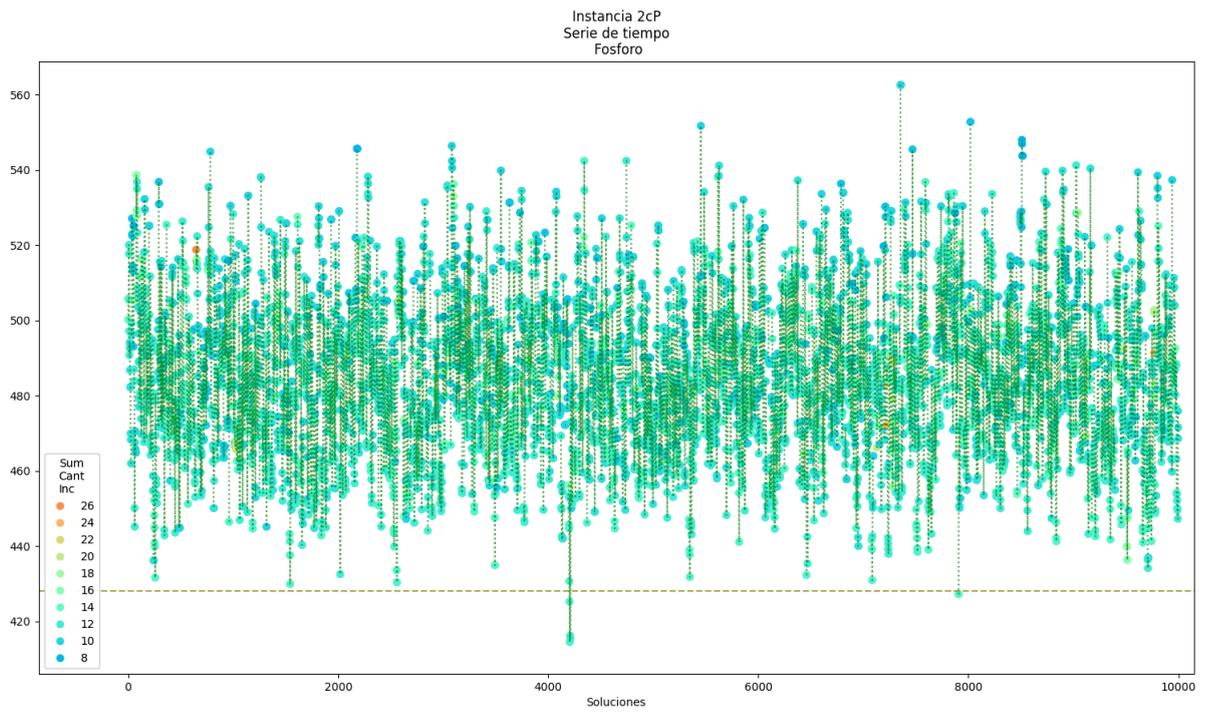


Figura 5 - Series de tiempo, fósforo contra Incumplimientos, productor 2

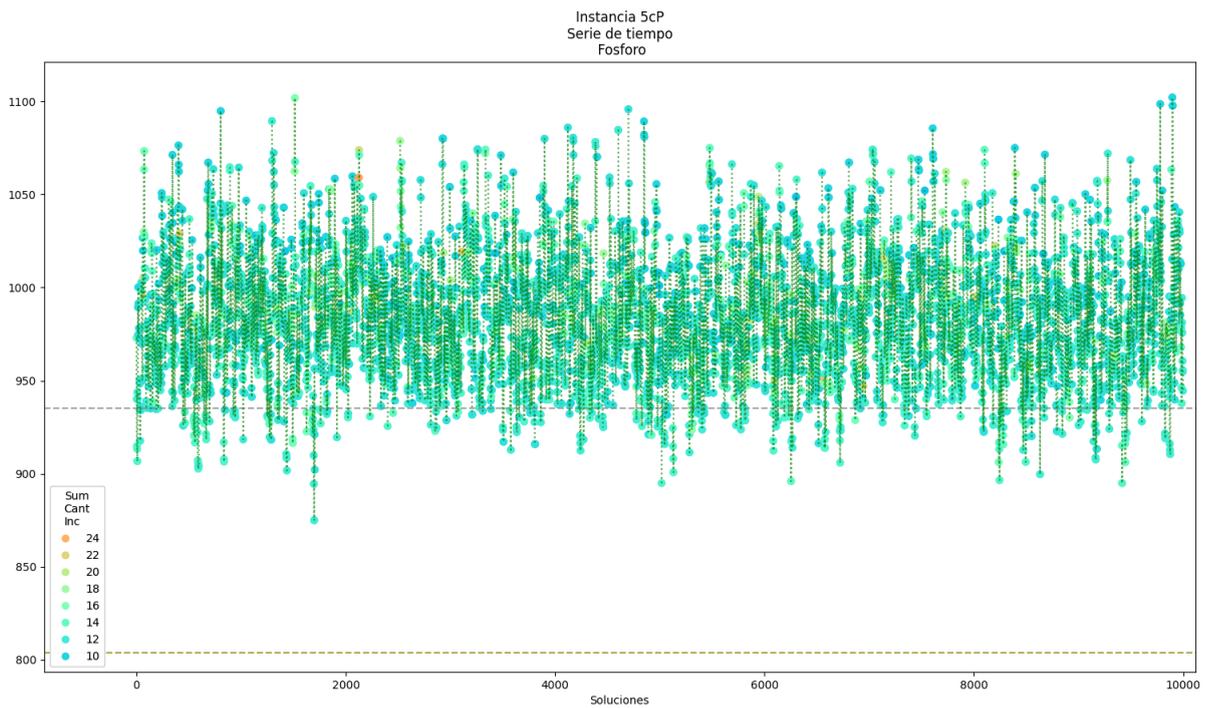


Figura 6 - Series de tiempo, fósforo contra Incumplimientos, productor 5

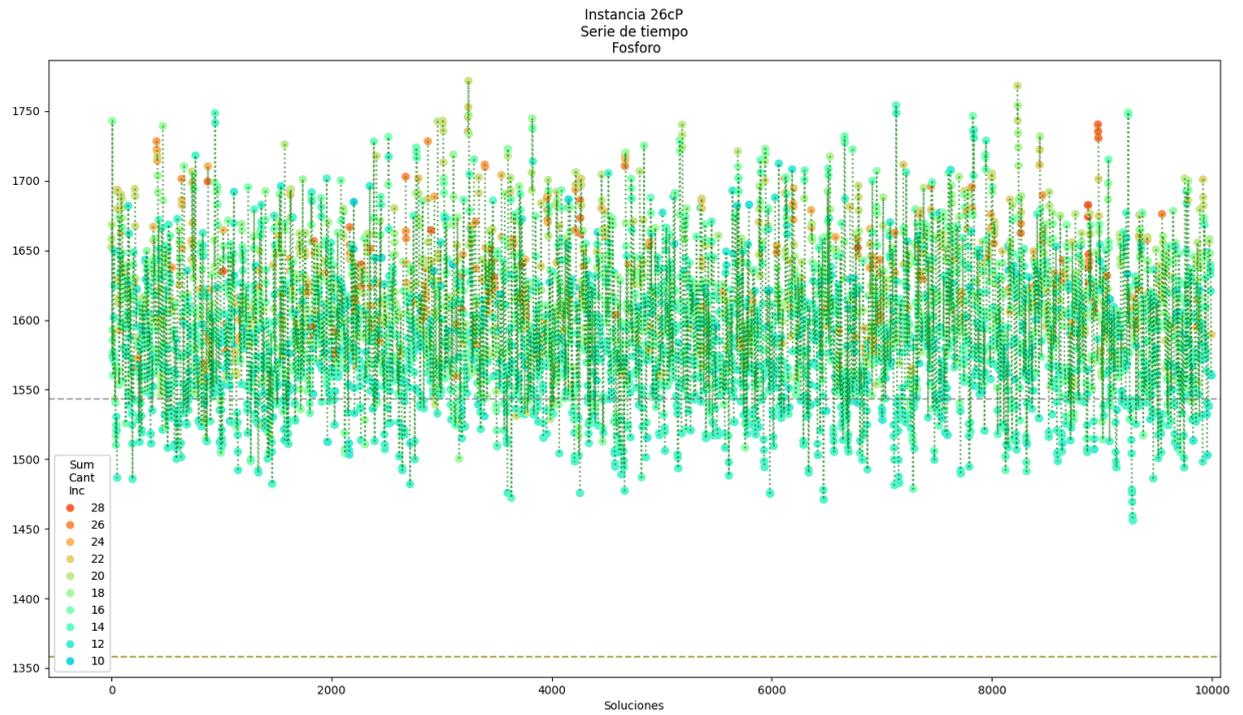


Figura 7 - Series de tiempo, fósforo contra Incumplimientos, productor 26

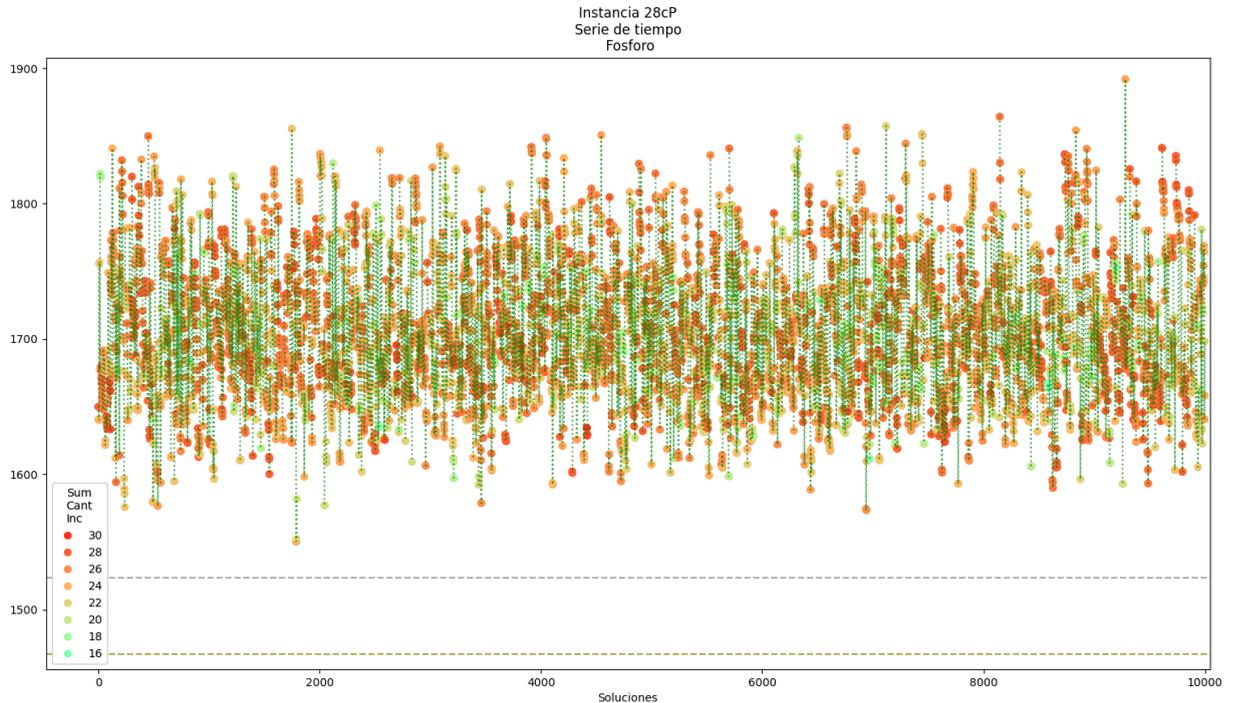


Figura 8 - Series de tiempo, fósforo contra Incumplimientos, productor 28

Pudiéndose notar en las figuras 4 a 8, un comportamiento cíclico, o un patrón que coincide con cada iteración del GRASP y permanece constante a lo largo de las 10.000 soluciones evaluadas. Esto permite enfocar las gráficas de las figuras 9 a 13 solo en las primeras 1000 soluciones evaluadas, al considerarlas un conjunto representativo.

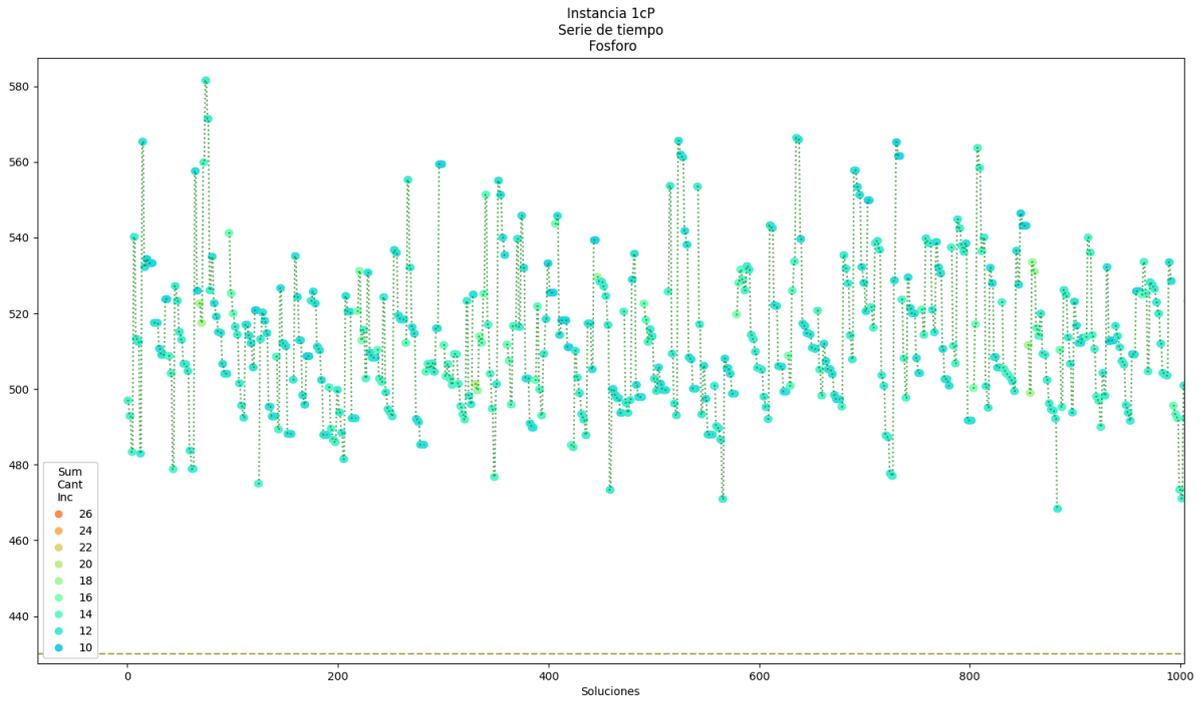


Figura 9 - Series de tiempo, fósforo contra Incumplimiento, primeras 1000 soluciones, productor 1

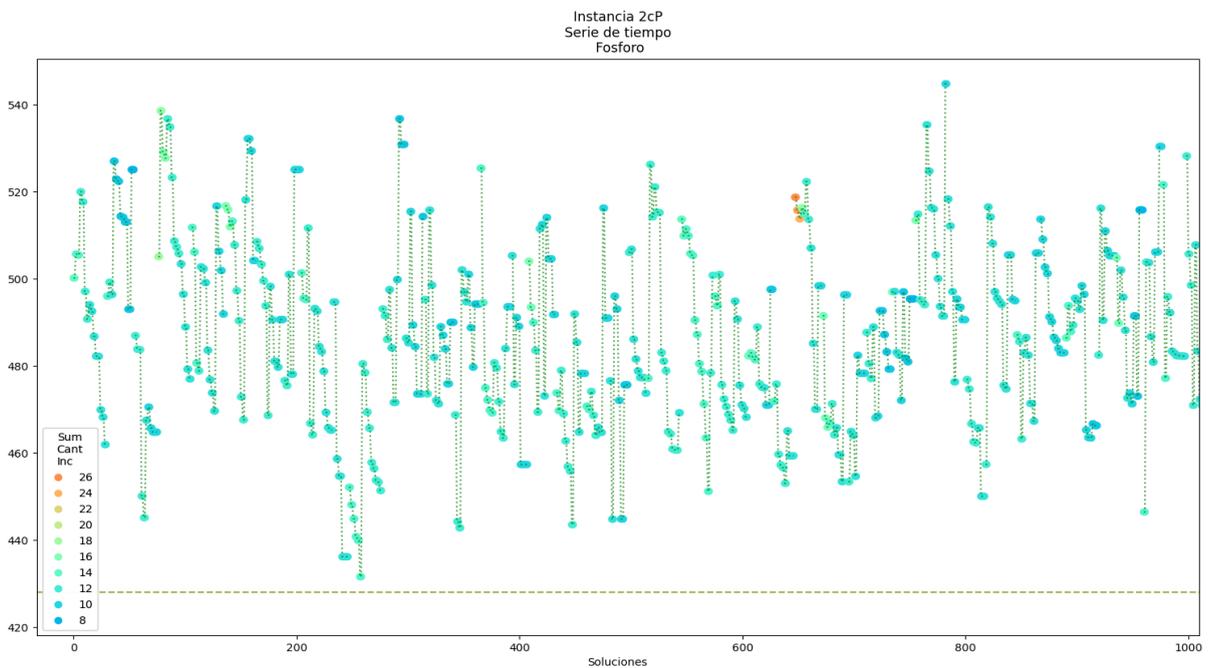


Figura 10 - Series de tiempo, fósforo contra Incumplimiento, primeras 1000 soluciones, productor 2

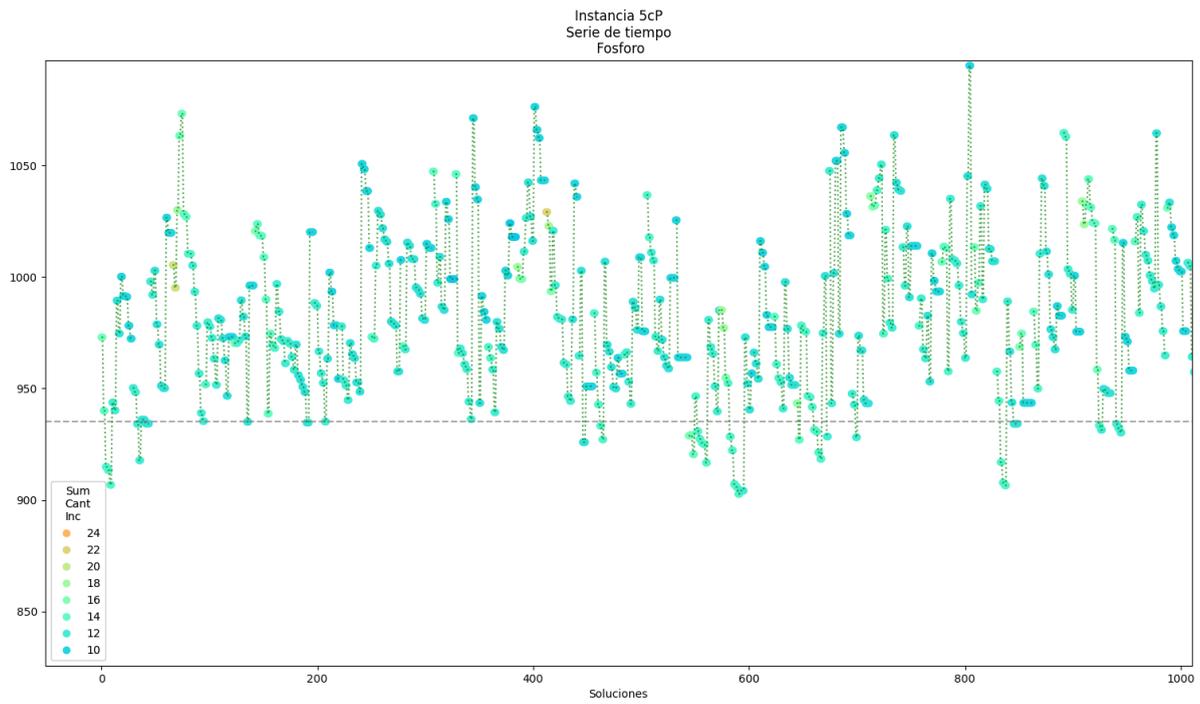


Figura 11 - Series de tiempo, fósforo contra Incumplimiento, primeras 1000 soluciones, productor 5

