

UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Matemática

Herramientas de programación matemática para mejorar la gestión pública y privada

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área de Matemática

Nazareno Angel Faillace Mullen

Director de tesis: Dr. Guillermo Durán

Consejero de Estudios: Dr. Pablo Groisman

Lugar de Trabajo: Instituto de Cálculo, Facultad de Ciencias Exactas y Naturales

Buenos Aires, 2025

Herramientas de programación matemática para mejorar la gestión pública y privada

Resumen: Los problemas de optimización combinatoria basados en instancias reales suelen presentar grandes desafíos por la numerosa cantidad de restricciones presentes. En esta tesis desarrollamos modelos matemáticos para abordar dos problemas, motivados por casos de estudio reales: logística en recolección de residuos en la gestión pública y planificación de producción en el ámbito privado.

La gestión eficiente de residuos urbanos tiene un importante impacto ambiental y económico. Utilizando como caso de estudio a los partidos bonaerenses de Berazategui y de Quilmes, desarrollamos herramientas para optimizar la recolección de residuos. Los objetivos principales son reducir el impacto ambiental, mejorar la calidad del servicio y de las condiciones laborales mediante la distribución equitativa de trabajo entre los recolectores y el diseño de recorridos eficientes. Enfocamos el problema de encontrar rutas óptimas como el *Mixed General Routing Problem with Turn Penalties*, el cual es un problema NP-Hard. Para hallar soluciones eficientes en tiempo razonable, desarrollamos un método basado en el *Asymmetrical Travelling Salesman Problem* y lo resolvemos con Programación Lineal Entera Mixta. Con respecto a la zonificación, proponemos dos enfoques de solución con el objetivo de lograr una mejor distribución en el trabajo: por un lado, una heurística *greedy* que construye una nueva zonificación y, por el otro, algoritmos basados en *Steepest-Ascent Hill Climbing*, *Simulated Annealing* y *Variable Neighbourhood Search* que modifican la zonificación existente. En Berazategui, logramos una reducción del $22,6\,\%$ en la distancia promedio recorrida y una disminución del $69,9\,\%$ en la disparidad de carga laboral, mientras que en Quilmes la mejora ha sido del 10,93 % y de $47\,\%$, respectivamente.

Con respecto a la planificación de producción, analizamos el *Unrelated Parallel Machine Problem* tomando como caso de estudio una fábrica de cerramientos de botellas. Se trata de un problema multiobjetivo: minimizar la demora en la finalización de los trabajos, la sobreproducción y el tiempo de inactividad de las máquinas. Debido a la complejidad del problema, lo dividimos en etapas y empleamos Programación Lineal Entera Mixta para resolverlas. En la primera, una minimización lexicográfica asigna los moldes a las máquinas; en la segunda, corregimos los tiempos de parada de la máquina para cambios de molde; y en la tercera, diferenciamos los productos para cada molde. Presentamos resultados para instancias de hasta 100 trabajos y una comparación entre el resultado de nuestro modelo y la planificación manual realizada por el personal de la fábrica para una instancia real, mostrando una disminución del $100\,\%$ y del $84{,}5\,\%$ en la demora en la entrega y en la sobreproducción, respectivamente, manteniendo nulo tiempo de inactividad de las máquinas.

Palabras clave: Programación Lineal Entera Mixta, Mixed General Routing Problem, Asymmetric Travelling Salesman Problem, Variable Neighbourhood Search, recolección de residuos municipales, heurísticas, ruteo de vehículos, Unrelated Parallel Machine Problem, planificación de horarios, optimizacion multiobjetivo.

Mathematical programming approaches to improve public and private sector management

Abstract: Combinatorial optimization problems based on real-world instances often present significant challenges due to the numerous constraints involved. In this thesis, we develop mathematical models to address two problems, motivated by real-world case studies: waste collection logistics in public management and production planning in the private sector.

Efficient urban waste management has a significant environmental and economic impact. Using the municipalities of Berazategui and Quilmes as case studies, we developed tools to optimize waste collection. The main objectives are to reduce environmental impact, improve service quality, and enhance working conditions by achieving a fairer distribution of work among collection crews and designing efficient routes. We formulate the routing problem as a *Mixed General Routing Problem with Turn Penalties* , which is NP-Hard. To obtain efficient solutions within reasonable computational times, we propose a method based on the *Asymmetric Travelling Salesman Problem* , solved through Mixed-Integer Linear Programming (MILP). For the zoning problem, we propose two approaches aimed at improving workload distribution: a greedy heuristic that generates new zones, and metaheuristic algorithms— *Steepest-Ascent Hill Climbing*, *Simulated Annealing*, and *Variable Neighbourhood Search* —that modify existing ones. In Berazategui, we achieved a 22.6 % reduction in the average distance traveled and a 69,9 % decrease in workload imbalance, while in Quilmes, the improvements were 10,93 % and over 47 %, respectively.

Regarding production planning, we analyze the *Unrelated Parallel Machine Problem*, using a bottle closures manufacturing facility as a case study. This is a multi-objective problem: minimizing job completion delay, overproduction, and machine idle time. Due to its complexity, we divide the problem into stages and solve them using MILP. The first stage applies a lexicographic minimization to assign molds to machines; the second adjusts downtime for mold changes; and the third differentiates products by mold. We present results for instances with up to 100 jobs and compare our model's output to the manual scheduling performed at the factory for a real instance, demonstrating significant improvements across the defined objectives: a 100 % and a 84,5 % reduction in job completion delay and overproduction, respectively, while keeping a null machine idle time.

Keywords: Mixed-Integer Linear Programming, Mixed General Routing Problem, Asymmetric Travelling Salesman Problem, Variable Neighbourhood Search, municipal waste collection, heuristics, vehicle routing, Unrelated Parallel Machine Problem, scheduling, multiobjetive optimization.

Agradecimientos

Son muchas las personas que me han acompañado en este proceso, por lo cual espero ser justo y no olvidarme de nadie. En primer lugar quiero agradecerle a mi mamá y a mi hermano por haber sido quienes estuvieron conmigo durante toda esta etapa. Además de muchas alegrías, hemos trascurrido tiempos difíciles, así que sin su apoyo, sin su motivación, sin sus consejos y sin su ayuda, todo este trabajo no hubiera sido posible o hubiera sido muchísimo más difícil de realizar. Mi madre siempre me ha ayudado a orientarme en lo momentos de abrumamiento y en mi hermano encuentro un ejemplo de tenacidad y perseverancia. También le agradezco a mi papá, quien si bien hace ya diez años que no me acompaña de manera física, también lo sentí presente en los momentos en los que más lo necesitaba. La compañía y el cariño de mis tíos y de mis primos también es algo que valoro mucho. De igual manera, mis amigues, Octi, Vic, Fede, Marian y Lucho también fueron un apoyo muy importante, distrayéndonos con muchas risas y compartiendo experiencias.

Le agradezco a Willy por su guía a lo largo de todos estos años, por su confianza en mi trabajo y por la posibilidad de participar en proyectos de optimización aplicada como los que me cautivaron cuando cursé Investigación Operativa, allá por 2014. A pesar de su responsabilidad enorme con esta Facultad, siempre pude contar con él cuando necesité ayuda o consejo. Admiro mucho su cualidad resolutiva de problemas. Siempre que nos encontramos con un obstáculo, me ayudó a resolverlo eficientemente y con calma.

Quiero agradecerle también al Instituto de Cálculo por su calidez desde el primer día. A les directores: a Dani, a Guille y a Coti, por tomar en cuenta mis opiniones y mis preferencias para la docencia. A les becaries por ser un muy lindo grupo y por toda la buena onda. A todes mis compañeres de docencia, en especial a Santi Laplagne, a Mariela Sued y a Iván Monardo. A las secretarias: Lida, Sole, Deby y Clara, quienes me tuvieron infinita paciencia a lo largo de todos estos años. A quienes colaboraron conmigo en la investigación. También agradezco al jurado de esta tesis, Carola, Nino y Cristián por la dedicación al evaluar mi trabajo.

Sería injusto no agradecer a CONICET por financiar la beca que hizo posible el desarrollo de este trabajo y a la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires por brindar el lugar de trabajo y la prestigiosa formación que la caracteriza. Sin la educación pública y la inversión en Ciencia y Tecnología, no hubiera logrado recibirme de licenciado, mucho menos de doctor. Por esta razón considero fundamental defender estas instituciones, repudiar su vaciamiento y reclamar por mayor prioridad en las políticas nacionales.

¡Muchísimas gracias a quienes han transitado conmigo este camino!

A la inteligencia de mi madre.

A la creatividad de mi padre.

A la perseverancia de mi hermano.

Índice

Introducción I
Estructura de la tesis II

I Recolección de residuos 1

1. Introducción 2

2. Descripción de los casos de estudio 3
2.1. Municipalidad de Berazategui 3
2.2. Municipalidad de Quilmes 4
2.3. El Achique y el *Mixed General Routing Problem* 6

3. Estado del Arte 8

4. Definiciones 11
4.1. Conceptos preliminares 11
4.2. El plano como un grafo 13
4.3. Grafo de zonificación 16
4.4. Digrafo de ruteo 18

5. Creación de una nueva zonificación 20
5.1. Fase I 24
5.2. Fase II 33

6. Modificación de una zonificación vigente 37
6.1. Método basado en Hill Climbing 46
6.2. Método basado en Simulated Annealing 48
6.3. Método basado en Variable Neighbourhood Search 50

7. Optimización de las rutas 53
7.1. Enfoque $F_4 \neq \emptyset$ 56
7.2. Enfoque $F_4 = \emptyset$ 63

8. Asignación a zonas 77

9. Aplicación a caso de estudio: Berazategui 80
9.1. Zonificación 81
9.2. Ruteo 83
9.3. Asignación a zonas 85

10. Aplicación a caso de estudio: Quilmes 88
10.1. Medida de carga laboral 88
10.2. Resultados computacionales 89

11. Discusión, conclusiones y trabajo futuro 101

II Planificación de la producción **105**

1. Introducción **106**

2. Descripción del Problema **109**

3. Estado del Arte **111**

4. Enfoque de solución **113**
4.1. Etapa 1 - Asignación de moldes 113
4.2. Etapa 2 - Ajuste de los tiempos de parada 121
4.3. Etapa 3 - Determinación del color para cada molde 126

5. Experimentación computacional **132**
5.1. Instancias 132
5.2. Resultados y discusión 133

6. Conclusiones **140**

Introducción

La Investigación Operativa es un campo de estudio interdisciplinario que busca desarrollar técnicas y herramientas para asistir en la toma de decisiones. Si bien sus raíces datan de hace varios siglos [53], su desarrollo cobró especial relevancia durante la Segunda Guerra Mundial por la necesidad de un enfoque científico en la administración eficiente de recursos escasos [67]. Los resultados fueron tan satisfactorios que desde aquel momento su aplicación se ha extendido a diversos ámbitos como, entre otros, la salud [106, 14, 70, 18, 89], el deporte [81, 42, 101, 40, 84], la educación [22, 78, 113, 39], la logística [21, 59, 52, 6, 33, 45], la recolección de residuos [9, 82, 90, 5, 13, 129] y la producción industrial [24, 49, 65, 55, 121]. En los dos últimos se encuadra el trabajo llevado a cabo en esta tesis. Por un lado, abordamos la optimización de la recolección de residuos sólidos urbanos (RSU) en dos municipios del Gran Buenos Aires, con el objetivo de desarrollar herramientas que tuvieran en cuenta las restricciones propias de cada uno, pero que al mismo tiempo fueran sencillas de generalizar para ser aplicadas en el resto de la provincia e incluso en el resto del país. Por otro lado, proponemos un enfoque de solución para la planificación de producción en una fábrica de cerramientos de botellas, que también puede extenderse a industrias de otros rubros con procesos de elaboración con características similares.

En Argentina la población urbana alcanza el 91 %, siendo superada con 99,9 % en el Gran Buenos Aires y 93 % en el interior de la Provincia de Buenos Aires [107]. De acuerdo con ello, se estima que dicha provincia concentra aproximadamente el 40 % de la producción de residuos del país, generando alrededor de veinte mil toneladas por día, cuyo 36 % es producido en el área conurbana [93]. Por lo tanto, como en otras ciudades contemporáneas, la gestión de RSU se considera un desafío debido al rápido crecimiento en la cantidad de deshechos causado por la urbanización y el crecimiento demográfico [12]. En general, de las actividades asociadas a los RSU, las de recolección y transporte representan el mayor porcentaje del costo total [119]. Las ciudades de ingresos medios destinan 50–60 % de su presupuesto de gestión de residuos a la recolección, mientras que las ciudades de bajos ingresos pueden llegar a gastar el 80 % [117, 68]. Por lo tanto, mejoras en estos aspectos tienen consecuencias económicas inmediatas para los municipios. Por otro lado, la producción y gestión de RSU emiten contaminantes que contribuyen al cambio climático e impactan sobre la salud pública [125]. En lo que respecta a la cuestión ambiental, la traza de rutas eficientes para los camiones recolectores repercute en la disminución de consumo de combustible y de emisión de gases de efecto invernadero. En cuanto a la salud pública, una gestión ineficiente de RSU conlleva a la acumulación de deshechos, lo cual atrae vectores de enfermedades y obstruye los desagües [72]. Otra consecuencia es que los trabajadores de la recolección de RSU tienden a tener una alta tasa de riesgo de accidentes laborales y de infecciones, muy por encima de la media de la población [61]. Por todas estas razones, la organización y la optimización de los sistemas de recolección de RSU tienen efectos muy positivos en la salud pública, la calidad de vida de los trabajadores, el ambiente y la economía de una ciudad.

Por su parte, los beneficios de una programación de producción eficiente abarcan aspectos económicos y organizativos. En primer lugar, se reducen los tiempos muertos y las ineficiencias operativas, lo que se traduce en la disminución de costos, especialmente en rubros donde los recursos son caros o limitados [103]. Además, las empresas que adoptan estrategias para planificar eficientemente su producción mejoran el cumplimiento de los plazos de entrega y,

por ende, aumenta la satisfacción de sus clientes [114]. Otra ventaja es el incremento de la calidad del proceso productivo: al establecer prioridades y secuencias óptimas, se pueden reducir errores asociados a cuellos de botella o interrupciones en la línea de producción. Finalmente, contar con una herramienta eficiente para la programación de la producción mejora la adaptabilidad a condiciones cambiantes, como por ejemplo, el ingreso de una nueva orden de trabajo o la inhabilitación de una máquina por una falla técnica.

Aunque los problemas de recolección de residuos y de planificación de la producción pertenecen a contextos aplicados muy diferentes, ambos pueden abordarse mediante un conjunto común de herramientas analíticas provenientes de la teoría de grafos y de la programación matemática. En particular, muchos de los subproblemas que surgen en ambos contextos, como la secuenciación de tareas o la búsqueda de caminos mínimos, pueden resolverse con algoritmos en grafos. A su vez, la formulación de estos problemas en términos de Programación Lineal Entera Mixta (MILP, por sus siglas en inglés) permiten considerar las restricciones operativas y los objetivos múltiples que los caracterizan. Por lo tanto, más allá de sus diferencias, ambos casos de estudio se inscriben en una misma línea de investigación orientada al diseño de herramientas de optimización combinatoria aplicadas a problemas reales de decisión.

Como mostraremos más adelante en las revisiones bibliográficas, para ambos problemas existe una vasta literatura que estudia diversas formulaciones teóricas y aplicaciones a casos reales. Sin embargo, al resolver situaciones concretas como las que tratamos aquí, muchas veces hace falta incluir restricciones particulares que hacen necesario el desarrollo de nuevas herramientas. Por lo tanto, el objetivo general de esta tesis es elaborar tales herramientas que permitan resolver problemas específicos en nuestro país y promover la aplicación de la Investigación Operativa tanto en la gestión pública como en la gestión privada.

Estructura de la tesis

El resto de la tesis se divide en dos capítulos. El Capítulo I está dedicado a los trabajos realizados en el área de la optimización de recolección de RSU. En ambos casos, el objetivo principal es disminuir la disparidad en la carga laboral entre los trabajadores. Como beneficio adicional, se obtienen recorridos eficientes para los camiones.

La introducción (Sección 1) caracteriza en líneas generales la gestión de RSU en la Provincia de Buenos Aires. En la Sección 2 se describen los casos de estudio de Berazategui y de Quilmes, para contextualizar las herramientas desarrolladas. En la Sección 3 se lleva a cabo una revisión de la literatura y se explica cuáles son los aportes de esta tesis. Seguidamente, en la Sección 4 se enumeran definiciones de conceptos preliminares y de terminología que se utilizarán a lo largo del capítulo. Las siguientes dos secciones están dedicadas a la zonificación. La Sección 5 describe el diseño de las zonas de recolección mediante una heurística que busca equilibrar la carga laboral y generar zonas de forma cuadrada. Por su parte, la Sección 6 detalla los algoritmos de búsqueda local para la modificación de la zonificación vigente. La Sección 7 está abocada a dos enfoques para hallar rutas eficientes, que están basados en la concatenación de caminos mínimos mediante la solución de un modelo MILP. La asignación de trabajadores a zonas es abordada en la Sección 8. Posteriormente, en la Sección 9 se comparten los resultados obtenidos de aplicar la heurística de diseño de zonificación, la optimización de rutas y la asignación de choferes (con los métodos de las Secciones 5, 7.1 y 8) al

caso de Berazategui, mientras que en la Sección 10 se muestran los resultados de modificar la zonificación vigente en Quilmes (con los procedimientos de las Secciones 6 y 7.2). El capítulo finaliza con la Sección 11, donde se comparten las conclusiones obtenidas a partir de los dos proyectos y se identifican posibles líneas de trabajo a futuro.

En el Capítulo II se halla el trabajo que realizamos en planificación de la producción utilizando a TAPI S.A., una empresa de cerramientos de botellas, como caso de estudio (Sección 1). El objetivo fue elaborar una herramienta que permitiera programar la producción en un horizonte de a lo sumo un mes satisfaciendo las restricciones propias del proceso de manufactura, las cuales son detalladas en la Sección 2. Se trata de un problema multiobjetivo donde se busca minimizar la demora en la terminación de los trabajos, la sobreproducción y el tiempo de inactividad de las máquinas. La Sección 3 consiste en la revisión bibliográfica y los aportes de esta tesis. En la Sección 4 se explica el enfoque de solución que se basa en dividir el problema en tres etapas: en la primera se emplea un modelo MILP y minimización lexicográfica para asignar moldes a las máquinas; en la segunda, se utiliza un modelo MILP para ajustar los tiempos de parada de las máquinas debido al cambio de moldes y, en la tercera etapa, se desarrolla un MILP para determinar la asignación de colores que diferencia a los productos elaborados con el mismo molde. La Sección 5 muestra los resultados de aplicar nuestro enfoque de solución a instancias de hasta 100 trabajos y el impacto de distintas características del problema en el desempeño del modelo. Finalmente, en la Sección 6 se comparten las conclusiones obtenidas a partir de este trabajo.

Hasta el momento, esta tesis ha dado lugar a las publicaciones que tuvieron a Berazategui [10] y a TAPI S.A. [41] como casos de estudio. El trabajo con la herramienta motivada por el caso de Quilmes y los aprendizajes obtenidos a partir de la implementación de la solución propuesta para Berazategui ha sido recientemente aceptado en el *XIII Latin American Algorithms, Graphs, and Optimization Symposium* (LAGOS 2025) y será publicado en los *proceedings* del simposio [32].

Capítulo I

Recolección de residuos

1. Introducción

Este capítulo está dedicado a las herramientas que desarrollamos para la optimización de la recolección de residuos con los objetivos de equilibrar la carga laboral de los recolectores y minimizar la longitud de las rutas de los vehículos. En el periodo 2011-2022, la población de la Provincia de Buenos Aires ha registrado un crecimiento del 12,4 %. Más precisamente, en los municipios de la provincia que conforman el Área Metropolitana de Buenos Aires (AMBA), el crecimiento poblacional fue del 9,8 % [108]. En general, la Ciudad Autónoma de Buenos Aires y los municipios del AMBA descargan los residuos en los centros de procesamiento de la Coordinación Ecológica Área Metropolitana Sociedad del Estado (CEAMSE). Según los datos proporcionados por este organismo, la cantidad de residuos generada por los habitantes de AMBA se mantuvo en ascenso constante desde 2015 hasta 2018, donde alcanzó el récord anual de 7,71 millones de toneladas [94]. Sin embargo, a partir de 2019 este monto disminuyó hasta estabilizarse alrededor de los 6,37 millones de toneladas anuales en 2024 [28].

Actualmente la gestión de los RSU en los partidos del AMBA recae en los gobiernos locales, representando un problema por falta de medios técnicos y financieros. La recolección de RSU requiere la articulación de recursos presupuestarios, humanos y logísticos que muchas veces presentan desafíos debido a marcos normativos poco claros y falta de coordinación entre jurisdicciones [94]. Para 2040 la generación de residuos en el país podría llegar a los 25 millones de toneladas anuales, representando un aumento del 25 % con respecto a 2020 [94]. Por esta razón, las herramientas que presentamos en este capítulo buscan asistir en la mejora de la eficiencia de la recolección de residuos y proveer soluciones a la gestión de RSU. Como ya hemos mencionado, tomaremos como casos de estudio a los municipios de Berazategui y de Quilmes, ambos ubicados en el zona sur del AMBA.

El resto del capítulo está estructurado de la siguiente manera. En la Sección 2 describimos las características de ambos casos de estudio. En la Sección 3 realizamos una revisión de la literatura en problemas de recolección de residuos y aplicaciones, mientras que en la Sección 4 definimos los conceptos básicos de este capítulo. La Sección 5 y la Sección 6 presentan los enfoques de creación y modificación de zonificaciones, respectivamente. Por su parte, la Sección 7 abarca las estrategias para hallar rutas eficientes y la Sección 8 el procedimiento para asignar grupos de recolectores a zonas. La Sección 9 y la Sección 10 reportan los resultados obtenidos en ambos casos de estudio. Finalmente, en la Sección 11 compartimos nuestras conclusiones y definimos algunos lineamientos para trabajo a futuro.

2. Descripción de los casos de estudio

2.1. Municipalidad de Berazategui

La municipalidad de Berazategui está ubicada en la zona sur del Gran Buenos Aires, a 26 kilómetros de la ciudad de Buenos Aires (Figura 1). Abarca una superficie de 217 km^2 y cuenta con una población de alrededor de 358.000 habitantes [75].



Figura 1: Ubicación del Partido de Berazategui

Los residuos domiciliarios del municipio se recogen en cestos particulares en la vereda, con una flota de 24 camiones que recorren un total de aproximadamente 15000 cuadras diarias, segmentadas en 42 zonas. Cada equipo de recolección está compuesto por un chofer y dos o tres recolectores que agrupan y suben las bolsas de residuos al camión. La jornada se divide en dos turnos: el turno mañana (desde las 4 hasta las 10 hs.) y el turno tarde (desde las 10 hasta las 16 hs.). En general, cada chofer trabaja durante ambos turnos, por lo que tiene asignada dos zonas por día, a diferencia de los recolectores que atienden una zona por día. La asignación de zonas es fija para todos los trabajadores. El itinerario típico del chofer comienza en la Dirección de Higiene Urbana (DHU), el ente del municipio encargado de realizar las tareas asociadas a la recolección de RSU. Con el primer grupo de recolectores, recogen los residuos de la primera zona asignada al chofer. Luego, el equipo regresa a la DHU y se realiza el recambio de recolectores para servir a la zona del turno tarde. Una vez finalizada esta tarea, el chofer deja al equipo de recolectores en la DHU y se dirige a la planta de tratamiento de la CEAMSE de Almirante Brown, ubicada a 23 km de distancia del Municipio. Finalmente, su jornada laboral termina al regresar el camión a la DHU. El itinerario es interrumpido si la capacidad de carga del camión es alcanzada antes de finalizar la recolección. En ese caso, el camión debe dirigirse a CEAMSE a descargar y luego reanuda el recorrido, provocando un aumento significativo en la duración de la jornada laboral, el combustible requerido y el

desgaste del vehículo.

La recolección se lleva a cabo dirariamente de lunes a sábado. Por esta razón, los lunes la cantidad de residuos suele ser significativamente mayor y todos los camiones deben realizar dos viajes a la planta de CEAMSE. Como esta situación es inevitable debido al tamaño de la flota de vehículos, el análisis de este trabajo se enfocó en la situación de los demás días de la semana. Aunque los resultados obtenidos no evitarían dos viajes a la planta de procesamiento los lunes, ciertas ventajas, como las rutas más eficientes, aún impactarían en la reducción de la longitud de la jornada laboral de los trabajadores.

Las autoridades de la DHU nos convocaron para mejorar el servicio de RSU. Antes de comenzar el proyecto, la zonificación y la asignación de choferes a zonas era realizada manualmente y no había rutas predeterminadas, sino que estaban libradas al criterio de cada conductor. Las ineficiencias que este tipo de organización conlleva dieron lugar a una serie de dificultades cuya solución requería de un enfoque más metódico. En primer lugar, el trabajo estaba mal distribuido entre choferes y entre recolectores, causa de reclamos por parte de los trabajadores cuyas jornadas laborales duraban varias horas más que las de sus compañeros y por parte de los vecinos, pues en ciertas ocasiones los equipos de recolección no lograban terminar de recorrer las zonas durante su turno. Por otra parte, algunas zonas estaban formadas por manzanas que se encontraban en distintas partes de la ciudad, lo que causaba demoras en el traslado y mayor consumo de combustible. Finalmente, la asignación ineficiente de choferes ocasionaba que algunos debieran realizar dos viajes a CEAMSE ciertos días además de los lunes. Por lo tanto, el objetivo radicaba en diseñar una nueva zonificación que balanceara la cantidad de kilómetros recorridos por los recolectores, obtener rutas eficientes y distribuir a los choferes de manera tal que la distribución de trabajo sea lo más justa posible.

En el desarrollo del enfoque de solución tuvimos en cuenta ciertas restricciones comunicadas por parte de la DHU. En primer lugar, la cantidad de zonas debía mantenerse en 42. Algunos de barrios del municipio, como El Pato y Juan María Gutiérrez, la zona céntrica y los barrios cerrados tenían esquemas de recolección especiales fuera del alcance de este proyecto. Debido a esto, la Municipalidad solicitó que dichas zonas y los camiones que tienen asignados no fueran modificados. Sin embargo, sí nos solicitaron el diseño de rutas para el barrio Juan María Guitérrez y la zona céntrica. Además, debíamos mantener la asignación de dos zonas por chofer, salvo para los casos excepcionales de los barrios antes mencionados. Adicionalmente, se valoraban dos características deseables de la solución. La primera, que ningún camión debiera descargar más de una vez por día en la planta de CEAMSE de martes a sábado. La segunda, que las zonas fueran conexas, es decir, que estuvieran formadas por manzanas adyacentes y no por conjuntos de manzanas en distintas partes del municipio.

2.2. Municipalidad de Quilmes

La municipalidad de Quilmes también está ubicada en la zona sur del Gran Buenos Aires, al norte de la municipalidad de Berazategui (Figura 2). Tiene una mayor densidad poblacional, con alrededor de 633.000 habitantes [75] distribuidos en 125 km^2.

Cuenta con una flota de 39 camiones de 8 toneladas de capacidad, y está dividida en 84 zonas de recolección. Como en el caso de Berazategui, los vecinos depositan sus residuos en cestos particulares ubicados en la vereda. Cada zona es recorrida durante alguno de los tres

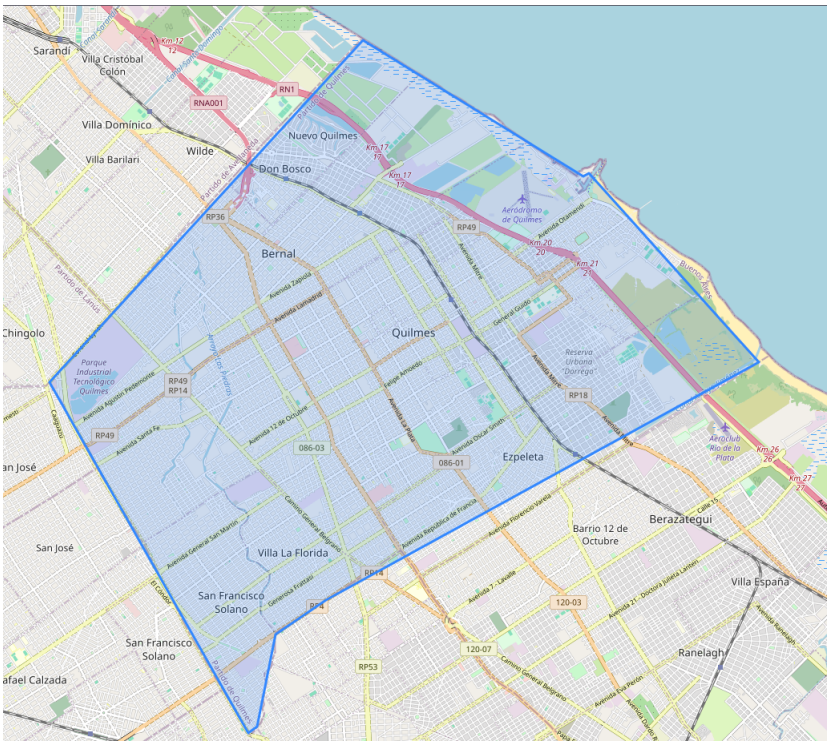


Figura 2: Ubicación del Partido de Quilmes

turnos: el turno mañana (de 4 a 12 hs), el turno tarde (de 12 a 18 hs) o el turno noche (de 18 a 24 hs). El turno asignado a cada zona está fijo, pues se espera que los vecinos saquen la basura en el intervalo de tiempo correspondiente para evitar la acumulación de residuos fuera del horario de recolección. A diferencia de lo que ocurría en Berazategui, tanto los choferes como los recolectores a pie recorren una sola zona por día. La asignación del equipo de recolección a cada zona también está fija y su itinerario comienza en el predio de la Secretaría de Ambiente y Gestión Integral de Residuos (GIRSU), que es la dependencia municipal encargada de la limpieza de la ciudad. Luego de recorrer la zona asignada, descargan el camión en una de las dos plantas de tratatamiento de CEAMSE más cercanas: la ubicada en Almirante Brown o la que se encuentra al sur de la Ciudad de Buenos Aires. Finalmente, el equipo regresa al predio de GIRSU para finalizar su jornada.

Como en el caso de Berazategui, la recolección de residuos ocurre de lunes a sábado. Sin embargo, como la capacidad de los camiones excede la demanda de cada zona, no descargan más de una vez por turno, incluso los lunes.

Las autoridades de GIRSU nos solicitaron asistencia para mejorar el servicio de recolección del municipio, que resultaba ineficiente por la organización manual de la zonificación y la falta de rutas establecidas para los choferes. Como ocurría en Berazategui, la distribución de la carga laboral desbalanceada era causa de reclamos por parte de los trabajadores. Ciertos días, algunas zonas no podían ser recorridas en su totalidad. En estos casos, dependiendo de la disponibilidad de camiones, se enviaba un *refuerzo*, es decir, un equipo de recolectores de otro turno que recorriera la sección de la zona afectada. Si por alguna razón desde GIRSU no

podían enviar un refuerzo, los vecinos presentaban quejas por el servicio ineficiente. Además de resolver o paliar estos problemas, un enfoque metódico como el que se puede obtener mediante el desarrollo de herramientas matemático-computacionales podría definir rutas más eficientes y así disminuir los costos del servicio de recolección.