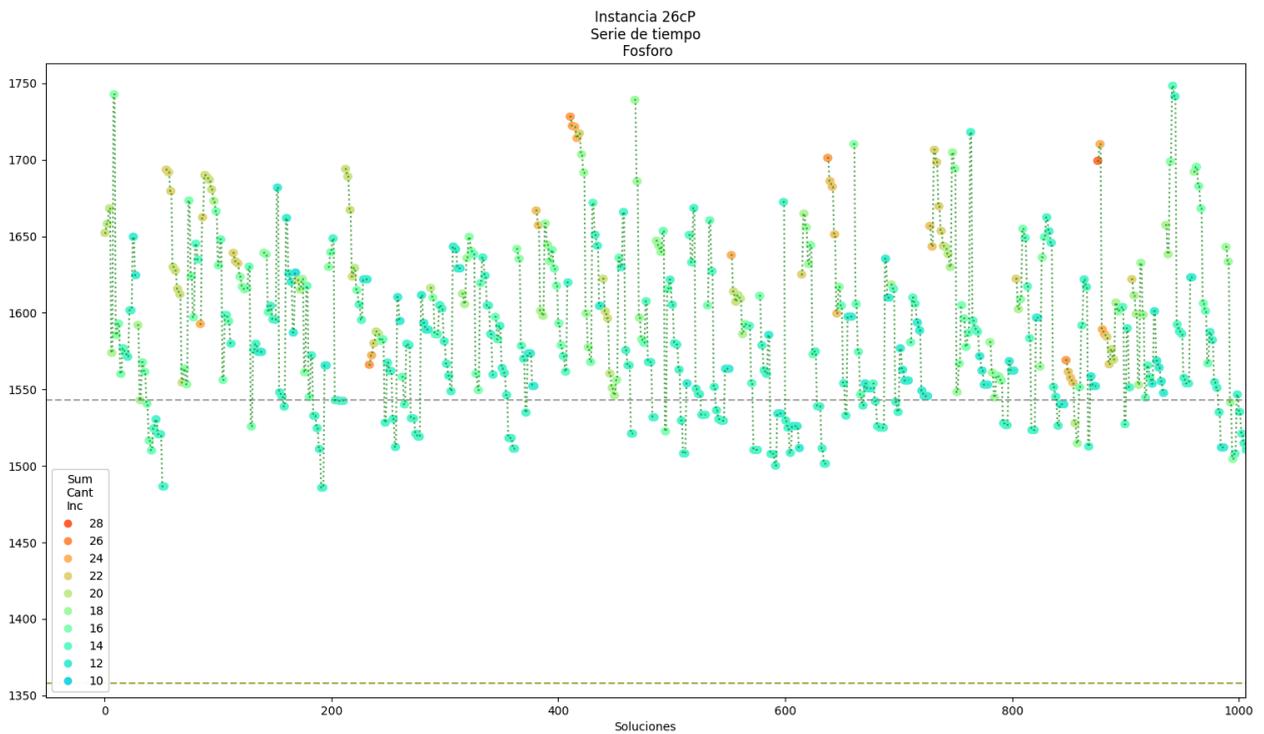


Figura 12 - Series de tiempo, fósforo contra Incumplimiento, primeras 1000 soluciones, productor 26

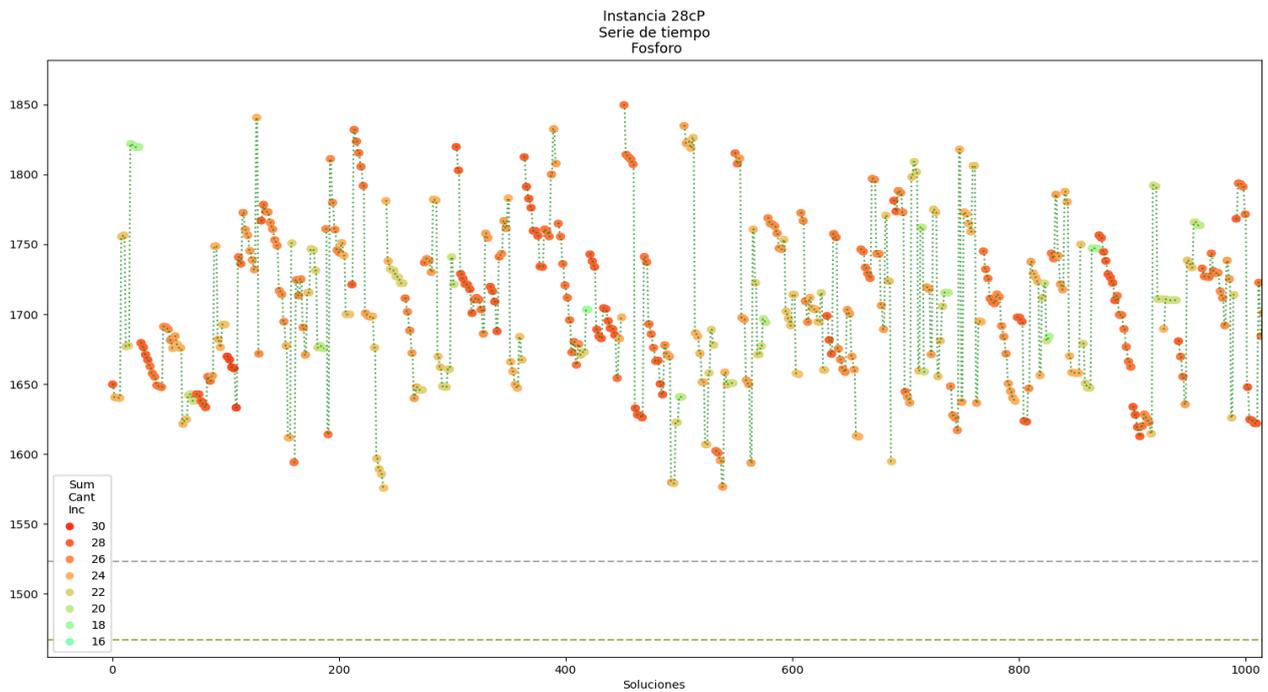


Figura 13 - Series de tiempo, fósforo contra Incumplimiento, primeras 1000 soluciones, productor 28

Puede observarse durante las búsquedas locales en cada iteración dentro del GRASP, una mejora, generalmente paulatina, en la calidad de las soluciones, dada tanto en la disminución del fósforo, y/o en una disminución de la cantidad de incumplimientos, logrando llevar este número a 10 en las instancias pequeñas y en la instancia mediana, y hasta un 16 para la instancia grande.

También es importante mencionar la notable disminución en la calidad de las soluciones a mayor tamaño de los productores, esta disminución se observa tanto en la solución inicial como en las encontradas durante la búsqueda.

Cantidad de usos distintos

Las figuras 14 a 18 contrastan el fósforo solo contra los incumplimientos de cantidades de usos distintos.

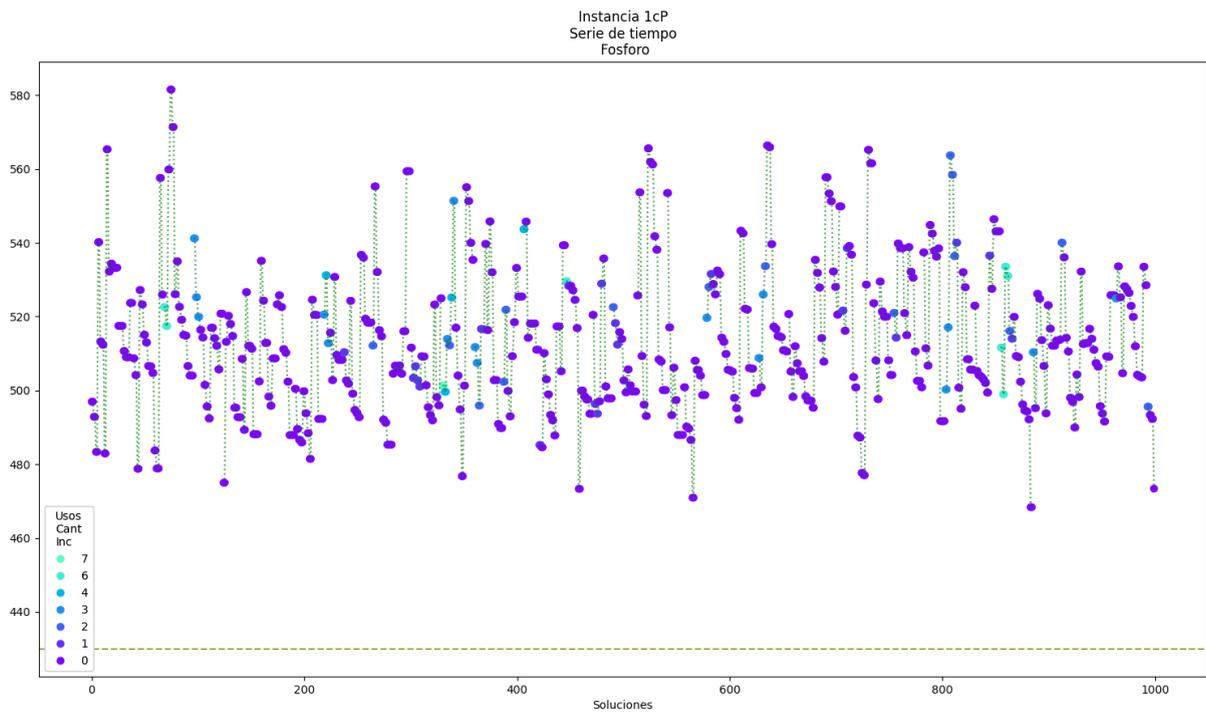


Figura 14 - Series de tiempo, fósforo contra Incumplimientos de Cantidad de usos distintos, primeras 1000 soluciones, productor 1

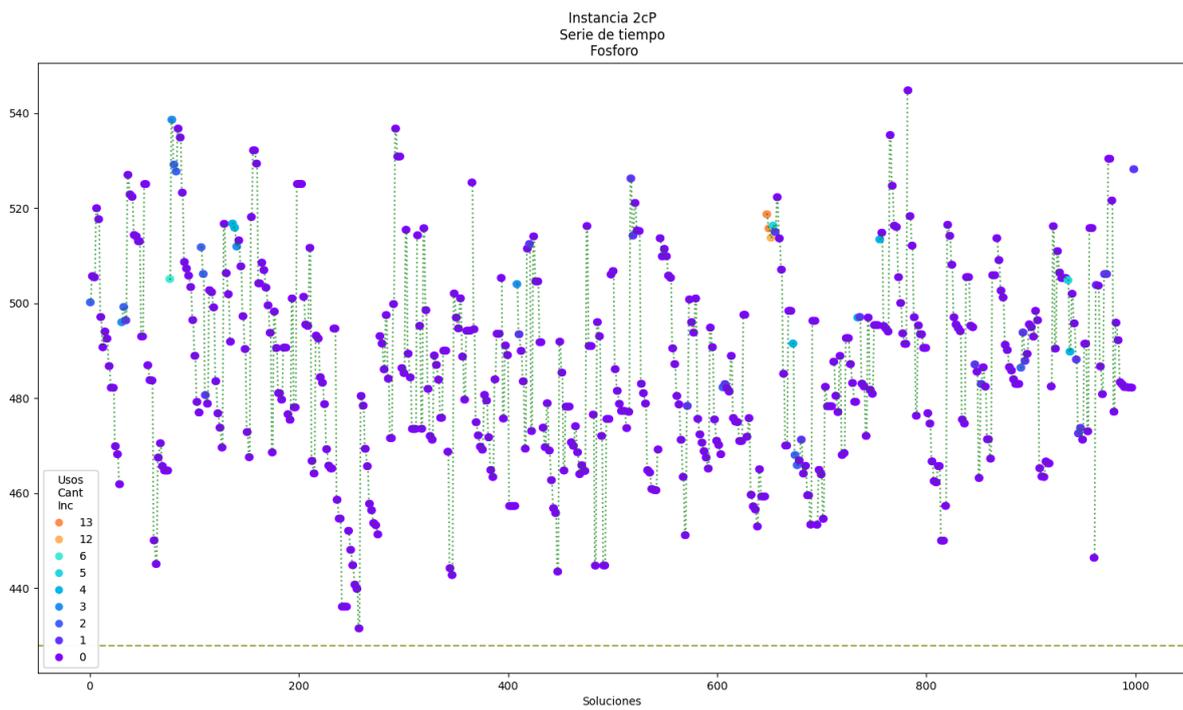


Figura 15 - Series de tiempo, fósforo contra Incumplimientos de Cantidad de usos distintos, primeras 1000 soluciones, productor 2

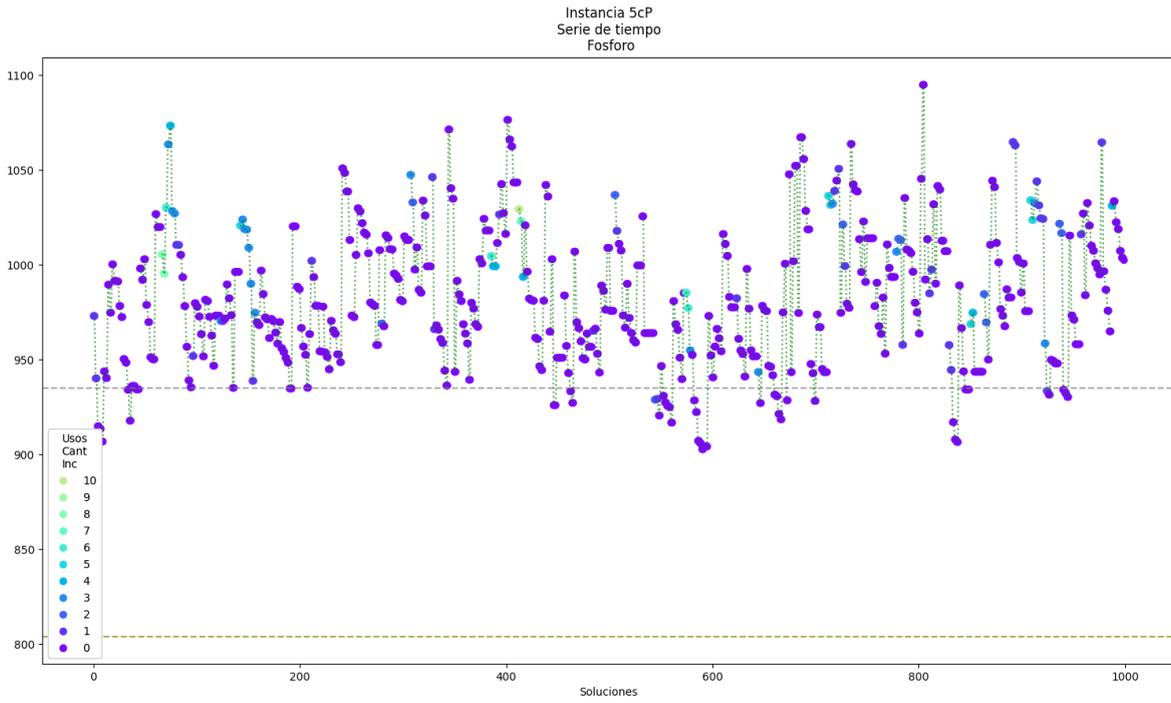


Figura 16 - Series de tiempo, fósforo contra Incumplimientos de Cantidad de usos distintos, primeras 1000 soluciones, productor 5

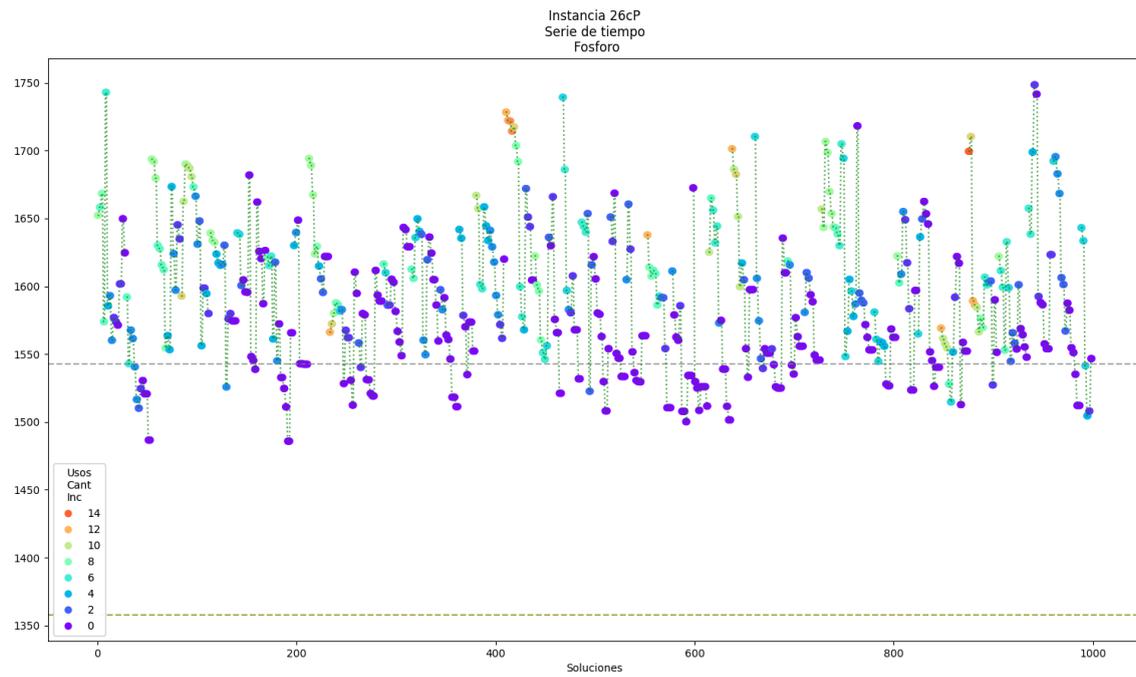


Figura 17 - Series de tiempo, fósforo contra Incumplimientos de Cantidad de usos distintos, primeras 1000 soluciones, productor 26

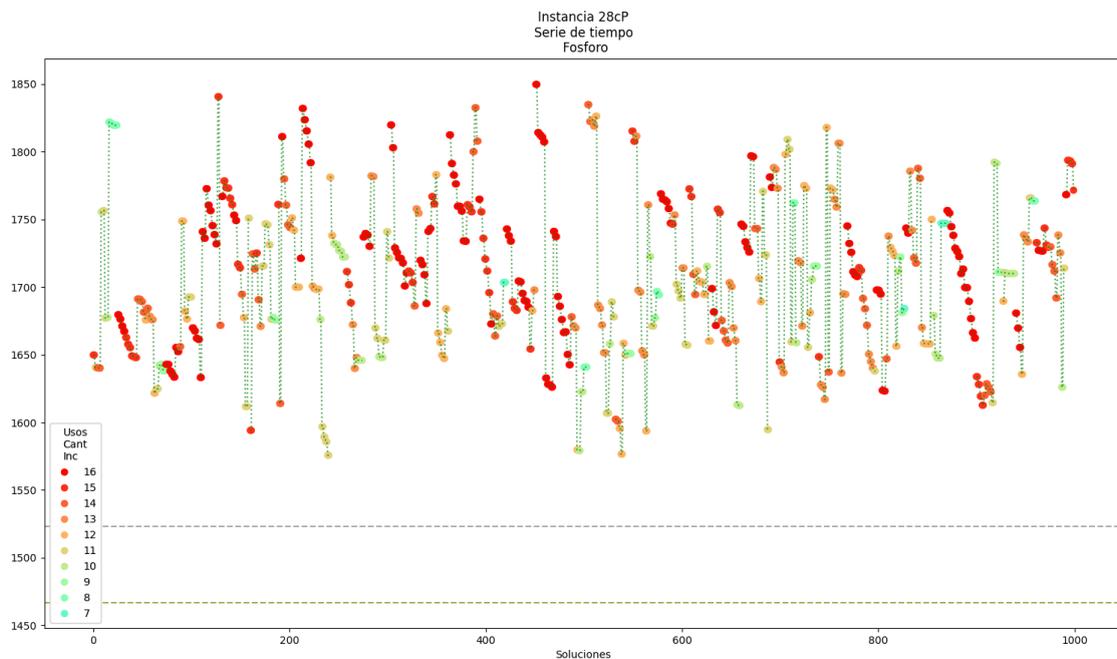


Figura 18 - Series de tiempo, fósforo contra Incumplimientos de Cantidad de usos distintos, primeras 1000 soluciones, productor 28

En las figuras 14 a 18 puede apreciarse la capacidad del algoritmo para mejorar soluciones con incumplimiento, encontrando vecinos con incumplimientos reducidos durante cada búsqueda local. Logrando encontrar para las instancias pequeñas y medianas, soluciones sin incumplimientos. Pero siendo incapaz de mantener el desempeño para la instancia grande representada por el productor 28. Esto puede deberse a una cantidad insuficiente de búsquedas locales dentro de cada GRASP para explotar sus espacios de soluciones (de naturaleza más compleja que las de los productores más pequeños).

Productividad

Las figuras 19 a 23 contrastan el fósforo solo contra los incumplimientos de Productividad.