Entonces, el objetivo principal del trabajo era diseñar una zonificación que balanceara la duración de la jornada laboral de los trabajadores, sujeto a una serie de restricciones. La primera de ellas era respetar la cantidad de zonas de recolección, el tamaño y la capacidad de la flota. Ciertos barrios, como el centro de la ciudad y el barrio Lynch y los barrios populares como Villa Azul y Villa Itatí, quedaron fuera del alcance de este proyecto debido a su esquema de recolección diferenciada. Si bien no era una restricción dura, una característica deseable de la solución era que la jornada del equipo de recolección no superara las cuatro horas. Por otro lado, si bien había zonas no conexas, no era un requerimiento que todas lo fueran. Como objetivo secundario, buscamos definir las rutas de los camiones para que sus recorridos fueran eficientes.

2.3. El Achique y el *Mixed General Routing Problem*

Ambos casos de estudio contaban con una particularidad que debimos tener en cuenta al momento de diseñar rutas eficientes: la técnica de recolección conocida como *achique*. Este método consiste en que uno o dos recolectores acompañen al camión, recogiendo los residuos de la calle que transita. Mientras tanto, el resto de los recolectores se adelantan, juntan los residuos de las cuadras transversales y los acercan a la esquina por donde pasa el camión. Cuando el vehículo llega a dicha esquina, los recolectores que lo acompañan le cargan las bolsas. Esta estrategia evita que el camión deba pasar por el frente de todas las casas, implicando una disminución del tiempo de recolección. En la Figura 3, presentamos un ejemplo gráfico del achique: las cuadras *achicadas* son las naranjas mientras que las celestes son transitadas por el camión. Es importante notar que si una cuadra es demasiado larga, no puede ser achicada, ya que en general los recolectores no pueden cargar tantos residuos hasta la esquina. Mediante el diálogo con las autoridades correspondientes, consideramos que en el caso de Berazategui una cuadra es demasiado larga si supera los 130 metros mientras que en Quilmes el límite es de 140 metros.

Como detallaremos en la Sección 4, una manera de modelar el plano de la ciudad es con un grafo mixto $G = (V, A, E)$ donde el conjunto de vértices V representa el conjunto de las esquinas, es decir, la intersección de las calles; el arco dirigido $(u, v) \in A$ si y sólo si la cuadra que une la esquina u con la esquina v se puede recorrer en un solo sentido, de u a v; y la arista $\{u, v\} \in E$ si y sólo si la cuadra que une la esquina u con la esquina v se puede recorrer en ambos sentidos. Con la estrategia del achique, el camión debe recorrer todas las esquinas de la zona asignada y las cuadras con longitud mayor al umbral establecido en cada caso. Por lo tanto, debemos encontrar un recorrido de distancia mínima que pase obligatoriamente, al menos una vez, por un subconjunto $\hat{V} \subset V$ de vértices y por un subconjunto $(\hat{A} \cup \hat{E}) \subset (A \cup E)$ de arcos y aristas. Esta es justamente la definición del *Mixed General Routing Problem* (MGRP) [30].

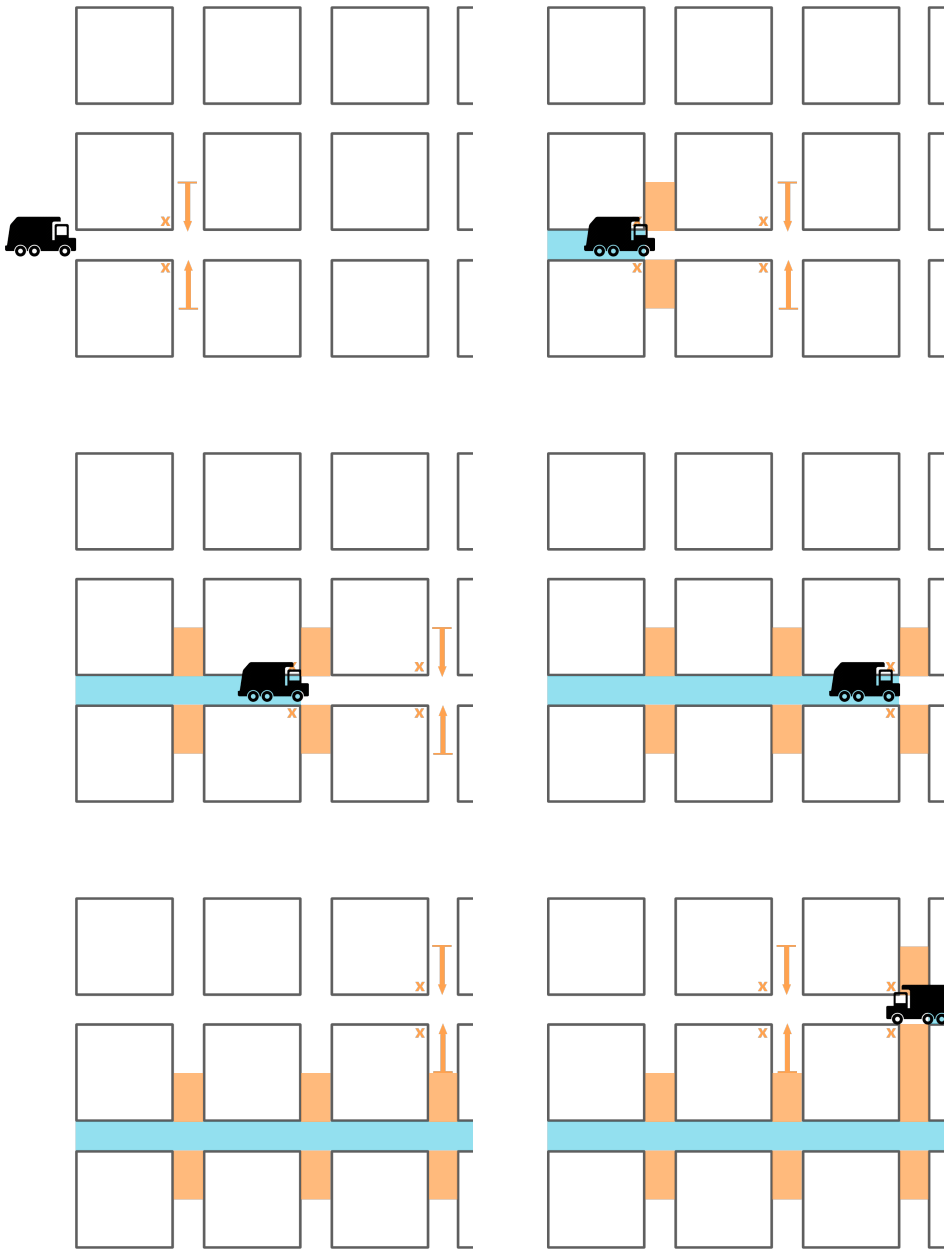


Figura 3: Ejemplo gráfico del achique. En celeste, las cuadras recolectadas directamente por el camión y en naranja, las achicadas. Las flechas indican cómo los recolectores que se adelantan acercan los residuos a la esquina por donde pasará el camión.

3. Estado del Arte

La optimización de la recolección de RSU generalmente aborda la división de la ciudad en zonas de recolección (zonificación, más conocida en la literatura como *districting*) y el diseño de rutas eficientes para los vehículos. En su definición más amplia, la zonificación consiste en agrupar unidades básicas (por ejemplo, clientes, calles o manzanas) en clusters geográficos más grandes, llamados zonas o distritos [80]. Generalmente, se busca que la zonificación cuente con tres características. La primera es el equilibrio: que las zonas sean lo más parecidas posible en cuanto a la carga laboral que demandan. La segunda es la compacidad: se espera que la forma geométrica de cada zona no sea demasiado distorsionada para simplificar la definición de sus límites. La última característica deseable es que las zonas sean contiguas, es decir, que estén formadas por un único bloque geográfico sin fragmentaciones ni dispersiones.

Por otra parte, cómo se modele la búsqueda de rutas eficientes depende del problema a resolver. Según Hess et. al [66] los problemas de ruteo en recolección de RSU pueden ser modelados, entre otros, como:

- *Network design and Location Routing Problems* (LRP): para los casos en los que se decide la ubicación de cestos o contenedores de residuos y se define la secuencia óptima que los visita a todos.
- *Arc Routing Problems* (ARP): cuando se debe recorrer todas las cuadras de áreas urbanas muy pobladas; en general también se tiene en cuenta la capacidad del vehículo recolector o de los centros de procesamiento de residuos, dando lugar al problema conocido como *Capacitated Arc Routing Problem* (CARP).
- *Vehicle Routing Problems* (VRP): el modelo más común para recolección de RSU y el utilizado cuando se deben visitar nodos dispersos en un área, por ejemplo en la recolección de residuos comerciales.
- *Mixed General Routing Problems* (MGRP): se modela sobre un grafo mixto en el que se deben visitar arcos y nodos; si bien suele ser el más apropiado para resolver problemas de recolección teniendo en cuenta la logística en una ciudad, es el menos encontrado en la literatura.

La zonificación y el ruteo han sido encarados desde distintos enfoques debido a su complejidad computacional [100, 116, 79]. Uno de los primeros reportes bibliográficos sobre recolección de residuos donde se exploran diferentes técnicas matemáticas para resolver esta clase de problemas fue realizada en 1974 por Beltrami y Bodin [9]. Desde entonces, dada la gran cantidad de potenciales aplicaciones y la versatilidad de éstas, numerosas investigaciones fueron realizadas y publicadas, algunas de las cuales se mencionan a continuación.

Male y Liebman [90] definen uno de los primeros algoritmos diseñados para generar simultáneamente la zonificación y el ruteo de vehículos para un sistema de recolección de residuos, bajo la suposición de que todas las calles son doble mano y cada cuadra tiene asociados un costo por ser transitada y cierta cantidad de residuos para recolectar. Algunos trabajos posteriores incluyen el de Hanafi et al. [63], donde la sectorización se lleva a cabo utilizando un algoritmo de búsqueda local. Teixeira et al. [118] desarrollan una heurística y una medida para el esfuerzo que requiere la recolección de cada zona y Kim et al. [82] modelan el problema de

ruteo con ventanas de tiempo y obtienen soluciones utilizando una extensión del algoritmo de inserción de Solomon [116]. Mourão et al. [98] implementan una heurística en dos fases para zonificación y ruteo y un método de inserción para resolver ambos problemas simultáneamente, mientras que Constantino et al. [27] afrontan la zonificación y el ruteo conjuntamente definiendo zonas a partir de rutas con la menor cantidad de sobreposición entre ellas. Gliesch et al. desarrollan métodos de zonificación basados en la construcción de soluciones iniciales y su posterior mejora mediante algoritmos iterativos con Búsqueda Tabú [57, 56]. Encontramos también trabajos donde el ruteo es modelado como ARP y se consideran restricciones en el marco de la logística de la ciudad. Por ejemplo, Cortinhal et al. [31] buscan minimizar la distancia de recorrido y equilibrar la carga laboral, teniendo en cuenta la capacidad de los vehículos y el tiempo. Jin et al. [77] proponen un enfoque basado en heurísticas y programación dinámica para una versión del ARP, donde además de la capacidad de los vehículos y las ventanas de tiempo para el servicio, consideran penalidades por no permitir el estacionamiento durante horarios de alta demanda en las cuadras a ser recolectadas.

Por otra parte, encontramos en la literatura diversas aplicaciones de técnicas de optimización matemática sobre problemas similares de numerosas ciudades del mundo. Algunos ejemplos son los trabajos de Eisenstein e Iyer [44] en Chicago, Estados Unidos, donde el peso y el tiempo requeridos para recolectar los residuos de una manzana son modelados como una variable aleatoria y el ruteo como un proceso de decisión de Markov; de Chang y Lu [23] en Kaohsiung, Taiwán y de Huang et al. [71] en Hamilton-Wentworth, Canadá, donde se utilizan técnicas de programación lineal entera que contemplan parámetros regidos por incertidumbre; de Mourão y Almeida [99] en Lisboa, Portugal, donde se presentan dos métodos de acotación inferior para el CARP que emplean programación lineal entera mixta; y de Arribas et al. [5] en Santiago de Chile, Chile, para el cual las zonas son diseñadas a través de la clusterización de los contenedores de residuos, los ruteos son diseñados con el uso de Sistemas de Información Geográfica (GIS) y un modelo de programación lineal entera determina la flota óptima.

También se ha aplicado a diferentes localidades argentinas: en la Ciudad de Buenos Aires, Bonomo et al. [13] proponen una reducción del problema de ruteo al Travelling Salesman Problem y se utiliza Concorde para obtener rutas eficientes que minimicen el trabajo mecánico del vehículo; en Morón, Provincia de Buenos Aires Braier et al. [15] modelan el problema como un caso generalizado del Open Rural Postman Problem y las rutas son diseñadas mediante un modelo de programación lineal entera; en San Miguel de Tucumán, Tucumán, Bianchettu et al. [11] proponen una heurística para generar una zonificación de manera iterativa e incremental, mientras que el problema del ruteo es modelado como el Chinese Poostman Problem [85]; en Trenque Lauquen, Provincia de Buenos Aires, Delle Donne et al. [34] desarrollan una heurística para la zonificación del barrido de hojas, se utiliza programación lineal entera y Concorde para hallar rutas eficientes para los barrenderos y camiones recolectores, respectivamente.

El Mixed General Routing Problem (MGRP) fue formulado por Corberán et al. en [30], donde también estudian al polihedro asociado y describen familias de desigualdades que inducen facetas, las cuales extienden luego en [29]. Soler et al. [115] definen el Mixed General Routing Problem with Turn Penalties (MGRPTP), una generalización del MGRP que penaliza los giros y prohíbe aquellos que no están permitidos por normas de tránsito. En dicho trabajo, proponen una transformación polinómica del MGRPTP al Asymmetric Travelling Salesman

Problem (ATSP), mediante la sucesiva transformación del problema original al Generalized Mixed Rural Postman Problem with Turn Penalties y al Asymmetric Generalized Travelling Salesman Problem. En [17] se plantea el Mixed Capacitated General Routing Problem with Turn Penalties (MCGRPTP), y se propone transformarlo al Asymmetric Capacitated Vehicle Routing Problem, para el cual pueden aplicarse algoritmos ya desarrollados en la literatura. Más recientemente, en [2] se presenta otro enfoque de solución para el MCGRPTP, basado en la metaheurística conocida como Slack Induction by String Removals [26]. Pourhejazy et al [105], modelan la recolección de residuos electrónicos como un Capacitated General Routing Problem y proponen un modelo matemático así como también un proceso de construcción de soluciones iniciales y su posterior mejora con Búsqueda Tabú.

Esta tesis aborda el problema de optimizar la recolección de RSU en dos municipalidades, contribuyendo a la literatura con nuevas metodologías fácilmente generalizables a otros casos. Recordemos que en Berazategui distribuir la carga laboral requiere un enfoque que considere independientemente la cantidad de trabajo de los recolectores y la de los choferes, pues recorren distinta cantidad de zonas diariamente. En los trabajos citados esta distinción no está presente y se trata directamente la distribución del trabajo entre los equipos de recolección. Para abordar este problema, primero diseñamos una nueva zonificación y luego buscamos una ruta eficiente para cada zona. En el proceso de zonificación proponemos una nueva medida para cuantificar cuánto se asemeja la forma de una zona a un cuadrado. Además, como detallaremos más adelante, la duración de la jornada laboral de los choferes depende de dos variables independientes, por lo que la asignación de choferes a zonas es un problema de optimización multiobjetivo.

Por su parte, el enfoque de solución motivado por el caso de estudio en Quilmes considera simultáneamente la zonificación y el ruteo. Sin embargo, a diferencia de las publicaciones citadas anteriormente, no se busca generar una nueva zonificación, sino modificar iterativamente la existente. Proponemos métodos basados en Steepest-Ascent Hill Climbing [92], Simulated Annealing [92] y Variable Neighbourhood Search [95] con el objetivo de reducir la disparidad en la distribución del trabajo y facilitar la implementación de la solución obtenida. Además, presentamos una medida para la carga laboral como combinación lineal de la cantidad de residuos recolectados y la distancia recorrida.

Adicionalmente, como hemos mencionado en la sección anterior, la estrategia del achique hace que el problema de ruteo pueda modelarse como un MGRP, los cuales no son comunes en la literatura de optimización de recolección de residuos [66]. El MGRP y su variante con penalización de giros son problemas NP-hard [29, 115], por lo tanto, debido a que el tamaño de las instancias que aparecen en nuestros casos de estudio es considerablemente mayor que el de los experimentos computacionales presentes en la literatura revisada, desarrollamos un enfoque basado en el Asymmetric Traveling Salesman Problem que permite obtener soluciones eficientes en un tiempo razonable. Además, proponemos un nuevo algoritmo para buscar caminos mínimos con caminos prohibidos penalizando giros mediante una transformación polinomial del digrafo que modela al plano.

4. Definiciones

4.1. Conceptos preliminares

Debido a que la definición de algunos conceptos elementales de Teoría de Grafos pueden variar según la fuente bibliográfica, esta subsección busca evitar las ambigüedades que esto pudiera generar. Asimismo, definiremos formalmente el TSP y el MGRPTP.

En esta tesis, con *grafo* nos referiremos a un grafo simple (cada par de vértices está conectado por a lo sumo una arista y no hay *loops*). Asimismo, entendemos que un *digrafo* es un grafo dirigido simple (no hay múltiples arcos con los mismos vértices de partida y de llegada y no hay loops) y que en un *grafo mixto* cada par de vértices puede estar conectado por, a lo sumo, una arista o dos arcos con sentido opuesto, y no tiene loops. Por otra parte notaremos con (u, v) al arco que conecta a u con v y $\{u, v\}$ notará la arista que conecta a esos vértices.

Definición 1 (Camino y camino simple)**.** Un camino en G es una secuencia de vértices $p = (v_1, \ldots, v_n)$ no necesariamente distintos tales que:

- si $G = (V, E)$ es un grafo, $\{v_i, v_{i+1}\} \in E \; \forall \, i \in \{1, \ldots, n-1\}$
- si $G = (V, A)$ es un digrafo, $(v_i, v_{i+1}) \in A \; \forall \, i \in \{1, \ldots, n-1\}$
- si $G = (V, A, E)$ es un grafo mixto, $(v_i, v_{i+1}) \in A \vee \{v_i, v_{i+1}\} \in E \; \forall \, i \in \{1, \ldots, n-1\}$

Si los vértices son todos distintos, diremos que p es un *camino simple*.

Notación. Sea $G = (V, A, \omega)$ un digrafo pesado ($\omega \colon A \to \mathbb{R}$) y $\gamma = (v_1, \ldots, v_n)$ un camino en G, notaremos:

$$\omega(\gamma) = \sum_{i=1}^{n-1} \omega(v_i, v_{i+1})$$

Definición 2 (Circuito)**.** Un circuito C en G es un camino $C = (v_1, \ldots, v_n)$ tal que $v_n = v_1$ y no tiene aristas o arcos repetidos.

Definición 3 (Ciclo)**.** Un ciclo C en G es un camino $C = (v_1, \ldots, v_n)$ tal que $v_n = v_1$ y $v_1, \ldots, v_{n-1}$ son todos distintos.

Definición 4 (Distancia usual)**.** Sea G un grafo, digrafo o grafo mixto, diremos que la *distancia usual* entre dos vértices u, v es $n \in \mathbb{N}$ si y solo si existe un camino desde u hasta v con exactamente n arcos y/o aristas y no existe ningún camino desde u hasta v con $k < n$ arcos y/o aristas. Notaremos la distancia usual desde u hasta v como $d(u, v)$. Si no existiera un camino que conecte a u con v, notaremos $d(u, v) = +\infty$.

Definición 5 (Vecindario)**.** Definimos la noción de vecindario para los distintos tipos de grafos:

1. Sea $G = (V, E)$ un grafo, definimos el *vecindario* de v como:

$$N_G^U(v) \coloneqq \{w \in V \colon \{w, v\} \in E\}$$

2. Sea $G = (V, A)$ un digrafo, definimos el *vecindario de entrada* y el *vecindario de salida* de v, respectivamente:
$$N_G^{in}(v) \coloneqq \{u \in V : (u, v) \in A\}$$
$$N_G^{out}(v) \coloneqq \{u \in V : (v, u) \in A\}$$

3. Sea $G = (V, A, E)$ un grafo mixto, definimos el *vecindario* de v como:
$$N_G(v) \coloneqq N_G^U(v) \cup N_G^{in}(v) \cup N_G^{out}(v)$$
y al *vecindario cerrado* de v:
$$N_G[v] \coloneqq N_G(v) \cup \{v\}$$

Si el contexto no se presta para ambigüedad, omitiremos el subíndice G.

Definición 6 (k-Vecindario)**.** Sea $G = (V, A, E)$ un grafo, digrafo o grafo mixto, definimos el *k-vecindario* de v como:
$$N_G^{(k)}(v) \coloneqq \{u \in V : 0 < d(v, u) \leq k\}$$

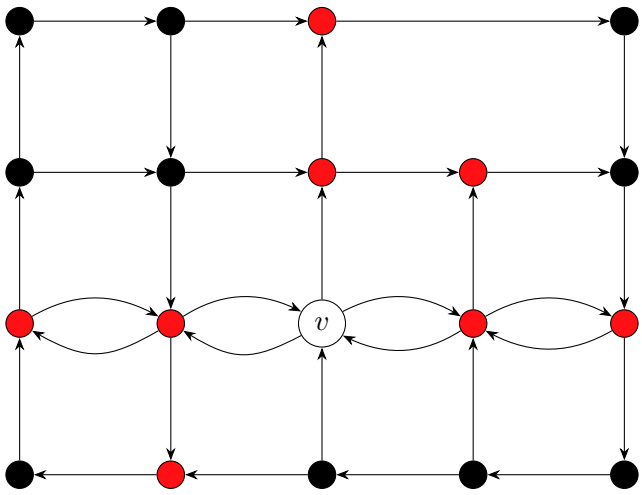


Figura 4: En rojo, $N_G^{(2)}(v)$, el 2-vecindario de v

Definición 7 (Excentricidad)**.** Sea $G = (V, E)$ un grafo conexo definimos la excentricidad de v como:
$$\varepsilon(v) = \max\{d(v, w) : w \in V\}$$

Definición 8 (Digrafo traspuesto)**.** Sea $G = (V, A, \omega)$ un digrafo pesado, su digrafo traspuesto $G^T = (V, C, \varrho)$ tiene el mismo conjunto de vértices y:

- $(v, w) \in C \Leftrightarrow (w, v) \in A$
- $\varrho(v, w) = \omega(w, v)$

4.2. El plano como un grafo

Obtuvimos los planos de los municipios en OpenStreetMap [102], una base de datos cartográficos abierta, construida y actualizada por voluntarios. Para visualizarlos, utilizamos Java OpenStreetMap Editor[1] (JOSM). El plano es esencialmente un archivo en formato `xml` que consta principalmente de tres tipos de objetos: `node`, `way` y `relation`. Cada `node` representa un punto con coordenadas de latitud y longitud y tiene asignado un número de identificación único. Opcionalmente, puede tener atributos para marcar alguna característica como la presencia de un semáforo, por ejemplo. Un `way` es una sucesión ordenada de `nodes`. Por ejemplo, si sus atributos indican que es una calle, especifican también en qué sentido se la puede recorrer, su nombre, su tipo (residencial, avenida, autopista, etc.), entre otras características. Un `way` también puede utilizarse para indicar vías de tren, caminos internos de parques, senderos peatonales, etc. Finalmente, como su nombre lo indica, un `relation` relaciona `ways` y/o `nodes`. Por ejemplo, un `relation` puede agrupar varios `way` para señalar los límites de un barrio o puede agrupar a dos `way` con un `node` para indicar un giro prohibido.

Utilizando la herramienta `osmfilter`[2] nos quedamos solo con los `way` que representan calles que el camión puede recorrer y con sus respectivos `nodes`. Dicha herramienta también nos permitió eliminar todas las `relations` que no representaran restricciones de tránsito. De esta manera, podemos definir formalmente cada objeto del plano filtrado.

Definición 9. Un `node` se puede representar como un par ordenado $(id, coords, atributos)$ donde $id \in \mathbb{Z}$ es el número de identificación, $coords \in \mathbb{R}^2$ es un par ordenado con las coordenadas de latitud y longitud, y *atributos* es un conjunto de pares ordenados con el nombre de cada atributo y su valor. Si no tiene atributos, $atributos = \emptyset$.

Ejemplo 1. Un `node` dado por $(5, (-34{,}5439, -58{,}4403), \{(\text{semáforo, sí})\})$ indica que su número de identificación es 5, sus coordenadas son $(-34{,}5439, -58{,}4403)$ y allí se encuentra un semáforo.

Definición 10. Sea N un conjunto de `nodes`, un `way` se puede representar con un par ordenado $(s, atributos)$ donde $s \in N^p$ $(p \geq 2)$ es una secuencia ordenada de p `nodes` y *atributos* es un conjunto de pares ordenados con el nombre de cada atributo y su valor. En este caso $atributos \neq \emptyset$ pues hay dos atributos que siempre están presentes: el tipo de calle y si es de una sola mano. Si *atributos* indica que la calle es de una sola mano, el sentido viene dado por el orden de s.

Notación. Notamos como $seq(w)$ a la secuencia ordenada de `nodes` del `way` w.

Ejemplo 2. Un `way` dado por $((n_1, n_2, n_3, n_4), \{(\text{tipo, avenida}), (\text{una sola mano, sí})\})$ representa una avenida de una sola mano que pasa por los nodos n_1, n_2, n_3 y n_4 en ese orden.

Definición 11. Sean N un conjunto de `nodes` y W un conjunto de `ways` con `nodes` en N, un `relation` se puede representar con una terna de tres `ways` o de dos `ways` y un `node`: $(a_1, a_2, a_3) \in W \times (N \cup W) \times W$. Si $a_2 \in N$, entonces a_2 es el primer o último elemento de $seq(a_1)$ y el primer o último elemento de $seq(a_3)$. Si $a_2 \in W$, entonces su secuencia de nodos tiene longitud exactamente 2 y cada uno de ellos debe ser el primer o último elemento de

[1] `https://josm.openstreetmap.de/`

[2] `https://wiki.openstreetmap.org/wiki/Osmfilter`

$seq(a_1)$ o de $seq(a_3)$. Se utilizan los `relation` para representar una prohibición de giro entre dos o tres `ways`.

Ejemplo 3. Sean dos `ways`:

$$w_1 = ((n_1, n_2, n_3), \{\text{(tipo, avenida), (una sola mano, no)}\})$$
$$w_2 = ((n_1, n_4, n_5, n_6), \{\text{(tipo, residencial), (una sola mano, sí)}\})$$

el `relation` $r = (w_1, n_1, w_2)$ indica que el giro de w_1 hacia w_2 pasando por n_1 está prohibido.

Ejemplo 4. Sean tres `ways`:

$$w_1 = ((n_1, n_2, n_3), \{\text{(tipo, avenida), (una sola mano, sí)}\})$$
$$w_2 = ((n_4, n_3), \{\text{(tipo, residencial), (una sola mano, no)}\})$$
$$w_3 = ((n_4, n_5, n_6, n_7), \{\text{(tipo, avenida), (una sola mano, sí)}\})$$

el `relation` $r = (w_1, w_2, w_3)$ indica que no se puede ir desde w_1 hacia w_3 pasando por w_2. Esto se utiliza generalmente para prohibir un giro en U en un boulevard.

Ahora estamos en condiciones de modelar el plano como un grafo mixto pesado.

Definición 12 (Grafo mixto pesado del plano)**.** Sea (N, W, R) un plano con N su conjunto de `nodes`, W su conjunto de `ways` y R su conjunto de `relations`, tal que cada $n \in N$ forma parte de la secuencia de al menos un `way`, definimos el grafo mixto pesado del plano como $P = (V_P, A_P, E_P, \omega_P)$ donde:

- conjunto de vértices: $V_P = N$. Para cada vértice utilizamos como etiqueta al número de identificación del `node` correspondiente.
- conjunto de arcos (dirigidos): $(i, j) \in A_P$ si y solo si existe $w \in W$ tal que en su secuencia i antecede directamente a j y los atributos de w indican que es de una sola mano.
- conjunto de aristas (no dirigidas): $\{i, j\} \in E_P$ si y solo si existe $w \in W$ tal que en su secuencia i antecede directamente a j y los atributos de w indican que es de doble mano.
- función de peso: $\omega_P \colon (A_P \cup E_P) \to \mathbb{R}_{\geq 0}$ mide la distancia geográfica entre los vértices del arco o la arista utilizando sus coordenadas de latitud y longitud y la Fórmula de Haversine [123].

A partir de R, definimos el conjunto de caminos prohibidos F como:

- $(i, j, k) \in F$ si y solo si existe $(w_1, j, w_2) \in R$ tal que cumple simultáneamente:
  1. $seq(w_1) = (n, \ldots, i, j) \vee seq(w_1) = (j, i, \ldots, n)$
  2. $seq(w_2) = (m, \ldots, k, j) \vee seq(w_2) = (j, k, \ldots, m)$
- $(i, j, k, \ell) \in F$ si y solo si existe $(w_1, w_2, w_3) \in R$ tal que cumple simultáneamente:
  1. $seq(w_2) = (j, k) \vee seq(w_2) = (k, j)$
  2. $seq(w_1) = (n, \ldots, i, j) \vee seq(w_1) = (j, i, \ldots, n)$
  3. $seq(w_3) = (m, \ldots, \ell, k) \vee seq(w_3) = (k, \ell, \ldots, m)$

Por convención en la representación de los planos en OSM, en P a lo sumo puede haber una arista o un arco que conecte dos nodos. Es decir, debe valer lo siguiente:

- $(i,j) \in A_P \Rightarrow (j,i) \notin A_P$
- $(i,j) \in A_P \Rightarrow \{i,j\} \notin E_P$
- $\{i,j\} \in E_P \Rightarrow (i,j) \notin A_P \wedge (j,i) \notin A_P$

También es importante notar que los datos que se descargan de OSM deben ser elegidos de tal manera que todos los vértices cuyas coordenadas se encuentren dentro de los límites geográficos del municipio pertenezcan a la misma componente fuertemente conexa de P. En nuestro caso, utilizando JOSM, incluímos en el plano las calles que se encontraran en un radio de 500 metros por fuera del límite del municipio.

A continuación, mostramos un ejemplo de cómo se confecciona el grafo mixto pesado del plano a partir de los datos de un archivo `.osm`. En pro de simplificar el ejemplo, utilizaremos la distancia euclídea en vez de la fórmula de Haversine. En la Figura 5 mostramos el grafo mixto pesado que corresponde a los datos proporcionados por un archivo `.osm` con los siguientes `nodes`:

$$\begin{aligned}
&\mathtt{n}_0 = (0,(0,0),\emptyset) && \mathtt{n}_6 = (6,(22,8),\emptyset)\\
&\mathtt{n}_1 = (1,(12,0),\emptyset) && \mathtt{n}_7 = (7,(32,8),\emptyset)\\
&\mathtt{n}_2 = (2,(22,0),\emptyset) && \mathtt{n}_8 = (8,(0,18),\emptyset)\\
&\mathtt{n}_3 = (3,(32,0),\emptyset) && \mathtt{n}_9 = (9,(12,18),\emptyset)\\
&\mathtt{n}_4 = (4,(0,8),\emptyset) && \mathtt{n}_{10} = (10,(32,18),\emptyset)\\
&\mathtt{n}_5 = (5,(12,8),\{\texttt{(semáforo, sí)}\})
\end{aligned}$$

los siguientes `ways`:

$$\begin{aligned}
&\mathtt{w}_1 = ((\mathtt{n}_3,\mathtt{n}_2,\mathtt{n}_1,\mathtt{n}_0),\{\texttt{(tipo, residencial), (una sola mano, sí)}\})\\
&\mathtt{w}_2 = ((\mathtt{n}_0,\mathtt{n}_4,\mathtt{n}_8),\{\texttt{(tipo, residencial), (una sola mano, sí)}\})\\
&\mathtt{w}_3 = ((\mathtt{n}_8,\mathtt{n}_9,\mathtt{n}_{10}),\{\texttt{(tipo, residencial), (una sola mano, sí)}\})\\
&\mathtt{w}_4 = ((\mathtt{n}_{10},\mathtt{n}_7),\{\texttt{(tipo, residencial), (una sola mano, sí)}\})\\
&\mathtt{w}_5 = ((\mathtt{n}_7,\mathtt{n}_3),\{\texttt{(tipo, residencial), (una sola mano, no)}\})\\
&\mathtt{w}_6 = ((\mathtt{n}_7,\mathtt{n}_6,\mathtt{n}_5),\{\texttt{(tipo, avenida), (una sola mano, no)}\})\\
&\mathtt{w}_7 = ((\mathtt{n}_5,\mathtt{n}_4),\{\texttt{(tipo, avenida), (una sola mano, no)}\})\\
&\mathtt{w}_8 = ((\mathtt{n}_1,\mathtt{n}_5),\{\texttt{(tipo, residencial), (una sola mano, no)}\})\\
&\mathtt{w}_9 = ((\mathtt{n}_5,\mathtt{n}_9),\{\texttt{(tipo, residencial), (una sola mano, no)}\})\\
&\mathtt{w}_{10} = ((\mathtt{n}_2,\mathtt{n}_6),\{\texttt{(tipo, residencial), (una sola mano, si)}\})
\end{aligned}$$

y las siguientes `relations`:

$$\mathtt{r}_1 = (\mathtt{w}_6,\mathtt{n}_5,\mathtt{w}_8) \quad \mathtt{r}_2 = (\mathtt{w}_7,\mathtt{n}_5,\mathtt{w}_9)$$

El conjunto de caminos prohibidos es $F = \{(6,5,1),(4,5,9)\}$. Cada arco, arista y vértice de P guardan los atributos y coordenadas de sus respectivos `ways` o `nodes`, según corresponda. A partir del grafo mixto pesado que modela al plano, definiremos los grafos que utilizaremos como base para la zonificación y para el ruteo.

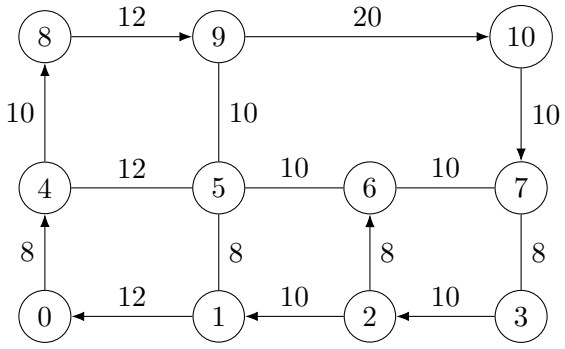


Figura 5: Grafo mixto pesado correspondiente al ejemplo de archivo `.osm`

4.3. Grafo de zonificación

Definición 13 (Grafo de zonificación)**.** Sea $P = (V_P, A_P, E_P, \omega_P)$ el grafo mixto pesado que representa al plano, llamaremos *grafo de zonificación* al grafo no dirigido pesado $S = (V_S, E_S, \omega_s)$ donde:

- $V_S = V_P$
- $E_S = E_P \cup \{\{i,j\} \colon (i,j) \in A\}$
- $\omega_S(i,j) = \begin{cases} \omega_P(\{i,j\}) & \text{si } \{i,j\} \in E_P \\ \omega_P((i,j)) & \text{si } (i,j) \in A_P \end{cases}$

Puesto que cada vértice de S tiene coordenadas de latitud y longitud, S puede representarse en $\mathbb{R}^2$. Con el propósito de utilizar un lenguaje coloquial que simplifique la explicación de nuestros enfoques de solución para la zonificación, a continuación relacionamos distintos elementos de S con sus representaciones en $\mathbb{R}^2$ y en el plano.

Definición 14 (Esquina)**.** Cada $v \in V_S$ es una *esquina* y notamos (lat_v, lon_v) a las coordenadas de latitud y longitud de v, respectivamente.

Definición 15 (Cuadra)**.** Cada $e = \{v, w\} \in E_S$ es una *cuadra*.

Definición 16 (Ángulo entre aristas)**.** Sean S el grafo de zonificación, $e_1 = \{i,j\} \in E_S$, $e_2 = \{k,j\} \in E_S$ tales que $k \neq i$, definimos el álgulo entre e_1 y e_2 como:

$$angulo(e_1, e_2) \coloneqq \begin{cases} A(e_1,e_2) + 2\pi & \text{si } A(e_1,e_2) \leq -\pi \\ A(e_1,e_2) - 2\pi & \text{si } A(e_1,e_2) > \pi \\ A(e_1,e_2) & \text{c.c.} \end{cases}$$

donde:

$$A(e_1, e_2) \coloneqq atan2(lon_k - lon_j, lat_k - lat_j) - atan2(lon_i - lon_j, lat_i - lat_j)$$

con $atan2$ la función arcotangente de dos parámetros, que para $(x,y) \neq (0,0)$, $atan2(y,x)$ devuelve el ángulo entre el eje positivo x y la recta que pasa por (x,y) y el origen. Observar que, como $Im(atan2) \subseteq (-\pi, \pi]$, entonces $Im(A) \subseteq (-2\pi, 2\pi]$. Por eso se realiza el ajuste necesario para que $Im(angulo) \subseteq (-\pi, \pi]$ y así facilitar la interpretación.

En la Figura 6 mostramos un ejemplo del resultado de la función $A(e_1, e_2)$ y de la función $angulo(e_1, e_2)$ para $e_1 = \{u, v\}, e_2 = \{v, w\}$, donde u, v, w son vértices con coordenadas $(-2, 1), (0, 0)$ y $(-1, -2)$ respectivamente. En este ejemplo, $A(e_1, e_2) = -\frac{3}{2}\pi$ y $angulo(e_1, e_2) = \frac{1}{2}\pi$. De esta manera, la función *angulo* permite interpretar mejor el giro en el recorrido $u \to v \to w$.

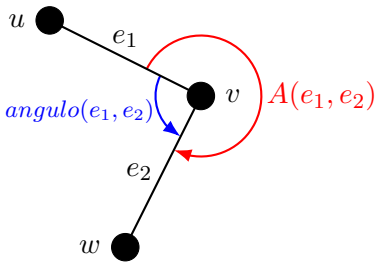


Figura 6: Ejemplo de la función *angulo* para dos aristas e_1, e_2 tales que $A(e_1, e_2) \leq -\pi$.

Definición 17 (Manzana)**.** Una *manzana* es un conjunto de aristas de un ciclo $C = (v_1, \ldots, v_n)$ en S que cumple:

$$\{v_{i+1}, v_{i+2}\} = \min_{e \in E_S \,:\, v_{i+1} \in e} angulo(\{v_i, v_{i+1}\}, e) \quad \forall i \in \{1, \ldots, n-2\}$$

En la Figura 7 mostramos ejemplos de una esquina, una cuadra y una manzana.

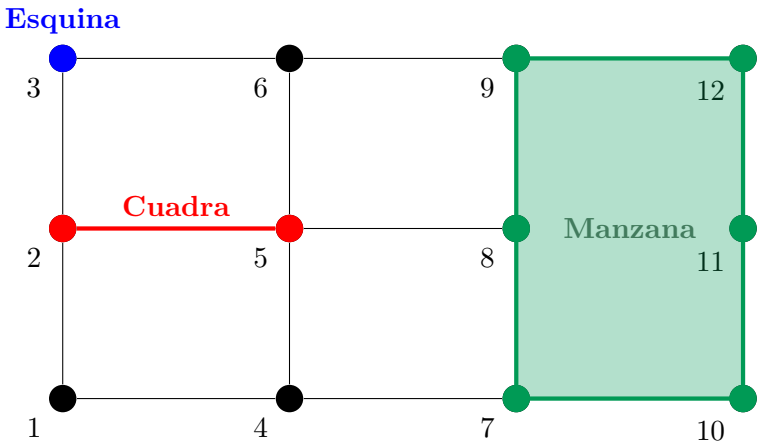


Figura 7: Ejemplos de una esquina, una cuadra y una manzana.

Para reconocer todas las manzanas en S, implementamos el algoritmo presentado en [127] que propone que cada cuadra $\{i, j\} \in E_S$, suponiendo sin pérdida de generalidad que el número de identificación de i es menor al de j, puede ser recorrida por su lado derecho $(i \to j)$ y por su lado izquierdo $(j \to i)$. De esta manera, se pueden hallar las manzanas comenzando un recorrido por cada lado de cada cuadra.

Definición 18 (Adyancencia de manzanas)**.** Dos manzanas M_1 y M_2 son adyacentes si y solo si $M_1 \cap M_2 \neq \emptyset$, es decir, tienen al menos una cuadra en común.

Definición 19 (Zona)**.** Una *zona* es un conjunto de manzanas.

Definición 20 (Cuadras de la zona)**.** Sea $Z = \cup_{i=1}^{n} M_i$ una zona, se definen *las cuadras de la zona* como el conjunto de cuadras que pertenecen a, por lo menos, una manzana de Z:

$$cuadras(Z) \coloneqq \{e \in E_S \colon \exists\, i \in \{1, \ldots, n\} \text{ tal que } e \in M_i\}$$

Definición 21 (Frontera de la zona)**.** Sea Z una zona, definimos la frontera de Z como el conjunto de cuadras que pertenecen a exactamente una manzana de Z:

$$\partial Z \coloneqq \{e \in E_S \colon \exists!\, i \in \{1, \ldots, n\} \text{ tal que } e \in M_i\}$$

Definición 22 (Zonas disjuntas)**.** Sean Z_1 y Z_2 dos zonas, diremos que son disjuntas si y solo si no tienen ninguna manzana en común.

Definición 23 (Zonas adyacentes)**.** Sean Z_1 y Z_2 dos zonas, diremos que son adyacentes si y solo si $\partial Z_1 \cap \partial Z_2 \neq \emptyset$

Definición 24 (Manzanas exteriores de una zona)**.** Sean S el grafo de zonificación, $\mathcal{M}$ el conjunto de manzanas de S y Z una zona, definimos el conjunto de las manzanas exteriores de Z como:

$$\mathcal{E}(Z) = \{B \in Z \colon \exists\, e \in B \text{ tal que } e \in \partial Z\}$$

Definición 25 (Manzanas adyacentes a una zona)**.** Sean S el grafo de zonificación, $\mathcal{M}$ el conjunto de manzanas de S y Z una zona, definimos el conjunto de las manzanas adyacentes a Z como:

$$\mathcal{A}(Z) = \{B \in \mathcal{M} \colon B \notin Z \wedge \exists\, e \in B \text{ tal que } e \in \partial Z\}$$

Definición 26 (Zonificación)**.** Una zonificación $\mathcal{Z}$ es un conjunto de zonas tales que, si $Z, Z' \in \mathcal{Z}$ y $Z \neq Z'$, entonces $Z \cap Z' = \emptyset$.

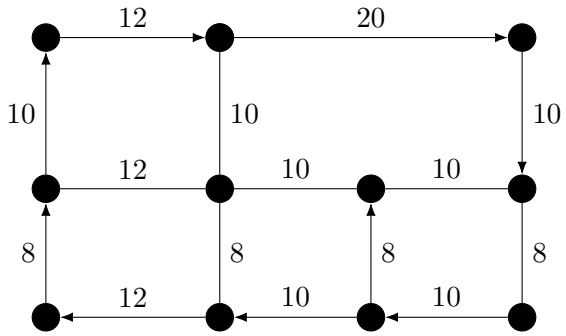
4.4. Digrafo de ruteo

A partir del grafo mixto del plano podemos definir el digrafo de ruteo.

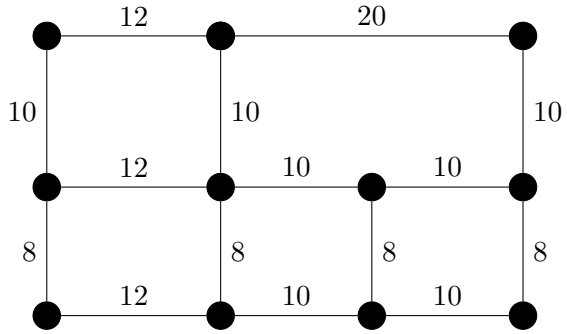
Definición 27 (Digrafo de ruteo)**.** Sea $P = (V_P, A_P, E_P, \omega_P)$ el grafo mixto del plano, definimos $R = (V_R, A_R, \omega_R)$ el digrafo de ruteo donde:

- $V_R = V_P$
- $A_R = A_P \cup \{(u, v), (v, u) \colon \{u, v\} \in E_P\}$
- $\omega_R(u, v) = \begin{cases} \omega_P((u, v)) & (u, v) \in A_P \\ \omega_P(\{u, v\}) & \{u, v\} \in E_P \end{cases}$

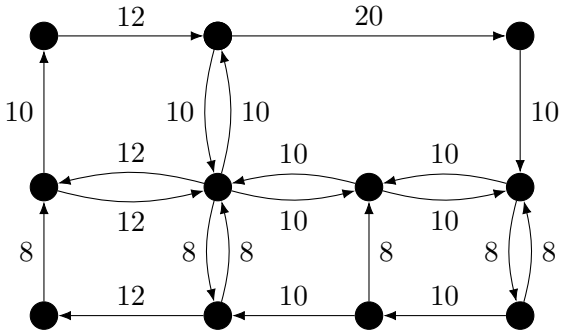
En la Figura 8 mostramos un ejemplo del grafo mixto de un plano, su grafo de zonificación y su digrafo de ruteo.



(a) P grafo mixto del plano.



(b) Grafo de zonificación correspondiente a P.



(c) Digrafo de ruteo correspondiente a P.

Figura 8: Ejemplo del grafo mixto de un plano, su grafo de zonificación y su digrafo de ruteo.

5. Creación de una nueva zonificación

En esta sección describimos el método que desarrollamos para elaborar una nueva zonificación, con el objetivo de balancear la carga laboral según algún criterio como, por ejemplo, la distancia que deben recorrer los recolectores a pie o la cantidad de residuos recolectados. Como objetivo secundario busca que la forma de las zonas sea parecida a un cuadrado para que sus límites sean fáciles de recordar para los choferes. Utiliza las manzanas como unidades básicas a partir de las cuales se construyen las zonas de recolección, siguiendo las definiciones de la Sección 4.3. Además, garantiza que las zonas obtenidas no estén formadas por grupos de manzanas no conectados entre sí.

Para explicar el procedimiento del método, utilizamos como criterio de carga laboral al metraje, es decir, la distancia que deben recorrer los recolectores a pie en cada zona.

Definición 28 (Metraje de una zona)**.** Sea $S = (V_S, E_S, \omega_S)$ el grafo de zonificación y sea Z una zona, definimos el *metraje* de Z como la suma de la longitud de sus cuadras:

$$m(Z) \coloneqq \sum_{e \in cuadras(Z)} \omega_S(e)$$

Observación 1. Si una cuadra pertenece a la frontera de dos zonas, su metraje se suma al metraje de cada zona.

Para cuantificar la similitud de la frontera de una zona a un cuadrado, tomamos en cuenta dos medidas. Cabe aclarar que utilizamos el término de medida en el sentido amplio de la palabra, no en el sentido estrictamente matemático ya que las mediciones que consideramos se aplican sobre conjuntos que no conforman una σ-álgebra en $\mathbb{R}^2$. La primera medida es la desarrollada por Rosin y Žunić [111] en el contexto de procesamiento de imágenes. Sea $A \subset \mathbb{R}^2$ acotado y tal que ∂A sea una curva cerrada y simple, los autores exponen una medida $\mathcal{Q}$ para determinar cuán parecida es ∂A a un cuadrado. Esta medida cumple las siguientes propiedades:

- $\mathcal{Q}(A) \in (0, 1]$ para todo A.
- $\mathcal{Q}(A) = 1 \iff \partial A$ es un cuadrado.
- $\mathcal{Q}(A)$ es invariante respecto a traslaciones, rotaciones y escalas.

La medida $\mathcal{Q}$ viene dada por:

$$\mathcal{Q}(A) = \frac{\sqrt{2}}{3} \frac{area(A)}{\min\limits_{\alpha \in [0, 2\pi]} \iint\limits_{A(\alpha)} (|x| + |y|) dx dy} \tag{1}$$

donde $A(\alpha)$ denota la rotación de A en un ángulo α alrededor de su centroide.

La otra medida considerada fue la medida de convexidad planteada por Delle Donne et. al [34] la cual, en el contexto de un problema de barrido y recolección de hojas, fue desarrollada para formar conjuntos de manzanas adyacentes cuya frontera se pareciera lo más posible a un

cuadrilátero. Sea $A \subset \mathbb{R}^2$ acotado tal que ∂A es una curva cerrada y simple y sea $conv(A)$ su cápsula convexa, la medida de convexidad $\mathcal{C}$ viene dada por:

$$\mathcal{C}(A) = \frac{area(A)}{area(conv(A))} \tag{2}$$

Proposición 1 (Propiedades de $\mathcal{C}$)**.** Sea $A \subset \mathbb{R}^2$ acotado tal que ∂A es una curva cerrada y simple, entonces $\mathcal{C}(A) \in (0, 1]$ y $\mathcal{C}$ es invariante por rotaciones, traslaciones y escalas.

Demostración. Puesto que $A \subseteq conv(A)$, de la definición de $\mathcal{C}$ se deduce que $\mathcal{C}(A) \in (0, 1]$, y que $\mathcal{C}(A) = 1$ si A es convexo. En particular, si A es un cuadrilátero, $\mathcal{C}(A) = 1$.

Probemos ahora que es invariante por transformaciones afines. Sea $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ definida como $f(x) = kRx + b$ con $k \in \mathbb{R}$, $b \in \mathbb{R}^2$ y $R \in \mathbb{R}^{2\times 2}$ una matriz de rotación. Veamos primero que $conv(f(A)) = f(conv(A))$:

$$\begin{aligned}
&z \in conv(f(A)) \Leftrightarrow \exists\, x_1, x_2 \in f(A), \lambda \in [0,1] \text{ tales que } z = \lambda x_1 + (1-\lambda)x_2 \\
&\Leftrightarrow \exists\, y_1, y_2 \in A, \lambda \in [0,1] \text{ tales que } z = \lambda f(y_1) + (1-\lambda) f(y_2) = f(\underbrace{\lambda y_1 + (1-\lambda)y_2}_{\in conv(A)}) \\
&\Leftrightarrow z \in f(conv(A))
\end{aligned}$$

Como el área es invariante por rotación y traslación y es cudrática en el factor de escala, se tiene que:

$$\mathcal{C}(f(A)) = \frac{area(f(A))}{area(conv(f(A)))} = \frac{k^2 area(A)}{k^2 area(conv(A))} = \mathcal{C}(A)$$

□

Ambas medidas son aplicables a una zona cuya frontera define una curva cerrada y simple. Sin embargo, en pruebas experimentales, ninguna de las dos brindó resultados completamente satisfactorios. Por un lado, $\mathcal{Q}$ prácticamente ignora pequeñas perturbaciones en la frontera de una zona con decenas de manzanas. Por otro lado, $\mathcal{C}$ otorga valores muy altos a zonas cuya frontera se asemeja a un rectángulo alargado. Por esta razón, decidimos utilizar como cuantificación de cuadratura de una zona a una combinación convexa entre $\mathcal{Q}$ y $\mathcal{C}$.

Definición 29 (Cuadratura de una zona)**.** Sea Z una zona tal que su frontera describe una curva cerrada simple en $\mathbb{R}^2$, definimos la *cuadratura* de Z como:

$$\mathcal{S}_\lambda(Z) \coloneqq \lambda\mathcal{Q}(Z) + (1-\lambda)\mathcal{C}(Z)$$

con $\lambda \in (0, 1)$, $\mathcal{Q}$ la fórmula de la Ecuación (1) y $\mathcal{C}$ la de Ecuación (2).

El valor de λ puede ser establecido experimentalmente según las particularidades del caso de estudio. A continuación, demostramos que $\mathcal{S}_\lambda$ mantiene las propiedades de $\mathcal{Q}$.

Proposición 2 (Propiedades de $\mathcal{S}_\lambda$)**.** Sea $A \subset \mathbb{R}^2$ acotado tal que su frontera es una curva cerrada simple, la medida de cuadratura $\mathcal{S}_\lambda$ cumple las siguientes propiedades:

1. $\mathcal{S}_\lambda(A) \in (0, 1]$
2. $\mathcal{S}_\lambda(A) = 1 \Leftrightarrow \partial A$ es un cuadrado.

3. $\mathcal{S}_\lambda(A)$ es invariante respecto a traslaciones, rotaciones y escalas.

Demostración. Sea $\lambda \in (0,1)$:

1. $0 \overset{\substack{\lambda>0\\ \mathcal{Q}(A)>0\\ \mathcal{C}(A)>0}}{<} S_\lambda = \lambda \underbrace{\mathcal{Q}(A)}_{\leq 1} + (1-\lambda)\underbrace{\mathcal{C}(Z)}_{\leq 1} \leq \lambda + (1-\lambda) = 1$

2. $\Rightarrow$) Por contrarrecíproco, supongamos que ∂A no es un cuadrado, entonces:

$$S_\lambda = \lambda \underbrace{\mathcal{Q}(A)}_{<1} + (1-\lambda)\mathcal{C}(A) < \underbrace{\lambda}_{<1} + \underbrace{(1-\lambda)}_{<1}\underbrace{\mathcal{C}(A)}_{\leq 1} \leq 1$$

   $\Leftarrow$) Si ∂A es un cuadrado, por propiedades de $\mathcal{Q}$, $\mathcal{Q}(A) = 1$ y, como $conv(A) = A$, se tiene también que $\mathcal{C}(A) = 1$. Por lo tanto, $S_\lambda(A) = 1$.

3. Sea $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ tal que $f(x) = aRx + b$ con $a \in \mathbb{R}_{\neq 0}$, $b \in \mathbb{R}^2$ y R una matriz de rotación. Como $\mathcal{Q}$ y $\mathcal{C}$ son invariantes respecto a f, S_λ también lo es:

$$S_\lambda(f(A)) = \lambda\mathcal{Q}(f(A)) + (1-\lambda)\mathcal{C}(f(A)) = \lambda\mathcal{Q}(A) + (1-\lambda)\mathcal{C}(A) = S_\lambda(A)$$

□

En la Figura 9 mostramos dos ejemplos para ilustrar cómo $\mathcal{S}_\lambda$ representa mejor nuestro criterio de cuadratura para la forma de la frontera de la zona que utilizar individualmente $\mathcal{C}$ o $\mathcal{Q}$. En la Figura 9a, la forma de Z se asemeja a un rectángulo, por lo que el valor de $\mathcal{C}(Z)$ es cercano a 1, mientras que la similitud a un rectángulo en vez de a un cuadrado está penalizada por $\mathcal{S}$. Por su parte, en Figura 9b, $\mathcal{S}$ penaliza la irregularidad de la forma de la frontera de Z mejor que $\mathcal{Q}$.

Una vez definidas las medidas de metraje y de cuadratura de una zona, estamos en condiciones de describir el procedimiento de zonificación. Comenzamos definiendo el grafo de manzanas y el concepto de zona conexa.

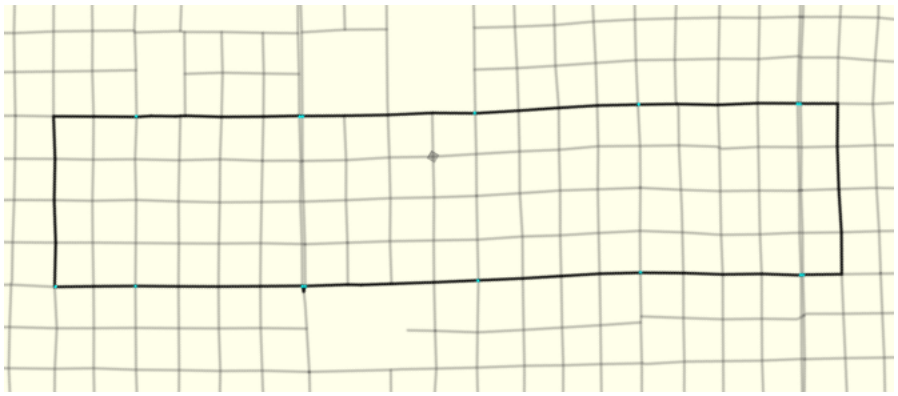
Definición 30 (Grafo de manzanas)**.** Sea $S = (V_S, E_S, \omega_S)$ el grafo de zonificación, definimos $G = (V_G, E_G)$ al grafo de manzanas donde:

- cada vértice de V_G representa a una manzana de S.
- $\{u, v\} \in E_G \Leftrightarrow$ la manzana a la que representa v es adyacente a la que representa u

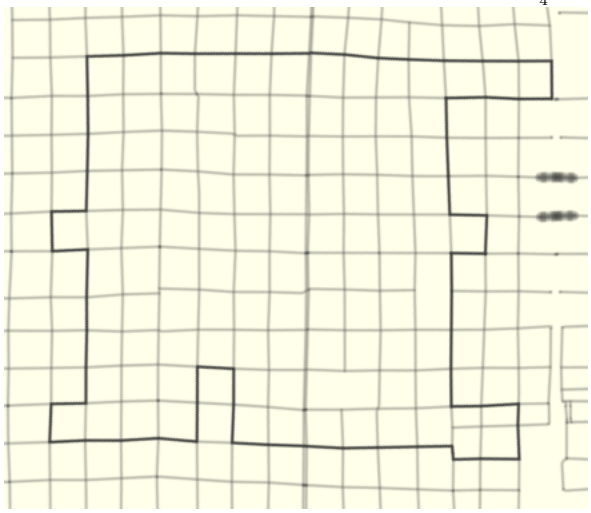
Notación. Notaremos con B_v a la manzana correspondiente al vértice $v \in V_G$ y con v_B al vértice correspondiente a la manzana B.

Definición 31 (Zona conexa)**.** Sea Z una zona y G el grafo de manzanas, diremos que Z es una zona *conexa* si el subgrafo inducido en G por $\{v \in V_G\colon B_v \in Z\}$ es conexo.

Como buscamos que las zonas sean conexas, sin pérdida de generalidad podemos suponer que G es conexo. De lo contrario, basta aplicar el procedimiento de zonificación a cada componente conexa de G.



(a) Para esta zona Z, $\mathcal{C}(Z) = 0{,}961$, $\mathcal{Q}(Z) = 0{,}608$ y $\mathcal{S}_{\frac{3}{4}}(Z) = 0{,}696$



(b) Para esta zona Z, $\mathcal{Q}(Z) = 0{,}962$, $\mathcal{C}(Z) = 0{,}79$ y $\mathcal{S}_{\frac{3}{4}}(Z) = 0{,}919$

Figura 9: Comparaciones entre $\mathcal{C}$ y $\mathcal{S}$ y entre $\mathcal{Q}$ y $\mathcal{S}$.

Elaborar una nueva zonificación con N zonas de recolección, de manera tal que cada una tenga distancia de recolección similar es equivalente a encontrar una partición balanceada de los vértices de G en N conjuntos. Dado que se ha demostrado que no hay un algoritmo polinomial para este tipo de problemas (salvo que P = NP) [3], y considerando además que se busca la forma de las zonas sea lo más cuadrada posible, decidimos desarrollar una heurística que permita obtener un resultado de buena calidad en un período de tiempo razonable.

La heurística de zonificación consta de dos fases. La Fase I diseña una zonificación que prioriza el equilibrio en el metraje, mientras que la Fase II busca mejorar la cuadratura de las zonas. A continuación describimos cada una.

5.1. Fase I

En la Tabla 1 resumimos la notación que utilizaremos para esta etapa.

Notación para la primera fase de la heurística	
S	Grafo de zonificación.
G	Grafo de manzanas.
$\mathcal{M}$	conjunto de manzanas de S.
B_v	la manzana representada por $v \in V_G$.
v_B	el vértice que representa a la manzana B.
$m(Z)$	metraje de Z.
$\mathcal{B}(F)$	sea $F = (V_F, E_F)$ un subgrafo inducido en G, $\mathcal{B}(F) \coloneqq \{B_v \in \mathcal{M} \colon v \in V_F\}$
$m(F)$	sea $F = (V_F, E_F)$ un subgrafo inducido en G, notamos $m(F) \coloneqq m(\mathcal{B}(F))$.
$\mathcal{A}(Z)$	el conjunto de manzanas adyacentes a la zona Z.
m_{MIN}	el metraje mínimo deseable para una zona.
N	cantidad de zonas a definir.
$\mathcal{Z}$	la zonificación.
$\mathcal{S}$	la medida de cuadratura.

Tabla 1: Notación para la heurística de zonificación

En el Algoritmo 1 presentamos el pseudocódigo para la rutina de la Fase 1. Requiere cuatro parámetros de entrada: $\mathcal{M}$ el conjunto de manzanas de S, G el correspondiente grafo de manzanas, N el número de zonas a crear y m_{MIN} el metraje mínimo deseable para una zona. Si solo falta definir una zona, la construye con todas las manzanas en $\mathcal{B}(G)$ (l. 1–2). De lo contrario, inicializa una zonificación vacía y una nueva zona Z con la manzana correspondiente al vértice de mayor excentricidad en G (l. 4–6). Mientras el metraje de Z no alcance el valor ideal $\frac{m(G)}{N}$ y mientras queden al menos $N - 1$ manzanas para poder definir las $N - 1$ zonas restantes, agrega a Z la manzana aledaña B que maximice $\mathcal{S}_\lambda(Z \cup \{B\})$ (l. 7–9). A continuación, añade Z a la zonificación y elimina de V_G los vértices correspondientes a las manzanas de Z (l. 10–11).