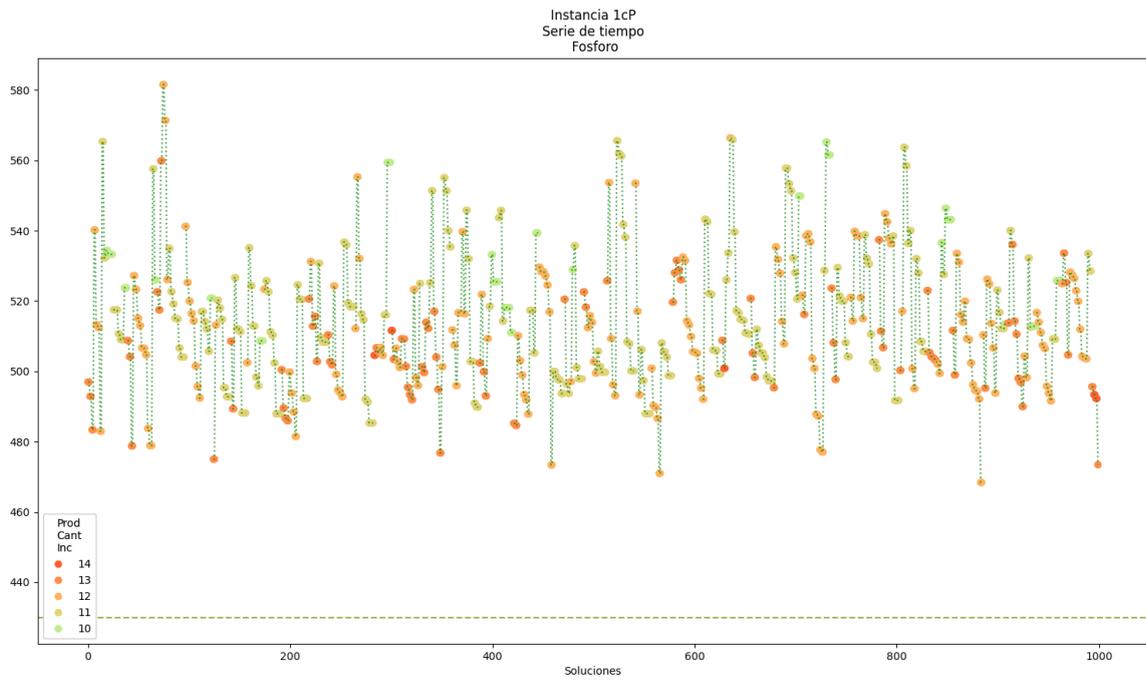


Figura 19 - Series de tiempo, fósforo contra Incumplimientos de Productividad, primeras 1000 soluciones, productor 1

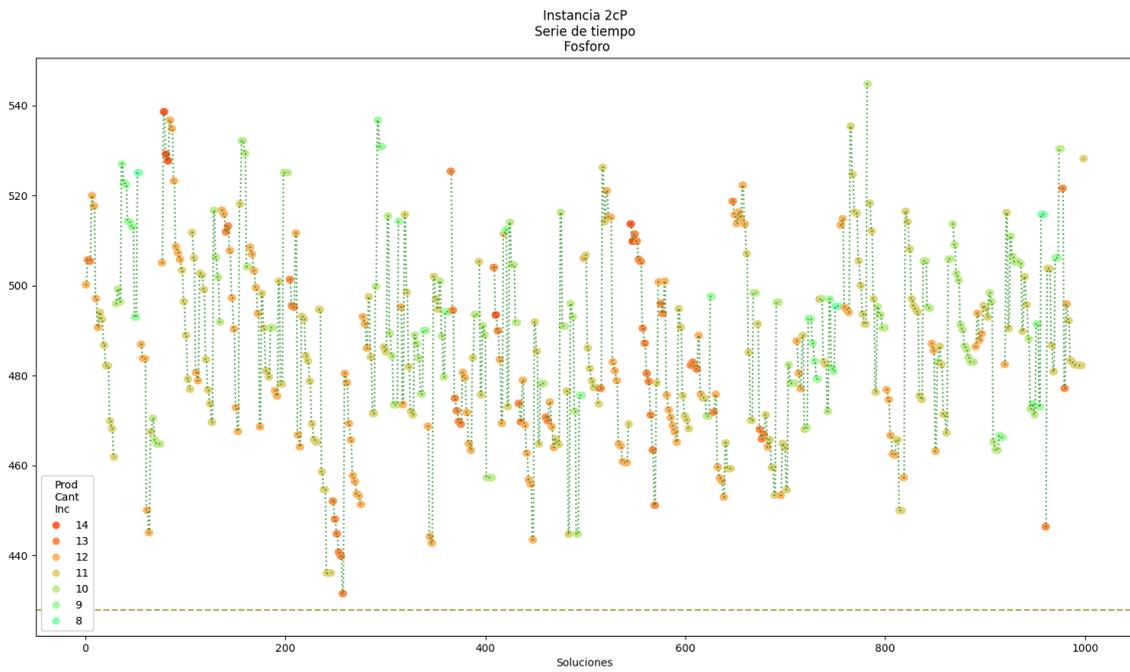


Figura 20 - Series de tiempo, fósforo contra Incumplimientos de Productividad, primeras 1000 soluciones, productor 2

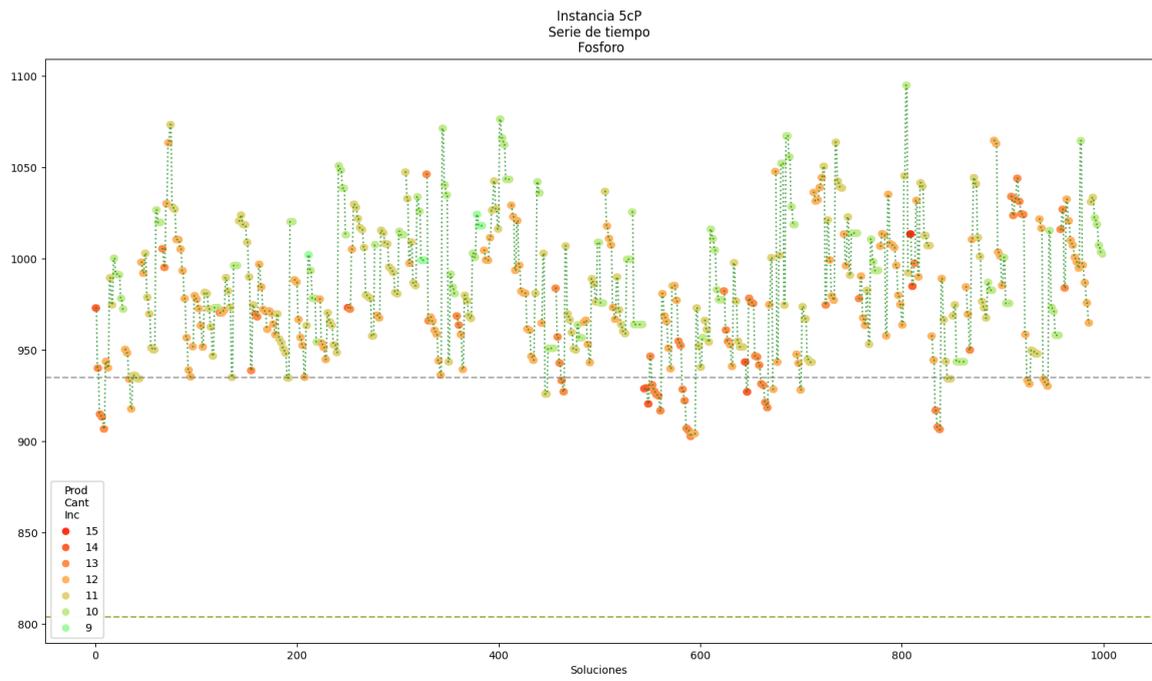


Figura 21 - Series de tiempo, fósforo contra Incumplimientos de Productividad, primeras 1000 soluciones, productor 5

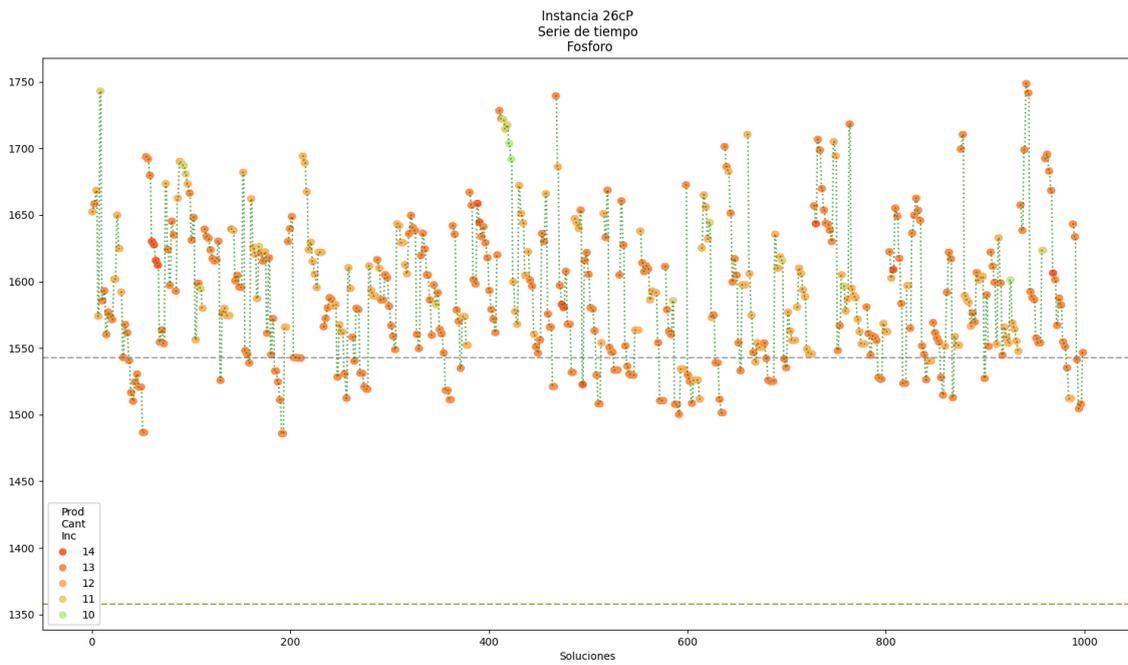


Figura 22 - Series de tiempo, fósforo contra Incumplimientos de Productividad, primeras 1000 soluciones, productor 26

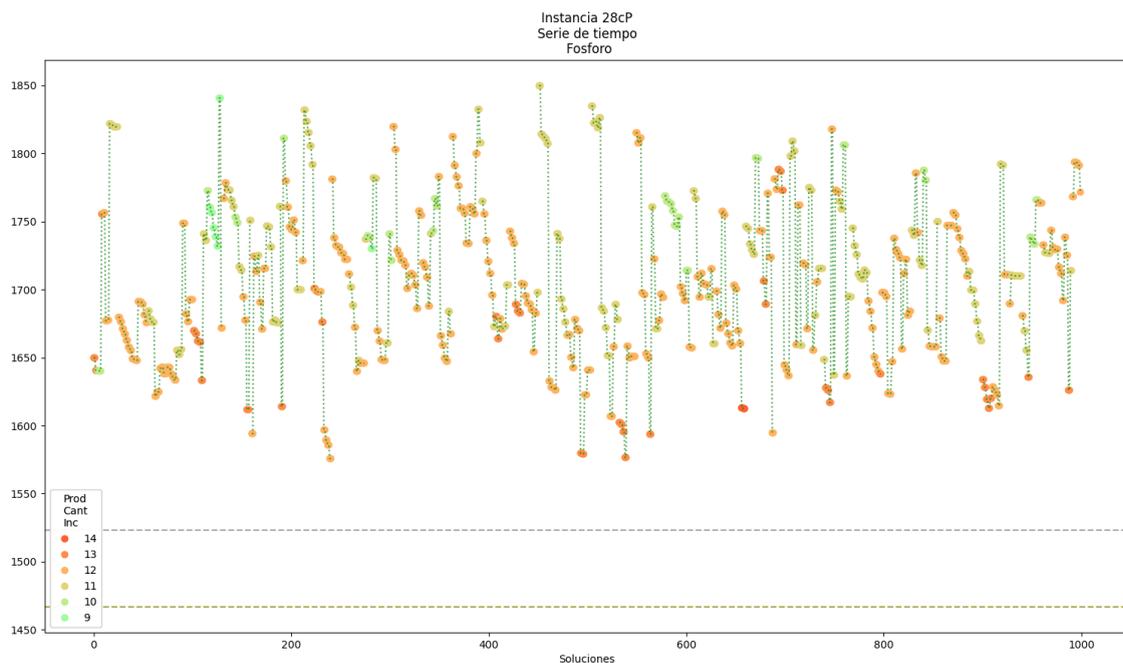


Figura 23 - Series de tiempo, fósforo contra Incumplimientos de Productividad, primeras 1000 soluciones, productor 28

En las Figuras 19 a 23 es notable una mayor dificultad para reducir los incumplimientos de productividad. Logrando reducirlas, en el mejor de los casos, a un 50% del máximo. Es posible volver a intuir una relación directa entre la reducción del fósforo y el aumento de incumplimientos de productividad, o que de forma similar al productor grande en los incumplimientos del uso, las iteraciones del GRASP no realizan suficientes búsquedas locales para explotar las regiones exploradas. También puede notarse en las figuras 21 y 22, que para los productores 5 y 26, el algoritmo encontró soluciones con mejor exportación de fósforo que el óptimo, pero que dicha mejora solo puede obtenerse a costa de los incumplimientos.

Experimento 2: Desempeño de la búsqueda local.

Para las ejecuciones del algoritmo de los tres productores pequeños, con las configuraciones que reportaron un mejor desempeño para el productor 5 (mejor promedio y mejor solución), se presenta en la tabla 4, qué porcentaje de las iteraciones del GRASP encontraron (luego de su búsqueda local) una solución que mejore alguno de sus criterios (menor exportación de fósforo, menor incumplimiento de cantidad de usos y menor incumplimiento de productividad).

Productor	Configuración	Mejoras Fósforo	Mejoras Usos	Mejoras Productividad
1	Promedio	40,26%	0,10%	1,50%
1	Mejor	37,16%	0,40%	1,10%
2	Promedio	48,65%	0,10%	1,00%
2	Mejor	51,95%	0,50%	1,00%
5	Promedio	45,55%	0,20%	1,20%
5	Mejor	48,75%	0,20%	1,10%

Tabla 4 - Desempeño del Local Search.

Puede observarse en la tabla 4 un buen porcentaje de mejoras al comparar el fósforo, pero un porcentaje casi nulo de mejoras en las cantidades de incumplimientos de los usos y la productividad (ambos atributos discretos). Lo que puede indicar que el procedimiento de construcción genera soluciones con una cantidad de incumplimientos baja respecto a sus vecinos, por lo que la búsqueda local no logra encontrar vecinos con menor cantidad de incumplimientos.

Experimento 3: Aumento de la exploración en la búsqueda local.

Buscando comprender el comportamiento, se revisa en la tabla 5 cuál es la causa por la que se detiene la búsqueda local. Siendo dos los posibles motivos de la finalización de la búsqueda: alcanzar la cantidad máxima de soluciones a comparar (`max_cantidad_FI` en el pseudocódigo 4), y alcanzar la cantidad máxima de soluciones evaluadas sin obtener mejora (`strikes_FI` en el pseudocódigo 4). Se compara en la tabla 5 qué porcentaje de las búsquedas fue detenido por cada motivo.

Productor	Configuración	Cantidad Máxima de Búsquedas	Cantidad de comparación sin mejoras
1	Promedio	35,86%	64,14%
1	Mejor	36,26%	63,74%
2	Promedio	35,96%	64,04%
2	Mejor	33,97%	66,03%
5	Promedio	36,86%	63,04%
5	Mejor	32,17%	67,73%

Tabla 5 - Criterio de finalización del local search.

En la tabla 5 se puede apreciar una disparidad entre ambas condiciones de paradas, pero se mantienen las proporciones entre los diferentes productores.

En la tabla 6 se muestra cómo aumentar a 5 la Cantidad de comparaciones, se logra alterar dichas proporciones pero sin conseguir que esto reporte mejoras significativas en los resultados en el desempeño de las búsquedas locales para ninguno de los 3 criterios.

Productor	Configuración	Cantidad Máxima de Búsquedas	Cantidad de comparación sin mejoras	Mejoras Fósforo	Mejoras Usos	Mejoras Productividad
1	Promedio	59,24%	40,66%	39,66%	0,00%	0,80%
1	Mejor	58,84%	41,06%	40,16%	0,40%	0,40%
2	Promedio	60,14%	39,86%	47,95%	0,00%	1,40%
2	Mejor	58,54%	41,46%	50,55%	0,40%	1,10%
5	Promedio	56,54%	43,46%	45,25%	0,20%	1,30%
5	Mejor	56,24%	43,76%	47,75%	0,30%	1,30%

Tabla 6 - Desempeño del local search con parada luego de 5 comparaciones sin mejora.

En la tabla 7 se aumenta la cantidad de comparaciones a 10, se puede ver que aproximadamente el 95% de las veces, la condición de parada es la cantidad máxima de búsquedas, las cuales siguen sin reportar una mejora en el desempeño del algoritmo.

Productor	Configuración	Cantidad Máxima de Búsquedas	Cantidad de comparación sin mejoras	Mejoras Fósforo	Mejoras Usos	Mejoras Productividad
1	Promedio	94,71%	5,19%	39,76%	0,00%	0,60%
1	Mejor	95,70%	4,20%	41,16%	0,70%	1,40%
2	Promedio	96,00%	4,00%	48,25%	0,00%	1,10%
2	Mejor	95,40%	4,60%	48,45%	0,20%	1,60%
5	Promedio	96,10%	3,90%	45,25%	0,00%	0,50%
5	Mejor	95,20%	4,80%	47,65%	0,40%	0,60%

Tabla 7 - Desempeño del local search con parada luego de 10 comparaciones sin mejora.

Experimento 4: Actualización alternativa de pesos.

Buscando comparar contra un método de referencia, se implementa una alternativa para la actualización de los pesos, fuertemente basada en el artículo "A Tabu Search for the Vehicle Routing Problem" (Gendreau et al., 1994), un método con eficacia probada en otros problemas con características similares (en particular restricciones difíciles de cumplir mediante heurísticas, y factibilidad que compite contra la optimalidad), en el cual los pesos asociados a un criterio se actualizan (dividiendo o multiplicando por 1,1) cuando llegan a t comparaciones consecutivas con o sin mejora. Luego de t soluciones sin mejora se multiplica el peso de un criterio para que sea priorizado, mientras que luego de t comparaciones con mejora se divide para que deje de ser priorizado.

Con esta configuración alternativa, se ejecuta el algoritmo para cada productor pequeño con 1000 iteraciones del GRASP (`max_cantidad` en el pseudocódigo 2), (evaluando en cada una de ellas más de 23.500 soluciones), reportando los resultados

en la tabla 8, donde puede observarse que el nuevo método no obtiene desempeño superior destacable a la configuración propuesta originalmente.

Productor	Mejoras Fósforo	Mejores Usos	Mejoras Productividad
1	37,16%	0,40%	1,10%
2	51,95%	0,50%	1,00%
5	46,95%	0,20%	1,10%

Tabla 8 - Desempeño Actualización alternativa de pesos para $t = 3$.

Definir t como una proporción de la 'cantidad máxima de búsquedas' sería una buena forma de compatibilizar ambos métodos de actualización, pero a costo de volverlos más complejos.

3.2.4 - Conclusiones de los experimentos.

Tomando como ejemplo los productores de la cuenca del río Santa Lucía, utilizando para ellos el algoritmo con cada configuración paramétrica, y generalizando sobre los resultados de las búsquedas, es posible sugerir configuraciones paramétricas que beneficien el desempeño del algoritmo según el tamaño de la entrada.

El proceso de exploración del algoritmo parece ser constante para diferentes ejecuciones independientes, permitiendo tomar cantidades reducidas de iteraciones dentro del GRASP como muestras representativas. La calidad de las soluciones iniciales (base en el pseudocódigo 2) varía, pero está fuertemente ligada al tamaño de la instancia a resolver.

En muestras suficientemente grandes puede observarse que el mecanismo de actualización de pesos permite escapar a óptimos locales, no teniendo problemas en abandonar una solución con buenos valores por otra que posee peores valores en alguno de los criterios, encontrando luego una tercera solución con mejores valores para todos los criterios.

El tamaño de la entrada complejiza el problema, y reduce el desempeño del algoritmo. El algoritmo tiene mayor éxito en reducir el fósforo y los incumplimientos de cantidad de usos distintos, que en reducir los incumplimientos de productividad (se conjetura por la relación directa entre exportación de fósforo y producción de materia seca, como se puede observar a partir de los parámetros del problema).