Puede ocurrir que al eliminar vértices de V_G, G deje de ser conexo. Por lo tanto, en las líneas 12–14 clasifica las componentes conexas de G en componentes *grandes* (con metraje mayor o igual que m_{MIN}) y componentes *pequeñas* (con metraje menor que m_{MIN}). En las líneas 15–16 se reclasifican las componentes pequeñas hasta que la cantidad total de manzanas

Algoritmo 1: `RUTINA_FASE_I`

Entrada: G grafo de manzanas
N cantidad de zonas a definir
$\mathcal{M}$ conjunto de manzanas
m_{MIN} metraje mínimo deseable para una zona

Salida: $\mathcal{Z}$ zonificación

si $N = 1$ **entonces**
  $\mathcal{Z} \leftarrow \mathcal{B}(G)$
en otro caso
  $\mathcal{Z} \leftarrow \emptyset$
  $\bar{v} \leftarrow$ vértice con excentricidad máxima en G
  $Z \leftarrow \{B_{\bar{v}}\}$
  mientras $m(Z) < \frac{m(G)}{N}$ **y** $|Z| < |V_G| - (N-1)$ **hacer**
    $\tilde{B} \leftarrow \arg\max_{B \in \mathcal{A}(Z)} \mathcal{S}(Z \cup \{B\})$
    $Z \leftarrow Z \cup \{\tilde{B}\}$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{Z\}$
  $G' \leftarrow$ subgrafo inducido en G por $V_G - \{v_B \colon B \in Z\}$
  $CCG \leftarrow$ componentes conexas de G'
  $G_{LARGE} \leftarrow \{G_j \in CCG \colon m(G_j) \geq m_{MIN}\}$
  $G_{SMALL} \leftarrow CCG \setminus G_{LARGE}$
  mientras $\sum_{j \colon G_j \in G_{LARGE}} |V_{G_j}| < N - 1$ **hacer**
    Agregar la componente de mayor metraje de G_{SMALL} a G_{LARGE} y quitarla de G_{SMALL}
  $SOL \leftarrow$ solución de MILP1 (SOL_j: cantidad de zonas a definir en $\mathcal{B}(G_j)$)
  para cada $G_j \in G_{LARGE}$ **hacer**
    si $SOL_j > 0$ **entonces**
      $\mathcal{Z} \leftarrow \mathcal{Z} \cup$ `RUTINA_FASE_I`$(G_j, SOL_j, \mathcal{M}, m_{MIN})$
    en otro caso
      $G_{SMALL} \leftarrow G_{SMALL} \cup \{G_j\}$
  mientras $G_{SMALL} \neq \emptyset$ **hacer**
    $H \leftarrow$ la componente con mayor metraje en G_{SMALL}
    $G_{SMALL} \leftarrow G_{SMALL} - H$
    $A \leftarrow$ zonas de $\mathcal{Z}$ adyacentes a $\mathcal{B}(H)$
    subzonas $\leftarrow$ `RUTINA_FASE_I`$(H, \text{mín}\{|A|, |V_H|\}, \mathcal{M}, m_{MIN})$
    subzonas_validas $\leftarrow$ subzonas que limitan con alguna zona de $\mathcal{Z}$
    $SOL \leftarrow$ solución de MILP2 para las subzonas en *subzonas_validas*
    Anexar cada subzona válida a la zona en A indicada por SOL
    Agregar cada componente conexa del subgrafo inducido por $H - \cup_{\{B \in s \colon s \in subzonas - subzonas_validas\}} \{v_B\}$
devolver $\mathcal{Z}$

de las componentes grandes superen $N-1$. De esta manera, se asegura que en las componentes grandes podremos definir las $N-1$ zonas restantes. Un modelo MILP, que llamamos MILP1 y que describiremos más adelante, decide cuántas zonas definir en cada componente grande (l. 17). La rutina se aplica recursivamente a cada componente grande donde la solución indique que se debe definir al menos una zona (l. 19–20). Las componentes grandes en las cuales no se defina ninguna zona pasan a ser consideradas componentes pequeñas (l. 21–22).

En las líneas 23–31, se reparten las manzanas de las componentes pequeñas entre las zonas aledañas. Se comienza eligiendo la componente pequeña con mayor metraje H, quitándola de G_{SMALL} e identificando sus zonas adyacentes (l. 24–26). Se ejecuta recursivamente la rutina para definir subzonas en las manzanas representadas por H y distinguen aquellas que limiten con una zona de $\mathcal{Z}$ como *subzonas válidas* (l. 27–28). Posteriormente, resuelve un modelo MILP, que llamamos MILP2 y describiremos más adelante, para determinar a qué zona se anexará cada subzona válida (l. 29). En la línea 30 se anexan las subzonas válidas según lo indicado por la solución de MILP2. Como pueden haber quedado subzonas que originalmente no lindaban con ninguna zona, los vértices que representan a sus manzanas forman al menos una componente pequeña de H. Por eso, en la línea 31, se agregan esas componentes a G_{SMALL}. Finalmente, en la línea 32 termina la ejecución devolviendo una zonificación que cubre todas las manzanas representadas por V_G en N zonas conexas.

En la Figura 10 mostramos un ejemplo de cómo progresa el algoritmo para generar tres zonas, en un caso donde no aparecen componentes pequeñas. Por su parte, en la Figura 11 ilustramos cómo procede el bucle que reparte las manzanas de las componentes pequeñas entre sus respectivas zonas adyacentes.

A continuación, presentamos los modelos MILP auxiliares empleados en el Algoritmo 1. El primero, MILP1, decide cuántas zonas se crearán en cada una de las componentes grandes de G. La Tabla 2 describe los conjuntos, parámetros y variables de decisión del modelo, que formulamos como:

$$\text{mín} \quad z + M\left(\sum_{i\in I}\sum_{k\in K-K^{(i)}} k\theta_{ik}\right) \tag{3}$$

$$\text{s.a:} \quad \sum_{i\in I}\sum_{k\in K} k\theta_{ik} = c \tag{4}$$

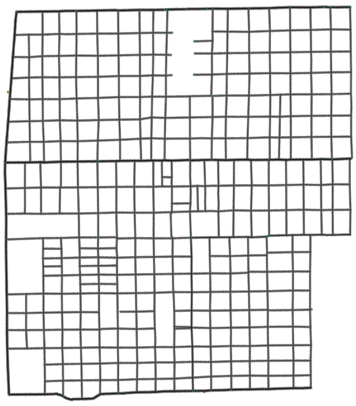
$$\sum_{k\in K} \theta_{ik} = 1 \qquad i \in I \tag{5}$$

$$\sum_{\substack{k\in K\\ k>q_i}} \theta_{ik} = 0 \qquad i \in I \tag{6}$$

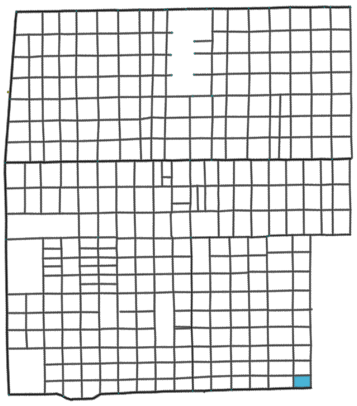
$$\sum_{\substack{k\in K\\ k\geq 1}} \frac{m_i}{k}\theta_{ik} - \sum_{\substack{k\in K\\ k\geq 1}} \frac{m_j}{k}\theta_{jk} \leq z \qquad i,j \in I \tag{7}$$

$$\theta_{ik} \in \{0,1\} \quad i \in I,\ k \in K$$

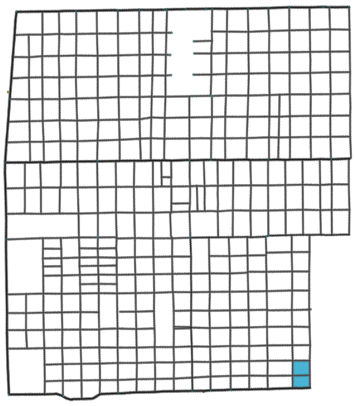
$$z \in \mathbb{R}_{\geq 0}$$



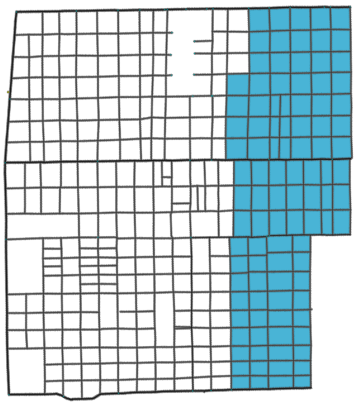
Sector del plano a zonificar con 3 zonas.



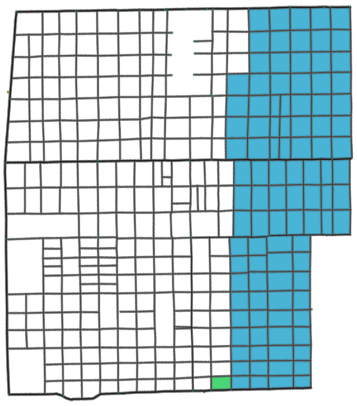
A partir de una manzana con máxima excentricidad, se inicializa Z_1.



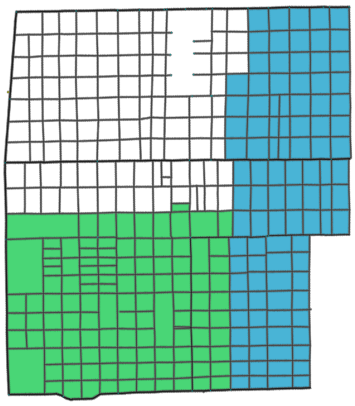
Se añade la manzana adyacente a Z_1 que maximice $\mathcal{S}_\lambda$.



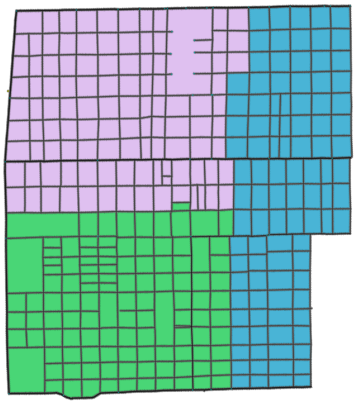
Iterativamente, se añaden manzanas adyacentes a Z_1 que maximicen $\mathcal{S}_\lambda$.



A partir de una manzana con máxima excentricidad de la única componente grande, se inicializa Z_2.

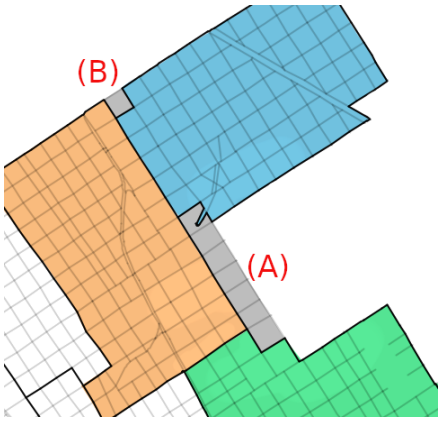


Se añaden iterativamente manzanas a Z_2 que maximicen $\mathcal{S}_\lambda$.

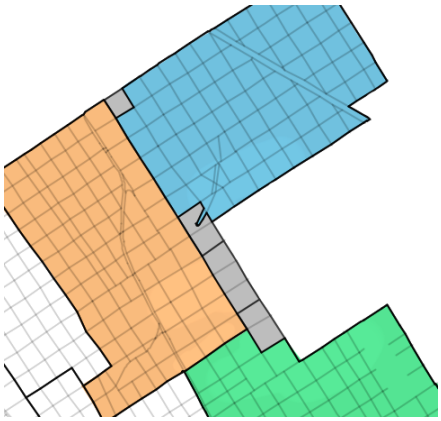


La última zona se define como el conjunto de manzanas que todavía no forman parte de la zonificación.

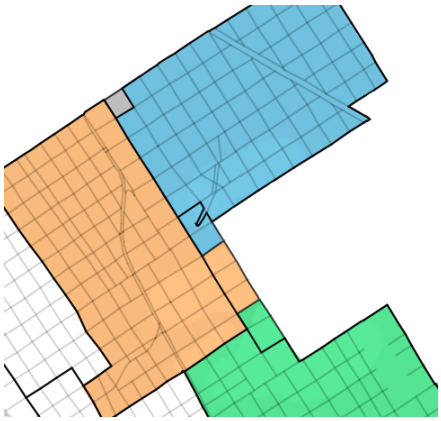
Figura 10: Ejemplo de progresión del algoritmo.



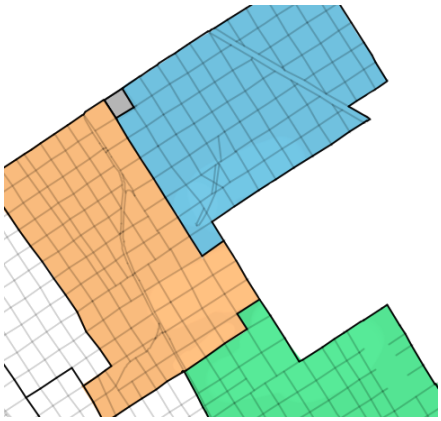
Dos componentes pequeñas (en gris) deben repartir sus manzanas entre sus zonas adyacentes. Comenzamos con (A).



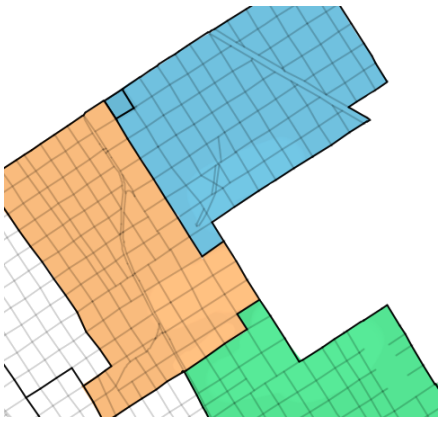
Se aplica el Algoritmo 1 para definir tantas subzonas como zonas adyacentes tenga la componente. En este caso, tres.



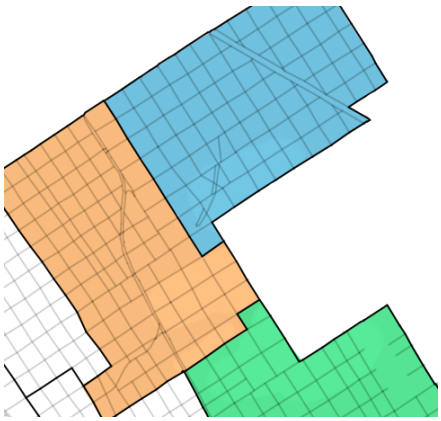
MILP2 asigna cada subzona válida a una de las zonas adyacentes.



Cada zona anexa la subzona asignada. El mismo procedimiento se aplica a la componente (B).



Como sólo hay una manzana, se le asigna la zona adyacente con menor metraje.



La zona anexa la manzana asignada.

Figura 11: Ejemplo de aplicación de procedimiento de repartición de manzanas de componentes pequeñas.

Notación	Descripción
Conjuntos y parámetros	
I	Conjunto de índices de las componentes grandes de G, ordenadas decrecientemente según su metraje
c	Cantidad de zonas a ser generadas
m_{MIN}	Metraje mínimo deseable para una zona
m_i	Metraje de la i-ésima componente grande de G
q_i	cantidad de manzanas de la componente i
M	Una constante lo suficientemente grande
K	$\{0, 1, \ldots, c\}$
$K^{(i)}$	$\{k \in K\colon\ k \cdot m_{MIN} \leq m_i\} \quad i \in I$
Variables	
θ_{ik}	Indica si en la i-ésima componente se definirán k zonas ($\theta_{ik} \in \{0, 1\}$)
z	Variable auxiliar para minimizar la máxima diferencia de metraje entre zonas definidas en diferentes componentes. ($z_i \in \mathbb{R}_{\geq 0}$)

Tabla 2: Conjuntos, parámetros y variables de MILP1

La función objetivo (3) minimiza la máxima diferencia de metraje entre las zonas que se definirán en cada componente grande, suponiendo que si en la componente i se definen k zonas, entonces cada una de ellas tendrá un mentraje de $\frac{m_i}{k}$. El segundo término de la función objetivo penaliza la creación de zonas con metraje menor a m_{MIN}: si $\theta_{ik} = 1$ con $k \notin K^{(i)}$, significa que en la componente i se crearán potencialmente k zonas con metraje menor que m_{MIN}. Con respecto a las restricciones, (4) asegura que se crearán la cantidad de zonas de recolección necesarias; (5) garantiza que en cada componente la cantidad de zonas a crear es única; (6) acota la cantidad de zonas que pueden ser creadas en una componente dado que cada zona debe estar compuesta por al menos una manzanay (7) acota inferiormente a z con la diferencia entre cada par de componentes del metraje esperado de las zonas que se definirán en cada una.

Observar que sin la restricción (4), el modelo sería siempre factible: basta tomar, por ejemplo, $\theta_{i1} = 1$ para cada $i \in I$ y $z \geq \text{máx}_{i,j \in I} \frac{m_i - m_j}{k}$. A continuación, probamos una condición necesaria y suficiente para la factibilidad de MILP1, que no resulta demasiado restrictiva para aplicaciones reales.

Proposición 3. El modelo MILP1 es factible si y solo si la cantidad total de manzanas de las componentes grandes es mayor o igual a c (i.e. $\sum_{i \in I} q_i \geq c$).

Demostración. $\Rightarrow$) Sea $(\theta^*, z^*) \in \{0, 1\}^{|I| \times |K|} \times \mathbb{R}_{\geq 0}$ una solución factible de MILP 1, se cumple la restricción (4). Como cada zona está formada por al menos una manzana, se deduce que la cantidad total de manzanas de las componentes grandes es mayor o igual a c.

$\Leftarrow$) Basta hallar una solución factible $(\theta^*, z^*) \in \{0, 1\}^{|I| \times |K|} \times \mathbb{R}_{\geq 0}$ para MILP1. Para cada $i \in I$, recordemos que $\mathcal{B}(G_i)$ es el conjunto de manzanas que representan los vértices de la i-ésima componente. Como $\sum_{i \in I} q_i \geq c$, tomamos un subconjunto $B \subseteq \cup_{i \in I} \mathcal{B}(G_i)$ tal que

$|B| = c$.

Para cada $i \in I$ definimos:

$$\theta^*_{ik} = \begin{cases} 1 & \text{si } k = |B \cap \mathcal{B}(G_i)| \\ 0 & \text{c.c.} \end{cases}$$

Por definición, θ^* cumple las restricciones (5) y (6). Además, cumple (4):

$$\sum_{i \in I} \sum_{k \in K} k\theta^*_{ik} = \sum_{i \in I} |B \cap \mathcal{B}(G_i)| = |B| = c$$

Por otra parte, si $z^* = \text{máx}_{i,j \in I} \sum_{\substack{k \in K \\ k \geq 1}} \frac{m_i}{k} \theta^*_{ik} - \sum_{\substack{k \in K \\ k \geq 1}} \frac{m_j}{k} \theta^*_{jk}$, entonces se cumple la restricción (7). Por ende, (θ^*, z^*) es una solución factible de MILP1. □

El segundo modelo auxiliar, MILP2, es utilizado para repartir las subzonas entre las zonas limítrofes. La Tabla 3 describe los conjuntos, parámetros y variables del modelo, cuya formulación es la siguiente:

Notación	Descripción
Conjuntos	
I	Conjunto de subzonas
J	Conjunto de zonas entre las que se repartirán las subzonas
Parámetros	
δ_{ij}	vale 1 si y solo si la subzona $i \in I$ limita con la zona $j \in J$
m_i	Metraje de la subzona $i \in I$
$\bar{m}_j$	Metraje de la zona $j \in J$ sin anexar ninguna subzona
Variables	
x_{ij}	Vale 1 si y solo si la subzona i es anexada a la zona j ($x_{ij} \in \{0, 1\}$)
z	Acota superiormente la diferencia de metraje entre cada par de zonas en J luego de llevarse a cabo la asignación de subzonas. ($z \in \mathbb{R}_{\geq 0}$)

Tabla 3: Conjuntos, parámetros y variables de MILP2

$$\begin{aligned}
\text{mín} \quad & z & & & & (8) \\
\text{s.t.:} \quad & x_{ij} & \leq & \delta_{ij} & i \in I,\ j \in J & (9) \\
& \sum_{j \in J} x_{ij} & = & 1 & i \in I & (10) \\
& \left(\sum_{i \in I} m_i x_{ij} + \bar{m}_j \right) - \left(\sum_{i \in I} m_i x_{ik} + \bar{m}_k \right) & \leq & z & (j,k) \in J^2 & (11) \\
& x_{ij} & \in & \{0,1\} & i \in I,\ j \in J & \\
& z & \in & \mathbb{R}_{\geq 0} & &
\end{aligned}$$

La función objetivo (8) minimiza la disparidad entre la zona con mayor metraje y la de menor metraje. Las restricciones (9) previenen que una subzona sea anexada a una zona no limítrofe; (10) aseguran que cada subzona debe ser anexada a exactamente una zona y (11) acotan inferiormente a z con la diferencia de metraje post-asignación de subzonas entre cada par de zonas en J. A continuación, mostramos que una condición necesaria y suficiente para que MILP2 sea factible es que cada subzona limite con al menos una zona ya definida.

Proposición 4. MILP2 es factible si y solo si para cada $i \in I$ vale que $\sum_{j \in J} \delta_{ij} \geq 1$.

Demostración. $\Rightarrow$) Supongamos que existen $x \in \{0,1\}^{|I| \times |J|}$, $z \in \mathbb{R}_{\geq 0}$ que cumplen (9)–(11). Luego:

$$\sum_{j \in J} \delta_{ij} \overset{(9)}{\geq} \sum_{j \in J} x_{ij} \overset{(10)}{=} 1 \quad \forall\, i \in I$$

$\Leftarrow$) Construyamos una solución $(x^\star, z^\star)$ para el modelo. Si para cada $i \in I$ vale que $\sum_{j \in J} \delta_{ij} \geq 1$, como $\delta_{ij} \in \{0,1\}$, para cada $i \in I$ existe $j(i) \in J$ tal que $\delta_{ij(i)} = 1$. Luego, basta tomar $x^\star \in \{0,1\}^{|I| \times |J|}$ con:

$$x^\star_{ij} = \begin{cases} 1 & \text{si j = j(i)} \\ 0 & \text{c.c.} \end{cases}$$

Es inmediato ver que $x^\star$ cumple (9) y (10).

Finalmente, definimos:

$$z^\star = \max_{j,k \in J} \left(\sum_{i \in I} m_i x^\star_{ij} + \bar{m}_j \right) - \left(\sum_{i \in I} m_i x^\star_{ik} + \bar{m}_k \right)$$

Así, se cumple (11) y $z^* \geq 0$.

Hemos construido entonces una solución a MILP2, por lo que el modelo es factible.

□

Con esto concluimos la descripción de la Fase I de nuestro enfoque de solución para la zonificación. Observar que el procedimiento se puede aplicar fácilmente con criterios de equilibrio de zonificación que sean diferentes a los adoptados aquí. Por ejemplo, se podría utilizar la cantidad de residuos recolectados en lugar del metraje. A continuación demostramos que el procedimiento permite obtener una zonificación con la cantidad de zonas conexas indicadas.

Proposición 5. Sean $S = (V_S, E_S, \omega_s)$ el grafo de zonificación, $\mathcal{M}$ el conjunto de las manzanas de S, $G = (V_G, E_G)$ el correspondiente grafo de manzanas y m_{MIN} el metraje mínimo deseable para una zona. Si G es conexo, entonces para todo $1 \leq N \leq |V_G|$, `RUTINA_FASE_I`(G, N, $\mathcal{M}$,m_{MIN}) termina y proporciona una zonificación con N zonas conexas que cubre todas las manzanas representadas por V_G.

Demostración. Lo demostraremos por inducción fuerte en $k = |V_G|$, considerando:

$P(k)$: para todo $1 \leq N \leq k$ `RUTINA_FASE_I`$(G,\ N,\ \mathcal{M})$ termina y proporciona una zonificación con N zonas conexas que cubre todas las manzanas representadas por V_G.

$P(1)$: como $1 \leq N \leq 1$, entonces sólo hay que probar que el enunciado es verdadero para $N = 1$. Se cumple la condición de la línea 1, por lo tanto la zonificación queda definida por la única manzana que representa el vértice de G. La ejecución continúa a la línea 30 y termina. La zonificación obtenida cubre la manzana representada por V_G y tiene una zona conexa, pues consta de una única manzana.

$P(\ell) \forall\, \ell < k \Rightarrow P(k)$: para el caso $N = 1$, como se cumple la condición de la línea 1, la única zona queda conformada por todas las manzanas representadas por V_G. Como G es conexo, la zona también lo es. La ejecución continúa en la línea 32, donde finaliza devolviendo una zonificación que cubre todas las manzanas representadas por V_G y está compuesta de una zona conexa.

Veamos el caso $N \geq 2$. Como el condicional de la línea 1 no se cumple, la ejecución continúa en la línea 3. La línea 6 inicializa Z con una manzana, por lo tanto Z es conexa. El bucle de las líneas 7–9 termina: una de las condiciones impone una cota superior a la cantidad de manzanas de Z, que aumenta de manera estrictamente monótona en cada iteración. Esta condición también garantiza que al finalizar el bucle queden al menos $N - 1$ manzanas sin zonificar. Por otra parte, como en cada iteración Z incorpora una de sus manzanas adyacentes, Z se mantiene conexa. Luego, se la añade a $\mathcal{Z}$.

La ejecución prosigue con la identificación y la clasificación de las componentes conexas. Probemos que el bucle de las líneas 15–17 concluye en, a lo sumo, $|G_{SMALL}|$ iteraciones. Luego de ese número de iteraciones, $G_{LARGE} = CCG$. Por condición del bucle de las líneas 7–9, hay al menos $N - 1$ manzanas para definir las $N - 1$ zonas restantes. Por lo tanto, deja de cumplirse la condición y el bucle termina. Observar que se cumple la condición necesaria y suficiente para que MILP1 sea factible (Proposición 3), pues la cantidad total de manzanas de las componentes de G_{LARGE} es mayor o igual que $N - 1$. Entonces SOL está bien definida en la línea 17.

En las líneas 18–22, en cada componente grande se definen la cantidad de zonas indicada por la solución de MILP1. Si $SOL_j \geq 1$, al ser SOL solución de MILP1, cumple que $1 \leq SOL_j \leq |V_{G_j}| < |V_G| = k$. Por hipótesis inductiva, `RUTINA_FASE_I`$(G_j, SOL_j, \mathcal{M})$ devuelve una zonificación que cubre todas las manzanas que representan V_{G_j}, en SOL_j zonas conexas. Estas zonas son agregadas a $\mathcal{Z}$, por lo tanto todas sus zonas siguen siendo conexas. Si $S_j = 0$, las componentes grandes pasan a ser consideradas componentes pequeñas. Al terminar de ejecutarse el bucle de las líneas 18–22, $\mathcal{Z}$ consta de N zonas conexas, pero faltan cubrir las manzanas de las componentes en G_{SMALL}.

Antes de probar que el bucle de las líneas 23–31 termina, demostramos la correctitud de su contenido. Primero, en la línea 26, notemos que $A \neq \emptyset$. Como H es conexo, si ninguna de las manzanas que representan V_H fueran adyacentes a alguna manzana de alguna zona de $\mathcal{Z}$, entonces H sería una componente conexa de G y $H \neq G$. Esto sería absurdo pues G es conexo. Entonces, existe al menos una zona adyacente a las manzanas que representa

V_H. Continuamos con la definición de subzonas, en la línea 27. Como $|V_H| < |V_G| = k$ y $1 \leq \text{mín}\{|A|, |V_H|\} \leq |V_H|$, por hipótesis inductiva, `RUTINA_FASE_I`$(H, \text{mín}\{|A|, |V_H|\}, \mathcal{M})$ devuelve una zonificación de todas las manzanas que representan V_H en $\text{mín}\{|A|, |V_H|\}$ zonas conexas. Como todas las manzanas representadas por V_H forman parte de alguna subzona y existe al menos una zona adyacente a las manzanas de H, entonces $subzonas_validas \neq \emptyset$. Por Proposición 4, MILP2 resulta factible y SOL está bien definido en la línea 29.

Por lo argumentado en el párrafo anterior, en cada iteración del bucle se define al menos una subzona válida con al menos una manzana que es anexada a alguna zona adyacente. Por lo tanto, al momento de comenzar el bucle, su número de iteraciones está acotado superiormente por $\sum_{j:\, G_j \in G_{SMALL}} |V_{G_j}|$. Además, al anexar una subzona a una zona, esta se mantiene conexa. Finalmente, notar que al finalizar el bucle de las líneas 23–31, todas las manzanas representadas por V_G forman parte de una zona. De lo contrario, las manzanas que no formaran parte de una zona representarían una componente en G_{SMALL}, pero al terminar el bucle, $G_{SMALL} = \emptyset$.

Finalmente, el algoritmo devuelve $\mathcal{Z}$ una zonificación que consta de N zonas conexas que cubren todas las manzanas representadas por V_G. □

5.2. Fase II

Como el criterio de parada de la generación de zonas está relacionado con el metraje, puede ser que la forma de la frontera pueda mejorarse. Por esta razón, desarrollamos la Fase II de la heurística, cuyo objetivo es minimizar el número de vértices del polígono que describe la frontera de cada zona mediante la transferencia de manzanas entre zonas adyacentes. Consiste esencialmente en la aplicación del *Steepest Ascent Hill Climbing* con Búsqueda Tabú [92], un algoritmo de búsqueda local que parte de una solución inicial y, en cada iteración, evalúa todos los vecinos de la solución actual, eligiendo aquel que no se encuentre en la lista tabú y tenga el mejor valor en la función objetivo. A continuación introducimos algunas definiciones necesarias para explicar el procedimiento de la Fase II.

Definición 32 (α-vértice)**.** Sea Z una zona y sean $e_1 = \{i, j\}$, $e_2 = \{j, k\}$ aristas tales que $e_1, e_2 \in \partial Z$, sea $\alpha \in (0, \frac{\pi}{2}]$, diremos que j es un *α-vértice* de ∂Z si y solo si

$$-\pi + \alpha \leq angulo(e_1, e_2) \leq \pi - \alpha$$

Notación. Notaremos como $v_\alpha(Z)$ a la cantidad de α-vértices en ∂Z

Como la transformación que sufre cada zona impacta en su metraje, es prudente establecer un límite para cuánto puede crecer o encogerse una zona con respecto a su estado inicial. Para esto definimos un radio aceptable de modificación.

Definición 33 (r-radio)**.** Sean G el grafo de manzanas y $r \geq 1$, definimos el *r-radio* de una zona Z como:

$$R_r(Z) = \{\hat{B} \in \mathcal{A}(Z) \colon \min_{B \in Z}\{d(v_{\hat{B}}, v_B)\} \leq r\} \cup \{\hat{B} \in Z \colon \min_{B \in \mathcal{E}(Z)}\{d(v_{\hat{B}}, v_B)\} \leq r - 1\}$$

donde d es la distancia usual en el grafo G, $\mathcal{A}(Z)$ es el conjunto de manzanas adyacentes a Z, $\mathcal{E}(Z)$ el conjunto de manzanas exteriores[3] de Z y v_B es el vértice de G que le corresponde a

[3] Definición 24

la manzana B.

En la Figura 12 se muestra un ejemplo del 2-radio de una zona. Vale aclarar que $R_r(Z)$ se calcula sólo al comienzo del procedimiento. A lo largo del algoritmo, a Z sólo pueden anexarse las manzanas que se encuentren en $R_r(Z) \cap \mathcal{A}(Z)$ y solo se le pueden quitar las manzanas en $R_r(Z) \cap \mathcal{E}(Z)$.

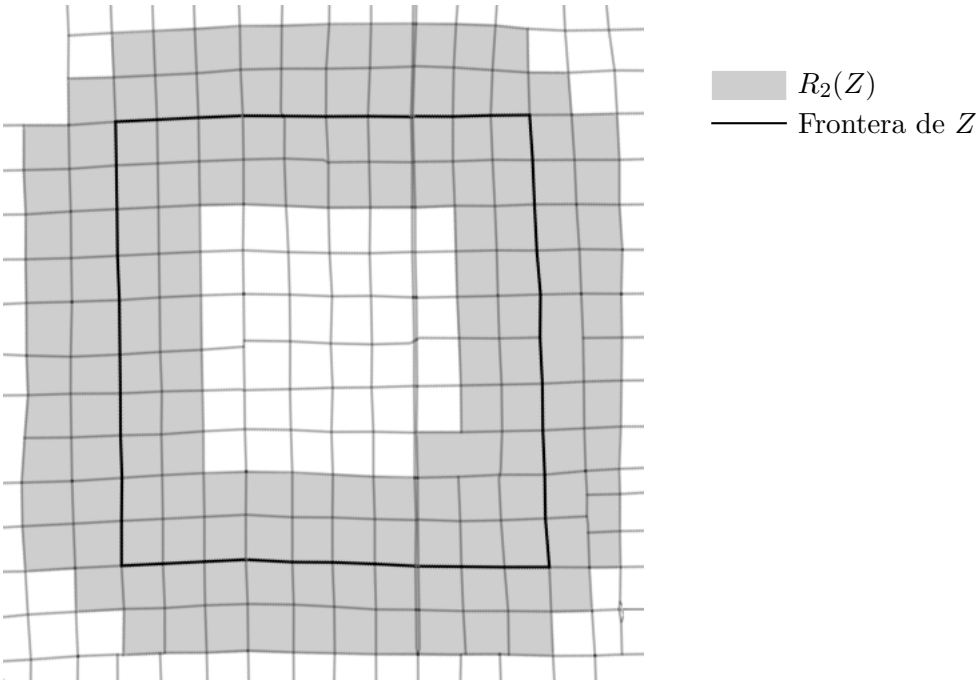


Figura 12: El 2-radio de una zona.

Notación. Sean $\mathcal{Z}$ una zonificación obtenida en la Fase I y $\mathcal{M}$ el conjunto de manzanas del plano, notamos $W : \mathcal{M} \to \mathcal{Z}$ a la función que, para cada $B \in \mathcal{M}$ devuelve la zona de $\mathcal{Z}$ a la que pertenece B. W está bien definida pues la zonificación obtenida en la Fase I es una partición de $\mathcal{M}$.

Notación. Sean X e Y dos zonas y sea $B \in X$, notamos la transferencia de B desde X hacia Y como (X, Y, B).

En el Algoritmo 2 describimos la heurística de la Fase II. Comienza inicializando $R_r(Z)$ para cada zona Z, la lista tabú y el contador de iteraciones (l. 1–4). En cada iteración, para cada zona, registra su cantidad de α-vértices inicial e inicializa el mejor cambio (l. 7–8). A continuación, en las líneas 9–14, para cada B manzana adyacente a la zona dentro del radio aceptable de modificación que no desconecte a la zona vecina, evalúa si agregarla a Z disminuye la cantidad de α-vértices. En caso afirmativo, registra la mejora. Luego remueve a B de Z. En las líneas 15–20 realiza un procedimiento análogo, evaluando si ceder manzanas logra disminuir la mejor cantidad de α-vértices obtenida hasta el momento. Finalmente, se efectúa la transferencia de manzanas, si existiera, que más reduce la cantidad de α-vértices de Z y se la registra en la lista tabú. El procedimiento finaliza al alcanzar el número máximo de iteraciones o cuando durante una iteración no se registraron cambios en ninguna zona.

Algoritmo 2: `RUTINA_FASE_II`

Entrada: $\mathcal{M}$ el conjunto de manzanas del grafo de zonificación
$\mathcal{Z}$ una zonificación
G grafo de manzanas
r parámetro para definir el r-radio
α ángulo para definir los α-vértices
$iter_max$ cantidad máxima de iteraciones

Salida: $\mathcal{Z}$ zonificación

para cada $Z \in \mathcal{Z}$ **hacer**
  $\rho_Z \leftarrow R_r(Z)$ `// Se utiliza` G `para definir` $R_r(Z)$
$T \leftarrow$ inicializar lista tabú vacía de longitud $|\mathcal{Z}|$
$it \leftarrow 1$
mientras $it \leq iter_max$ y al menos cambia una zona **hacer**
  para cada $Z \in \mathcal{Z}$ **hacer**
    $V \leftarrow v_\alpha(Z)$
    $mejor_cambio \leftarrow \emptyset$
    para cada $B \in \rho_Z \cap \mathcal{A}(Z)$ tal que $W(B) - \{B\}$ es conexo y no vacío **hacer**
      $Z \leftarrow Z \cup \{B\}$
      si $v_\alpha(Z) < V$ y $(Z, W(B), B) \notin T$ **entonces**
        $V \leftarrow v_\alpha(Z)$
        $mejor_cambio \leftarrow (W(B), Z, B)$
      $Z \leftarrow Z - \{B\}$
    para cada $B \in \rho_Z \cap \mathcal{E}(Z)$ tal que $Z - \{B\}$ es conexo y no vacío **hacer**
      $Z \leftarrow Z - \{B\}$
      si $v_\alpha(Z) < V$ y $(W(B), Z, B) \notin T$ **entonces**
        $V \leftarrow v_\alpha(Z)$
        $mejor_cambio \leftarrow (Z, W(B), B)$
      $Z \leftarrow Z \cup \{B\}$
    si $mejor_cambio \neq \emptyset$ **entonces**
      Actualizar T agregando $mejor_cambio$
      Efectuar la transferencia de manzanas indicada por $mejor_cambio$
  $it \leftarrow it + 1$
devolver $\mathcal{Z}$

Proposición 6. Sean $S = (V_S, E_S, \omega_s)$ el grafo de zonificación, $\mathcal{M}$ su conjunto de manzanas, G el correspondiente grafo de manzanas y $\mathcal{Z}$ una zonificación de las manzanas de $\mathcal{M}$. Dados $r \in \mathbb{N}, \alpha \in (0, \frac{\pi}{2}], iter_max \in \mathbb{N}$, `RUTINA_FASE_II`$(\mathcal{M}, G, \mathcal{Z}, r, \alpha, iter_max)$ termina y devuelve una zonificación en $|\mathcal{Z}|$ zonas conexas que cubre las mismas manzanas de $\mathcal{M}$ que $\mathcal{Z}$.

Demostración. Las primeras cuatro líneas inicializan los r-radios de las zonas, la lista tabú y el contador de iteraciones. Para probar que el bucle de las líneas 5–24 termina, basta ver que en cada iteración se puede alcanzar la línea 24. De esta manera, se garantiza la terminación del bucle en, a lo sumo, $iter_max$ iteraciones.

El bucle de las líneas 6–23 termina, pues la cantidad de zonas de $\mathcal{Z}$ es finita y es constante. En las primeras dos líneas de este bucle se inicializan la mejor cantidad de α-vértices de Z y el mejor cambio. El bucle de las líneas 9–14 termina porque el conjunto $\rho_Z \cap \mathcal{A}(Z)$ es finito. También es importante notar que la segunda restricción garantiza que $W(B)$ siga siendo conexa y que la transferencia no elimine a $W(B)$ en caso de que conste de una sola manzana. Además, la línea 14 deshace el cambio a Z realizado en la línea 10, por lo que Z es igual a antes de comenzar el bucle. Lo análogo ocurre con el bucle de las líneas 15–20.

Finalmente, en las líneas 21–23, de existir, se efectúa el cambio que más disminuya la cantidad de vértices de Z y se actualiza la lista tabú. Notar que si Z transfiere una manzana a otra zona, esa manzana sigue siendo cubierta por la zonificación resultante de ese cambio. Lo análogo ocurre si a Z se le transfiere una manzana desde otra zona. Además, se garantiza que la transferencia de la manzana no desconecte a Z ni a su manzana vecina, así como también se asegura que ninguna zona será vacía. Por lo tanto, al finalizar cada iteración del bucle 6–23, la zonificación consta de tantas zonas conexas como la zonificación de entrada y cubre las mismas manzanas de $\mathcal{M}$.

Como en cada iteración se puede alcanzar la línea 24, el bucle de las líneas 5–24 termina y la rutina devuelve una zonificación que cubre las mismas manzanas de $\mathcal{M}$ que la zonificación inicial. □

Una vez finalizada la Fase II, ya contamos con una zonificación del plano construida con un método que prioriza la distribución equitativa del metraje y busca que las fronteras de las zonas tengan una forma sencilla de recordar para los choferes.

6. Modificación de una zonificación vigente

Como la resistencia al cambio puede comprometer la implementación de los resultados, desarrollamos un método desde la perspectiva de mejorar la recolección realizando cambios a las zonas vigentes, en vez de elaborar una zonificación completamente nueva. De esta manera, existe la posibilidad de que no todos los equipos de recolección deban modificar su zona y, para quienes deban hacerlo, se espera que los cambios no sean tan drásticos. Por esta razón, buscamos que, en la medida de lo posible, las zonas afectadas conserven una forma similar a la original. Para esto, definimos el lado de una zona utilizando los α-vértices (Definición 32).

Definición 34 (α-lado de una zona)**.** Sean $S = (V_S, E_S, \omega_S)$ el grafo de zonificación, $\alpha \in (0, \frac{\pi}{2}]$ y Z una zona en S, un camino $p = (v_1, \ldots, v_n)$ es un *α-lado* de Z si y solo si $\{\{v_i, v_{i+1}\}\}_{i=1}^{n-1} \subseteq \partial Z$, v_1 y v_n son α-vértices y v_i no es α-vértice para $i = 2, \ldots, n-1$.

Si bien la definición de lado depende del ángulo α elegido, debido a que el valor de ese parámetro no varía a lo largo del procedimiento y no incurrimos en ambigüedad, por simplicidad nos referiremos a los lados de Z en vez de los α-lados de Z. En la Figura 13 mostramos un ejemplo de una zona y sus cuatro lados tomando $\alpha = \frac{1}{2}\pi$. A continuación definimos dos conceptos más que serán claves para describir los procedimientos de modificación a la zonificación.

Definición 35 (Sub-lado de una zona)**.** Sea Z una zona y sea $\ell = (v_1, \ldots, v_n)$ un lado de Z, decimos que s_ℓ es un sublado de ℓ si y solo si s_ℓ es un subcamino de ℓ, es decir, existen $i, k \in \{1, \cdots n\}$, $i < k$ tales que $s_\ell = (v_i, v_{i+1}, \ldots, v_k)$.

Definición 36 (Manzanas adyacentes a un camino)**.** Sean S el grafo de zonificación, $\mathcal{M}$ el conjunto de manzanas de S y $\ell = (v_1, \ldots, v_n)$ un camino en S. Definimos el conjunto de *manzanas adyacentes a ℓ* como:

$$\mathcal{M}_\ell \coloneqq \{B \in \mathcal{M} \colon \exists\, i \in \{1, \ldots, n\} \text{ tal que } \{v_i, v_{i+1}\} \in B\}$$

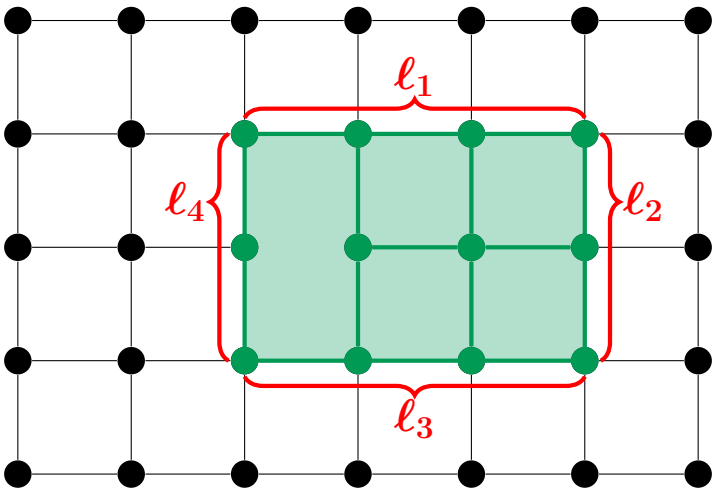


Figura 13: Una zona Z y sus cuatro $\frac{1}{2}\pi$-lados

La modificación de una zona Z consiste en la expansión o contracción por uno de sus lados o sub-lados. Ahora bien, si Z forma parte de una zonificación $\mathcal{Z}$, es deseable que su modificación

de lugar a otra zonificación $\mathcal{Z}'$ cuyas zonas abarquen exactamente las mismas manzanas que las zonas de $\mathcal{Z}$. De esta manera, ninguna manzana pierde el servicio de recolección y no se incluyen manzanas que se deseen excluir (por ejemplo, porque representan terrenos no urbanizados o lindantes con autopistas). Para esto, definimos formalmente las funciones de modificación de $\mathcal{Z}$ según la expansión o la contracción de Z por su (sub-)lado ℓ.

Definición 37 (Zona adyacente por un (sub-)lado)**.** Sea S un grafo de zonificación, $\mathcal{M}$ su conjunto de manzanas, Z una zona en S y sea $\ell = (v_1, \ldots, v_n)$ un (sub-)lado de Z. Diremos que una zona Z' de S es adyacente a Z por ℓ si y solo si existe $i \in \{1, \ldots, n-1\}$ tal que $\{v_i, v_{i+1}\} \in \partial Z'$.

Notación. Sea $\mathcal{Z}$ una zonificación de S, notaremos $ADY_{(\mathcal{Z},Z,\ell)}$ al conjunto de zonas en $\mathcal{Z}$ adyacentes a Z por el (sub-)lado ℓ.

Notación. Sea Z una zona, notaremos $\mathcal{M}_\ell^Z$ a las manzanas en $\mathcal{M}_\ell \cap Z$ que pertenezcan al conjunto de manzanas adyacentes de exactamente una zona en $ADY_{(\mathcal{Z},Z,\ell)}$:

$$\mathcal{M}_\ell^Z = \{B \in \mathcal{M}_\ell \cap Z \colon |\{W \in ADY_{(\mathcal{Z},Z,\ell)} \colon B \in \mathcal{A}(W)\}| = 1\}$$

Observar que, por definición de $\mathcal{M}_\ell^Z$, si $W_1, W_2 \in ADY_{(\mathcal{Z},Z,\ell)}$, con $W_1 \neq W_2$, entonces:

$$(\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2)) = \emptyset \tag{12}$$

Definición 38. Sea S el grafo de zonificación, $\mathcal{M}$ el conjunto de manzanas de S, Z una zona de S, $\mathcal{Z}$ una zonificación, Θ el conjunto de zonificaciones de S y $\Theta_Z \subset \Theta$ el conjunto de zonificaciones de S de las que Z forma parte. Definimos *la modificación mediante la expansión de Z por su (sub-)lado ℓ* como $EXP_{(Z,\ell)} \colon \Theta_Z \to \Theta$ dada por:

$$EXP_{(Z,\ell)}(\mathcal{Z}) = \zeta_1 \cup \zeta_2 \cup \zeta_3$$

donde:

- $\zeta_1 = \mathcal{Z} - \big(\{Z\} \cup ADY_{(\mathcal{Z},Z,\ell)}\big)$ es un conjunto de zonas sin las zonas que son afectadas por la modificación de Z.
- $\zeta_2 = \Big\{Z \cup \Big(\bigcup_{W \in ADY_{(\mathcal{Z},Z,\ell)}} (W \cap M_\ell)\Big)\Big\}$ es el singleton con la expansión de Z, que resulta de agregarle a Z las manzanas en $\mathcal{M}_\ell$ de las zonas adyacentes por ℓ.
- $\zeta_3 = \{W - \mathcal{M}_\ell \colon W \in ADY_{(\mathcal{Z},Z,\ell)}\}$ es el conjunto de las zonas que resultan de quitarles las manzanas de $\mathcal{M}_\ell$ a las zonas adyacentes a Z por ℓ.

Análogamente, definimos *la modificación mediante la contracción de Z por su (sub-)lado ℓ* como $CON_{(Z,\ell)} \colon \Theta_Z \to \Theta$ dada por:

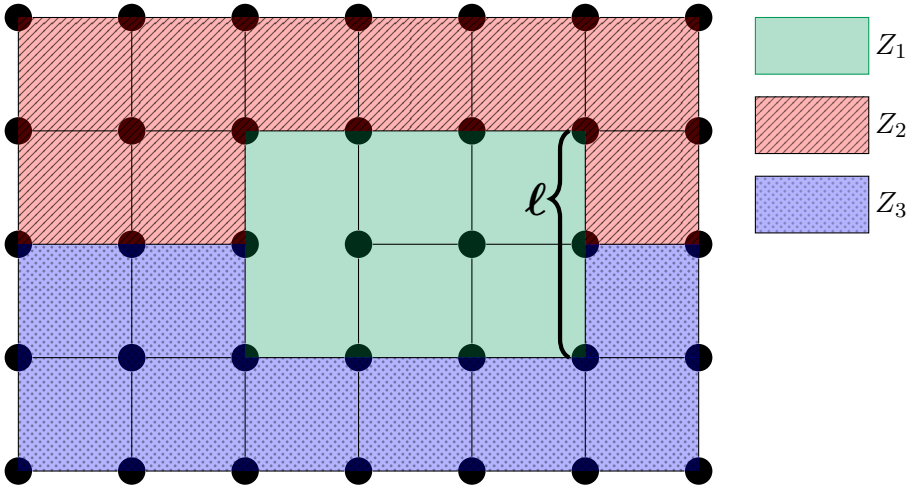
$$CON_{(Z,\ell)}(\mathcal{Z}) = \zeta_1 \cup \zeta_4 \cup \zeta_5$$

donde:

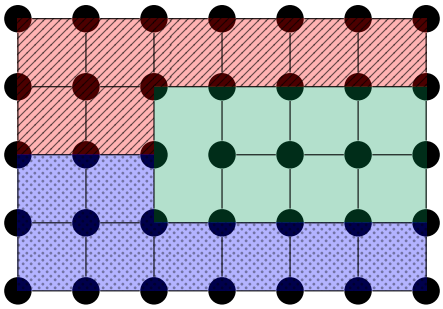
- $\zeta_4 = \Big\{Z - \Big(\bigcup_{W \in ADY_{(\mathcal{Z},Z,\ell)}} (\mathcal{A}(W) \cap \mathcal{M}_\ell^Z)\Big)\Big\}$ es el singleton con la contracción de Z, que resulta de quitarle a Z las manzanas que se encuentran en la intersección $\mathcal{M}_\ell^Z$ con cada una de sus zonas adyacente por ℓ.

- $\zeta_5 = \{W \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W)) \colon W \in ADY_{(\mathcal{Z},Z,\ell)}\}$ es el conjunto de las zonas que resultan de agregarle a cada zona adyacente a Z por ℓ las manzanas adyacentes que se encontraran en $\mathcal{M}_\ell^Z$.

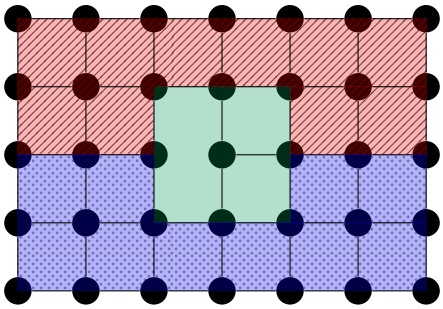
En Figura 14 mostramos un ejemplo de una zonificación y los resultados de aplicar las funciones de modificación.



(a) Una zonificación $\mathcal{Z}$ donde marcamos un lado ℓ de la zona Z_1



(b) $EXP_{(Z_1,\ell)}(\mathcal{Z})$



(c) $CON_{(Z_1,\ell)}(\mathcal{Z})$

Figura 14: Ejemplo de una zonificación (a), su modificación por expansión de Z_1 por su lado ℓ (b) y su modificación por contracción de Z_1 por su lado ℓ (c).

Proposición 7. Sean Z una zona y ℓ uno de sus (sub-)lados, las funciones de modificación $EXP_{(Z,\ell)}$ y $CON_{(Z,\ell)}$ están bien definidas y devuelven una zonificación cuyas zonas abarcan las mismas manzanas que su argumento.

Demostración. Sea $\mathcal{Z}$ una zonificación tal que $Z \in \mathcal{Z}$, supondremos que $ADY_{\mathcal{Z},Z,\ell} \neq \emptyset$, de lo contrario es fácil ver que $EXP_{(Z,\ell)}(\mathcal{Z}) = CON_{(Z,\ell)}(\mathcal{Z}) = \mathcal{Z}$.

Veamos que $\mathcal{Z}' = EXP_{(Z,\ell)}(\mathcal{Z})$ es una zonificación. Para esto, basta ver que todas las manzanas de $\mathcal{Z}'$ son disjuntas dos a dos. Utilizaremos la misma notación de $\zeta_1, \ldots, \zeta_5$ que en la definición.

Observemos que ζ_1 es una zonificación pues $\zeta_1 \subset \mathcal{Z}$. Luego, todas las zonas en ζ_1 son disjuntas dos a dos. Sea X la zona de ζ_2, para $Y \in \zeta_1$ tenemos que:

$$\begin{aligned} Y \cap X &= Y \cap \left(Z \cup \left(\bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} (W \cap \mathcal{M}_\ell)\right)\right) = \\ &= \underbrace{(Y \cap Z)}_{=\emptyset} \cup \left(Y \cap \left(\bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} (W \cap \mathcal{M}_\ell)\right)\right) = \\ &= \bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} (\underbrace{(W \cap \mathcal{M}_\ell) \cap Y)}_{\subseteq W \cap Y = \emptyset} = \emptyset \end{aligned}$$

Por lo tanto, las zonas de $\zeta_1 \cup \zeta_2$ son disjuntas dos a dos. Ahora bien, sean W_1, $W_2 \in ADY_{\mathcal{Z},Z,\ell}$:

$$(W_1 - \mathcal{M}_\ell) \cap (W_2 - \mathcal{M}_\ell) \subseteq W_1 \cap W_2 = \emptyset$$

Luego, todas las zonas de ζ_3 son disjuntas dos a dos. Sean $Y \in \zeta_1$, $W \in ADY_{\mathcal{Z},Z,\ell}$:

$$Y \cap (W - \mathcal{M}_\ell) \subseteq Y \cap W = \emptyset$$

Por ende, las zonas de $\zeta_1 \cup \zeta_3$ son disjuntas dos a dos. Finalmente, observemos que, para $X \in \zeta_2$, $\tilde{W} \in ADY_{\mathcal{Z},Z,\ell}$:

$$\begin{aligned} X \cap (\tilde{W} - \mathcal{M}_\ell) &= \left(Z \cup \left(\bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} (W \cap \mathcal{M}_\ell)\right)\right) \cap (\tilde{W} - \mathcal{M}_\ell) = \\ &= \underbrace{(Z \cap (\tilde{W} - \mathcal{M}_\ell))}_{\subseteq Z \cap \tilde{W} = \emptyset} \cup \left(\bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} (W \cap \mathcal{M}_\ell) \cap (\tilde{W} - \mathcal{M}_\ell)\right) = \\ &= \bigcup_{W \in ADY_{\mathcal{Z},Z,\ell}} \left(\underbrace{(W \cap \mathcal{M}_\ell)}_{\subseteq \mathcal{M}_\ell} \cap (\tilde{W} - \mathcal{M}_\ell)\right) = \emptyset \end{aligned}$$

De esta manera, se ve que las zonas de $\zeta_2 \cup \zeta_3$ son disjuntas dos a dos. Queda demostrado entonces que $\mathcal{Z}'$ es zonificación.

Veamos ahora que $\mathcal{Z}$ y $\mathcal{Z}'$ cubren las mismas manzanas. Para esto, basta ver que, para $B \in \mathcal{M}$, B pertenece a una zona de $\mathcal{Z}$ si y solo si B pertenece a una zona de $\mathcal{Z}'$.

$\Rightarrow$) Si $B \notin Z$ y $B \notin W \,\forall\, W \in ADY_{\mathcal{Z},Z,\ell}$, entonces existe $Y \in \zeta_1$ tal que $B \in Y$. Como $\zeta_1 \subset \mathcal{Z} \cap \mathcal{Z}'$, entonces B pertenece a una zona de $\mathcal{Z}'$. Si $B \in Z$ entonces $B \in X$ para $X \in \zeta_2$. Si $B \in W$ para algún $W \in ADY_{\mathcal{Z},Z,\ell}$:

- $B \in W - \mathcal{M}_\ell$, entonces existe $Y \in \zeta_3$ tal que $B \in Y$.
- $B \in W \cap \mathcal{M}_\ell$, entonces $Y \in X$.

En todos los casos, B pertenece a una zona de $\mathcal{Z}'$.

$\Leftarrow$) Sea $B \in \mathcal{M}$ que pertenece a una zona $Y \in \mathcal{Z}'$. Si $Y \in \zeta_1$, como $\zeta_1 \subset \mathcal{Z} \cap \mathcal{Z}'$, entonces B pertenece a una zona de $\mathcal{Z}$. Si $Y \in \zeta_3$, existe $W \in ADY_{\mathcal{Z},Z,\ell}$ tal que $Y = W - \mathcal{M}_\ell$. En particular, $B \in W \in \mathcal{Z}$. Por último, si $Y \in \zeta_2$, entonces $B \in Z$ o $B \in W$ para algún $W \in ADY_{\mathcal{Z},Z,\ell}$. En ambos casos, B pertenece a una zona de $\mathcal{Z}$.

Queda entonces demostrado que $EXP_{(Z,\ell)}$ está bien definida y que $EXP_{(Z,\ell)}(\mathcal{Z})$ es una zonificación cuyas zonas abarcan las mismas manzanas que $\mathcal{Z}$.

A continuación probamos que $\mathcal{Z}' = CON_{(Z,\ell)}(\mathcal{Z})$ es una zonificación y sus zonas abarcan las mismas manzanas que $\mathcal{Z}$.

Ya hemos probado que las zonas de ζ_1 son disjuntas dos a dos. Veamos que cada $Y \in \zeta_1$ es disjunta con la zona en ζ_4:

$$Y \cap \left(Z - \left(\bigcup_{W \in ADY_{(\mathcal{Z},Z,\ell)}} (\mathcal{A}(W) \cap \mathcal{M}_\ell^Z) \right) \right) \subseteq Y \cap Z = \emptyset$$

A continuación veamos que las zonas de ζ_5 son disjuntas dos a dos. Sean W_1, $W_2 \in ADY_{\mathcal{Z},Z,\ell}$, queremos ver que:

$$(W_1 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1))) \cap (W_2 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) = \emptyset$$

Esto es equivalente a que valga simultáneamente:

(a) $W_1 \cap (W_2 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) = \emptyset$

(b) $(\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap (W_2 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) = \emptyset$

Probemos ambos, comenzando con (a):

$$\begin{aligned} W_1 \cap (W_2 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) &= \underbrace{(W_1 \cap W_2)}_{=\emptyset} \cup (W_1 \cap (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) \subseteq \\ &\subseteq W_1 \cap M_\ell^Z \subseteq W_1 \cap Z = \emptyset \end{aligned}$$

Y siguiendo con (b):

$$\begin{aligned} (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap (W_2 \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2))) = & \quad ((\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap W_2)) \cup \\ & \underbrace{((\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_2)))}_{=\emptyset \text{ por (12)}} = \\ = & \quad (\mathcal{M}_\ell^Z \cap \mathcal{A}(W_1)) \cap W_2) \subseteq \mathcal{M}_\ell^Z \cap W_2 \subseteq \\ \subseteq & \quad Z \cap W_2 = \emptyset \end{aligned}$$

Por lo tanto, las zonas de ζ_5 son disjuntas dos a dos.

Demostremos que cada zona de ζ_5 es disjunta con la zona de ζ_4. Sea $Y \in ADY_{(\mathcal{Z},Z,\ell)}$:

$$\begin{aligned} \Big(Z - \Big(\textstyle\bigcup_{W \in ADY_{(\mathcal{Z},Z,\ell)}} (\mathcal{M}_\ell^Z \cap \mathcal{A}(W))\Big)\Big) \cap (Y \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y))) \subseteq \\ \subseteq \quad (Z - (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y))) \cap (Y \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y))) = \\ = \quad \underbrace{((Z - (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y))) \cap Y)}_{\subseteq Z \cap Y = \emptyset} \cup \underbrace{((Z - (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y))) \cap (\mathcal{M}_\ell^Z \cap \mathcal{A}(Y)))}_{=\emptyset} = \emptyset \end{aligned}$$

Finalmente, corroboramos que las zonas de ζ_1 son disjuntas con las de ζ_5. Sean $Y \in \zeta_1$ y $W \in ADY_{(\mathcal{Z},Z,\ell)}$:

$$Y \cap (W \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W))) = \underbrace{(Y \cap W)}_{=\emptyset} \cup \underbrace{(Y \cap (\mathcal{M}_\ell^Z \cap \mathcal{A}(W)))}_{\subseteq Y \cap Z = \emptyset} = \emptyset$$

Por lo tanto, $\mathcal{Z}' = CON_{(Z,\ell)}(\mathcal{Z})$ es una zonificación. Veamos que cubre las mismas manzanas que $\mathcal{Z}$: $B \in \mathcal{M}$ pertenece a una zona de $\mathcal{Z}$ si y sólo si pertenece a una zona de $\mathcal{Z}'$.

$\Rightarrow$) Si $B \notin Z$ y $B \notin W \,\forall\, W \in ADY_{\mathcal{Z},Z,\ell}$, entonces existe $Y \in \zeta_1$ tal que $B \in Y$. Como $\zeta_1 \subset \mathcal{Z} \cap \mathcal{Z}'$, entonces B pertenece a una zona de $\mathcal{Z}'$. Si $B \in W$ para alguna $W \in ADY_{\mathcal{Z},Z,\ell}$, entonces $B \in W \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W)) \in \zeta_5$. Si $B \in Z$, pueden ocurrir dos situaciones:

(a) $B \in Z - \left(\bigcup_{W \in ADY_{(\mathcal{Z},Z,\ell)}} (\mathcal{M}_\ell^Z \cap \mathcal{A}(W))\right) \in \zeta_4$

(b) $\exists W \in ADY_{\mathcal{Z},Z,\ell}$ tal que $B \in \mathcal{M}_\ell^Z \cap \mathcal{A}(W)$. Entonces, $B \in W \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W)) \in \zeta_5$

Por lo tanto, cada manzana cubierta por $\mathcal{Z}$ está cubierta por $\mathcal{Z}'$.

$\Leftarrow$) Sea B una manzana perteneciente a una zona Y en $\mathcal{Z}'$. Si $Y \in \zeta_1$, como $\zeta_1 \subset \mathcal{Z} \cap \mathcal{Z}'$, entonces B pertenece a una zona de $\mathcal{Z}$. Si $Y \in \zeta_4$, como $Y \subseteq Z$, tenemos que $B \in Z \in \mathcal{Z}$. Por último, si $Y \in \zeta_5$, existe $W \in ADY_{\mathcal{Z},Z,\ell}$ tal que $Y = W \cup (\mathcal{M}_\ell^Z \cap \mathcal{A}(W))$. Por ende, $B \in W$ o $B \in Z$. En ambos casos, pertenece a una zona de $\mathcal{Z}$. □

Luego de definir cómo se modifica una zonificación a través de la expansión o la contracción de una zona, estamos en condiciones de describir los procedimientos iterativos que proponemos para disminuir la disparidad en la distribución del trabajo. Presentamos tres algoritmos distintos, cada uno basado en reconocidas metaheurísticas de optimización: *Steepest Ascent Hill Climbing* [92], *Simulated Annealing* (SA) [92] y *Variable Neighbourhood Search* (VNS) [95].