La búsqueda local tiene una baja tasa de éxito para la reducción (estricta) de incumplimientos (la naturaleza discreta de esta medida puede influir en los bajos valores). Ni aumentar la cantidad de comparaciones sin éxito (strikes) antes de finalizar una búsqueda local, ni utilizar una actualización de pesos alternativa logró una mejora notable en el desempeño del algoritmo para instancias pequeñas.

Finalmente se compara en la tabla 9 el desempeño del algoritmo en las búsquedas de duración moderada, realizadas para encontrar las configuraciones a sugerir, comparado con el resultado óptimo hallado por la formulación exacta.

	Productor de ejemplo	Cantidad Pixeles	Mejor Promedio GRASP (kg P/ha)	Incumplimientos Promedio	Solución exacta (kg P/ha)	Diferencia Porcentual Entre Heurística y Exacta
Instancias Pequeñas	5	10	983,58	25,94%	804,00	22,34%
Instancias Medias	26	15	1.547,05	32,19%	1.354,00	14,26%
Instancias Grandes	28	29	1.613,88	49,06%	1.467,00	10,01%

Tabla 9 - Desempeño del algoritmo comparado con la formulación exacta.

En la tabla 9 puede apreciarse que, ya en búsquedas de duración moderada, el algoritmo es capaz de encontrar soluciones con una buena exportación de fósforo para los tres tamaños de instancias, a costo de un promedio de incumplimientos alto. También es notable que según crece el tamaño del productor, la diferencia en fósforo mejora, pero los incumplimientos empeoran.

4 - Desarrollo del Sistema de Soporte a las Decisiones

Un Sistema de Soporte a las Decisiones, o DSS por sus siglas en inglés (Decision Support System), es un conjunto de procedimientos basados en modelos para procesar datos y juicios para ayudar a un administrador en su toma de decisiones (Little, 1970).

Enfocado más específicamente a la agricultura, un DSS para la agricultura, o ADSS por sus siglas en inglés, se puede definir como un sistema humano-informático que utiliza datos de diversas fuentes, con el objetivo de proporcionar a los agricultores una lista de consejos para respaldar su toma de decisiones en diferentes circunstancias (Z. Zhai et al., 2020). Un ADSS también puede ayudar a tomar decisiones para lograr un mejor desempeño en las tareas futuras (Alenljung, 2008).

En este proyecto se desarrolla un ADSS que manipula una gran cantidad de datos (parcelas, productores, usos, restricciones), y retorna un conjunto de posibles planificaciones de cultivos. Este proyecto sigue un modelo de desarrollo iterativo incremental. En este modelo primero se realiza un relevamiento de requerimientos en términos de casos de uso, luego para cada caso de uso se realizan cuatro etapas:

- etapa de análisis: en la cual se modela la funcionalidad a realizar, y se especifica el comportamiento del sistema.
- etapa de diseño: en la que se especifica la estructura interna del sistema.
- etapa de implementación: en la que se codifica en lenguaje de programación lo especificado en la etapa de diseño.
- etapa de prueba (testing): en la cual se prueba la funcionalidad implementada.

Luego de realizar estas cuatro etapas para todos los casos de uso, se realizaron pruebas generales de todas las funcionalidades de la aplicación.

A continuación se muestran las iteraciones realizadas en orden cronológico, en este modelo iterativo incremental, en un alto nivel de agregación:

- Esqueleto de la interfaz (ventana principal de la aplicación con pestañas con información de prueba).
- Pestaña de usos, y sus ventanas correspondientes.
- Pestaña de regiones, y sus ventanas correspondientes.
- Pestaña de configuraciones paramétricas, y sus ventanas correspondientes.
- Pestaña de problemas, y sus ventanas correspondientes.
- Capacidad de editar los componentes existentes (usos, regiones, configuraciones paramétricas, problemas).
- Capacidad de duplicar y borrar los componentes existentes.
- Agregar un gestor de dependencias al proyecto.
- Cargar regiones a partir de un archivo shape.
- Mostrar en un mapa una región cargada en la herramienta.
- Ejecución de un problema (ventana con barra de porcentaje de ejecución)
- Integración de la interfaz con el algoritmo legado.
- Creación un archivo jar ejecutable
- Guardar soluciones no dominadas al ejecutar un problema.
- Pestaña de resultados, y sus ventanas correspondientes.
- Mostrar en un mapa una solución (una planificación de rotación de cultivos).
- Mostrar gráficas con información relevante de cada solución.
- Guardar archivos de texto con información relevante de cada solución.

4.1 - Requerimientos y Análisis

Los requerimientos se obtienen mediante entrevistas realizadas a Lorena Rodríguez-Gallego y Oscar Blumetto, contrapartes de CURE e INIA respectivamente, luego priorizados y acotados por Antonio Mauttone, en su rol de supervisor en este proyecto de grado, según el alcance del proyecto, sumados a los requerimientos relevados durante las etapas iniciales del proyecto por quienes escriben. Estos se encuentran detallados a continuación en un alto nivel de agregación.

4.1.1 - Requerimientos no funcionales

- Implementar una aplicación de escritorio monousuario, utilizando Java (ya que se contaba con un algoritmo legado implementado en este lenguaje), preferentemente en Windows. Es importante destacar que la aplicación no tiene por qué ser web, ni móvil.
- Esta aplicación debe contar con una interfaz gráfica con la que el usuario pueda utilizar todas sus funcionalidades. Esta interfaz debe ser sencilla y fácil de utilizar.

4.1.2 - Requerimientos funcionales

- La herramienta debe permitir cargar y editar nuevos usos, además de permitir editar los usos precargados.
- Además, se debe poder importar las regiones (incluyendo productores y píxeles) desde archivos .shape generados en programas SIG.
- El algoritmo de búsqueda debe ser parametrizable, permitiendo cambiar valores referentes a la heurística GRASP.
- El problema a resolver debe ser parametrizable, permitiendo configurarlo de manera de poder generar diferentes instancias de este.
- Al ejecutar un problema para obtener un resultado, se debe ver el avance de la ejecución del algoritmo, como también tener la posibilidad de cancelarlo.
- A partir de una solución, la herramienta debe poder generar: gráficas con información útil para el usuario, y un mapa interactivo que facilite la visualización de la solución.
- La herramienta debe poder generar para una solución, archivos de reporte que faciliten el análisis de la planificación presentada. A la exportación de las gráficas (en formatos de imágenes) y los mapas (en formato shape), se le deben sumar archivos de texto que releven información útil para el usuario.
- Se debe agregar al algoritmo la posibilidad de almacenar las mejores soluciones encontradas, recolectando la mejor solución sin incumplimientos y tres listas de mejores soluciones según los criterios que dan calidad a la solución. Estas listas almacenen las 10 mejores soluciones no dominadas (para cada criterio) encontradas durante la búsqueda.

La especificación detallada de los casos de uso fue determinada por los autores de este proyecto, en base a los requerimientos funcionales (especificados con bajo nivel de detalle) y al conocimiento de la forma de interacción con el algoritmo de optimización en su versión anterior (operación por línea de comandos, en base a archivos de entrada), y fue validada con el supervisor.

4.1.3 - Modelo Conceptual

Se realiza un modelado conceptual partiendo del conocimiento de la instancia del problema trabajado y la formulación matemática disponible, identificando los siguientes conceptos principales y cómo se relacionan. En la figura 24 se presenta el diagrama del modelo en Lenguaje Unificado de Modelado (UML por sus siglas en inglés), en el que se muestran las relaciones entre los conceptos. En un alto nivel se puede ver que cada productor es dueño de un conjunto de píxeles, y estos píxeles pertenecen a una única región. Además, una región (con todos sus píxeles), una configuración paramétrica, y un conjunto de usos conforman un problema, el cual, luego de ser ejecutado, asigna usos a plantar en cada píxel de la región en cada estación, generando un resultado, que a su vez es un conjunto de soluciones.

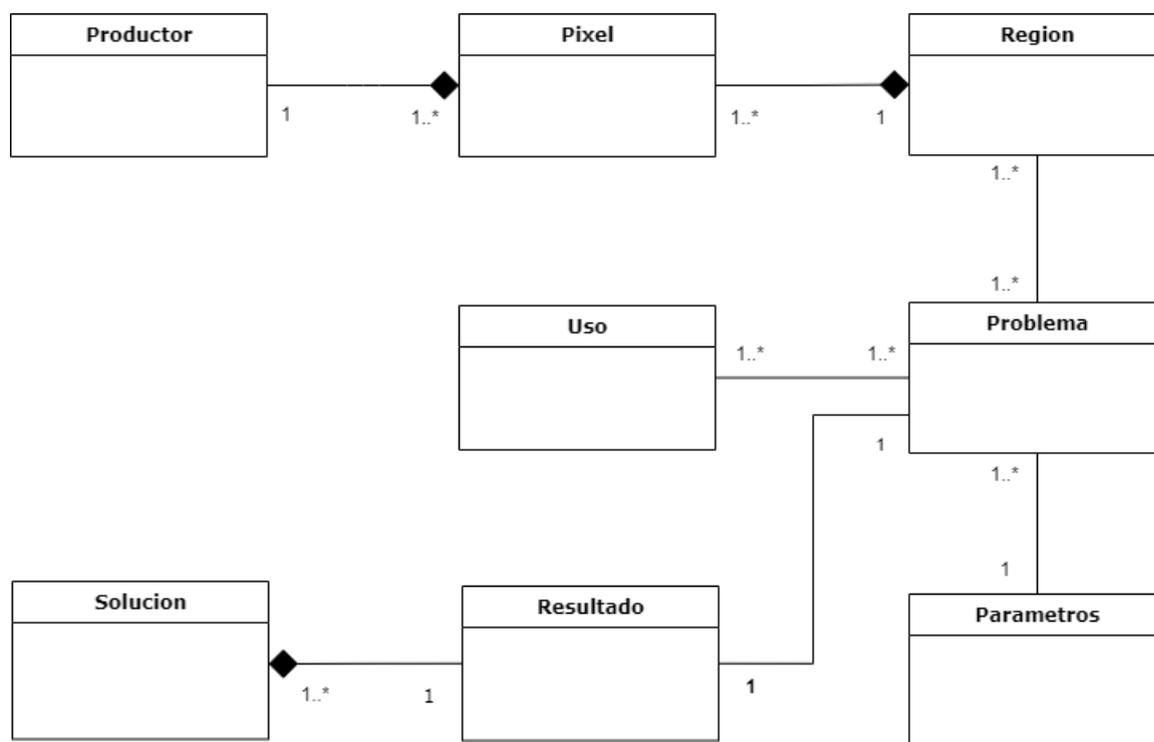


Figura 24 - Diagrama de Modelo Conceptual en UML.

4.1.4 - Casos de uso

Los siguientes casos de uso son obtenidos a partir del relevamiento de requerimientos. Estos son una especificación detallada de los requerimientos

funcionales, a partir de los cuales, junto con el modelo conceptual, se realiza el diseño. Es importante aclarar que la aplicación no debe contar con un sistema de administración de usuarios, cualquier persona que utilice la herramienta debe poder contar con todas sus funcionalidades.

Nombre	1) Agregar uso
Actores	Usuario
Descripción	El caso de uso comienza cuando un Usuario desea crear un uso. Para esto el Sistema autogenera un número de uso, luego el Usuario ingresa un nombre, selecciona una primera estación, ingresa una duración, la exportación de fósforo por estación, la productividad por estación, y selecciona los siguientes usos de un uso. Finalmente se le pide al usuario que confirme el ingreso creando el nuevo uso. En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar el ingreso del uso.

Para evitar tener que ingresar todos los datos de un uso nuevamente, y también para evitar equivocarse en el proceso, es deseable poder copiar un uso y luego modificarlo.

Nombre	2) Copia Modificada de un uso
Actores	Usuario
Descripción	El caso de uso comienza cuando un usuario desea copiar un uso. El Sistema lista todos los usos, y una vez que el usuario selecciona uno, se desencadena el caso de uso Agregar uso, teniendo todos sus atributos con valores por defecto iguales a los del uso copiado, (salvo el identificador), con la posibilidad de modificarlos. Finalmente se pide al usuario que confirme la copia modificada de un uso. En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la copia modificada del uso.

Nombre	3) Quitar uso
Actores	Usuario
Descripción	El caso de uso comienza cuando un usuario desea quitar un uso. El Sistema lista todos los usos, y el usuario selecciona uno para ser removido. Sólo se permitirá remover usos que no pertenezcan a otros componentes (problemas, regiones, soluciones, resultados). Finalmente se pide al usuario que confirme el quitar el uso. En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar el quitar uso.

Nombre	4) Modificar siguientes usos a un uso
Actores	Usuario
Descripción	El caso de uso comienza cuando un usuario desea modificar los siguientes usos de un uso (esto refiere a la restricción de respetar la secuencia de cultivos durante toda la planificación). Para esto se listan todos los usos del Sistema, el usuario puede marcar o desmarcar cada uso, seleccionando así los usos siguientes que desee. Finalmente se pide al usuario que confirme la modificación de los siguientes usos. En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la modificación

	de los Sigüientes usos del uso.
--	---------------------------------

Nombre	5) Importar región
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando el usuario desea importar una región, para esto se le permite seleccionar un archivo de formato shape (.shp) desde una ventana de navegación. El archivo debe contar con una colección de píxeles, cada uno de ellos debe tener un identificador numérico, un número de productor al que pertenece, su superficie en hectáreas, y un uso heredado (esto es: el número del uso que está plantado actualmente en dicho píxel y la cantidad de estaciones que lleva plantado dicho uso).</p> <p>El Sistema lee el archivo, creando un píxel para cada elemento de la colección. Luego de generado todos los píxeles, genera los productores dueños de estos, y la región que los contiene. En caso de que el formato sea correcto y la información coherente, se despliega un resumen de la información leída (cantidad de píxeles, cantidad de productores, etc.), y se permitirá confirmar la importación de la nueva región (pidiendo para esto un nombre de región, y asignándole un identificador a esta).</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la importación de una nueva región.</p>

Nombre	6) Seleccionar región de un problema
Actores	Usuario
Descripción	<p>Se despliega un menú que lista todas las regiones (en el que se puede seleccionar la región con la que trabajar de entre todas las creadas) junto con un botón que permite ir al caso de uso Importar región. Además de un botón de aceptar y cancelar.</p>

Nombre	7) Seleccionar usos de un problema
Actores	Usuario
Descripción	<p>Se despliega un menú que lista todos los usos (en el que se puede seleccionar o deseleccionar cada uso) junto con un botón que permite ir al caso de uso Importar región. Además de un botón de aceptar y cancelar.</p> <p>En caso de dejar píxeles del problema cuyo uso heredado no posea sigüientes usos que pertenezcan al problema, el Sistema advertirá de la situación.</p> <p>Finalmente se pide al usuario que confirme la modificación de los usos del problema.</p>

Nombre	8) Armar problema
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando un usuario desea crear un problema.</p> <p>Para esto el usuario recibe un número de problema, y debe ingresar un nombre, y una duración. Luego, el usuario podrá continuar agregando una región al problema, iniciando el caso de uso Seleccionar región de un problema, o agregando los usos del problema iniciando el caso de uso Seleccionar usos de un problema, luego de esto el usuario podrá aceptar el agregar o cancelar el agregado de la región o de los usos.</p> <p>Por último el usuario deberá ingresar las restricciones del problema (de momento mínima y máxima cantidad de usos, ya que las restricciones de productividad son abordadas por el productor).</p> <p>Finalmente se pide al usuario que confirme el armado del problema.</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el</p>

	error en caso de que lo haya, y el usuario puede decidir cancelar el Armado del problema.
--	---

Nombre	9) Modificar problema
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando un usuario desea modificar un problema , que aún no ha sido ejecutado.</p> <p>Para esto el Sistema lista los atributos del problema permitiendo modificarlos, estos atributos son: el nombre, la duración de la planificación en estaciones, la estación inicial, y las restricciones de los productores (la mínima y máxima cantidad de usos distintos por estación, y la mínima productividad por estación).</p> <p>El Sistema también muestra las regiones (permitiendo al usuario cambiar su selección), y muestra la lista de usos seleccionados (también permitiendo al usuario cambiarlos, iniciando el caso de uso Modificar usos de un problema).</p> <p>Finalmente se pide al usuario que confirme la modificación del problema.</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la modificación del problema.</p>

Nombre	10) Quitar problema
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando un usuario desea eliminar un problema que no ha sido ejecutado. Para esto se listan todos los problemas del Sistema junto a un resumen de su región, sus usos y un botón para eliminarlo.</p> <p>En caso de seleccionar el botón el Sistema pide una confirmación y en caso de aceptarla, elimina los datos del problema.</p>

Nombre	11) Agregar configuración paramétrica
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando un usuario desea crear una configuración paramétrica.</p> <p>Para esto el usuario recibe un número de configuración paramétrica, y debe ingresar un nombre. Luego de esto puede seleccionar si definir una semilla o dejarla aleatoria, cantidad de soluciones a generar, cantidad máxima de mejoras a permitir, factor de relajación, y de peso de las restricciones.</p> <p>Finalmente se le pide al usuario que confirme la creación de la configuración paramétrica.</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la creación de la configuración paramétrica.</p>

Nombre	12) Seleccionar configuración paramétrica para una Ejecución
Actores	Usuario
Descripción	<p>El caso de uso comienza cuando un usuario desea seleccionar una configuración paramétrica para un problema que no ha sido ejecutado.</p> <p>El sistema lista las configuraciones paramétricas, permitiendo al usuario modificarlas, y seleccionar una para ejecutarla junto al problema.</p> <p>En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la selección de la configuración paramétrica.</p>

Nombre	13) Realizar Ejecución
Actores	Usuario
Descripción	El caso de uso comienza cuando un usuario desea ejecutar un problema. En caso de que el problema no tenga una configuración paramétrica seleccionada se invoca el caso de uso Seleccionar configuración paramétrica, y en caso de que se haya seleccionado una configuración paramétrica se ejecuta el problema con la misma, devolviendo un resultado. En todos los casos el Sistema realiza los controles pertinentes, indicando el error en caso de que lo haya, y el usuario puede decidir cancelar la ejecución del problema.

Nombre	14) Mostrar resultado
Actores	Usuario
Descripción	El caso de uso comienza cuando un usuario desea ver el resultado de una ejecución de un problema. El Sistema lista los problemas ejecutados junto con sus configuraciones paramétricas correspondientes, permitiendo al usuario seleccionarlos. Al seleccionar un problema y una configuración paramétrica asociada a este, el Sistema mostrará una breve descripción de ambos (para el problema, su nombre, y el nombre de su región asociada, y para la configuración paramétrica, su nombre y semilla asociada). También se le dará al usuario la posibilidad de acceder al problema y a la configuración paramétrica de manera de poder ver todos sus detalles. Además se mostrará una solución sin incumplimientos (esto quiere decir que cumple todas sus restricciones), y tres listas de soluciones con incumplimientos ordenadas por menor cantidad de exportación de fósforo, menor cantidad de incumplimientos de cantidad de usos, y menor cantidad de incumplimientos de productividad. Estas soluciones con incumplimientos son soluciones no dominadas entre sí. Cada solución mostrada se describe en el caso de uso Mostrar solución.

Nombre	15) Mostrar solución
Actores	Usuario
Descripción	Cada solución deberá mostrar la cantidad de fósforo exportado y la cantidad incumplimientos de las restricciones (en caso de incumplir), además le debe dar la posibilidad al usuario de visualizar dicha solución en un mapa, ver gráficas que describen la solución (gráficas de productividad contra fósforo por estación, y de superficie de uso por estación), y guardar archivos de texto que describan detalladamente la solución (usos asignados por estación por píxel, cantidad de usos distintos por estación por productor, productividad sobre área total por estación por productor, fósforo exportado sobre área total por estación por productor).

4.1.5 - Definición de Conceptos del Dominio

A continuación se vuelven a mencionar varios conceptos ya definidos en la sección 2.4.1 Conceptos, los cuales, en este contexto se enfocan principalmente en la herramienta que resuelve el problema planteado.

- **Pixel:** Parcela, mínima unidad de tierra productiva a planificar ($j \in J$ en la formulación implícita).
- **Región:** Conjunto de píxeles (J en la formulación implícita).
- **Productor:** Dueño de varias parcelas, cada parcela pertenece a un único productor ($p \in P$ en la formulación implícita).