La idea detrás de los tres métodos es, en cada iteración, expandir la zona con menor carga laboral (Z_{MIN}) y contraer la de mayor carga laboral (Z_{MAX}), de manera tal que la brecha entre ambas disminuya y se logre una distribución del trabajo más equitativa. Al tratarse de algoritmos iterativos, quisiéramos evitar que la modificación realizada en una iteración deshaga el cambio de la iteración anterior. Para esto consideramos el *enfriamiento*, inspirado en la lista tabú. Si una zona es expandida, su enfriamiento de expansión aumenta en un valor *cd*, y significa que, en lo posible, no podrá ser modificada por la expansión de otra zona en las siguientes *cd* iteraciones. Análogo para la contracción. Si la zona a ser modificada tiene a todas sus zonas adyacentes con valores positivos de enfriamiento, se le permitirá modificarse sobre aquella(s) de valor mínimo de enfriamiento. Los contadores de enfriamiento para la expansión y la contracción se inicializan en 0 para todas las zonas.

Es importante notar que suponemos que la zonificación es no vacía, toda zona tiene al menos una zona adyacente y que el grafo de manzanas (Definición 30) es conexo. Por otra parte, sea Φ la función que mide la carga laboral de una zona, utilizaremos como función objetivo F a la diferencia entre la máxima y la mínima carga laboral de las zonas de la zonificación:

$$F(\mathcal{Z}) \coloneqq \max_{W \in \mathcal{Z}} \Phi(W) - \min_{W \in \mathcal{Z}} \Phi(W)$$

F está bien definida para toda zonificación $\mathcal{Z} \neq \emptyset$. Proponemos dos criterios de parada: cantidad máxima de iteraciones y límite de tiempo.

Con el fin de simplificar los pseducódigos que detallan el procedimiento de cada método, en la Tabla 4 describimos la notación de los inputs más comunes.

Notación	Descripción
$\mathcal{Z}$	una zonificación
Z	una zona
Φ	función que mide la carga laboral de una zona
α	el ángulo que define los α-lados de una zona
ℓ	un (sub-)lado
expandir	booleano que indica si se va a expandir una zona o se va a contraer
cooldown	lista de enfriamiento
$cooldown^{(e)}$	lista de enfriamiento de expansión
$cooldown^{(c)}$	lista de enfriamiento de contracción
cd	cantidad por la que se aumenta el enfriamiento de una zona
iter_max	cantidad máxima de iteraciones
tiempo_limite	tiempo límite para el método

Tabla 4: Notación y descripción de las entradas comunes a los métodos de modificación de zonificación.

Funciones en común

Comenzamos con la función `modificar` (Algoritmo 3) que devuelve la zonificación que resulta de expandir o contraer una zona por uno de sus (sub-)lados. La Proposición 7 demuestra la correctitud de este algoritmo.

Algoritmo 3: `modificar`

Entrada: $\mathcal{Z}$, Z, ℓ, *expandir*
Salida: $\mathcal{Z}'$ zonificación modificada

si *expandir* **entonces**
  $\mathcal{Z}' \leftarrow EXP_{(Z,\ell)}(\mathcal{Z})$
en otro caso
  $\mathcal{Z}' \leftarrow CON_{(Z,\ell)}(\mathcal{Z})$
devolver $\mathcal{Z}'$

Podría ocurrir que se quisiera modificar una zona por uno de sus lados, pero que una parte de este sea adyacente a una zona que no tenga valor de enfriamiento mínimo. Resulta conveniente entonces identificar los sub-lados según a qué zona son adyacentes. Para esto utilizamos `sub_lados_adyacencia`, descripto en Algoritmo 4. En la Figura 15 mostramos un ejemplo del resultado del algoritmo al ser aplicado al lado de una zona.

Proposición 8. `sub_lados_adyacencia`$(Z, \ell, ADY_{\mathcal{Z},Z,\ell})$ termina y devuelve un conjunto de sub-lados de ℓ tal que cada uno es adyacente a, a lo sumo, una zona en $ADY_{\mathcal{Z},Z,\ell}$.

Demostración. Como ℓ está conformado por esquinas pertencientes a cuadras de la frontera de Z, entonces cada una de sus cuadras pertenece a la frontera de, a lo sumo, una zona en $ADY_{\mathcal{Z},Z,\ell}$. Por lo tanto, no hay ambigüedad en la definición de las variables `adyacente` y `nueva_adyacente` (líneas 3 y 7).

Algoritmo 4: `sub_lados_adyacencia`

Entrada: Z una zona
$\ell = (v_1, \ldots, v_n)$ un lado de Z
A el conjunto $ADY_{\mathcal{Z},Z,\ell}$
Salida: S conjunto de (sub-)lados de Z

```
1  S ← ∅
2  s ← (v_1, v_2)
3  adyacente ← W ∈ A tal que {v_1, v_2} ∈ ∂W (∅ si no existe tal W)
4  i ← 2
5  mientras i ≤ n − 1 hacer
6  |  s' ← (v_i, v_{i+1})
7  |  nueva_adyacente ← W ∈ A tal que {v_i, v_{i+1}} ∈ ∂W (∅ si no existe tal W)
8  |  si adyacente ≠ nueva_adyacente entonces
9  |  |  S ← S ∪ {s}
10 |  |  s ← s'
11 |  |  adyacente ← nueva_adyacente
12 |  en otro caso
13 |  |  Agregar v_{i+1} al final de s
14 |  i ← i+1
15 devolver S
```

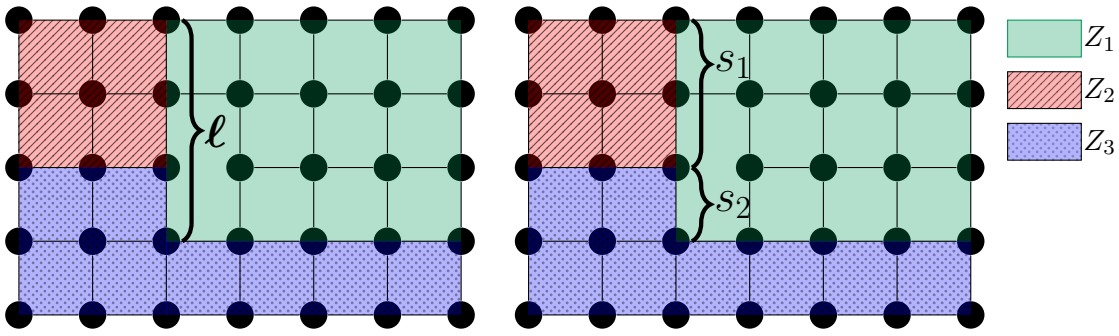


Figura 15: Sub-lados obtenidos luego de aplicar `sub_lados_adyacencia` al lado ℓ de la zona Z_1.

El algoritmo termina, pues la cantidad de iteraciones del único bucle (líneas 5–14) está acotada por la longitud de ℓ. Por otra parte, debido al condicional de la línea 8, si $\{v_i, v_{i+1}\}$ es adyacente a la misma zona que las aristas de s (o si, al igual que s, no es adyacente a ninguna zona), entonces v_{i+1} pasa a formar parte de s. En caso contrario, se añade s al conjunto de sub-lados S y se inicializa un nuevo sub-lado (v_i, v_{i+1}). Por esta razón, al finalizar el procedimiento, dado un sub-lado $s \in S$ no puede ocurrir que sus aristas sean adyacentes a distintas zonas. □

Dada una zona Z queremos quedarnos con los (sub-)lados que limiten con las zonas de enfriamiento mínimo y que la modificación no provoque la eliminación de una zona. Para esto utilizamos la función `filtrar_lados`, descripta en Algoritmo 5

Algoritmo 5: `filtrar_lados`

Entrada: $\mathcal{Z}$, Z, α, *cooldown*, *expandir*
Salida: L' (sub-)lados de Z que lindan con las zonas de enfriamiento mínimo

1 $L \leftarrow$ α-lados de Z
2 $L' \leftarrow \emptyset$
3 `min_cooldown` $\leftarrow \min_{\{W \in \mathcal{Z}:\ W \text{ adyacente a } Z\}} cooldown[W]$
4 **para** $\ell \in L$ **hacer**
5 $\quad S \leftarrow$ `sub_lados_adyacencia`$(Z, \ell, ADY_{\mathcal{Z},Z,\ell})$
6 $\quad S' \leftarrow \{s \in S:\ s \text{ linda con } W \in ADY_{\mathcal{Z},Z,\ell} \text{ tal que } cooldown[W] =$ `min_cooldown`$\}$
7 $\quad$ **si** expandir **entonces**
8 $\quad\quad S' \leftarrow \{s \in S' : |EXP_{Z,s}(\mathcal{Z})| = |\mathcal{Z}|\}$
9 $\quad$ **en otro caso**
10 $\quad\quad S' \leftarrow \{s \in S' : |CON_{Z,s}(\mathcal{Z})| = |\mathcal{Z}|\}$
11 $\quad$ **si** $S = S'$ **entonces**
12 $\quad\quad L' \leftarrow L' \cup \{\ell\}$
13 $\quad$ **en otro caso**
14 $\quad\quad L' \leftarrow L' \cup S'$
15 **devolver** L'

Proposición 9. `filtrar_lados`$(\mathcal{Z}, Z, \alpha, cooldown, expandir)$ termina y devuelve un conjunto de lados y sub-lados de Z que lindan con las zonas de enfriamiento mínimo y tales que modificar Z por cualquiera de ellos según lo que indique *expandir* no reduce la cantidad de zonas.

Demostración. Como suponemos que cada zona en $\mathcal{Z}$ tiene al menos una zona adyacente, entonces la variable `min_cooldown` está bien definida (línea 3). Además, la cantidad de iteraciones del bucle de las líneas 4–14 está acotado por la longitud de L, que es finita. Por lo tanto el bucle termina y también el algoritmo.

En cada iteración, se define S el conjunto de sub-lados de ℓ según adyacencia. En las líneas 6–10 se considera el conjunto S' de los sub-lados que lindan con alguna zona de enfriamiento mínimo y que expandir o contraer (según indique *expandir*) Z por cualquier de ellos no reduzca la cantidad de zonas. Notar que si un sub-lado no linda con ninguna zona, no pertenece a

S'. Finalmente, en las líneas 11–14, si todos los sub-lados de ℓ pertenecen a S', entonces se agrega ℓ al conjunto L'; de lo contrario, se agregan los sublados que pertenezcan a S'.

De esta manera, se garantiza que, al final del bucle, L' sea un conjunto de lados y (sub-)lados que lindan con zonas de enfriamiento mínimo y que modificar Z por cualquier de ellos no elimina una zona. □

Los algoritmos sobre los que basamos nuestros métodos requieren de la definición del vecindario de una solución, la cual presentamos a continuación.

Definición 39 (Vecindario de una zonificación)**.** Sea $\mathcal{Z}$ una zonificación, definimos el vecindario de $\mathcal{Z}$ como las zonificaciones que pueden obtenerse a partir de expandir o contraer una zona de $\mathcal{Z}$ por alguno de sus (sub-)lados:

$$\mathcal{N}(\mathcal{Z}) \coloneqq \{\mathcal{Z}' \colon \exists\, Z \in \mathcal{Z}, \ell \text{ (sub-)lado de } Z \text{ tales que } \mathcal{Z}' = CON_{(Z,\ell)}(\mathcal{Z}) \vee \mathcal{Z}' = EXP_{(Z,\ell)}(\mathcal{Z})\}$$

Finalmente, para los métodos basados en SA y VNS implementamos `vecino_aleatorio` (Algoritmo 6), cuya correctitud se deduce de la de `modificar` y `filtrar_lados`.

Algoritmo 6: `vecino_aleatorio`

Entrada: $\mathcal{Z}$, Z, α, *cooldown*, *expandir*
Salida: $\mathcal{Z}'$ zonificación
1 $L \leftarrow$ `filtrar_lados`$(\mathcal{Z}, Z, \alpha, cooldown, expandir)$
2 $\ell \leftarrow$ elemento de L elegido al azar
3 $\mathcal{Z}' \leftarrow$ `modificar`$(\mathcal{Z}, Z, \ell, expandir)$
4 **devolver** $\mathcal{Z}'$

6.1. Método basado en Hill Climbing

La metaheurística conocida como *Steepest-Ascent Hill Climbing* es un método *greedy* en el que en cada iteración se calcula el valor de la función objetivo de todos los vecinos y se realiza el moviemiento hacia el que presente la mejora más significativa. En el contexto de nuestra aplicación, expande Z_{MIN} por el (sub-)lado que minimice F sobre todos los (sub-)lados de Z_{MIN}. Análogo para la contracción de Z_{MAX}. En primer lugar definimos `mejor_modificacion` (Algoritmo 7) que realiza la mejor modificación, si la hubiere, en términos de disminuir el valor de F. Es importante notar que si bien no aumentamos el valor del enfriamiento para Z, la entrada de *cooldown* es incluida pues es necesaria para las funciones `modificar` y `filtrar_lados`.

Proposición 10. `mejor_modificacion`$(\mathcal{Z}, Z, \alpha, cooldown, expandir)$ termina y devuelve una zonificación $\mathcal{Z}'$ tal que $F(\mathcal{Z}') \leq F(\mathcal{Z})$.

Demostración. El bucle de las líneas 4–9 finaliza, pues su cantidad de iteraciones está acotada por $|L|$. Dentro de él, solo se aceptan modificaciones que mejoran el valor de la función objetivo. Por lo tanto, si existieran tales modificaciones, es claro que $F(\mathcal{Z}') < F(\mathcal{Z})$. Si no existiera ninguna, por línea 2, $F(\mathcal{Z}') = F(\mathcal{Z})$. □

Algoritmo 7: `mejor_modificacion`

Entrada: $\mathcal{Z}$, Z, α, *cooldown*, *expandir*
Salida: zonificación

1 $mejor_gap \leftarrow F(\mathcal{Z})$
2 $mejor_zonificacion \leftarrow \mathcal{Z}$
3 $L \leftarrow$ `filtrar_lados`$(\mathcal{Z}, Z, \alpha, cooldown, expandir)$
4 **para cada** $\ell \in L$ **hacer**
5 $\quad \mathcal{Z}' \leftarrow$ `modificar`$(\mathcal{Z}, Z, \ell, expandir)$
6 $\quad gap \leftarrow F(\mathcal{Z}')$
7 $\quad$ **si** $gap < mejor_gap$ **entonces**
8 $\quad\quad mejor_gap \leftarrow gap$
9 $\quad\quad mejor_zonificacion \leftarrow \mathcal{Z}'$
10 **devolver** $mejor_zonificacion$

Algoritmo 8: `HILL_CLIMBING`

Entrada: $\mathcal{Z}$, α, *iter_max*, *tiempo_limite*
Salida: $\mathcal{Z}$ zonificación modificada

1 $it \leftarrow 1$
2 $t \leftarrow 0$
3 Inicializar $cooldown^{(e)}$ y $cooldown^{(c)}$
4 **mientras** $it \leq iter_max$ **y** $t < tiempo_limite$ **hacer**
5 $\quad Z_{MIN} \leftarrow \arg\min_{W \in \mathcal{Z}} \Phi(W)$
6 $\quad \mathcal{Z} \leftarrow$ `mejor_modificacion`$(\mathcal{Z}, Z_{MIN}, \alpha, cooldown^{(e)}, \texttt{True})$
7 $\quad Z_{MAX} \leftarrow \arg\max_{W \in \mathcal{Z}} \Phi(W)$
8 $\quad \mathcal{Z} \leftarrow$ `mejor_modificacion`$(\mathcal{Z}, Z_{MAX}, \alpha, cooldown^{(c)}, \texttt{False})$
9 $\quad it \leftarrow it + 1$
10 $\quad t \leftarrow$ tiempo trascurrido
11 **devolver** $\mathcal{Z}$

En Algoritmo 8 describimos el método basado en Steepest-Ascent Hill Climbing para obtener una zonificación con mejor valor en la función objetivo que la inicial.

Proposición 11. `HILL_CLIMBING`$(\mathcal{Z}, \alpha, iter_max, tiempo_limite)$ termina y devuelve una zonificación $\mathcal{Z}'$ tal que $F(\mathcal{Z}') \leq F(\mathcal{Z})$.

Demostración. Basta ver que al finalizar el bucle de las líneas 4–10 el valor de la función objetivo decrece o se mantiene igual. Sea F_1 el valor de $F(\mathcal{Z})$ en la línea 5. Sea F_2 el valor de $F(\mathcal{Z})$ luego de aplicar `mejor_modificacion` (línea 6), por Proposición 10, $F_2 \leq F_1$. Análogamente, sea F_3 el valor de $F(\mathcal{Z})$ luego de aplicar `mejor_modificacion` (línea 8), $F_3 \leq F_2$. Por lo tanto, al finalizar el bucle, $F(\mathcal{Z}) \leq F_1$. □

La principal desventaja del Steepest-Ascent Hill Climbing es que puede converger rápidamente a un mínimo local. Por lo tanto, proponemos otros dos métodos para intentar sortear

este obstáculo.

6.2. Método basado en Simulated Annealing

Simulated Annealing (SA) es una metaheurística de búsqueda local que, además de aceptar movimientos que mejoren el valor de la función objetivo, introduce la probabilidad de aceptar aquellos que la empeoren y, de esta manera, escapar de los mínimos locales. Dicha probabilidad disminuye a medida que aumentan las iteraciones. Así, al comienzo, SA se asemeja más a un paseo aleatorio y, luego de un determinado número de iteraciones, termina pareciéndose a un algoritmo de tipo Hill Climbing, donde solo se aceptan movimientos beneficiosos.

Como su nombre lo indica, SA fue motivado por el procedimiento de recocido de metales [83, 132], que comienza a una alta temperatura y que se enfría paulatinamente para que el material adquiera distintas propiedas. Sea T la temperatura correspondiente a cierta iteración del método, la probabilidad de aceptar un movimiento en un problema de minimización viene dado por:

$$e^{\frac{-\Delta F}{T}}$$

donde $\Delta F = F(nueva_solucion) - F(solucion_actual)$. Cada cierto número de iteraciones, T disminuye su valor y, por lo tanto, también la probabilidad de aceptar un movimiento que empeore el valor de F.

En el Algoritmo 9 mostramos el procedimiento por el cual se modifica $\mathcal{Z}$ por un (sub-)lado al azar y se decide si aceptar la modificación según ΔF y el valor de T. En este caso, la zonificación obtenida no necesariamente tiene menor valor en F que la de entrada. La correctitud del algoritmo se deduce de `vecino_aleatorio`.

Algoritmo 9: `modificacion_aleatoria`

Entrada: $\mathcal{Z}$, Z, α, *cooldown*, *expandir*, *cd*,T
Salida: $\mathcal{Z}'$ zonificación

$mejor_gap \leftarrow F(\mathcal{Z})$
$\mathcal{Z}' \leftarrow$ `vecino_aleatorio`$(\mathcal{Z}, Z, \alpha, cooldown, expandir)$
$gap \leftarrow F(\mathcal{Z}')$
$p \leftarrow$ número aleatorio en $(0, 1)$
si $p < \exp\left(-\dfrac{gap - mejor_gap}{T}\right)$ **entonces**
    Aumentar enfriamiento de Z en $cd + 1$ en *cooldown*
    devolver $\mathcal{Z}'$
en otro caso
    devolver $\mathcal{Z}$

Para la aplicación del SA a nuestro problema, utilizamos un esquema exponencial para disminuir la temperatura y lo hacemos en cada iteración. Sea T_0 la temperatura inicial, la temperatura en la iteración k viene dada por:

$$T_k = T_{k-1}^{\delta}$$

donde $\delta \in (0, 1)$ es el parámetro que regula la disminución de T. En Algoritmo 10 describimos el procedimiento del método.

Algoritmo 10: `SIMULATED_ANNEALING`

Entrada: $\mathcal{Z}$, α , T, δ, *iter_max tiempo_limite*
Salida: $\mathcal{Z}$ zonificación

1 $it \leftarrow 1$
2 $t \leftarrow 0$
3 $mejor_solucion \leftarrow \mathcal{Z}$
4 $mejor_gap \leftarrow F(\mathcal{Z})$
5 Inicializar $cooldown^{(e)}$ y $cooldown^{(c)}$
6 **mientras** $it \leq iter_max$ **y** $t \leq tiempo_limite$ **hacer**
7 $\quad Z_{MIN} \leftarrow \arg\min_{W \in \mathcal{Z}} \Phi(W)$
8 $\quad \mathcal{Z} \leftarrow$ `modificacion_aleatoria`$(\mathcal{Z}, Z_{MIN}, \alpha, cooldown^{(e)}, \texttt{True}, T)$
9 $\quad$ **si** $F(\mathcal{Z})$ <mejor_gap **entonces**
10 $\quad\quad mejor_solucion \leftarrow \mathcal{Z}$
11 $\quad\quad mejor_gap \leftarrow F(\mathcal{Z})$
12 $\quad Z_{MAX} \leftarrow \arg\max_{W \in \mathcal{Z}} \Phi(W)$
13 $\quad \mathcal{Z} \leftarrow$ `modificacion_aleatoria`$(\mathcal{Z}, Z_{MAX}, \alpha, cooldown^{(c)}, \texttt{False}, T)$
14 $\quad$ **si** $F(\mathcal{Z})$ <mejor_gap **entonces**
15 $\quad\quad mejor_solucion \leftarrow \mathcal{Z}$
16 $\quad\quad mejor_gap \leftarrow F(\mathcal{Z})$
17 $\quad it \leftarrow it + 1$
18 $\quad t \leftarrow$ tiempo trascurrido
19 $\quad T \leftarrow T^{\delta}$
20 $\quad$ Disminuir en $cooldown^{(e)}$ y $cooldown^{(c)}$ el enfriamiento en 1 para todas las zonas
21 **devolver** *mejor_solucion*

Proposición 12. `SIMULATED_ANNEALING`$(\mathcal{Z}, \alpha, T, \delta, iter_max, tiempo_limite)$ termina y devuelve $\mathcal{Z}'$ una zonificación tal que $F(\mathcal{Z}') \leq F(\mathcal{Z})$.

Demostración. El algoritmo termina, pues la cantidad de iteraciones del bucle de las líneas 6–20 está acotado por *iter_max*.

En la línea 3 la mejor solución es inicializada como la zonificación inicial $\mathcal{Z}$. En el bucle de las líneas 6–20 la mejor solución sólo es actualizada si alguna modificación mejora el valor de F. Si la mejor solución no fue actualizada luego de `iter_max` iteraciones, $F(\mathcal{Z}') = F(\mathcal{Z})$. Si la mejor solución fue actualizada al menos una vez, $F(\mathcal{Z}') < F(\mathcal{Z})$. En todos los casos, $F(\mathcal{Z}') \leq F(\mathcal{Z})$. □

La ventaja de incluir un aspecto estocástico es tener la posibilidad de llegar a una mejor solución que a la del mínimo local más cercano. Sin embargo, su desempeño depende de la elección de los parámetros T y δ.

6.3. Método basado en Variable Neighbourhood Search

Variable Neighbourhood Search (VNS) es una metaheurística propuesta por Mladenović y Hansen [95] y su idea general consiste en cambiar el vecindario tanto en la fase de descenso en la que se busca un óptimo local como en la fase de perturbación para salir del valle en el que se encuentra dicho óptimo. Hay varios esquemas de VNS y nuestra adaptación se basa en el *Basic Variable Neighbourhood Search* (BVNS) [64]. Desde una solución s, se toma una solución al azar en su vecindario y se le aplica un algoritmo de búsqueda local. Si el resultado es una solución con mejor valor en la función objetivo que la de s, se mueve a esa solución y repite el procedimiento. En caso contrario, toma una solución del 2-vecindario de s (i.e. un vecino de un vecino) y le aplica un algoritmo de búsqueda local. Si luego de aplicar búsqueda local a tomar una solución en el k_{MAX}-vecindario de s no encuentra una mejor solución, vuelve a tomar una al azar en el vecindario de s. El criterio de parada es el límite de tiempo.

En el Algoritmo 11 describimos el procedimiento de búsqueda local. A diferencia del utilizado para el método basado en Steepest-Ascent Hill Climbing, no recorre todo el vecindario, sino que la iteración termina cuando encuentra una solución en el vecindario con mejor valor en la función objetivo. Sin embargo, la demostración de que devuelve una zonificación con menor o igual valor en F es análoga a la de la Proposición 10.

Por su parte, en Algoritmo 12 describimos el procedimiento inspirado en VNS que se aplicará a Z_{MIN} y a Z_{MAX} en cada iteración.

Proposición 13. `modificacion_vns`$(\mathcal{Z}, Z, \alpha, cooldown, expandir, k_{MAX}, iter_max)$ termina y devuelve una zonificación $\mathcal{Z}'$ tal que $F(\mathcal{Z}') \leq F(\mathcal{Z})$.

Demostración. Puesto que la cantidad de iteraciones del bucle 5–15 está acotada por k_{MAX} y `busqueda_local` termina, el bucle de las líneas 5–15 termina, y, por ende, el algoritmo también.

Veamos ahora que $F(\mathcal{Z}') \leq F(\mathcal{Z})$. En cada iteración del bucle, se lleva a cabo una búsqueda local a partir de una zonificación en el k-vecindario de $\mathcal{Z}$ (línea 8). Si el resultado de la búsqueda local brinda una zonificación $\mathcal{Z}'$ con menor valor de F, el bucle termina (líneas 10–12). Por lo tanto, al finalizar el algoritmo, $F(\mathcal{Z}') < F(\mathcal{Z})$. Si el resultado de la búsqueda local no brinda una zonificación con mejor valor en la función objetivo, se aumenta el valor de k (líneas 13–15). Si se agotan las k_{MAX} iteraciones sin hallar una mejor zonificación que $\mathcal{Z}$, por línea 15, al terminar el algoritmo $\mathcal{Z}' = \mathcal{Z}$ y vale que $F(\mathcal{Z}') = F(\mathcal{Z})$. En cualquier caso, el output de `modificacion_vns` es una zonificación $\mathcal{Z}'$ tal que $F(\mathcal{Z}') \leq F(\mathcal{Z})$. □

Finalmente, en el Algoritmo 13 describimos el procedimiento completo del método inspirado en VNS. Observar que básicamente sólo difiere del procedimiento de Hill Climbing (Algoritmo 8) en la función de modificación, por lo tanto, la demostración de su correctitud es análoga.

Es importante notar que los tres métodos iterativos devuelven una zonificación que tiene la misma cantidad de zonas que la zonificación inicial. Esto está garantizado por el uso de `filtrar_lados` para descartar las modificaciones que eliminarían una zona. Por otra parte, la definición de las funciones de expansión y contracción garantizan que la zonificación devuelta por cualquier de los tres métodos cubre las mismas manzanas que la zonificación incial.

Algoritmo 11: `busqueda_local`

Entrada: $\mathcal{Z}$, α, *cooldown*, *expandir*, *iter_max*
Salida: $\mathcal{Z}'$ zonificación

$mejor_gap \leftarrow F(\mathcal{Z})$
$it \leftarrow 1$
$\mathcal{Z}' \leftarrow \mathcal{Z}$
mientras $it \leq iter_max$ **hacer**
  si *expandir* **entonces**
    $Z \leftarrow \arg\min_{W \in \mathcal{Z}'} \Phi(W)$
  en otro caso
    $Z \leftarrow \arg\max_{W \in \mathcal{Z}'} \Phi(W)$
  $L \leftarrow$ `filtrar_lados`$(\mathcal{Z}', Z, \alpha, cooldown, expandir)$
  $mejora_gap \leftarrow$ `False`
  mientras $L \neq \emptyset$ **y no** $mejora_gap$ **hacer**
    $\mathcal{Z}'' \leftarrow \mathcal{Z}'$
    $\ell \leftarrow$ un elemento de L
    $L \leftarrow L - \{\ell\}$
    $\mathcal{Z}'' \leftarrow$ `modificar`$(\mathcal{Z}', Z, \ell, expandir)$
    $gap \leftarrow F(\mathcal{Z}'')$
    si $gap < mejor_gap$ **entonces**
      $mejora_gap \leftarrow$ `True`
      $\mathcal{Z}' \leftarrow \mathcal{Z}''$
  si no $mejora_gap$ **entonces**
    $it \leftarrow iter_max$
  $it \leftarrow it + 1$
devolver $\mathcal{Z}'$

Algoritmo 12: `modificacion_vns`

Entrada: $\mathcal{Z}$, Z, α, *cooldown*, *expandir*, k_{MAX}, *iter_max*
Salida: $\mathcal{Z}'$ zonificación

$gap_inicial \leftarrow F(\mathcal{Z})$
$mejora_gap \leftarrow$ `False`
$k \leftarrow 1$
$\mathcal{Z}' \leftarrow \mathcal{Z}$
mientras $k \leq k_{MAX}$ **y no** $mejora_gap$ **hacer**
  para cada $i \in \{1, \ldots, k\}$ **hacer**
    $\mathcal{Z}' \leftarrow$ `vecino_aleatorio`$(\mathcal{Z}', Z, \alpha, cooldown, expandir)$
  $\mathcal{Z}' \leftarrow$ `busqueda_local`$(\mathcal{Z}', \alpha, cooldown, expandir, iter_max)$
  $gap \leftarrow F(\mathcal{Z}')$
  si $gap < gap_inicial$ **entonces**
    $mejora_gap \leftarrow$ `True`
    Aumentar enfriamiento de Z en $cd + 1$ en *cooldown*
  en otro caso
    $k \leftarrow k + 1$
    $\mathcal{Z}' \leftarrow \mathcal{Z}$
devolver $\mathcal{Z}'$

Algoritmo 13: `VARIABLE_NEIGHBOURHOOD_SEARCH`

Entrada: $\mathcal{Z}$, Z, α, *iter_max*, *tiempo_limite*, *iter_max_bl*, *k_max*
Salida: $\mathcal{Z}$ zonificación

$it \leftarrow 1$
$t \leftarrow 0$
Inicializar $cooldown^{(e)}$ y $cooldown^{(c)}$
mientras $it \leq iter_max$ **y** $t \leq tiempo_limite$ **hacer**
  $Z_{MIN} \leftarrow \arg\min_{W \in \mathcal{Z}} \Phi(W)$
  $\mathcal{Z} \leftarrow$ `modificacion_vns`$(\mathcal{Z}, Z_{MIN}, \alpha, cooldown^{(e)}, \texttt{True}, iter_max_bl, k_max)$
  $Z_{MAX} \leftarrow \arg\max_{W \in \mathcal{Z}} \Phi(W)$
  $\mathcal{Z} \leftarrow$ `modificacion_vns`$(\mathcal{Z}, Z_{MAX}, \alpha, cooldown^{(c)}, \texttt{False}, iter_max_bl, k_max)$
  $it \leftarrow it + 1$
  $t \leftarrow$ tiempo trascurrido
  Disminuir en $cooldown^{(e)}$ y $cooldown^{(c)}$ el enfriamiento en 1 para todas las zonas
devolver $\mathcal{Z}$

7. Optimización de las rutas

En esta sección describiremos nuestro enfoque de solución para hallar rutas de recolección eficientes. Comenzamos con la definición formal del Asymmetric Traveling Salesman Problem y del Mixed General Routing Problem with Turn Penalties.

Definición 40 (Asymmetric Travelling Salesman Problem [110])**.** Sea $G = (V, A, \omega)$ un digrafo pesado completo con $\omega\colon A \to \mathbb{R}_{\geq 0}$, el *Asymmetric Travelling Salesman Problem* (ATSP) se trata de encontrar un ciclo $C = (v_1, \ldots, v_n)$ en G que visite todos los vértices de V y minimice $\sum_{i=1}^{n} \omega(v_i, v_{i+1})$

Definición 41 (Giro [115])**.** Sea $G = (V, A, E, \omega)$ un grafo mixto pesado, decimos que para un par $a, b \in A \cup E$ hay un giro de a hacia b si y solo si los extremos de a son $u, v \in V$, los extremos de b son $v, w \in V$ y existe el camino (u, v, w) en G.

Definición 42 (Mixed General Routing Problem with Turn Penalties [115])**.** Sea $G = (V, A, E, \omega)$ un grafo mixto pesado con $\omega\colon A \cup E \to \mathbb{R}_{\geq 0}$ y en el cual cada giro tiene asociada una penalidad (definida como $+\infty$ para giros prohibidos). Dados $E_R \subseteq E$, $A_R \subseteq A$ y $V_R \subseteq V$ el *Mixed General Routing Problem with Turn Penalties* (MGRPTP) se trata de encontrar un recorrido de costo mínimo que pase por todas las aristas de E_R, todos los arcos de A_R y todos los vértices de V_R al menos una vez.

Recordemos que debido a la técnica del achique, buscar una ruta óptima para una zona Z se puede modelar como una instancia del MGRPTP en P, el grafo mixto pesado del plano[4]. El trayecto debe visitar al menos una vez todos los vértices correspondientes a las esquinas de las cuadras de Z y recorrer al menos una vez las cuadras de Z cuya longitud sea mayor al umbral de achique. Nuestro enfoque de solución supone que la cantidad de residuos de ninguna zona supera la capacidad de los camiones. Por lo tanto, ignoramos el factor de capacidad al determinar la ruta eficiente de cada zona.

Como mencionamos en la revisión de la literatura, Soler et al. [115] proponen una transformación del grafo mixto a un digrafo completo, reduciendo el MGRPTP al ATSP. Sin embargo, este enfoque de solución no puede aplicarse a casos con giros prohibidos de más de tres vértices. Además, su experimentación computacional fue realizada sobre instancias donde la cantidad de aristas y arcos requeridos es mucho mayor que la de vértices, al contrario de lo que demandan nuestros casos de estudio. Como consecuencia, la cantidad de vértices del ATSP resultante de la transformación podría llegar a ser de un orden de magnitud mayor que el número de esquinas de la zona. Por otra parte, el procedimiento implicaría el cálculo de caminos mínimos sobre un grafo mixto cuyo tamaño es dos órdenes de magnitud mayor al de los utilizados en la experimentación computacional del citado trabajo. Por estas razones, proponemos una estrategia para hallar soluciones eficientes del MGRPTP que está basada en resolver el ATSP con restricciones adicionales. Las principales ventajas de nuestro enfoque sobre el desarrollado por Soler at al. radican en que la cantidad de vértices del ATSP es igual al número de esquinas de la zona, la aplicabilidad a casos donde hay giros prohibidos con tres o cuatro vértices y la búsqueda de caminos mínimos en grafos significativamente más pequeños que P. Si bien nuestro procedimiento no garantiza optimalidad, permite obtener rutas que disminuyen significativamente la distancia recorrida y organizar el achique.

[4]Definición 12

Recordemos que F es el conjunto de caminos prohibidos que representan violaciones de normas de tránsito y que cada uno de ellos consta de 3 o 4 vértices. De aquí en más notaremos con F_3 al conjunto de caminos prohibidos de 3 vértices y con F_4 al conjunto de caminos prohibidos con 4 vértices. Como el procedimiento para calcular caminos mínimos sin caminos prohibidos (s.c.p.) depende de F_4, desarrollamos dos metodologías para hallar rutas eficientes. La primera fue motivada por el caso de estudio de Berazategui, donde $F_4 \neq \emptyset$. En este caso consideramos que todos los giros tienen penalidad 0, salvo los giros prohibidos. Nos referiremos a esta metodología como *Enfoque* $F_4 \neq \emptyset$. Para los casos en los que $F_4 = \emptyset$, posterioremente desarrollamos el *Enfoque* $F_4 = \emptyset$. Esta estrategia aprovecha la ausencia de caminos prohibidos de 4 vértices para considerar penalidades no nulas para giros permitidos y para calcular los caminos mínimos s.c.p. más eficientemente. Además, distingue entre la distancia recorrida fuera y dentro de la zona de recolección para reflejar mejor el impacto de cada una en la carga laboral del equipo de recolectores.

A pesar de las diferencias entre ambos enfoques, la idea general detrás de ellos, que describimos en Algoritmo 14, es obtener una ruta eficiente a través de la concatenación de caminos mínimos entre las esquinas de la zona. En general, en esta sección, notaremos con s al vértice correspondiente a la esquina de partida de los camiones. Podría ser, por ejemplo, la esquina más próxima a la salida del Predio[5]. Asimismo, notaremos con t al vértice correspondiente a la esquina de llegada de los camiones. Podría representar, por ejemplo, la esquina más cercana al ingreso al Predio o la esquina más cercana al ingreso a la planta de tratamiento. Supondremos que $s \neq t$ y, si s y/o t son esquinas de una zona Z, las excluiremos del conjunto de esquinas de Z, al cual notamos V_Z.

Algoritmo 14: Método para hallar rutas eficientes

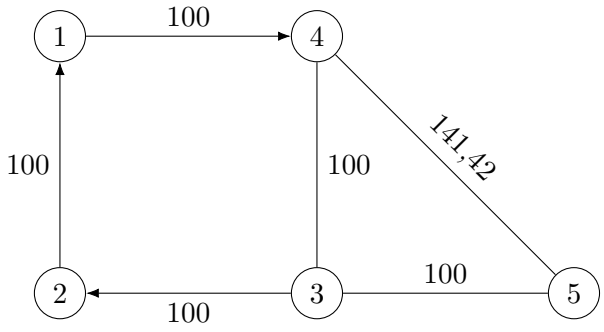
Entrada: R digrafo de ruteo correspondiente a P (Definición 27)
s vértice de partida de los camiones ($s \in V_R$)
t vértice de llegada de los camiones ($t \in V_R$)
F conjunto de caminos prohibidos
$\mathcal{Z}$ una zonificación

Salida: Una ruta eficiente para cada zona en $\mathcal{Z}$

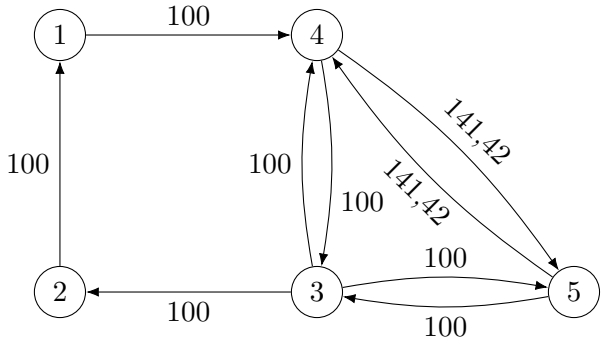
1 Calcular caminos mínimos sin caminos prohibidos (s.c.p.) desde s a todos los vértices de R y desde todos los vértices de R hasta t `// Se utiliza` F
2 **para cada** $Z \in \mathcal{Z}$ **hacer**
3 Definir $V_Z \subset V_R$ como los vértices que pertenecen a la zona Z
4 Calcular un camino mínimo s.c.p. entre cada par de vértices de V_Z `// Se utiliza` F
5 Elaborar un digrafo completo K cuyos vértices son $V_Z \cup \{s, t\}$ y el peso del arco (i, j) viene dado por la distancia del camino mínimo s.c.p. entre i y j
6 Resolver sobre K una instancia del ATSP con restricciones adicionales mediante un modelo MILP
7 Procesar la solución del modelo MILP para obtener una ruta eficiente en R que es factible para el MGRPTP
8 **devolver** conjunto de rutas eficientes para cada $Z \in \mathcal{Z}$

[5]Denominaremos *el Predio* al lugar donde se estacionan los camiones fuera del horario de recolección.

Veamos un ejemplo sencillo de la aplicación del método. En la Figura 16 mostramos el grafo mixto de un plano (Figura 16a) donde la zona Z está compuesta por la manzana formada por las aristas $\{3,4\},\{3,5\},\{5,4\}$. El vértice de partida de los camiones es 1 ($s=1$) y el de llegada es 2 ($t=2$). Los giros en U están prohibidos así como también el camino $(4,5,3)$. La arista $\{4,5\}$ debe formar parte del recorrido, pues supera el umbral de achique. La penalidad por giro es $p=0$. El digrafo de ruteo se muestra en la Figura 16b.



(a) P grafo mixto de un plano



(b) R digrafo de ruteo de P

Figura 16: Un grafo mixto de un plano y su correspondiente digrafo de ruteo.

Definiendo $V_Z=\{3,4,5\}$ y una vez calculados los caminos mínimos s.c.p. desde s a todo $v\in V_R$, desde todo $v\in V_R$ a t y entre cada par de vértices en V_Z, confeccionamos el digrafo completo K de la Figura 17. Determinamos que el peso de la arista $(2,1)$ sea 0, pues el recorrido termina en 2 y no tiene sentido sumar la distancia de 2 a 1. Si resolvemos el ATSP sobre K con s como vértice inicial, obtenemos una secuencia de vértices $(v_1,\ldots,v_5)$ de costo mínimo, con $v_1=s$. Siguiendo esta secuencia, podemos concatenar caminos mínimos s.c.p. para construir una ruta en P que pase al menos una vez por cada vértice de V_Z. Sin embargo, la solución del ATSP no garantiza que en algún momento se recorra la arista $\{4,5\}$, que en la concatenación de caminos mínimos s.c.p. no aparezcan caminos prohibidos ni que t sea el último vértice visitado antes de regresar a s. Por esta razón, agregamos restricciones al ATSP para asegurar que a partir de una solución se pueda construir una ruta en P que sea factible para el MGRPTP.

En nuestro ejemplo, la solución óptima para el ATSP con restricciones adicionales da lugar a la secuencia $(1,3,5,4,2)$. Construimos la ruta concatenando los caminos mínimos entre cada

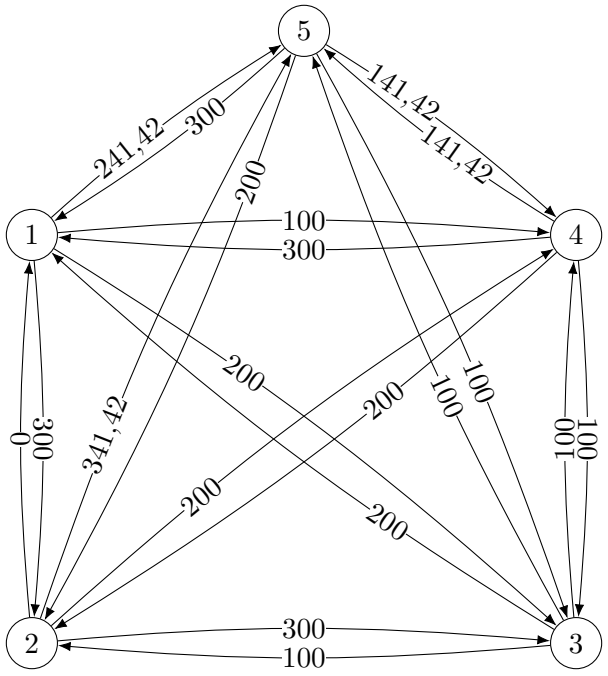


Figura 17: Digrafo pesado completo K, donde el peso de (u, v) viene dado por el costo del camino mínimo s.c.p. desde u hasta v en R.

par consecutivo de elementos de la secuencia y obtenemos:

$$(\mathbf{1}, 4, \mathbf{3}, \mathbf{5}, \mathbf{4}, 3, \mathbf{2})$$

que representa un recorrido en P que parte del vértice 1, visita todos los vértices de V_Z, recorre la arista $\{4, 5\}$ y termina en 2. El Enfoque $F_4 \neq \emptyset$ fue el que utilizamos para el caso de estudio de Berazategui y el Enfoque $F_4 = \emptyset$, para Quilmes. A continuación detallamos cada uno de ellos.

7.1. Enfoque $F_4 \neq \emptyset$

Caminos mínimos

Para calcular el camino mínimo s.c.p. entre dos vértices de un digrafo no alcanza con utilizar los algoritmos convencionales para hallar caminos mínimos, pues no necesariamente hay subestructura óptima. Ver, por ejemplo, la Figura 18. Si $(1, 2, 3)$ es un camino prohibido, el camino mínimo s.c.p. desde 1 hasta 3 es $(1, 2, 4, 5, 3)$. Por su parte, $(2, 4, 5, 3)$ es un camino s.c.p. desde 2 hasta 3, pero no es mínimo.

En este enfoque, para calcular los caminos mínimos s.c.p. entre vértices de un digrafo R utilizamos el algoritmo desarrollado por Di Puglia Pugliese y Guerreiro [36]. Se trata de un algoritmo basado en programación dinámica que permite hallar el camino mínimo (no necesariamente simple) desde un vértice fuente ζ al resto de los vértices del grafo sin transitar completamente por ninguno de los caminos prohibidos de F. Sean $G = (V, A)$ un digrafo, $k = \sum_{j \in V - \{\zeta\}} k_j$ donde k_j es la cantidad de caminos que existen en G que comienzan en ζ y

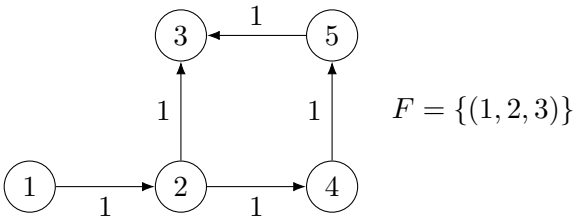


Figura 18: Con caminos prohibidos, la subestructura óptima no está garantizada.

terminan en j, los autores demuestran que el algoritmo tiene complejidad $\mathcal{O}(k^2|A||F|)$. Nos referiremos a este algoritmo como dPPG.

Notación. Notaremos como γ_{vw} a un camino mínimo s.c.p. entre v y w en R. Notaremos como su longitud de recorrido a:

$$\omega(\gamma) = \sum_{k=0}^{N-1} \omega_R(v_k, v_{k+1})$$

Estamos en condiciones de plantear cómo se resuelve la primera línea del Algoritmo 14:

1 Calcular caminos mínimos sin caminos prohibidos (s.c.p.) desde s a todos los vértices de R y desde todos los vértices de R hasta t

Primero, aplicamos dPPG a R con s como vértice fuente y F como conjunto de caminos prohibidos. Luego, aplicamos dPPG al digrafo transpuesto de R con t como vértice fuente y F' el conjunto de caminos con secuencia inversa a los de F. De esta manera, como demostramos a continuación, al invertir los caminos devueltos por el algoritmo se obtienen los caminos mínimos s.c.p. desde cada $v \in V_R - \{t\}$ hasta t en R.

Proposición 14. Sean $R = (V_R, A_R, \omega_R)$ el digrafo de ruteo, $R^T = (V_R, C, \varrho)$ su digrafo traspuesto, F el conjunto de caminos prohibidos y F' el conjunto de caminos en F con sentido inverso (i.e. $(v_1, v_2, \ldots, v_N) \in F'$ si y solo si $(v_N, \ldots, v_2, v_1) \in F$). Sean $u, v \in V_R$, si $\gamma = (v_1, \ldots, v_N)$ es un camino mínimo s.c.p. en F' desde u a v en R^T ($v_1 = u$, $v_N = v$), entonces $\overline{\gamma} = (v_N, \ldots, v_1)$ es un camino mínimo s.c.p. en F de v a u en R.

Demostración. Supongamos que $\overline{\gamma}$ tiene un camino en F como subcamino, es decir, que existen $i \in \{2, \ldots, N\}$, $k \in \{1, \ldots, i-1\}$ tales que $(v_i, v_{i-1}, \ldots, v_{i-k}) \in F$. Entonces $(v_{i-k}, \ldots, v_{i-1}, v_i) \in F'$, lo cual es absurdo pues γ no tiene a ningún camino en F' como subcamino. Luego, $\overline{\gamma}$ no tiene un camino en F como subcamino.

Supongamos ahora que existe $\overline{\eta}$ un camino mínimo sin caminos en F de v a u en R tal que $\omega(\overline{\eta}) < \omega(\overline{\gamma})$. Por definición, en R^T existe η el camino que une u con v en R^T tal que η es la secuencia en orden inverso de $\overline{\eta}$ y, por lo demostrado anteriormente, no tiene caminos en F' como subcaminos. Entonces vale que:

$$\omega(\overline{\gamma}) = \varrho(\gamma) \leq \varrho(\eta) = \omega(\overline{\eta}) < \omega(\overline{\gamma}) \quad \text{Abs!}$$

El absurdo proviene de suponer que $\overline{\gamma}$ no es un camino de distancia de recorrido mínima sin caminos en F como subcaminos. □

Una vez calculados los caminos mínimos desde s y hacia t, comienza el procedimiento sobre cada zona Z.

Construcción del grafo completo K

Veamos ahora cómo se resuelven los siguientes pasos del Algoritmo 14 para cada $Z \in \mathcal{Z}$:

3 Definir $V_Z \subset V_R$ como los vértices que pertenecen a la zona Z
4 Calcular un camino mínimo s.c.p. entre cada par de vértices de V_Z
5 Elaborar un digrafo completo K cuyos vértices son $V_Z \cup \{s, t\}$ y el peso del arco (i, j) viene dado por la distancia del camino mínimo s.c.p. entre i y j

Sea Z una zona de recolección, consideramos $V_Z \subset V_R$ al conjunto de vértices que son esquinas de las manzanas de Z y notamos $R[V_Z]$ al subgrafo inducido por V_Z. Podría ocurrir que en el subgrafo inducido $R[V_Z]$ existan $i, j \in V_Z$ tales que no exista un camino s.c.p. que los conecte. Por esta razón se considera el siguiente conjunto:

$$V_Z^{(r)} = \{i \in V_R \colon \exists j \in V_Z \ / \ d(i, j) \leq r\}$$

donde d es la distancia usual en el digrafo. Dado que es deseable que el camión no deba salir de la zona mientras la recorre, se comienza con $r = 0$ y se aplica dPPG en $R[V_Z^{(0)}]$ con fuente v para cada $v \in V_Z$. Mientras haya vértices que no puedan ser conectados por un camino s.c.p. se aumenta r y se ejecuta dPPG hasta hallar un r tal que en $R[V_Z^{(r)}]$ existe un camino s.c.p. entre todo par de vértices de V_Z. Tal r existe, pues suponemos que en el plano existe un camino s.c.p. entre cualquier par de esquinas. De esta manera, para cada $u, v \in V_Z,\ u \neq v$ se obtiene γ_{uv} el camino mínimo s.c.p. entre u y v en $R[V_Z^{(r)}]$ que no viola ninguna restricción de tránsito. Vale aclarar que para $u, v \in V_Z$ tales que $(u, v) \in A_R$, suponemos $\gamma_{uv} = (u, v)$.

Una vez calculados los caminos mínimos entre cada par de vértices de Z, construimos el digrafo completo K sobre el que resolveremos el ATSP con restricciones adicionales.

Definición 43 (Digrafo completo de la zona)**.** Sean Z una zona, $R = (V_R, A_R, \omega_R)$ el digrafo de ruteo, $V_Z \subset V_R$ los vértices correspondientes a las esquinas de las manzanas de Z, $s \in V_R$ la esquina de partida de los camiones y $t \in V_R$ la de llegada. Definimos el digrafo completo de la zona Z como $K = (V_K, A_K, \omega_K)$ con:

- $V_K = V_Z \cup \{s, t\}$
- $A_K = \{(u, v) \in V_K \times V_K \colon\ u \neq v\}$
- $\omega_K(u, v) = \begin{cases} \omega(\gamma_{uv}) & \text{si } (u, v) \neq (t, s) \\ 0 & \text{c.c.} \end{cases}$

Modelo MILP para hallar una ruta eficiente

Finalmente, abordamos los dos últimos pasos del Algoritmo 14:

6 Resolver sobre K una instancia del ATSP con restricciones adicionales mediante un modelo MILP
7 Procesar la solución del modelo MILP para obtener una ruta eficiente en R que es factible para el MGRPTP

Proponemos un modelo MILP para hallar la secuencia en la que se visitan las esquinas de Z recorriendo todas las cuadras que no pueden ser achicadas y sin violar restricciones de tránsito. En el caso en el que Z no es conexa, es razonable esperar que en un recorrido eficiente sus sectores sean visitados en orden. Luego, si una zona Z tiene k sectores, existen $k!$ permutaciones de orden de recorrido. El modelo se resuelve para cada una de esas permutaciones, manteniendo un registro de la ruta más corta para Z obtenida hasta el momento. Dada una permutación de orden de recorrido de sectores, si la relajación lineal del modelo provee una cota inferior que supera la longitud de la mejor ruta obtenida hasta el momento, entonces se interrumpe la resolución del modelo y se continúa con la siguiente permutación. En la Tabla 5 describimos los conjuntos, las constantes y las variables de decisión del modelo de ruteo y su correspondiente notación. Asimismo, presentamos su formulación:

Notación	Descripción
Conjuntos y constantes	
V_K	conjunto de vértices de K
V_Z	conjunto de vértices que representan a las esquinas de Z ($V_Z = V_K - \{s,t\}$)
τ	umbral de achique.
n	$\lvert V_K \rvert$
$\mathcal{L}$	$\{e \in A_P \cup E_P \colon \omega_P(e) > \tau$ y los extremos de e están en $Z\}$ el conjunto de cuadras de Z que superan el umbral de achique τ.
L_e	para cada $e \in \mathcal{L}$, $(u,v) \in A_K$ pertenece a L_e si y solo si alguno de los arcos que representan a e en R es parte de γ_{uv}.
V_{uv}	$\{w \in V_K - \{u,v\} \colon$ la concatenación de γ_{uv} con γ_{vw} tiene un camino prohibido$\}$
$F_4^{(Z)}$	caminos prohibidos con cuatro vértices en V_Z ($F_4^{(Z)} \subseteq F_4 \subseteq F$).
B_f	para cada $f = (u_1,u_2,u_3,u_4) \in F_4^{(Z)}$, $v \in V_K$ pertenece a B_f si y solo si el último arco de γ_{vu_2} es (u_1,u_2).
$\hat{B}_f$	para cada $f = (u_1,u_2,u_3,u_4) \in F_4^{(Z)}$, $v \in V_K$ pertenece a $\hat{B}_f$ si y solo si los dos últimos arcos de γ_{vu_3} son (u_1,u_2) y (u_2,u_3).
E_f	para cada $f = (u_1,u_2,u_3,u_4) \in F_4^{(Z)}$, $v \in V_K$ pertenece a E_f si y solo si el primer arco de γ_{u_3v} es (u_3,u_4).
$\hat{E}_f$	para cada $f = (u_1,u_2,u_3,u_4) \in F_4^{(Z)}$, $v \in V_K$ pertenece a $\hat{E}_f$ si y solo si los primeros dos arcos de γ_{u_2v} son (u_2,u_3) y (u_3,u_4).
S_i	conjunto de vértices del i-ésimo sector de Z ($S_i \subset V_Z$)
p	si Z tiene ξ sectores, p es una permutación de $(1,\ldots,\xi)$ que indica el orden de recorrido de los sectores de Z. Notamos con p_k a la k-ésima coordenada de p.
Variables	
x_{uv}	vale 1 si y solo si γ_{uv} forma parte del recorrido ($x_{uv} \in \{0,1\}$).
y_v	indica la posicion de v en el recorrido ($y_v \in \mathbb{R}_{\geq 0}$).

Tabla 5: Conjuntos, parámetros y variables modelo de ruteo

$$\text{mín} \quad \sum_{u \in V_K} \sum_{\substack{v \in V_K \\ v \neq u}} \omega_K(u,v) x_{uv} \tag{13}$$

s.a:

$$\sum_{u \in V_K} x_{uv} = 1 \qquad v \in V_K \tag{14}$$

$$\sum_{v \in V_K} x_{uv} = 1 \qquad u \in V_K \tag{15}$$

$$y_u - y_v + (n-1)x_{uv} + (n-3)x_{vu} \leq n-2 \qquad u,v \in V_K - \{s\} \tag{16}$$

$$-y_u + (n-3)x_{us} + \sum_{\substack{v \in V_K - \{s\} \\ v \neq u}} x_{vu} \leq -1 \qquad u \in V_K - \{s\} \tag{17}$$

$$y_u + (n-3)x_{su} + \sum_{\substack{v \in V_K - \{s\} \\ v \neq u}} x_{vu} \leq n-1 \qquad u \in V_K - \{s\} \tag{18}$$

$$\sum_{(u,v) \in L_e} x_{uv} \geq 1 \qquad e \in \mathcal{L} \tag{19}$$

$$x_{ts} = 1 \tag{20}$$

$$\sum_{w \in V_{uv}} x_{vw} \leq 1 - x_{uv} \qquad u,v \in V_K \tag{21}$$

$$\sum_{v \in B_f} x_{vu_2} + x_{u_2u_3} + \sum_{v \in E_f} x_{u_3v} \leq 2 \qquad f \in F_4^{(Z)} \tag{22}$$

$$\sum_{v \in \hat{B}_f} x_{vu_3} + \sum_{v \in E_f} x_{u_3v} \leq 1 \qquad f \in F_4^{(Z)} \tag{23}$$

$$\sum_{v \in B_f} x_{vu_2} + \sum_{v \in \hat{E}_f} x_{u_2v} \leq 1 \qquad f \in F_4^{(Z)} \tag{24}$$

$$\sum_{u \in S_{p_k}} \sum_{v \in S_{p_{k+1}}} x_{uv} = 1 \qquad 1 \leq k \leq \xi - 1 \tag{25}$$

$$\sum_{u \in S_{p_k}} \sum_{\substack{\ell=1 \\ \ell \neq k \\ \ell \neq k+1}}^{\xi} \sum_{v \in S_{p_\ell}} x_{uv} = 0 \qquad 1 \leq k \leq \xi - 1 \tag{26}$$

$$\sum_{u \in S_{p_\xi}} \sum_{k=1}^{\xi-1} \sum_{\substack{v \in S_{p_k} \\ v \neq t}} x_{uv} = 0 \tag{27}$$

$$x_{uv} \in \{0,1\} \qquad (u,v) \in A_K$$

$$y_u \in \mathbb{R}_{\geq 0} \qquad u \in V_K$$

La función objetivo (13) minimiza la longitud del circuito que visita cada vértice en K exactamente una vez. Las restricciones (14)–(18) corresponden al modelo de Desrochers y

Laporte [35] para resolver el ATSP. Las restricciones (14) y (15) aseguran que cada vértice en K sea visitado exactamente una vez, mientras que (16)–(18) evitan la formación de subtours. La restricción (19) asegura que los camiones recorren las cuadras que son demasiado largas para ser achicadas. La restricción (20) impone que t sea el último vértice visitado. Por construcción, γ_{ij} no tiene caminos prohibidos, pero al concatenarse con otro camino mínimo, se podría violar una restricción de tránsito. Por lo tanto, las restricciones (21)–(24) evitan esto al anticipar casos de rutas prohibidas con tres o cuatro vértices. Finalmente, (25)-(27) se utilizan sólo si Z tiene más de un sector. (25) y (26) garantizan que desde cada sector sólo se viaja exactamente una vez al que le sigue en el orden indicado por p. Y (27) indica que desde el último sector en ser recorrido no se puede viajar a ningún otro. A partir de la solución del modelo, la ruta Γ que recorre la zona Z se construye concatenando los caminos γ_{ij} tales que $x_{ij} = 1$, en el orden indicado por las variables y.

Definición 44 (Concatenación de caminos)**.** Dados dos caminos $\boldsymbol{\gamma} = (v_1, v_2, \ldots, v_N)$, $\eta = (u_1, \ldots, u_M)$ tales que $v_N = u_1$, definimos la concatenación de $\boldsymbol{\gamma}$ y η como:

$$\boldsymbol{\gamma}^\frown \eta \coloneqq (v_1, \ldots, v_N, u_2, \ldots, u_M)$$

Sea (x^*, y^*) una solución del modelo de ruteo, ordenamos los vértices de V_K según lo indicado por y^*:

$$\zeta \coloneqq (v_1, \ldots, v_n) \quad \text{con } v_1 = s \text{ y } v_n = t$$

Concatenando los caminos mínimos s.c.p. , obtenemos un camino en P, el grafo mixto del plano que representa una ruta para la zona Z :

$$\Gamma_\zeta \coloneqq \boldsymbol{\gamma}_{v_1 v_2}{}^\frown \boldsymbol{\gamma}_{v_2 v_3}{}^\frown \ldots {}^\frown \boldsymbol{\gamma}_{v_{n-1} v_n}$$

A continuación demostramos que Γ_ζ es una solución para el MGRPTP.

Proposición 15. Sea Γ la ruta para Z obtenida a partir de una solución del modelo de ruteo, entonces es una solución factible para el MGRPTP.

Demostración. Sea (x^*, y^*) una solución del modelo de ruteo y sea Γ la ruta de recolección en Z que induce, veamos que Γ es una solución factible para el MGRPTP. En primer lugar, por construcción del digrafo de ruteo R, Γ es un camino en P, el grafo mixto pesado del plano. Además, por construcción, ningún camino $\boldsymbol{\gamma}_{v_i v_{i+1}}$ tiene caminos prohibidos y, por las restricciones (21)–(24) del modelo, la concatenación de los caminos tampoco da lugar a giros prohibidos. Por lo tanto, Γ es un camino en P sin caminos prohibidos.

Veamos ahora que pasa al menos una vez por los vértices requeridos ($V_Z \cup \{s, t\}$) y por los arcos y aristas requeridos ($\mathcal{L}$). Γ visita al menos una vez todos los vértices de $V_Z \cup \{s, t\}$ pues $\Gamma = \cup_{i=1}^{n-1} \boldsymbol{\gamma}_{v_i v_{i+1}}$ con $v_1 = s, v_n = t$ y $v_i \in V_Z$ para $i = 2, \ldots, n-1$. Por lo tanto, los vértices requeridos son visitados en P. Además, por la restricción (19), se asegura que toda cuadra cuya longitud supere el umbral de achique forma parte de $\boldsymbol{\gamma}_{v_i v_{i+1}}$, para algún $i = 1, \ldots, n-1$. De esta manera, la ruta recorre todas las cuadras en $\mathcal{L}$. □

7.2. Enfoque $F_4 = \emptyset$

Este enfoque fue desarrollado posteriormente al Enfoque $F_4 \neq \emptyset$ con dos objetivos en mente. En primer lugar, facilitar la implementación de las rutas. Por esta razón, incluimos penalidad no nula para los giros permitidos para simplificar las maniobras que debe realizar el chofer del camión recolector. También distinguimos la distancia del recorrido fuera y dentro de la zona de recolección. En segundo lugar, mejoramos el desempeño del modelo MILP para la búsqueda de rutas. De esta manera podría utilizarse en métodos iterativos que permitieran, por ejemplo, zonificar y buscar rutas simultáneamente.

Caminos mínimos

Para hallar caminos mínimos s.c.p. , en el Enfoque $F_4 \neq \emptyset$ habíamos utilizado el algoritmo de Di Puglia Pugliese y Guerreiro que permite hallar caminos mínimos s.c.p. donde los caminos prohibidos tienen una cantidad arbitraria de vértices. Sin embargo, encontramos dos desventajas: su tiempo de cómputo no es favorable dentro de un método iterativo y resulta muy complejo añadir las penalidades por giro. Por estas razones, proponemos un algoritmo para calcular caminos mínimos s.c.p. considerando penalización por giro cuando $F_4 = \emptyset$, que se basa en transformar el digrafo y aplicar un algoritmo de caminos mínimos.