- **Uso:** Uso productivo que puede ser plantado en una parcela ($k \in K$ en la formulación implícita). Cuenta con los siguientes parámetros:
 - Una duración en cantidad de estaciones, la cual indica la duración de su ciclo productivo (esto es, el tiempo desde que se planta hasta que es sustituido por otra plantación ya que deja de ser rentable).
 - Un conjunto de posibles usos que pueden ser plantados a continuación de este ($K_{kd} \subset K$ en la formulación implícita).
 - Una cantidad de exportación de fósforo por estación (f_{kd} en la formulación implícita).
 - Una productividad por estación (g_{kd} en la formulación implícita).
- Además, un uso tiene las siguientes restricciones:
- Al momento de seleccionar un nuevo uso para ser plantado en su lugar, se debe elegir a alguno de los usos que pueden ser plantados a continuación del uso actual ($K_{kd} \subset K$ en la formulación implícita).
 - Solo se puede elegir un nuevo uso cuando el uso actual termina su ciclo productivo, no antes.
- **Problema:** Instancia del problema de Planificación de rotación de cultivos. Compuesta de una región en la que se debe planificar qué usos productivos plantar durante un conjunto de estaciones y el conjunto de especificaciones.
 - **Solución:** Una planificación de rotación de cultivos con algunos reportes asociados, por ejemplo, la cantidad de fósforo por hectárea que exportan todos los píxeles de cada productor en cada estación, o la productividad por hectárea que tienen todos los píxeles de cada productor en cada estación.
 - **Soluciones no dominadas:** Dados dos o más criterios, un conjunto de soluciones es no dominado cuando ninguna de ellas es mejor que cualquiera de las otras según estos criterios.
 - **Algoritmo de búsqueda:** Algoritmo heurístico que busca soluciones al problema de rotación de cultivos intentando cumplir las restricciones y minimizar la exportación de fósforo.
 - **Ejecución:** Ejecución del algoritmo de búsqueda para obtener soluciones a una Planificación de rotación de cultivos. Dos ejecuciones difieren según sus componentes asignados (usos, región, y/o parámetros asignados).
 - **Resultado:** Conjunto de mejores soluciones obtenidas por el algoritmo, incluye la mejor solución sin incumplimientos, y tres listas de soluciones no dominadas, ordenadas por los distintos criterios (mínimas exportaciones de fósforo, incumplimiento de productividad e incumplimiento de cantidad de usos.).
 - **Parámetros:** Configuración paramétrica del algoritmo que especifica su comportamiento.

4.2 - Diseño

4.2.1 - Arquitectura

Se opta por un diseño de arquitectura en tres capas: presentación, lógica y persistencia. Este proyecto se centra en el diseño e implementación de la capa de presentación, considerando un usuario final técnico con conocimientos agronómicos y

experiencia en programas SIG. En la capa lógica se agrega una interfaz que permite acceder a sus funcionalidades. Y finalmente la capa de persistencia se basa en la escritura y lectura de archivos de texto plano.

4.2.2 - Capa de presentación

Dada la separación conceptual definida se decide diseñar la herramienta en una sola ventana principal con una pestaña de inicio con información de referencia y cuatro pestañas que gestionen la creación, modificación, y eliminación de los componentes principales (usos, regiones, configuración paramétrica y problemas), y una pestaña para el historial de resultados donde acceder a las soluciones obtenidas en las búsquedas.

Este diseño se elige por ser simple y clásico, siendo similar a muchos menús de páginas webs y aplicaciones de escritorio con los cuales el usuario puede estar familiarizado, lo cual se espera facilite su uso. En particular, las pestañas se muestran como una columna en el lado izquierdo de la ventana y no como una fila en el margen superior, para darles una mejor visualización y acceso.

Glosario de elementos de la interfaz

Se detalla en la Tabla 10 un breve glosario de componentes mencionados durante el diseño de la interfaz gráfica.

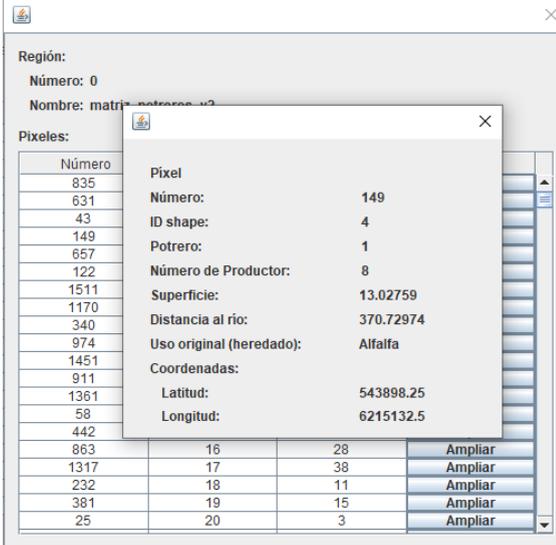
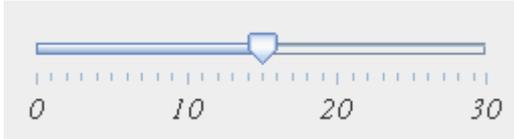
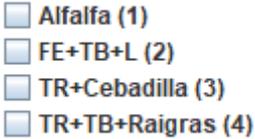
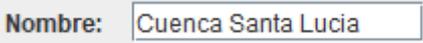
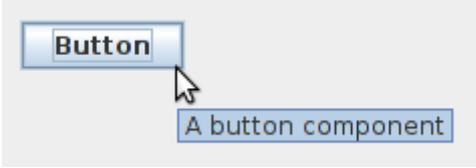
Nombre	Imagen	Descripción
Ventana tipo pop-up		Ventana emergente que aparece por encima del contenido
Slider bar		Herramienta de selección que representa un rango de valores como una barra, de la que el usuario puede elegir un único valor.
Radio button		Herramienta de selección que despliega todas las opciones como botones seleccionables, donde el usuario puede seleccionar solo una opción.
Checkbox		Herramienta que permite seleccionar o deseleccionar una o más opciones.
Input		Herramienta de ingreso de texto, caja que permite el ingreso de texto mediante escritura.
Hover		Herramienta de visualización, despliega información junto al puntero del mouse cuando este se ubica sobre un componente.

Tabla 10 - Glosario de componentes de la interfaz gráfica

Pestaña de Inicio

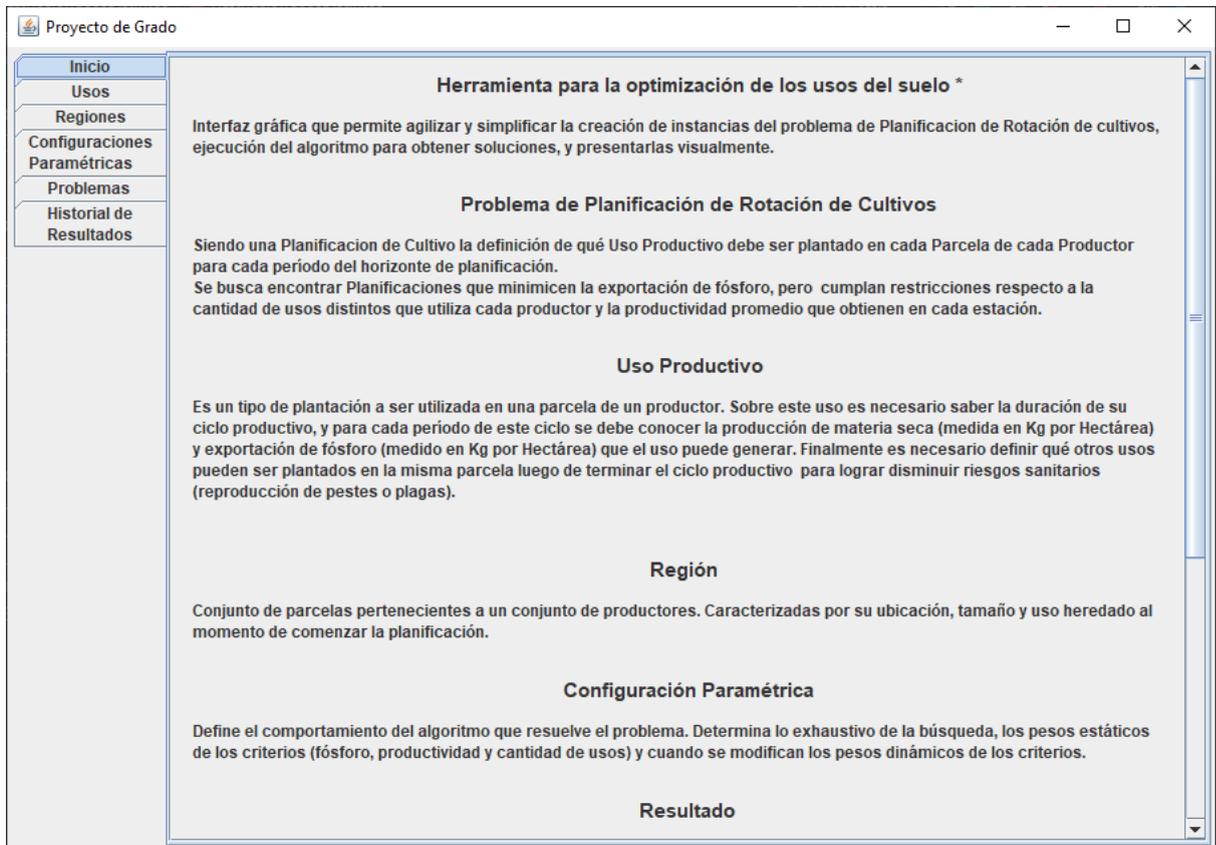
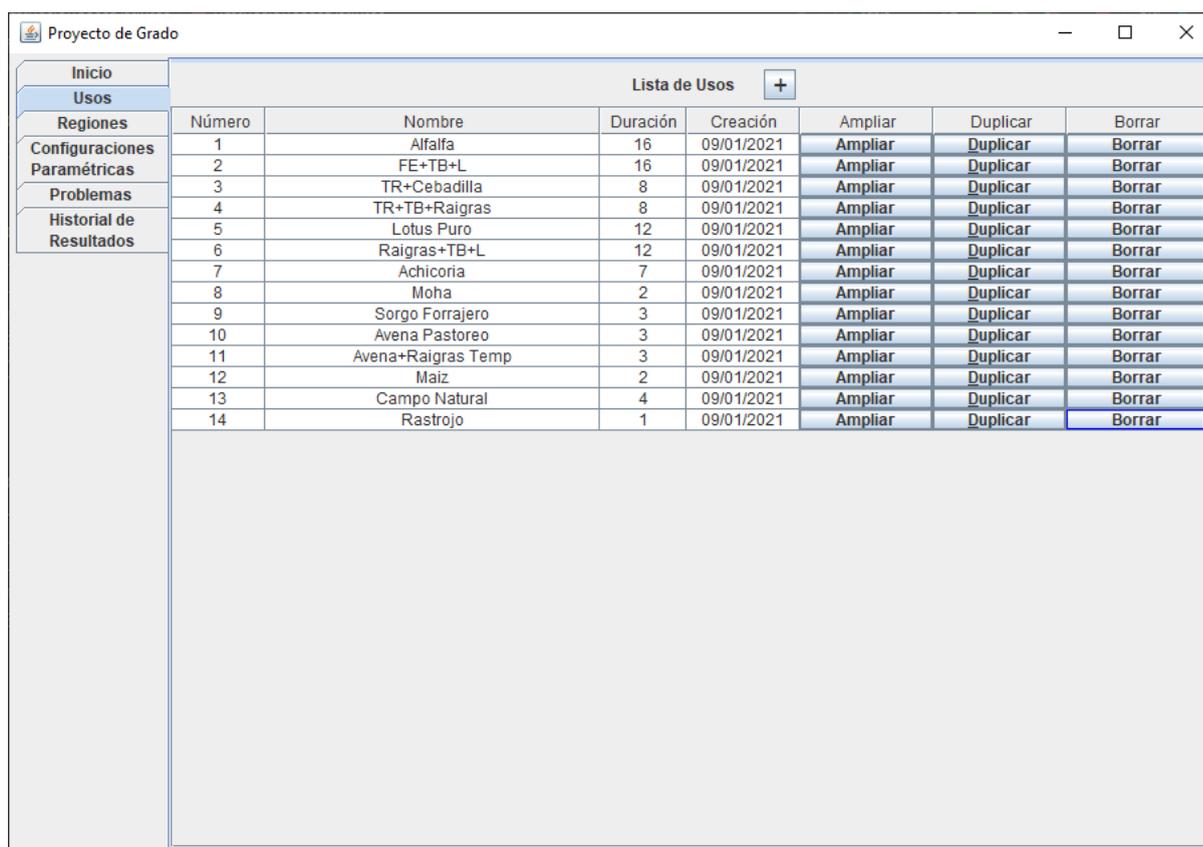


Figura 25 - Pestaña de Inicio

En la figura 25 se puede ver la pestaña de inicio. Esta pestaña es simplemente informativa, el usuario no puede interactuar con ella. En ella se definen varios conceptos necesarios para entender el problema y la herramienta implementada.

Pestaña de usos



	Número	Nombre	Duración	Creación	Ampliar	Duplicar	Borrar
Inicio							
Usos							
Regiones							
Configuraciones Paramétricas	1	Alfalfa	16	09/01/2021	Ampliar	Duplicar	Borrar
Problemas	2	FE+TB+L	16	09/01/2021	Ampliar	Duplicar	Borrar
Historial de Resultados	3	TR+Cebadilla	8	09/01/2021	Ampliar	Duplicar	Borrar
	4	TR+TB+Raigras	8	09/01/2021	Ampliar	Duplicar	Borrar
	5	Lotus Puro	12	09/01/2021	Ampliar	Duplicar	Borrar
	6	Raigras+TB+L	12	09/01/2021	Ampliar	Duplicar	Borrar
	7	Achicoria	7	09/01/2021	Ampliar	Duplicar	Borrar
	8	Moha	2	09/01/2021	Ampliar	Duplicar	Borrar
	9	Sorgo Forrajero	3	09/01/2021	Ampliar	Duplicar	Borrar
	10	Avena Pastoreo	3	09/01/2021	Ampliar	Duplicar	Borrar
	11	Avena+Raigras Temp	3	09/01/2021	Ampliar	Duplicar	Borrar
	12	Maiz	2	09/01/2021	Ampliar	Duplicar	Borrar
	13	Campo Natural	4	09/01/2021	Ampliar	Duplicar	Borrar
	14	Rastrojo	1	09/01/2021	Ampliar	Duplicar	Borrar

Figura 26 - Pestaña de usos

En la figura 26, pestaña de usos, se muestran los usos disponibles en la herramienta, los datos mostrados de cada uno son: su número (identificador del uso), nombre, duración en cantidad de estaciones, y su fecha de creación en formato DD/MM/AAAA. Cada uso en esta lista puede ser ampliado, lo que despliega una nueva ventana de tipo *pop-up* donde se muestran todos los atributos del uso. El uso también puede ser duplicado, creando una copia del uso (con otro número y nombre), y también puede ser borrado (a menos que esté asociado a un problema). En la parte superior de la pestaña, junto al título, se agrega un botón para cargar nuevos usos, simbolizado con un “+”.

Se elige este diseño por considerarse clásico, simple, y entendible. Otros posibles diseños podrían haber tenido el botón de “cargar nuevo uso” junto al nombre de la pestaña u oculto en un menú de 3 puntos, pero esto podría haber sido confuso o innecesario ya que no habrían más opciones dentro del menú. Otros diseños podrían haber mostrados otros atributos de los usos al igual que menos botones en cada fila, omitiendo el botón de “Ampliar” (teniendo que clicar en la fila para ampliar un uso), u omitiendo los botones de duplicar y/o borrar (dejándolos dentro de la ventana de “uso ampliado”).

Número: 1

Nombre: Alfalfa

Duración en estaciones: 16

Estaciones Iniciales: Primavera Verano Otoño Invierno

Sigüientes usos:

- Alfalfa (1)
- FE+TB+L (2)
- TR+Cebadilla (3)
- TR+TB+Raigras (4)
- Lotus Puro (5)

Exportación de Fósforo por estación (kg P/ha):

Estación 1	Estación 2	Estación 3	Estación 4
0.513	0.513	0.513	0.513
0.513	0.513	0.513	0.513
0.513	0.513	0.513	0.513
0.513	0.513	0.513	0.513

Productividad por estación (kg MS/ha):

Estación 1	Estación 2	Estación 3	Estación 4
0.0	0.0	3025.0	2475.0
3000.0	1200.0	4200.0	3600.0
1600.0	800.0	3200.0	2400.0
700.0	350.0	3500.0	2450.0

Cancelar **Aceptar**

Figura 27 - Ventana de uso ampliado

En la figura 27, se muestra la ventana de uso ampliado, la cual se utiliza para cargar nuevos usos y para ampliar los ya existentes.

El número del uso es su identificador (distinto para cada uso), es autogenerado, y se le mostrará al usuario cada vez que se haga referencia a un uso, junto con su nombre. El nombre del uso debe ser una o más palabras alfanuméricas. La “duración en estaciones” debe ser un número y define la cantidad de estaciones que este uso permanece plantado. Las “estaciones iniciales” son todas las posibles estaciones en las que este uso puede ser plantado. “Sigüientes usos” es la lista de los usos que pueden ser plantados luego de este. “Exportación de fósforo por estación” es una tabla de valores reales, la cantidad de fósforo exportado para cada estación en kilogramos de fósforo por hectárea; cada fila representa un año. “Productividad por estación” es una tabla también de valores reales donde cada fila representa la productividad del uso en kilogramos de materia seca por hectárea.

En cuanto al diseño elegido, se puede destacar al parámetro “Duración en estaciones”, el cual podría haber sido un *slider bar* con el que el usuario pudiese ingresar el valor adecuado, pero esto hubiese delimitado el valor máximo del parámetro o haber confundido por tener muchos valores posibles intentado mitigar el primer punto. También el parámetro “Estaciones iniciales” fue inicialmente pensado como un grupo de *radio buttons*, pero esto se cambió a un grupo de *checkboxes* ya que este parámetro puede contar con más de un valor (una o más estaciones iniciales). El diseño utilizado para los restantes parámetros, fue el más simple y entendible posible, una lista de *checkboxes* para los siguientes usos, y tablas para la exportación de fósforo y productividad mínima.

En caso de estar editando un uso, este no se podrá modificar si ya pertenece a un problema, el botón de “aceptar” quedará deshabilitado. Esto se debe a que, en caso de permitirlo, sería posible convertir soluciones válidas de un problema (encontradas previamente) en soluciones inválidas, al alterar las precedencias de usos o la duración en estaciones. También sería posible invalidar una solución al alterar los valores de productividad del uso, ya que si se disminuyese su productividad, se podría llegar a incumplir la restricción de mínima productividad por estación.

Pestaña regiones

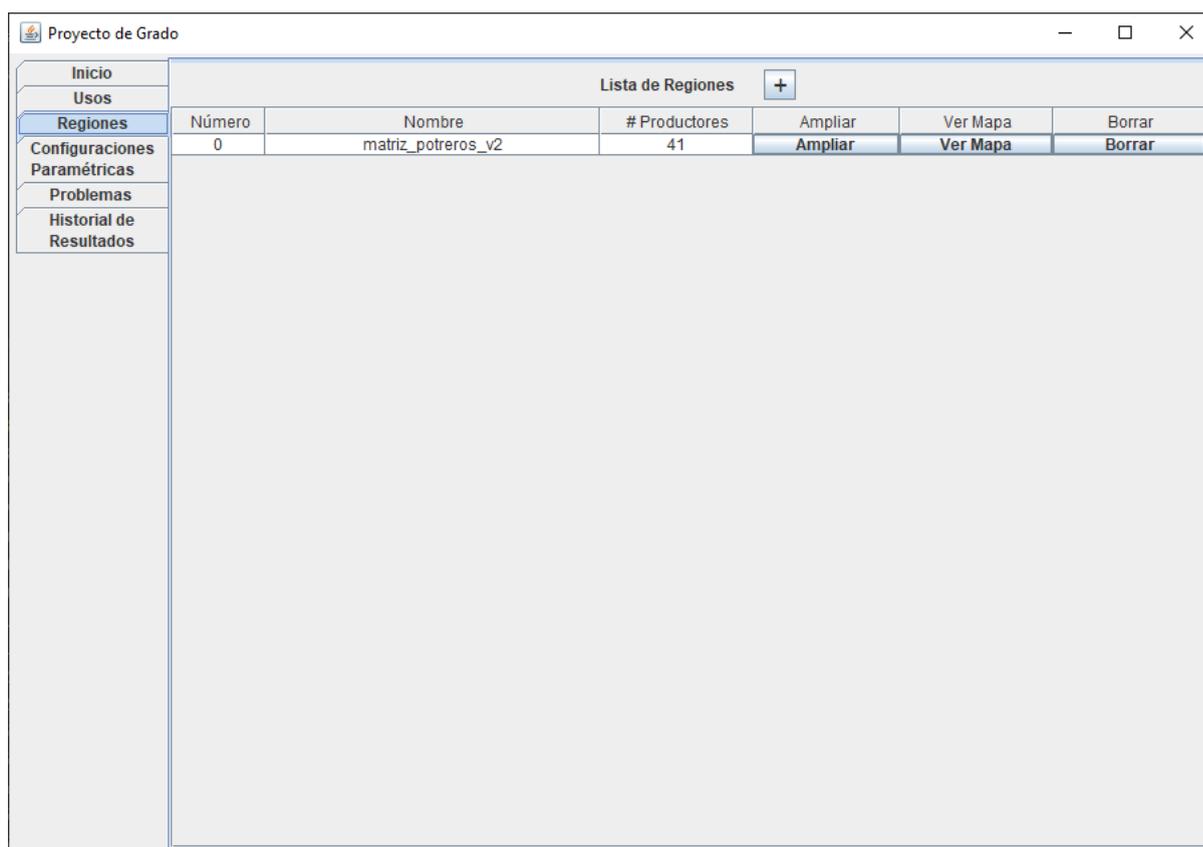


Figura 28 - Pestaña de regiones.

En la figura 28, pestaña regiones se utiliza el mismo diseño que en la pestaña de usos, en esta pestaña se muestran las regiones disponibles en la herramienta. Los datos mostrados de cada una son: su número (identificador de la región), nombre (obtenido del archivo shape del que fue cargado), y su cantidad de productores. Cada

región en esta lista puede ser ampliada, lo que despliega una nueva ventana de tipo *pop-up* donde se muestra una lista de todos sus productores, pudiendo acceder también a toda la información de cada productor, y de cada pixel en la región. La región puede ser borrada (a menos que esté asociada a un problema), y también puede ser visualizada en un mapa.

Al igual que en la pestaña de usos, junto al título se agrega un botón para cargar nuevas regiones, simbolizado con un “+”. Al clicar este botón, se despliega una ventana de navegación de carpetas del sistema de archivos con el que se puede elegir un archivo shape (extensión .shp) que contiene la nueva región a cargar. Además del archivo .shp, es necesario tener en la misma carpeta con el mismo nombre a los archivos de extensión .dbf y .shx (requisitos del formato shape), y es recomendable pero no necesario tener también los archivos .prj y .qix. (contienen datos redundantes que pueden ser accedidos por el usuario desde el visor de mapas).

Ventana de mapa de una región

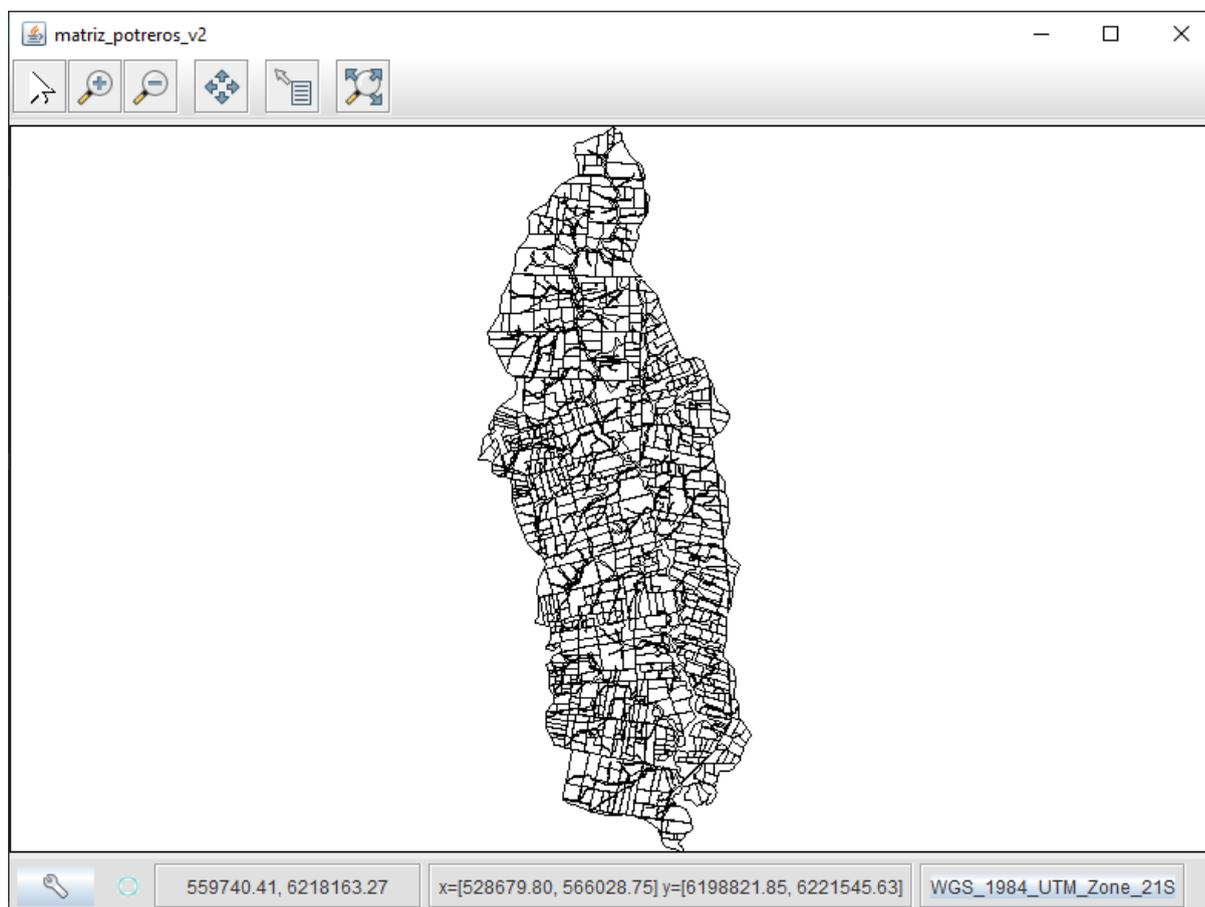


Figura 29 - Ventana de mapa de una región.

En la figura 29, se muestra la ventana de mapa de una región, en ella se carga el mapa en formato shape asociado a la región, en la parte superior se cuenta con algunas herramientas de visualización de mapas. Este paquete para manipular mapas es parte de la biblioteca GeoTools para Java, el cual fue elegido por considerarse

simple, potente, y por ser ampliamente utilizado tanto en la comunidad SIG, como en la comunidad Java.

Pestaña de configuraciones paramétricas

Lista de Parámetros de Configuración +					
Número	Nombre	Creación	Ampliar	Duplicar	Borrar
0	Configuración por Defecto	06/02/2021	Ampliar	Duplicar	Borrar
1	prueba rapida	23/02/2021	Ampliar	Duplicar	Borrar

Figura 30 - Pestaña de configuraciones paramétricas.

En la figura 30, pestaña de configuraciones paramétricas, se utiliza el mismo diseño que en la pestaña de usos y de regiones, en esta pestaña se muestran las configuraciones paramétricas disponibles en la herramienta. Los datos mostrados de cada una son: su número (identificador de la configuración), nombre, y su fecha de creación en formato DD/MM/AAAA. Cada configuración en esta lista puede ser ampliada, lo que despliega una nueva ventana de tipo *pop-up* donde se muestran todos sus atributos. La configuración también puede ser duplicada, creando una copia de sí misma (con otro número y nombre), y también puede ser borrada (a menos que esté asociada a un problema). En la parte superior de la pestaña, junto al título, se agrega un botón para cargar nuevas configuraciones paramétricas, simbolizado con un “+”.

Ventana de configuración paramétrica

The screenshot shows a window titled 'Configuración Paramétrica' with a close button (X) in the top right corner. The window contains the following fields and controls:

- Número:** 0
- Nombre:** Configuración por Defecto
- Fecha de Creación:** 06/02/2021
- Configuración:**
 - Semilla:** 0
 - Soluciones a generar GRASP:** 100
 - Soluciones a generar Local search:** 100
 - Mínima cantidad de First Improvements:** 10
 - Cantidad de strikes First Improvement:** 3
 - Máxima cantidad de First Improvements:** 15
 - Cantidad de intentos First Improvement:** 100
 - Peso global del Fósforo:** 1.0
 - Peso global de Usos:** 1.0
 - Peso global de la Productividad:** 1.0
 - Actualizar pesos del Local Search con mejora
 - Actualizar pesos del Local Search sin mejora
 - Actualizar pesos del GRASP con mejora
 - Actualizar pesos del GRASP sin mejora

At the bottom of the window, there are three buttons: 'Valores por defecto' (left), 'Cancelar' (right), and 'Guardar cambios' (right).

Figura 31 - Ventana de configuración paramétrica.

En la figura 31, se muestra la ventana de configuración paramétrica, esta se utiliza para cargar nuevas configuraciones paramétricas y para ampliar las ya existentes. Aquí se muestran todos los atributos de una configuración paramétrica. Todos los atributos

son requeridos, pudiendo dejarlos todos en su valor por defecto clickeando el botón “Valores por defecto”. El número de la configuración es su identificador (distinto para cada configuración), es autogenerated, y se le mostrará al usuario cada vez que se haga referencia a un uso, junto con su nombre. El nombre de la configuración debe ser una o más palabras alfanuméricas. Cada parámetro de la configuración tiene un *input* para ingresar su valor numérico (entero o real, dependiendo del parámetro, esto se puede diferenciar según su valor por defecto, si tiene o no número decimal), o un *checkbox* en caso de ser de tipo booleano. Además, cada parámetro tiene a su lado un botón simbolizado con “?” el cual despliega una ventana que explica su definición.

Este diseño fue elegido por considerarse intuitivo, y fácil de entender. Otros diseños posibles podrían no haber tenido el botón de explicación, mostrando el texto como un *hover* al pasar el mouse por encima del nombre del parámetro, pero esto quizás no hubiese sido muy claro, o hubiese quedado escondido para el usuario. También se podrían haber utilizado *slider bars* para elegir los valores de los parámetros numéricos, pero esto hubiese delimitado los posibles valores reales de cada parámetro, y también hubiese delimitado los valores mínimos y máximos de cada parámetro, pudiendo también llevar a pensar al usuario que esos valores no mostrados no son adecuados.

Pestaña de problemas

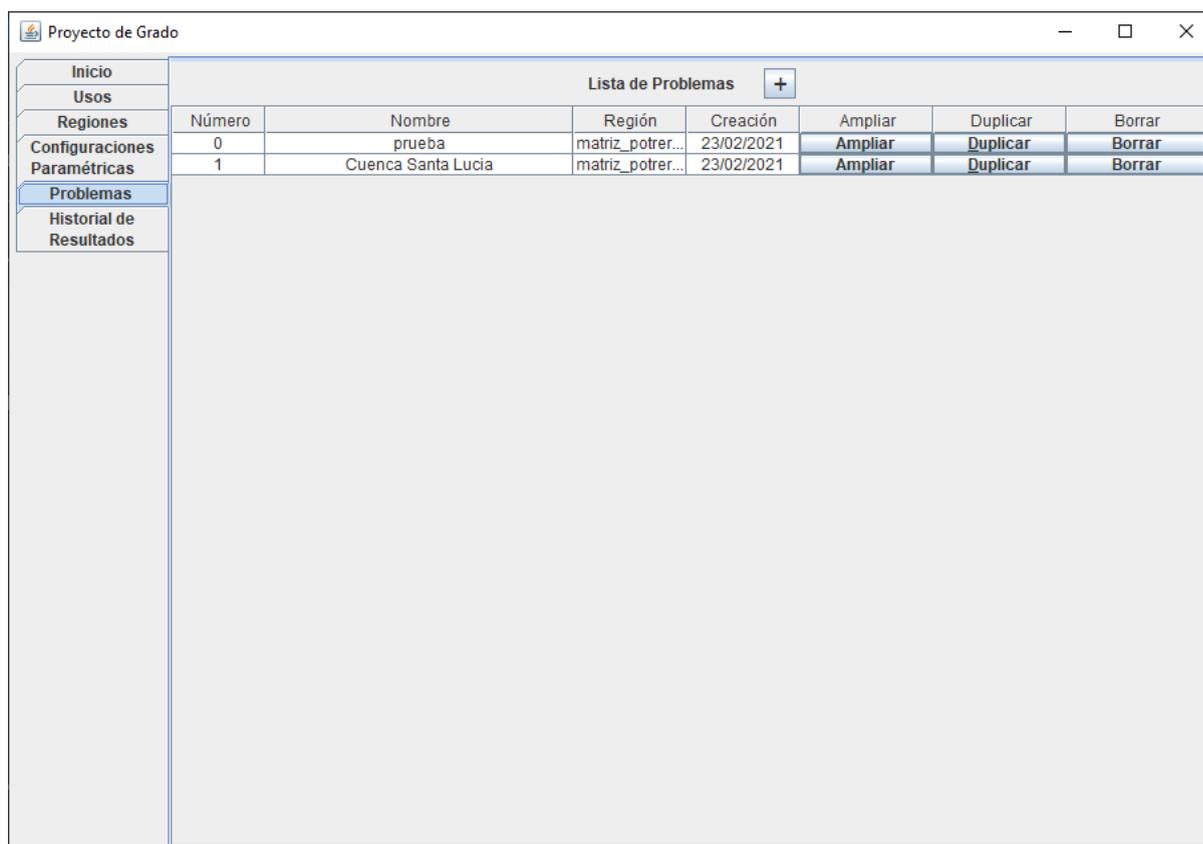


Figura 32 - Pestaña de problemas

En la figura 32, pestaña de problemas, se utiliza el mismo diseño que la pestaña de usos, de regiones, y de configuraciones paramétricas, y muestra los problemas cargados en la herramienta.

Ventana de problema

Problema

Número: 1

Nombre:

Fecha de Creación: 23/02/2021

Duración en estaciones:

Primer estación: Primavera Verano
 Otoño Invierno

Restricciones:

Mínima cantidad de usos distintos:

Máxima cantidad de usos distintos:

Productividad por estación (kg MS/ha):

Otoño	Invierno	Primavera	Verano
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

Figura 33 - Ventana de problema.

En la figura 33, ventana de problemas, se muestra un problema ampliado, se utiliza para cargar nuevos problemas y para mostrar los ya existentes. Todos sus atributos son requeridos. El número del problema es su identificador (distinto para cada problema), es autogenerado. El nombre del problema debe ser una o más palabras alfanuméricas.

El diseño de esta ventana es muy similar a la de la ventana de uso ampliado, cambiando la forma de ingresar el valor de la "Primer estación" a un *radio button*, ya que en este caso la planificación solamente puede comenzar por una única estación.

Además se encuentran presentes las restricciones de cantidad de usos distintos que se ingresan como valores numéricos, y los usos, la región, y la configuración paramétrica asociada al problema. Estos 3 últimos componentes pueden ser asociados al problema con sus respectivos botones desplegando 3 nuevas ventanas de tipo *pop-up*.

Ventana de selección de usos de un problema

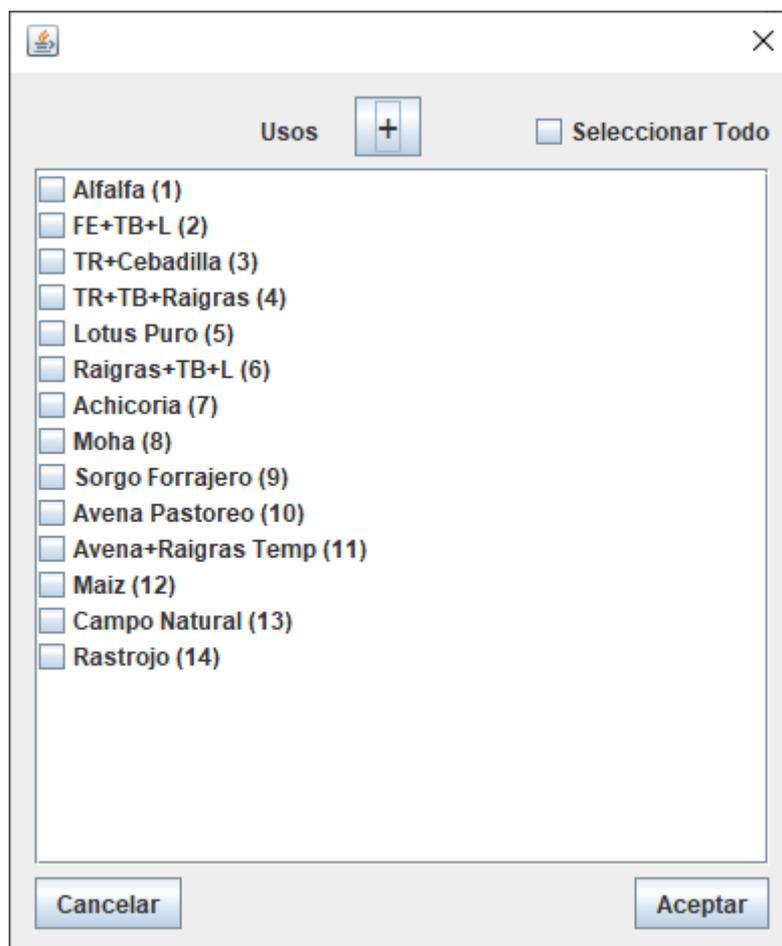


Figura 34 - Ventana de selección de usos de un problema.

En la figura 34, ventana de selección de usos de un problema, se pueden seleccionar los usos productivos que pertenecerán a un problema; cada uso está representado con su nombre, y su número entre paréntesis. Para ayudar al usuario a elegir más rápidamente los usos, se agregó el *checkbox* con el que se pueden seleccionar todos los usos de la lista. También se agregó el botón de “cargar nuevo uso” simbolizado con un “+”, el cual despliega la ventana de uso ampliado, permitiendo agregar un nuevo uso y asociarlo al problema.

El diseño de esta ventana se pensó para ser simple y similar a las anteriores ventanas de la herramienta, con un botón para agregar nuevos usos en la parte superior, y los botones de aceptar y cancelar en la parte inferior

Ventana de selección de región de un problema

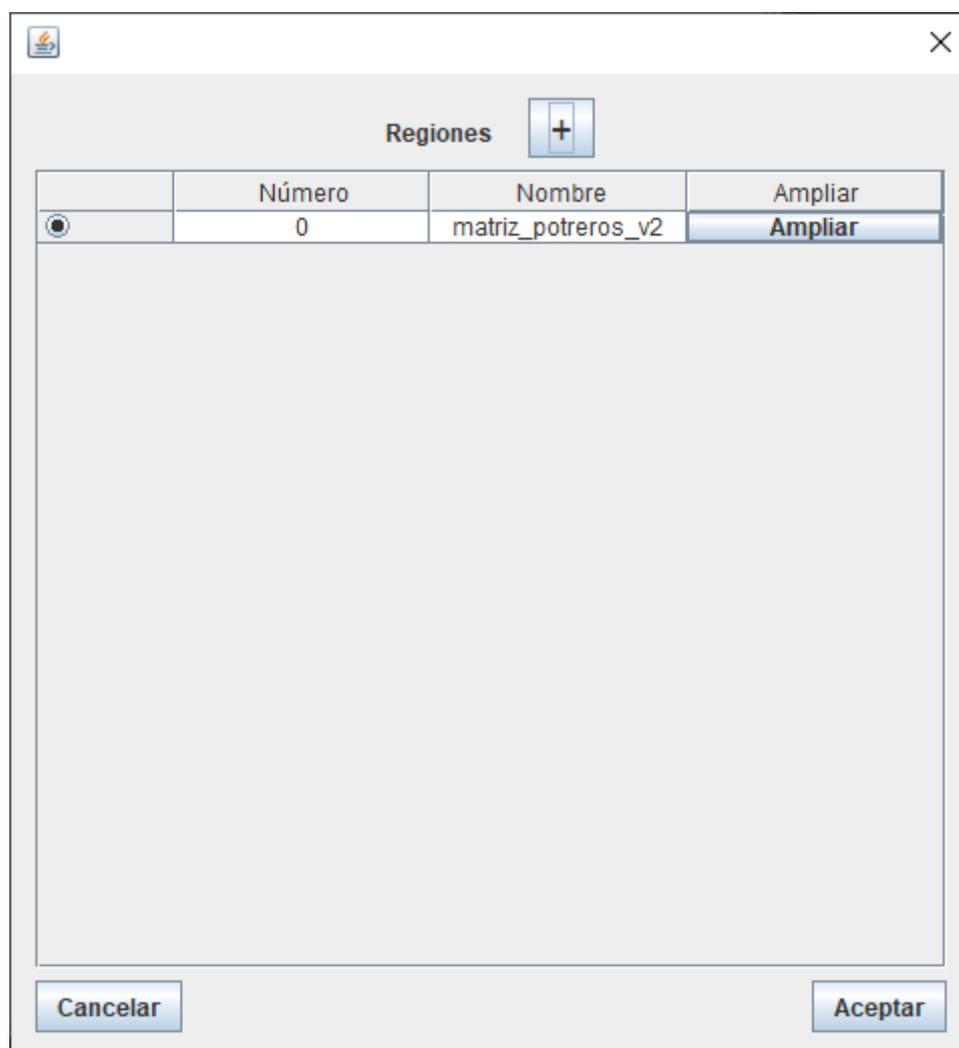


Figura 35 - Ventana de selección de región de un problema.

En la figura 35, ventana de selección de región de un problema, se puede seleccionar la región que pertenecerá a un problema; cada región está representada por su nombre y número.

Su diseño es muy similar a la ventana de asignación de usos a un problema, también contando con un botón de "cargar nueva región" para ayudar a simplificar las acciones del usuario.

Ventana de selección de configuración paramétrica de un problema

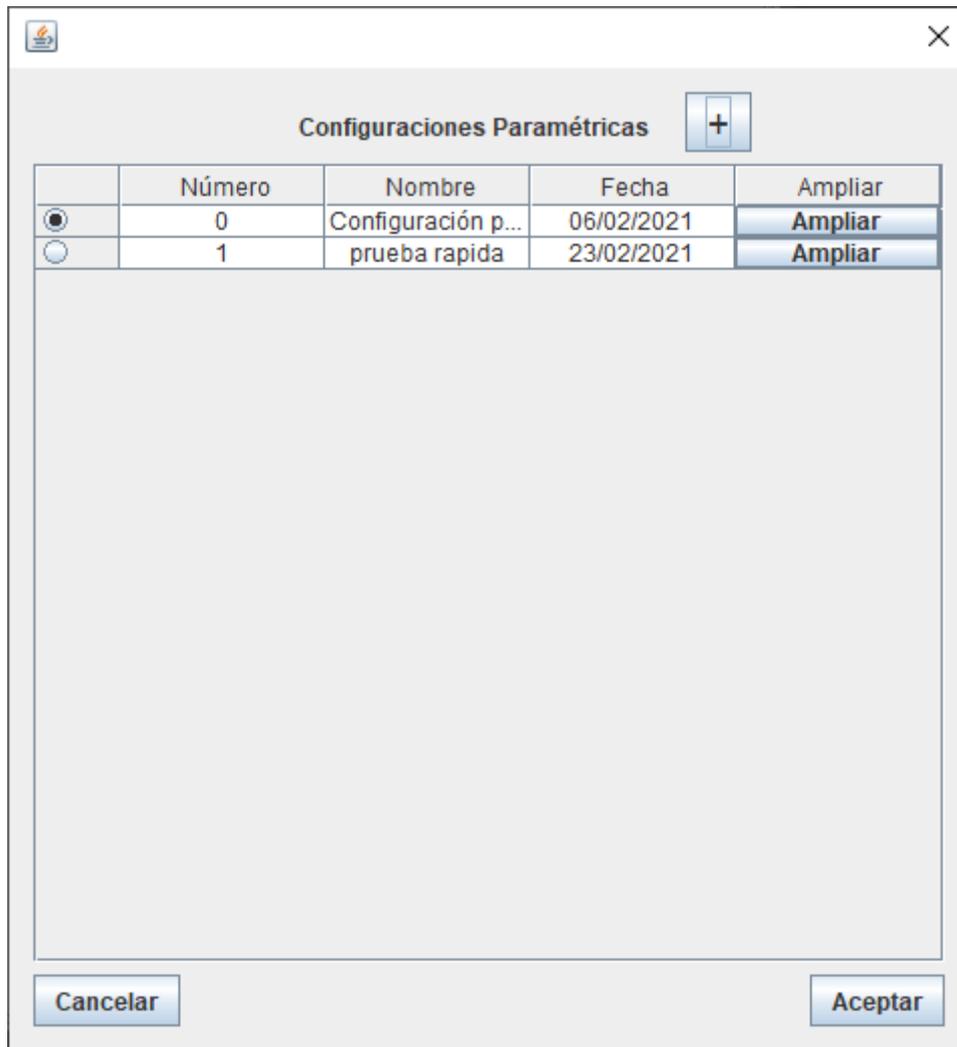


Figura 36- Ventana de selección de configuración paramétrica de un problema.

En la figura 36, ventana de Selección de configuración paramétrica de un problema, se puede seleccionar la configuración paramétrica que pertenecerá a un problema; cada configuración paramétrica está representada por su nombre, número, y fecha de creación.

Esta ventana cuenta con el mismo diseño que la ventana de asignación de región a un problema.

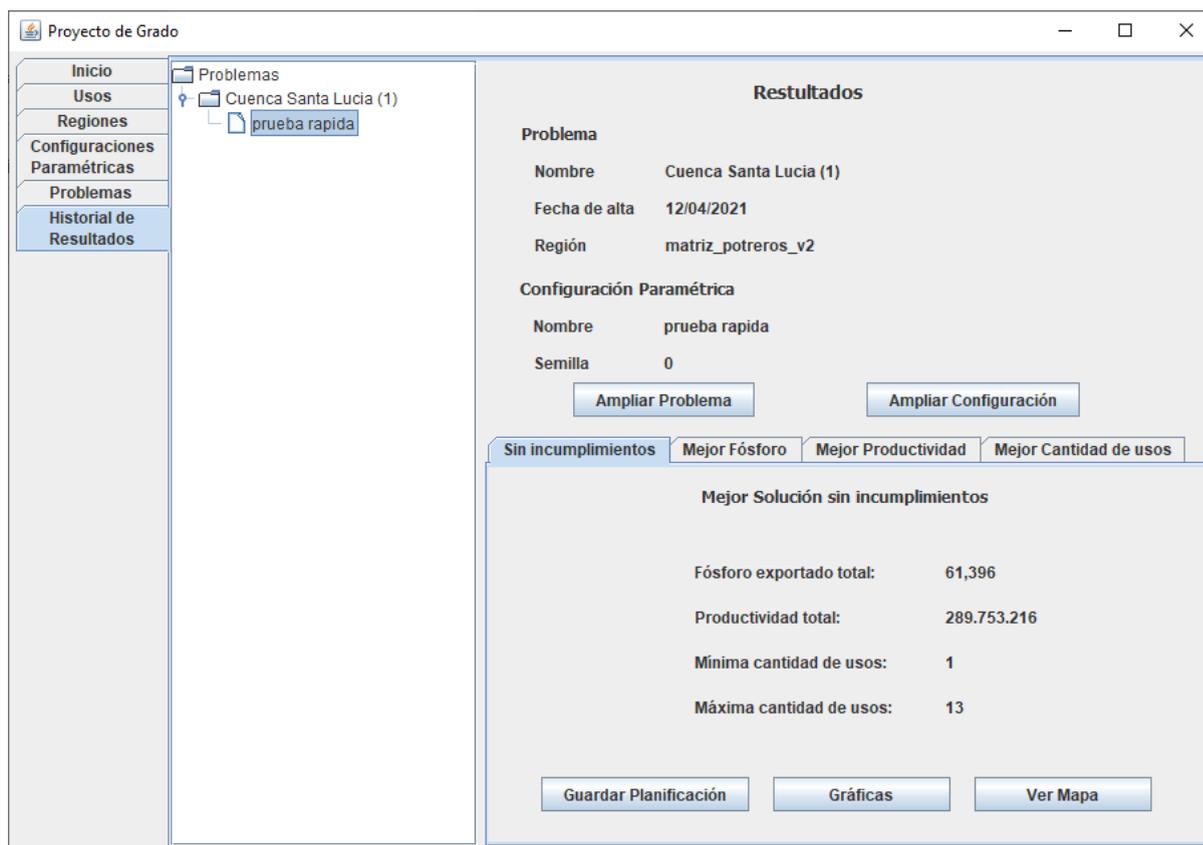


Figura 37 - Pestaña de historial de resultados.

En la figura 37, pestaña de historial de resultados, se muestran los resultados de todos los problemas cargados en la herramienta. En el lado izquierdo de la pestaña se puede ver una estructura arborescente donde el nodo principal contiene todos los problemas, y cada problema a su vez contiene configuraciones paramétricas. Esta estructura agrupa las configuraciones paramétricas de un problema en un mismo nodo. Al seleccionar una de estas configuraciones, se muestra en el lado derecho y arriba de la pantalla una breve descripción del problema, y de la configuración paramétrica (con la posibilidad de ampliar ambas con sus respectivos botones), y más abajo las soluciones encontradas. La solución sin incumplimientos se muestra en la primera pestaña, en la que se pueden ver cuatro propiedades importantes y tres botones (los cuales están disponibles para todas las soluciones presentadas), estos son “Guardar Planificación”, “Gráficas”, y “Ver Mapa”. El botón “Guardar Planificación” guarda en el mismo directorio donde está el problema, seis archivos de texto plano que tienen la planificación de la solución junto con información relativa a esta. El botón “Gráficas” despliega dos ventanas de tipo *pop-up* en las que se muestran la gráfica de eje dual con fósforo contra productividad por estación, y la gráfica de barras apiladas según la superficie de cada uso por estación. El botón “Ver Mapa” despliega la ventana de mapa de una solución.

Un diseño alternativo para la estructura de árbol mencionada anteriormente, fue el utilizar una lista de problemas (donde se mostraría a cada problema con su configuración asociada), pero con la estructura arborescente se pueden comparar más

fácilmente dos soluciones pertenecientes al mismo problema (con distintas configuraciones paramétricas). También, de esta manera se da un mejor uso del espacio vertical en esta sección ya que esta estructura muestra los nombres de las configuraciones paramétricas de un mismo problema debajo del nombre del problema y no al lado como se hubiese mostrado de haber sido una lista.

La sección superior derecha, donde se muestra una breve descripción del problema y de la configuración paramétrica, fue pensada para que tuviese un diseño muy similar a la ventana de problema para que sea fácil de entender por el usuario.

La sección inferior derecha debía ser separada en cuatro subsecciones, una para cada tipo de solución encontrada, por lo que se optó por utilizar pestañas horizontales clásicas para ahorrar lugar. En el caso de la mejor solución sin incumplimientos se optó por mostrar la información más relevante de la solución, que sería el valor a minimizar (el fósforo exportado total), y ambas restricciones (la productividad total, y la mínima y máxima cantidad de usos distintos). Además se muestran los tres botones ya mencionados, los cuales se utilizan en todas las soluciones. Las demás pestañas muestran las soluciones no dominadas, con incumplimientos, como se puede ver en la figura 38.

Sin incumplimientos		Mejor Fósforo	Mejor Productividad	Mejor Cantidad de usos		
#	Fósforo (tons)	# Inc. Prod.	# Inc. Usos.	Guardar	Gráficas	Mapa
1	62,458	0%	52,29%	Guardar	Gráficas	Mapa
2	61,813	0%	52,44%	Guardar	Gráficas	Mapa
3	61,697	0%	52,74%	Guardar	Gráficas	Mapa
4	61,449	0%	52,90%	Guardar	Gráficas	Mapa
5	61,035	0%	53,96%	Guardar	Gráficas	Mapa
6	61,002	0%	55,34%	Guardar	Gráficas	Mapa
7	60,988	0%	64,33%	Guardar	Gráficas	Mapa
8	60,965	0%	65,55%	Guardar	Gráficas	Mapa

Figura 38 - Pestañas de listas de soluciones no dominadas, con incumplimientos.

En la Figura 38, pestañas de listas de soluciones no dominadas, con incumplimientos, se muestra como cada pestaña tiene una lista de soluciones ordenada según cual sea su criterio, “Mejor Fósforo”, “Mejor Productividad”, y “Mejor Cantidad de usos”. Cada lista muestra la siguiente información: índice de la solución (la solución número uno es mejor que las demás para el criterio por el que está ordenado), Fósforo total exportado en toneladas, porcentaje de incumplimientos de productividad (esto es, el porcentaje de la cantidad de veces que se incumple en productividad, de un total de la máxima cantidad de veces que se puede incumplir en productividad, que sería cantidad de estaciones multiplicado por la cantidad de productores), porcentaje de incumplimientos de cantidad de usos distintos (calculado de la misma manera), y los

tres botones ya mencionados anteriormente: “Guardar Planificación”, “Gráficas”, y “Ver Mapa”. Los títulos de las columnas de estos datos debieron ser abreviados o acortados para que no haya un desbordamiento. Además, los títulos de cada columna tienen un texto explicativo de tipo *hover* que aparece al pasar el mouse por encima de cada uno.

Se eligió este diseño de listas porque parecía el más simple para mostrar varias soluciones; los atributos mostrados de cada solución, juntos con sus botones son los mismos a los mostrados en la solución sin incumplimientos.

Ventanas de errores

Cuando ocurre un error en la aplicación o se da un error de configuración, la herramienta muestra una ventana de error con los detalles del mismo. Por ejemplo, cuando se intenta crear un problema con el nombre vacío (figura 39), o cuando se quiere cargar un mapa que no tiene formato shape (figura 40).

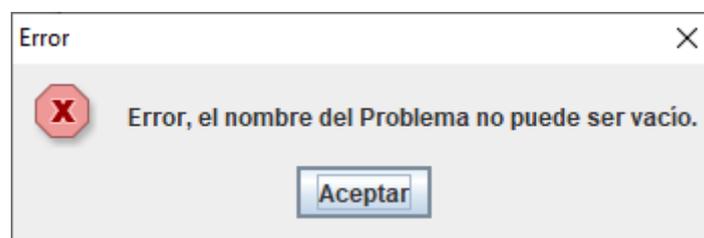


Figura 39 - Mensaje de error mostrado al usuario cuando se quiere crear un problema con nombre vacío.

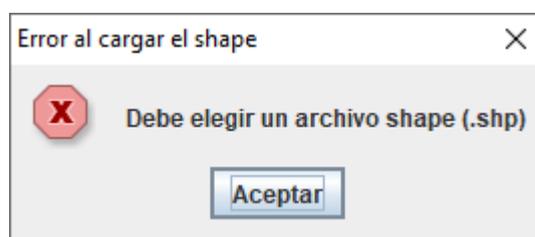


Figura 40 - Mensaje de error mostrado al usuario cuando se quiere cargar un mapa que no tiene formato shape.

La mayoría de estas ventanas se muestran al crear un componente, por ejemplo cuando un campo obligatorio no es llenado (como se muestra en la figura 39), o en una verificación de tipos (enteros y reales).

4.2.3 - Capa lógica

Partiendo de la herramienta GRASP diseñada para trabajar sobre un solo problema, la cual contemplaba sólo las clases: constantes, pixel, productor, uso, y solución, se abstrae un modelo más general que agrega una nueva clase, resultados, y se descompone la clase Constantes en las clases: problema, configuraciones paramétricas, y región (esto se debe a que anteriormente se trabajaba con un único problema, una única región, y una única configuración paramétrica que mantenían sus atributos en la clase Constantes).

Diagrama de clases

En la figura 41 se muestra un diagrama de clases simplificado mostrando sólo los atributos más importantes de cada clase, seguido de una explicación de sus clases.

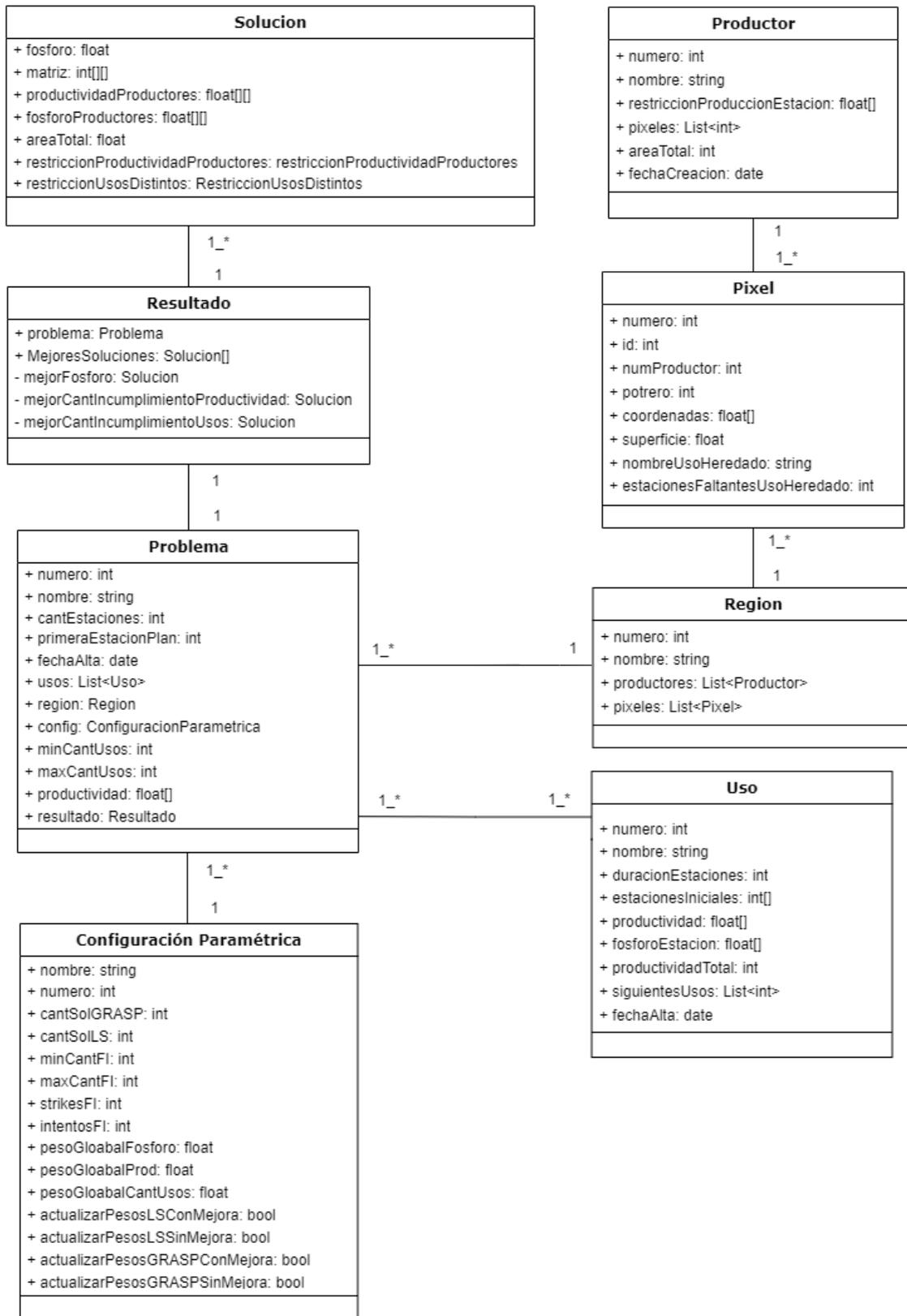


Figura 41 - Diagrama Clases simplificado.

La clase más importante, Problema, está compuesta de una lista de usos, una región, y una configuración paramétrica, y luego de ejecutarse genera un resultado que esencialmente es una colección de soluciones no dominadas (las mejores soluciones según fósforo, productividad, y usos distintos), y en caso de haber sido encontrada, la mejor solución sin incumplimientos. A su vez, cada región contiene píxeles, y cada píxel pertenece a un productor.

Se destaca la clase configuración paramétrica ya que es una clase nueva que no existía en el sistema legado. Esta permite la personalización de la búsqueda para adaptarla a las necesidades de cada instancia, intentando que la misma pueda afectar la replicabilidad de la búsqueda (semilla), controlar la duración de la ejecución (Cantidad de soluciones GRASP, soluciones a generar Local Search), balancear la exploración y explotación del espacio de soluciones (Mínima cantidad de First Improvement, Cantidad de strikes de First Improvement, Máxima cantidad de First Improvement, y Cantidad de intentos de First Improvement), modificar la importancia relativa de los distintos criterios (Peso Global de Fósforo, Peso Global de uso, y Peso Global de Productividad), y especificar el mecanismo de actualización de pesos (Actualizar pesos del Local Search y GRASP, con y sin mejora). Antes de utilizar esta clase, todos estos valores permanecían constantes durante la ejecución y debían ser cambiados al compilar la aplicación.

Para automatizar la importación de las regiones desde archivos shape es fundamental que todos los píxeles contengan los siguientes atributos (features de un SIG):

- ID: el identificador numérico de cada parcela, Pixel.numero.
- Productor: identificador numérico del productor al que pertenece la parcela, Productor.numero.
- POT: identificador numérico del potrero al que pertenece la parcela, Pixel.potrero.
- USO: nombre del uso plantado en la parcela cuando comienza la planificación, Uso.nombre.
- durUso: cantidad de estaciones que faltan para completar el ciclo productivo, Pixel.estacionesFaltantesUsoHeredado
- distance: distancia al río más cercano.
- Sup_ha: superficie de la parcela en hectáreas. Pixel.superficie.

Soluciones no dominadas

Con el objetivo de brindar solamente las mejores soluciones encontradas, se define el término "Soluciones no dominadas". Un conjunto de soluciones es no dominado cuando ninguna de ellas es mejor que cualquiera de las otras. Para que una solución sea considerada mejor que otra debe cumplir tres condiciones: tener menor (o igual) exportación de fósforo, tener una menor (o igual) cantidad de incumplimientos de usos distintos, y tener una menor (o igual) cantidad de incumplimientos de productividad. En las dos últimas condiciones se considera que el no tener incumplimientos de usos distintos o de productividad equivale a tener 0 incumplimientos (nunca menor a 0, en caso de cumplir una restricción con holgura). Esta herramienta brinda el resultado de un

problema como una colección de soluciones no dominadas. Estas soluciones se agrupan de la siguiente manera: la mejor solución sin incumplimientos (esto quiere decir que cumple las restricciones de mínima productividad y mínima cantidad de usos distintos), y tres listas cada una de las cuales conteniendo un máximo de 10 soluciones no dominadas ordenadas según los siguientes criterios: menor fósforo, menor cantidad de incumplimientos de productividad, y menor cantidad de incumplimientos de usos distintos, destacando que estas listas solamente contienen soluciones con incumplimientos (ya sea de productividad o de cantidad de usos distintos). De esta manera un resultado asociado a un problema contendrá, a lo sumo 31 soluciones, pudiendo no tener una solución sin incumplimientos (ya sea porque no exista una para el problema, o porque no se encontró), y pudiendo tener soluciones repetidas en estas tres listas de soluciones con incumplimientos. Esta funcionalidad es sugerida por los autores de este proyecto y validada con el supervisor, con la idea brindar al usuario una gran variedad de soluciones con incumplimientos, ya que los usuarios mencionaron que puede ser interesante contar con soluciones con baja cantidad de incumplimientos, y una gran minimización del fósforo exportado. Sumado a esto, se quiere que estas soluciones con incumplimiento no sean dominadas, contando así las mejores soluciones encontradas por el algoritmo.

4.2.4 - Capa de persistencia

Todos los componentes son persistidos en archivos de configuración. Estos son archivos de texto plano en formato JSON⁶, los cuales evitan que los componentes deban ser cargados nuevamente cada vez que se utiliza la herramienta. Cada vez que se crea o modifica un componente, este es almacenado/sustituido en su respectivo archivo, y cada vez que se borra un componente, este es eliminado de su respectivo archivo.

En la tabla 11 se pueden ver los archivos de configuración mencionados, estos se encuentran en el directorio “Resources”, y son nombrados por los componentes que almacenan, comenzando con un guión, y teniendo extensión txt (para evitar problemas de compatibilidad en distintos sistemas operativos).

-configParams.txt	Almacena las configuraciones paramétricas de la aplicación.
-problemas.txt	Almacena los problemas de la aplicación.
-regiones.txt	Almacena las regiones de la aplicación.
-usos.txt	Almacena los usos de la aplicación.

Tabla 11 - Archivos de configuración utilizados para la persistencia.

⁶ <https://www.json.org/json-en.html>

4.3 - Implementación

Se selecciona el Framework Swing⁷, dada la experiencia previa de los integrantes del proyecto y la mayor cantidad de artículos en StackOverflow⁸ sobre la alternativa JavaFX⁹.

Para el manejo de los componentes SIG se elige la biblioteca Geotools¹⁰, la cual es muy utilizada, tiene una gran comunidad, y en una primera prueba permitió implementar todas las funcionalidades deseadas.

La selección de Geotools forzó la adopción del manejador de dependencias Maven¹¹.

Para la creación y visualización de gráficas se elige la librería jFreeChart¹², por contar con muy buena documentación, ejemplos implementados de las gráficas requeridas (eje dual y barras apiladas), y compatibilidad con swing.

Para la persistencia de los componentes se evalúa la posibilidad de agregar una Base de Datos, pero quedando por fuera del alcance final se opta por la elaboración de archivos de texto como forma de almacenamiento.

4.4 - Ejecución de un caso típico de uso

La siguiente ejecución se considera una secuencia típica de casos de uso de la herramienta:

- Cargar uno o más usos nuevos en la Pestaña usos.
- Cargar una nueva región en su respectiva pestaña, importándola de un archivo shape.
- Cargar una nueva configuración paramétrica en su respectiva pestaña.
- En la Pestaña de problemas, crear un nuevo problema con los componentes previamente cargados, y ejecutarlo.
- Luego de finalizada la ejecución, visualizar el resultado del problema en la pestaña historial de resultados. En ella se debe buscar el resultado asociado al problema y configuración paramétrica. Luego de seleccionado el resultado, se podrá ver la mejor solución sin incumplimientos, y tres listas de soluciones no dominadas ordenadas según los distintos criterios.

Para comodidad de los usuarios finales se crea el documento **Guía de Uso** que detalla paso a paso la instalación de la herramienta, la creación de cada componente, ejecución de un problema, y lectura de los resultados. Este documento se describe en el anexo, sección Documentos adicionales.

⁷ <https://docs.oracle.com/javase/7/docs/api/javafx/swing/package-summary.html>

⁸ <https://stackoverflow.com/>

⁹ <https://openjfx.io/>

¹⁰ <https://geotools.org/>

¹¹ <https://maven.apache.org/>

¹² <https://www.jfree.org/jfreechart/>

4.5 - Despliegue

Para poder utilizar esta herramienta, se genera un archivo de despliegue jar, utilizando Maven. Con esto se intenta que cualquier usuario pueda utilizar la herramienta fácilmente, ya que el único requisito es tener instalado Java¹³ 1.8 o posterior. Este archivo jar debe ser ejecutado en una carpeta junto a la carpeta Resources donde se encuentran los distintos archivos de almacenamiento. Para ejecutarlo se puede agregar la variable de entorno¹⁴ necesaria para utilizar Java y correr el comando 'java -jar <nombre-del-jar.jar>', otra forma sería simplemente abrir el archivo .jar utilizando Java.

4.6 - Testeo y puesta en producción

A nivel interno al proyecto, se realizan:

- Pruebas de casos de uso, estas son pruebas de caja negra (no se toma en cuenta el código, ni la estructura de lo que se prueba) que toman como guía los casos de usos, simulando entradas válidas e inválidas, casos de borde, y comparando las salidas con las esperadas.
- Pruebas unitarias, que son pruebas de caja blanca (en estas se toma en cuenta la estructura del código, considerando los posibles flujos de ejecución) llevadas a cabo sobre un módulo concreto del sistema.
- Pruebas de integración (de caja negra), realizadas sobre la integración entre los módulos del sistema.
- Pruebas de cubrimiento (de caja blanca), en las cuales intentan ejecutar todas las líneas del código a probar.

A nivel externo, se cuenta con un usuario final, el Licenciado en Gestión Ambiental, Federico Weinstein, quien en el marco de su trabajo de Maestría "Modelación de los usos del suelo bajo un esquema de producción agrícola para disminuir la exportación de nutrientes a las aguas superficiales de la cuenca del río Santa Lucía" realiza pruebas de caja negra al utilizar esta herramienta para ejecutar nuevos problemas asociados al caso base y generando nuevos casos de estudio. Durante dicho trabajo ha ayudado a depurar la aplicación de fallas, encontrando errores. Además de que presenta un nuevo perfil de usuario, lo que aporta a adaptar el nivel de asistencia al usuario en la utilización de la herramienta, dado que la misma está concebida para difundir la aplicación de técnicas cuantitativas de gestión de usos de suelos productivos, en diferentes ámbitos de toma de decisión.

La interacción con el Lic. Weinstein ha llevado a agregar las siguientes mejoras a la herramienta:

- Corrección de un error en el que las soluciones generadas podrían tener usos seguidos no permitidos.
- Corrección de un error en el que no se podía mostrar una solución en un mapa si su problema asociado contaba con más de 30 usos productivos.
- Sugerencia de agregar títulos explicativos a ciertas ventanas de la herramienta que los tenían.

¹³ <https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>

¹⁴ https://www.java.com/es/download/help/path_es.html

- Corrección de un error en el que las soluciones no calculaban adecuadamente la productividad total.
- Corrección de un error en el que ciertos atributos de una región no eran correctamente cargados desde el archivo shape.
- Corrección de un error en el que las gráficas no mostraban adecuadamente la superficie total de las parcelas, ni las unidades adecuadas en los ejes.
- Sugerencia de agregar unidades de medida a ciertos valores que no los tenían.

5 - Conclusiones y trabajos a futuro

En este proyecto se creó un DSS para el Problema de Planificación de Rotación de Cultivos que implica:

1. Desarrollo del algoritmo de búsqueda GRASP: se partió de una primera versión, la cual se perfeccionó, en particular mediante un estudio computacional detallado (estudio paramétrico y estudio de comportamiento del algoritmo), y la corrección de algunos errores.
2. Una interfaz gráfica para la herramienta, esta interfaz brinda las siguientes funcionalidades adicionales:
 - el cargado de los componentes (usos, regiones, problemas) de forma manual en la interfaz
 - el nuevo componente llamado configuración paramétrica, el cual permite realizar búsquedas personalizadas.
 - la recopilación de varias soluciones no dominadas como resultado de una ejecución del algoritmo
 - la visualización del mapa de la región cargada
 - la visualización de las soluciones en un mapa
 - el exportado de una solución a archivos shape
 - la visualización de las soluciones en gráficas

Para desarrollar esta aplicación se utilizó Java, y su framework de interfaz gráfica Swing.

5.1 - Conclusiones

Los experimentos computacionales validaron el comportamiento del algoritmo y la necesidad de invertir tiempo en adaptar las configuraciones paramétricas sugeridas según la naturaleza de cada instancia a resolver, mejorando así el desempeño de la búsqueda.

Durante los experimentos computacionales fue posible sugerir valores por defecto de la configuración paramétrica según el tamaño de los productores, pero es difícil afirmar si los mismos no están fuertemente relacionados con el problema base (producción de pasturas para alimentación ganadera) y no serían efectivos con otro tipo de problemas (fruticultura, horticultura, floricultura, etc).

El diseño de la herramienta estuvo centrado en un usuario con un perfil técnico en Agronomía, capaz de crear nuevos componentes utilizando herramientas SIG para la definición de regiones, investigación académica para detallar los usos, y desarrollar un conocimiento empírico que le permita crear configuraciones paramétricas adaptadas a las instancias que lo ameriten.

Swing demostró ser un framework cómodo y potente, pudiendo resolver todos los problemas encontrados, aunque no es estéticamente la mejor opción (en caso de que ese criterio se vuelva importante se aconseja elegir un nuevo framework).

La biblioteca de manejo de mapas para Swing, Geotools, dió buenos resultados, se llegó a todos los requerimientos importantes, (cargar, colorear, y mostrar mapas), gracias a las guías y plantillas aportadas por la extensa comunidad que la utilizan. El

único inconveniente fue la necesidad de integrar Maven como gestor de proyecto (siendo la única forma de trabajar con el framework), el cual no fue sencillo de integrar, aunque resultó ser una herramienta adecuada con la cual trabajar.

La experiencia con JFreeChart fue sumamente positiva y se recomienda su uso para cualquier programador java con nociones básicas de estadística. Pero el análisis de datos y generación de reportes en la configuración paramétrica se utilizó matplotlib, pandas y numpy.

Demasiado tarde en el proceso de desarrollo se contempló la necesidad de persistir el estado de ejecución de la herramienta. De haberlo contemplado antes, hubiera sido deseable el diseño e implementación de una base de datos (probablemente) relacional que permitiera una mejor gestión de almacenamiento.

Se resalta también el uso que el Lic. Weinstein dio a la herramienta, el cual permitió ejecutar ciclos completos de la metodología de optimización subyacente, siendo un componente central en su tesis.

5.2 - Trabajos a futuro.

A lo largo del proyecto, debieron descartarse varios objetivos por falta de tiempo, o experiencia. Estos son detallados a continuación como oportunidades de mejora para un siguiente proyecto.

Modularización de algoritmo de resolución: Es deseable poder modularizar el algoritmo de resolución, siendo posible seleccionarlo como un atributo más del problema o de la configuración paramétrica para permitir a la herramienta utilizar nuevos algoritmos de resolución.

Carga de componentes con archivos de configuración (json, csv): Definir un estándar de objetos para archivos JSON y/o cabezales de archivos csv que permitan la importación más rápida de los componentes (usos y configuraciones paramétricas) y permitir su exportabilidad.

Sustituir archivos de configuración por BD: Se quisiera dejar de utilizar archivos en formato JSON para persistir componentes, y utilizar una base de datos en su lugar (no se analizó si hay ventajas o desventajas en un modelo relacional).

Ampliar la generación de reportes gráficos: Es de suponer que el contacto de la herramienta con usuarios permite relevar la necesidad de nuevos reportes gráficos. La biblioteca JFreeChart es un framework sencillo e intuitivo en el que se hace voto de confianza para implementar a bajo costo estas necesidades.

Compacidad: Agregar la noción de compacidad al algoritmo de resolución como una restricción adicional. Esto es, intentar agrupar los mismos usos en píxeles adyacentes, evitando que estos estén muy alejados entre sí, reduciendo costos operativos del seguimiento de los planes generados.

Pausar y reanudar ejecución: Dado que el algoritmo de resolución puede tomar varias horas en completar su ejecución, es deseable implementar la capacidad de pausar y reanudar ejecuciones ya que resultaría beneficiosa para equipos no dedicados a tiempo completo a la tarea.

Reportes del resultado como conjunto de soluciones: Un análisis holístico de las soluciones encontradas durante la búsqueda podría generar una mejor comprensión del espacio de soluciones.

Compatibilidad con otros archivos SIG: Si bien el formato shape está ampliamente extendido en la comunidad SIG, sería prudente compatibilizar la herramienta con algunos formatos alternativos como GeoJSON y KML.

Duplicación de grupos de usos: Para facilitar al usuario final la creación de nuevos escenarios sería conveniente incorporar a la herramienta la capacidad de trabajar con grupos de usos y duplicarlos como un conjunto consistente.

Referencias

- A. Araujo, A. Parrilla, S. Mancebo, V. Moncalvo, "Optimización de usos del suelo en el contexto de rotación de cultivos considerando aspectos productivos y medioambientales". Proyecto de Grado de Ingeniería de Producción de la Universidad de la República, Facultad de Ingeniería 2020.
- C. Blum, A. Roli, "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison". *ACM Comput. Surv.*, 35. 268-308, 2001.
- A. Barletta. "Modelos de optimización y multiatributo para la asignación de usos del suelo en la cuenca de la Laguna de Rocha". Proyecto Final de Licenciatura en Ciencias Biológicas de la Universidad de la República, Facultad de Ciencias 2017.
- C. Cabrera. "Optimización de usos del suelo para prevenir floraciones nocivas de fitoplancton en la Laguna de Rocha, Uruguay". Tesis de Maestría en Geociencias de la Universidad de la República, Facultad de Ciencias 2015.
- A. Castagna. "Modelación del uso de suelo en una cuenca lechera para minimizar la exportación de nutrientes hacia aguas superficiales". Tesis de Maestría en Ciencias Agrarias de la Universidad de la República, Facultad de Agronomía 2019.
- H. Clarke. "Combinatorial aspects of cropping pattern selection in agriculture". *European Journal of Operational Research*, 40(1), 70-77, 1989.
- M. Debeljak. "A Field-Scale Decision Support System for Assessment and Management of Soil Functions". *Frontiers in Environmental Science*, 7 (115), 2019.
- N. Detlefsen, A. Leck. "Modelling optimal crop sequences using network flows, *Agricultural Systems*", 94 (2), 566-572, 2007.
- T. Feo, M. Resende. "Greedy Randomized Adaptive Search Procedures". *Journal of Global Optimization*. 6 (2), 109–133, 1995.
- M. Gendreau, A. Hertz, G. Laporte. "A Tabu Search for the Vehicle Routing Problem". *Management Science*, 40(10), 1276-1290, 1994.
- S. Haakonsson, L. Rodríguez-Gallego, A. Somma, S. Bonilla, "Temperature and precipitation shape the distribution of harmful cyanobacteria in subtropical lotic and lentic ecosystems". *The Science of the total environment*, 609, 1132–1139, 2017.
- W. Klein, A. Stegeman. "Crop succession requirements in agricultural production planning", *European Journal of Operational Research*, 166 (2), 406-429, 2005.
- J. Li, D. Rodriguez, D. Zhang, K. Ma. "Crop rotation model for contract farming with constraints on similar profits", *Computers and Electronics in Agriculture*, 119, 12-18, 2015.
- F. Matonte. "Modelado del problema de Planificación de Rotación de Cultivos utilizando Programación Entera Mixta". Proyecto Final de Licenciatura en Computación de la Universidad de la República, Facultad de Ingeniería, 2019.
- M. Memmah, F. Lescourret, X. Yao, C. Lavigne, "Metaheuristics for agricultural land use optimization". *Agronomy for Sustainable Development* 35, 975–998, 2015.
- F. Quintans, "Informe Evolución de la Calidad de Agua en la cuenca del río Santa Lucía 5 años de información (2015 - 2019)", Ministerio de Vivienda y Ordenamiento Territorial, 2020.

- B. Recio. "A decision support system for farm planning using AgriSupport II". *Decision Support Systems* 36(2),189-203, 2003.
- A. Ridier, K. Chaib, C. Roussy. "A Dynamic Stochastic Programming model of crop rotation choice to test the adoption of long rotation under price and production risks". *European Journal of Operational Research*, 252(1), 270-279, 2016.
- L. Rodrigues, P. Michelon, M. Nereu, R. Silva. "Crop rotation scheduling with adjacency constraints", *Annals of Operations Research*. 190 (1), 165-180, 2011.
- L. Rodríguez-Gallego, A. Barletta, C. Cabrera, C. Kruk, M. Nin. "Establishing limits to agriculture and afforestation: A GIS based multi-objective approach to prevent algal blooms in a coastal lagoon". *Journal of Dynamics and Games*, 6(2), 159–178, 2019.
- L. Rodríguez-Gallego. "Eutrofización de las lagunas costeras de Uruguay : impacto y optimización de los usos del suelo". Tesis de Doctor en Ciencias Biológicas de la Universidad de la República, Facultad de Ciencias 2011.
- R. Rupnik, M. Kukar, P. Vračar, D. Košir, D. Pevec, Z. Bosnić. "AgroDSS: A decision support system for agriculture and farming", *Computers and Electronics in Agriculture*, 161, 260-271, 2019.
- Y. K. Sheng, S. Zhang. "Analysis of problems and trends of decision support systems development". In: 2009 International Conference on E-Business and Information System Security, 1216–1218, 2009.
- W. Steffen, A. Persson, L. Deutsch, J. Zalasiewicz, M. Williams, K. Richardson, C. Crumley, P. Crutzen, C. Folke, L. Gordon, M. Molina, V. Ramanathan, J. Rockström, M. Scheffer, H. Schellnhuber, U. Svedin. "The Anthropocene: From Global Change to Planetary Stewardship". *Ambio*, 40(7), 739–761, 2011.
- P. Taechatanasat y L. Armstrong. "Decision support system data for farmer decision making". *Proceedings of Asian Federation for Information Technology in Agriculture*, 472–486, 2014.
- F. Terribile. "A web-based spatial decision supporting system for land management and soil conservation". *Solid Earth*. 6, 903–928, 2015.
- Z. Zhai, J. Fernán, V. Beltran, N. Martínez. "Decision support systems for agriculture 4.0: Survey and challenges". *Computers and Electronics in Agriculture*, 170, 2020.
- J. Little, "Models and Managers: The Concept of A Decision Calculus," *Management Science*, Vol. 16, No. 8 (April, 1970), pp. B466-B48 5.
- B. Alenljung. "Envisioning a future decision support system for requirements engineering: a holistic and human-centered perspective". Ph.D. Dissertation Dept. Comput. Info. Sci., Linköping Univ., Linköping, Sweden (2008)

Anexo

Documentos adicionales

Se crean dos documento para ayudar al usuario:

- Guía de Pantallas¹⁵: que detalla el comportamiento de la herramienta, mediante un inventario de cada elemento de la interfaz gráfica, su motivo, y utilidad.
- Guía de Uso¹⁶: que detalla cómo instalar y utilizar la herramienta, explicando un caso de uso típico y los resultados de este. Además se crean 5 videos¹⁷ que muestran este caso de uso típico, y cómo cargar nuevos usos, regiones, configuraciones paramétricas, y problemas, también explicando los resultados de la ejecución.

¹⁵  Guía de Pantallas.pdf

¹⁶  Guía de uso.pdf

¹⁷ <https://drive.google.com/drive/folders/1ty5AANz0M8lZW8nFvnd6a9hTwINM1dWh?usp=sharing>