Definición 45 (Digrafo transformado)**.** Sean $R = (V_R, A_R, \omega)$ un digrafo, F el conjunto de caminos prohibidos, $\alpha \in (0, \frac{1}{2}\pi]$ el ángulo entre arcos que define los giros y $p \geq 0$ la penalidad por girar, consideramos el *digrafo transformado* $T_R = (V_T, A_T, \eta)$ donde:

- $V_T = \{v_u \colon v \in V_R \wedge u \in N_R^{in}(v)\} \cup \{v_{(\cdot)} \colon v \in V_R\}$
- $A_T = \{(v_u, w_v) \colon (u,v) \in A_R \wedge (v,w) \in A_R \wedge (u,v,w) \notin F\} \cup \{(v_{(\cdot)}, w_v) \colon v \in V_R \wedge w \in N_R^{out}(v)\}$
- $\eta(v_u, w_v) = \omega(v,w) + p \cdot \theta_\alpha(u,v,w)$
- $\eta(v_{(\cdot)}, w_v) = \omega(v,w)$

con

$$\theta_\alpha(u,v,w) = \begin{cases} 1 & \text{si } -\pi + \alpha \leq angulo((u,v),(v,w)) \leq \pi - \alpha \\ 0 & \text{c.c.} \end{cases}$$

y $N_R^{in}(v)$ y $N_R^{out}(v)$ los vecindarios de entrada y de salida[6] de v, respectivamente.

En pro de la simplicidad, de aquí en más diremos que (u,v,w) es un giro si y sólo si $\theta_\alpha(u,v,w) \neq 0$. En la Figura 19 mostramos un ejemplo de un digrafo y su correspondiente transformación. En este ejemplo, el camino (v,y,x) y todos los giros en U están prohibidos. Por otro lado, en Algoritmo 15 presentamos un procedimiento para obtener T a partir de R.

Proposición 16. Sean $R = (V_R, A_R, \omega)$ un digrafo, F el conjunto de caminos prohibidos ($F \neq \emptyset$), $\alpha \in (0, \frac{1}{2}\pi]$ el ángulo entre arcos que define los giros y $p \geq 0$ la penalidad por girar. Entonces `TRANSFORMAR_GRAFO`(R, F, α, p) termina, devuelve la transformación del digrafo y su complejidad temporal es $\mathcal{O}(|V_R|^3|F|)$.

Demostración. Comenzamos probando la correctitud del algoritmo. El bucle de las líneas 3–6 agrega los vértices a V_T. Este bucle y el bucle de las líneas 5–6 terminan pues su cantidad de

[6] Definición 5

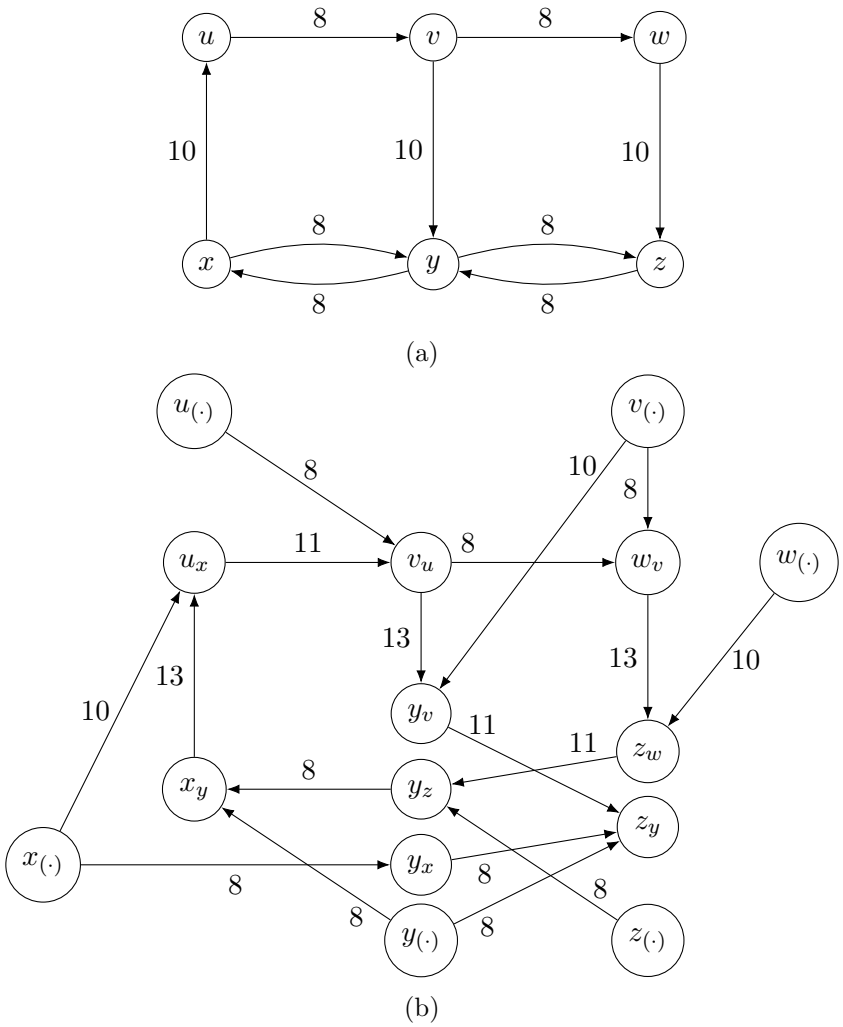


Figura 19: (a) Un ejemplo de un digrafo y (b) su transformación, considerando una penalidad $p = 3$ y ángulo de giro $\alpha = \frac{1}{5}\pi$. El camino (v, y, x) está prohibido, por eso no existe el arco (y_v, x_y). También están prohibidos todos los giros en U.

iteraciones está acotada por $|V_R|$. Al finalizar el bucle de las líneas 3–6, por la línea 4 se ha agregado a V_T el conjunto $\{v_{(\cdot)} \colon v \in V_R\}$ y por las lineas 5–6, se le ha agregado el conjunto $\{v_u \colon v \in V_R \wedge u \in N_R^{in}(v)\}$. Como V_T había sido inicializado vacío (línea 2), entonces coincide con el conjunto de vértices de la definición al concluir el bucle.

Analicemos el bucle de las líneas 7–12. Veamos que la definición de las aristas es correcta (líneas 9 y 12). En la línea 9, $v_{(\cdot)} \in V_T$ por la línea 4 y $w_v \in V_T$ por la línea 6 (pues si $w \in N_{out}^{(R)}$, entonces $v \in N_{in}^{(R)}(w)$), luego ambos vértices ya pertenecen a V_T y se puede agregar el arco $(v., w_v)$ a A_T. Además, se define el peso del arco en A_T según el peso de (v, w) en R. La prueba es análoga para la correcta definición de aristas en la línea 12. En ese caso, el peso de (v_u, w_v) se define como el peso de (v, w) en R más la penalidad por el giro (u, v, w), si existiera. El bucle termina pues su cantidad de iteraciones (y la de sus bucles anidados) está acotada por

Algoritmo 15: `TRANSFORMAR_DIGRAFO`

Entrada: R digrafo
F conjunto de caminos prohibidos
α ángulo de giro
p penalidad
Salida: T transformación de R

1 $A_T \leftarrow \emptyset$
2 $V_T \leftarrow \emptyset$
3 **para cada** $v \in V_R$ **hacer**
4 $\quad V_T \leftarrow V_T \cup \{v_{(\cdot)}\}$
5 $\quad$ **para cada** $u \in N_{in}^{(R)}(v)$ **hacer**
6 $\quad\quad V_T \leftarrow V_T \cup \{v_u\}$
7 **para cada** $v \in V_R$ **hacer**
8 $\quad$ **para cada** $w \in N_{out}^{(R)}(v)$ **hacer**
9 $\quad\quad$ Agregar $(v_{(\cdot)}, w_v)$ a A_T con peso $\eta(v_{(\cdot)}, w_v) \coloneqq \omega(v, w)$
10 $\quad\quad$ **para cada** $u \in N_{in}^{(R)}(v)$ **hacer**
11 $\quad\quad\quad$ **si** $(u, v, w) \notin F$ **entonces**
12 $\quad\quad\quad\quad$ Agregar (v_u, w_v) a A_T con peso $\eta(v_u, w_v) \coloneqq \omega(v, w) + p \cdot \theta_\alpha(u, v, w)$
13 **devolver** $T = (V_T, A_T, \eta)$

$|V_R|$. Notar además que por la línea 9 se agregan a A_T los arcos $\{(v_{(\cdot)}, w_v) \colon w \in N_R^{out}(v)\}$ y las líneas 10–12 agregan $\{(v_u, w_v) \colon (u, v) \in A_R \wedge (v, w) \in A_R \wedge (u, v, w) \notin F\}$. Como A_T había sido inicializado vacío, al finalizar el bucle coincide con el conjunto de aristas de la definición. Es directo ver que el peso de las aristas también coincide.

Con respecto a la complejidad del algoritmo, observemos que $|N_{in}(v)^{(R)}| \leq |V_R| - 1$ y $|N_{out}(v)^{(R)}| \leq |V_R| - 1$ para todo $v \in V_R$. Por ende, el bucle de las líneas 3–6 tiene complejidad $\mathcal{O}(|V_R|^2)$, pues tiene un bucle anidado. Por su parte, la complejidad del bucle de las líneas 7–12 es $\mathcal{O}(|V_R|^3|F|)$, pues son tres bucles anidados y decidir si $(u, v, w) \in F$ tiene complejidad $\mathcal{O}(F)$. Por ende, tenemos que la complejidad del algoritmo es:

$$\mathcal{O}(|V_R|^2 + |V_R|^3|F|) = \mathcal{O}(|V_R|^3|F|)$$

□

Corolario 1. Sean $R = (V_R, A_R, \omega)$ un digrafo y $T = (V_T, A_T, \eta)$ el correspondiente digrafo transformado, entonces $|V_T| = \mathcal{O}(|V_R|^2)$ y $|A_T| = \mathcal{O}(|V_R|^3)$.

Demostración. $|V_T| = \mathcal{O}(|V_R|^2)$ se deduce del cálculo de complejidad del Algoritmo 15. Recordemos que definimos A_T como:

$$A_T = \{(v_u, w_v) \colon (u, v) \in A_R \wedge (v, w) \in A_R \wedge (u, v, w) \notin F\} \cup \{(v_{(\cdot)}, w_v) \colon v \in V_R \wedge w \in N_R^{out}(v)\}$$

Por un lado, tenemos que:

$$|\{(v_{(\cdot)}, w_v) \colon v \in V_R \wedge w \in N_R^{out}(v)\}| = |A_R| \leq |V_R|^2$$

Por el otro:

$$|\{(v_u, w_v)\colon (u,v) \in A_R \wedge (v,w) \in A_R\}| = \left(\sum_{v \in V_R} \underbrace{|N_{in}^{(R)}(v)|}_{\leq |V_R|} \cdot \underbrace{|N_{out}^{(R)}(v)|}_{\leq |V_R|} \right) \leq |V_R|^3$$

Por ende:

$$|A_T| \leq |V_R|^3 - |F| + |V_R|^2 \leq |V_R|^3 + |V_R|^2$$

□

Como incluímos la penalización por giro, no buscamos minimizar ω al buscar caminos entre vértices en R, sino una función que toma en cuenta la distancia recorrida y la cantidad de giros del camino, que definimos a continuación.

Definición 46 (Función de costo con penalizacion por giro)**.** Sea $R = (V_R, A_R, \omega)$ un digrafo, p la penalidad por giro y $\alpha \in (0, \pi]$, para los caminos $\boldsymbol{\gamma} = (v^{(1)}, \ldots, v^{(n)})$ en R definimos el *costo con penalizacion por giro* como:

$$\hat{\omega}(\boldsymbol{\gamma}) = \begin{cases} \omega(\boldsymbol{\gamma}) & \text{si } n = 2 \\ \omega(\boldsymbol{\gamma}) + p \cdot \left(\sum_{i=1}^{n-2} \theta_\alpha(v^{(i)}, v^{(i+1)}, v^{(i+2)}) \right) & \text{si } n \geq 3 \end{cases}$$

Definición 47 (Camino de costo penalizado mínimo)**.** Sea $R = (V_R, A_R, \omega)$ un digrafo y $\boldsymbol{\gamma}$ un camino que une a $v \in V_R$ con $u \in V_R$, decimos que $\boldsymbol{\gamma}$ es de costo penalizado mínimo si y solo si minimiza $\hat{\omega}$ entre todos los caminos que unen a v con u en R.

A continuación enunciamos y demostramos proposiciones que relacionan los caminos de costo penalizado mínimos s.c.p. de un digrafo R con los caminos mínimos de su transformación T_R.

Proposición 17. Sean $R = (V_R, A_R, \omega)$ un digrafo y $T_R = (V_T, A_T, \eta)$ la transformación de R, $\boldsymbol{\gamma} = (v^{(1)}, \ldots, v^{(n)})$ es un camino s.c.p. en R si y solo si $\boldsymbol{\varphi} = (v_{(\cdot)}^{(1)}, v_{v^{(1)}}^{(2)}, \ldots, v_{v^{(n-1)}}^{(n)})$ es un camino en T_R. Además, $\hat{\omega}(\boldsymbol{\gamma}) = \eta(\boldsymbol{\varphi})$.

Demostración. Probémoslo primero para $n = 2$. Como todos los caminos prohibidos tienen 3 vértices, cualquier camino de dos vértices en R no tiene caminos prohibidos. Tenemos entonces que:

$$\begin{aligned} \boldsymbol{\gamma} = (v^{(1)}, v^{(2)}) \text{ es un camino en } R \quad &\iff (v^{(1)}, v^{(2)}) \in A_R \iff (v_{(\cdot)}^{(1)}, v_{v^{(1)}}^{(2)}) \in A_T \\ &\iff \boldsymbol{\varphi} = (v_{(\cdot)}^{(1)}, v_{v^{(1)}}^{(2)}) \text{ es un camino en } T_R \end{aligned}$$

Además, por definición de $\hat{\omega}$ y de T_R:

$$\hat{\omega}(\boldsymbol{\gamma}) = \omega(\boldsymbol{\gamma}) = \omega(v^{(1)}, v^{(2)}) = \eta(v_{(\cdot)}^{(1)}, v_{v^{(1)}}^{(2)}) = \eta(\boldsymbol{\varphi})$$

Veamos que vale para $n \geq 3$:

$\boldsymbol{\gamma} = (v^{(1)}, \ldots, v^{(n)})$ un camino s.c.p. en $R \iff (v^{(i)}, v^{(i+1)}) \in A_R$ para $1 \leq i \leq n-1$ y $(v^{(i)}, v^{(i+1)}, v^{(i+2)}) \notin F$ para $1 \leq i \leq n-2 \iff (v^{(1)}_{(\cdot)}, v^{(2)}_{v^{(1)}}) \in A_T$ y $(v^{(i)}_{v^{(i-1)}}, v^{(i+1)}_{v^{(i)}}) \in A_T$ para $2 \leq i \leq n-1 \iff \varphi = (v^{(1)}_{(\cdot)}, v^{(2)}_{v^{(1)}}, \ldots, v^{(n)}_{v^{(n-1)}})$ es un camino en T_R.

Probemos que $\hat{\omega}(\boldsymbol{\gamma}) = \eta(\varphi)$:

$$\begin{aligned}
\hat{\omega}(\boldsymbol{\gamma}) &= \omega(\boldsymbol{\gamma}) + p \cdot \sum_{i=1}^{n-2} \theta_\alpha(v^{(i)}, v^{(i+1)}, v^{(i+2)}) \\
&= \sum_{i=1}^{n-1} \omega(v^{(i)}, v^{(i+1)}) + p \cdot \sum_{i=2}^{n-1} \theta_\alpha(v^{(i-1)}, v^{(i)}, v^{(i+1)}) \\
&= \omega(v^{(1)}, v^{(2)}) + \sum_{i=2}^{n-1} \omega(v^{(i)}, v^{(i+1)}) + p \cdot \theta_\alpha(v^{(i-1)}, v^{(i)}, v^{(i+1)}) \\
&= \eta(v^{(1)}_{(\cdot)}, v^{(2)}_{v^{(1)}}) + \sum_{i=2}^{n-1} \eta(v^{(i)}_{v^{(i-1)}}, v^{(i+1)}_{v^{(i)}}) \\
&= \eta(\varphi)
\end{aligned}$$

□

Retomando el ejemplo de la Figura 19, el camino $\varphi = (u_{(\cdot)}, v_u, y_v, z_y)$ en T_R tiene asociado el camino $\boldsymbol{\gamma} = (u, v, y, z)$ en R y podemos corroborar que $\hat{\omega}(\boldsymbol{\gamma}) = \eta(\varphi) = 32$. Asimismo, el camino (x, u, v) en R tiene asociado el camino $(x_{(\cdot)}, u_x, v_u)$ en T_R.

Proposición 18. Sean $R = (V_R, A_R, \omega)$ un digrafo y $v, w \in V_R$ tales que $v \neq w$ y existe al menos un camino en R desde v a w. Sea $T_R = (V_R, A_R, \eta)$ la transformación de R y sea $x \in N_R^{in}(w)$ tal que el camino mínimo entre $v_{(\cdot)}$ y w_x minimiza η entre todos los caminos mínimos en T_R desde $v_{(\cdot)}$ a w_u con $u \in N_R^{in}(w)$. Sea $\varphi = (v^{(1)}_{(\cdot)}, v^{(2)}_{v^{(1)}}, \ldots, v^{(n)}_{v^{(n-1)}})$ tal camino, entonces $\boldsymbol{\gamma} = (v^{(1)}, v^{(2)}, \ldots, v^{(n-1)}, v^{(n)})$ es un camino de costo penalizado mínimo entre v y w en R.

Demostración. Si $\boldsymbol{\gamma}$ no fuera un camino de costo penalizado mínimo, existiría $\bar{\boldsymbol{\gamma}}$ otro camino entre v y w en R tal que $\hat{\omega}(\bar{\boldsymbol{\gamma}}) < \hat{\omega}(\boldsymbol{\gamma})$. Entonces, por Proposición 17, existe $\bar{\varphi}$ camino en T_R que une $v_{(\cdot)}$ con w_y tal que $y \in N_R^{in}(w)$ e $y \neq x$ Por lo tanto:

$$\eta(\bar{\varphi}) = \hat{\omega}(\bar{\boldsymbol{\gamma}}) < \hat{\omega}(\boldsymbol{\gamma}) = \eta(\varphi) \quad \text{Abs!}$$

El absurdo provino de suponer que $\boldsymbol{\gamma}$ no es un camino de costo penalizado mínimo. □

En consecuencia, para encontrar el camino de costo penalizado mínimo en R desde v hasta el resto de los vértices basta ejecutar un algoritmo de caminos mínimos en T_R con $v_{(\cdot)}$ como vértice fuente. Puesto que en el contexto de nuestro problema no hay ciclos negativos, elegimos el algoritmo de Dijkstra [37], cuya complejidad en T_R es $\mathcal{O}(|A_T| + |V_T| \log(|V_T|))$ si se implementa con un *heap* de Fibonacci [51].

En el Algoritmo 16 resumimos el procedimiento para calcular caminos de costo penalizado mínimos sin caminos prohibidos desde un vértice v a todos los vértices alcanzables desde él.

Algoritmo 16: `CAMINOS_CPMSCP`

Entrada: R digrafo
v vértice fuente, $v \in V_R$
F conjunto de caminos prohibidos
α ángulo de giro
p penalidad
Salida: C colección de caminos de costo penalizado mínimos desde v a los vértices alcanzables desde v

1 $T \leftarrow$ `TRANSFORMAR_DIGRAFO`(R, F, α, p)
2 $D \leftarrow$ Aplicar el algoritmo de Dijkstra a T con $v_{(\cdot)}$ como fuente
`/* notamos` $D[a,b]$ `al camino desde` a `a` b `hallado por Dijkstra y` $\eta_{[a,b]}$ `a su costo. Si no existe ningún camino,` $D[a,b] = \emptyset$ `y` $\eta_{[a,b]} = +\infty$. `*/`
3 $C \leftarrow$ colección vacía
4 **para cada** $w \in V_R$, $w \neq v$ **hacer**
5 u$\leftarrow \arg\min_{x \in N_{in}^{(R)}(w)} \eta_{[v_{(\cdot)}, w_x]}$
6 $C[v, w] \leftarrow$ camino en R relacionado a $D[v_{(\cdot)}, w_u]$
7 **devolver** C

Proposición 19. Sea $R = (V_R, A_R, \omega)$ un digrafo tal que para cada $w \in V_R$ vale que $N_{in}^{(R)}(w) \neq \emptyset$, $v \in V_R$, F el conjunto de caminos prohibidos ($F \neq \emptyset$), α el ángulo para determinar los giros y p la penalización por giro, entonces `CAMINOS_CPMSCP`(R, v, F, α, p) termina, devuelve caminos de costo penalizado mínimo s.c.p. en R desde v a todos los vértices alcanzables desde él y tiene complejidad $\mathcal{O}(|V_R|^3|F|)$.

Demostración. Puesto que R no tiene aristas con peso negativo y la penalidad por giro p es no negativa, entonces T_R no tiene aristas de peso negativo. Por lo tanto, no tiene ciclos negativos y se puede aplicar el algoritmo de Dijkstra. El bucle de las líneas 4–6 termina, pues su cantidad de iteraciones está acotada por $|V_R|$ y no contiene otro bucle. Con respecto a la línea 5, como $N_{in}^{(R)}(w) \neq \emptyset$ para todo $w \in V_R$, tomar argumento mínimo está bien definido.

Por Proposición 17, existe en R un camino s.c.p. desde v a w relacionado a $D[v_{(\cdot)}, w_u]$. Además, por Proposición 18, como $D[v_{(\cdot)}, w_u]$ es de costo mínimo, su camino relacionado en R es de costo penalizado mínimo s.c.p. Por lo tanto, de existir un camino entre v y w en R, en C se almacena un camino de costo penalizado mínimo s.c.p.

Con respecto a la complejidad, en términos de R, por Proposición 16 la línea 1 tiene complejidad $\mathcal{O}(|V_R|^3|F|)$. Utilizando el Corolario 1 y considerando la complejidad del algoritmo de Dijkstra, la complejidad de la línea 2 es:

$$\mathcal{O}(|A_T| + |V_T| \log(|V_T|)) = \mathcal{O}(|V_R|^3 + |V_R|^2 \log(|V_R|^2)) = \mathcal{O}(|V_R|^3)$$

Por su parte, el bucle de las líneas 4–6 tiene complejidad $\mathcal{O}(|V_R|^2)$, pues el bucle se ejecuta $|V_R| - 1$ veces y en el caso en el que R sea un digrafo completo, se debe calcular el argumento mínimo sobre $|N_R^{in}(w)| = |V_R| - 1$ elementos. Por ende, la complejidad del algoritmo es $\mathcal{O}(|V_R|^3|F|)$. □

Recordar que el algoritmo de Di Puglia Pugliese y Guerreiro (dPPG) utilizado para el Enfoque $F_4 \neq \emptyset$ tiene complejidad $\mathcal{O}(k^2|A_R||F|)$, con $k = \sum_{w \in V_R - \{v\}} k_w$ donde k_w es la cantidad de caminos que existen en R que comienzan en v y terminan en w. Si, como en el cálculo de la complejidad del Algoritmo 16, consideramos el caso en el que R es un grafo completo, entonces dPPG tendría complejidad factorial, pues k_w sería del orden de $(|V_R|-2)!$ (tan solo contemplando la cantidad de caminos simples). Si bien dPPG permite buscar caminos mínimos s.c.p. independientemente de la cantidad de vértices de los caminos en F, nuestro algoritmo aprovecha que sólo hay caminos prohibidos de tres vértices para calcular caminos mínimos s.c.p. en tiempo polinomial en $|V_R|$. Además, a diferencia de dPPG, nuestro algoritmo admite penalización de giro no nula.

Distancia ponderada

En este enfoque cuantificamos la distancia de una manera distinta. En primer lugar, la velocidad con la que el camión se desplaza fuera de su zona asignada es considerablemente mayor que dentro de ella. Por otra parte, el tipo de calle también incide sobre la ruta del camión cuando se dirige hacia la zona o hacia la planta de tratamiento: generalmente transitar por avenidas o autopistas es más rápido que por calles residenciales, por lo que puede ser preferible tomar una ruta sensiblemente más larga, pero en la que se pueda transitar con mayor velocidad. Otra variable que afecta la velocidad de desplazamiento es la cantidad de tránsito. Como nuestros casos de estudio no contamos con datos detallados sobre este atributo, suponemos que el tráfico depende del turno de recolección.

Para reflejar estas características, introducimos el concepto de *distancia ponderada*. Dentro de la zona de recolección, consideraremos que las cuadras tienen el largo en metros que les corresponde. Fuera de la zona de recolección, el largo de las cuadras estará multiplicado por una constante menor a 1 que depende del tipo de calle y del turno en el que se recolecta la zona.

Consideramos tres turnos: mañana (M), tarde (T) y noche (N). También clasificamos a las calles en avenidas (Av), autopistas (Au) y residenciales (Res). La constante que utilizamos en la ponderación es el cociente de la media de la velocidad dentro de la zona y la media de la velocidad fuera de la zona para el turno y tipo de calles correspondientes. De esta manera, obtenemos un digrafo de ruteo con distancia ponderada para cada turno.

Notación. Notamos con $v_{(q,c)}$, donde $c \in \{Av, Au, Res\}$ y $q \in \{M, T, N\}$, a la velocidad media en calles de tipo c en el turno q durante la recolección. Por otro lado, $\hat{v}_{(q,c)}$ es la velocidad media en calles de tipo c durante el turno q fuera de la zona de recolección.

Definición 48 (Digrafo de ruteo de turno)**.** Sea $R = (V_R, A_R, \omega)$ el digrafo de ruteo del plano. Para cada $q \in \{M, T, N\}$ definimos el *digrafo de ruteo del turno* q como $R_q = (V_R, A_R, \omega_q)$ con $\omega_q \colon A_R \to \mathbb{R}_{\geq 0}$ dada por:

$$\omega_q(e) = \frac{v_{(q,c(e))}}{\hat{v}_{(q,c(e))}} \omega(e)$$

donde $c(e)$ es el tipo de calle que indica el atributo de e.

Análogamente al Enfoque $F_4 \neq \emptyset$, este enfoque resuelve la primera línea del Algoritmo 14:

1 Calcular caminos mínimos sin caminos prohibidos (s.c.p.) desde s a todos los vértices de R y desde todos los vértices de R hasta t

invocando el Algoritmo 16 sobre cada digrafo de ruteo de turno R_q con s (vértice de partida de los camiones) como vértice fuente y sobre el transpuesto de R con t (vértice de llegada de los camiones) como vértice fuente.

Construcción del grafo completo K

Para cada $Z \in \mathcal{Z}$, encaramos ahora la resolución de los siguientes pasos del Algoritmo 14:

3 Definir $V_Z \subset V_R$ como los vértices que pertenecen a la zona Z
4 Calcular un camino mínimo s.c.p. entre cada par de vértices de V_Z
5 Elaborar un digrafo completo K cuyos vértices son $V_Z \cup \{s, t\}$ y el peso del arco (i, j) viene dado por la distancia del camino mínimo s.c.p. entre i y j

La introducción de la distancia ponderada requiere una definición del digrafo completo K distinta a la del Enfoque $F_4 \neq \emptyset$. En el caso en el que Z no fuera conexa[7], diremos que dos vértices de V_Z pertenecen al mismo sector si son esquinas de manzanas que pertenecen a la misma componente conexa del grafo de manzanas.

Para calcular el camino de costo penalizado mínimo s.c.p. entre cada par de vértices de V_Z, utilizamos el Algoritmo 16. Tal como ocurría en el Enfoque $F_4 \neq \emptyset$, consideramos

$$V_Z^{(r)} = \{i \in V_R \colon \exists j \in V_Z \ / \ d(i,j) \leq r\}$$

donde d es la distancia usual en el digrafo. Comenzamos con $V_Z^{(0)}$ y aumentamos r hasta que exista un camino de costo penalizado mínimo s.c.p. entre cada par de vértices de V_Z. Notamos con γ_{uv} al camino de costo penalizado mínimo s.c.p. entre u y v y, tal como hicimos en el Enfoque $F_4 \neq \emptyset$, suponemos que si $(u, v) \in A_R$, entonces $\gamma_{uv} = (u, v)$. Continuemos ahora con la elaboración del digrafo completo K.

Notación. Sea $R = (V_R, A_R, \omega_R)$ el digrafo de ruteo, notaremos γ_{uv} al camino de costo penalizado mínimo s.c.p. entre u y v en R y con $\omega(\gamma_{uv})$ a su costo. Análogamente, sea $q \in \{M, T, N\}$ un turno y $R_q = (V_R, A_R, \omega_q)$ el digrafo del turno q, notaremos con $\gamma_{uv}^{(q)}$ al camino de costo penalizado mínimo s.c.p. entre u y v en R_q y con $\omega_q(\gamma_{uv}^{(q)})$ a su costo.

Definición 49 (Digrafo completo de la zona)**.** Sean Z una zona que se recorre en el turno q, $R = (V_R, A_R, \omega_R)$ el digrafo de ruteo, $R_q = (V_R, A_R, \omega_q)$ digrafo del turno q, $V_Z \subset V_R$ los vértices correspondientes a las esquinas de las manzanas de Z, $s \in V_R$ la esquina de partida de los camiones y $t \in V_R$ la de llegada. Definimos el *digrafo completo de la zona Z* como $K = (V_K, A_K, \omega_K)$ con:

- $V_K = V_Z \cup \{s, t\}$
- $A_K = \{(u, v) \in V_K \times V_K \colon u \neq v\}$

[7] Definición 31

- $\omega_K(u,v) = \begin{cases} \omega(\gamma_{uv}) & \text{si } (u,v) \neq (t,s),\ \{u,v\} \subset V_Z \text{ y pertenecen al mismo sector} \\ 0 & \text{si } (u,v) = (t,s) \\ \omega_q(\gamma_{uv}^{(q)}) & \text{c.c.} \end{cases}$

Modelo MILP para hallar una ruta eficiente

Ahora nos abocaremos a resolver las últimas líneas del Algoritmo 14:

6 Resolver sobre K una instancia del ATSP con restricciones adicionales mediante un modelo MILP
7 Procesar la solución del modelo MILP para obtener una ruta eficiente en R que es factible para el MGRPTP

Para esto, elaboramos un modelo MILP similar al del Enfoque $F_4 \neq \emptyset$, agregando la penalidad por giro, redefiniendo una de sus restricciones para lograr una formulación más ajustada e introduciendo heurísticas para mejorar su rendimiento. Para esto, definimos algunos conjuntos y variables.

Sea $v \in V_R$, consideremos los siguientes conjuntos:

$$\begin{aligned} O_v &= \{(v,u) \in A_R \colon u \in N_R^{out}(v)\} \quad \text{(arcos salientes de } v) \\ I_v &= \{(u,v) \in A_R \colon u \in N_R^{in}(v)\} \quad \text{(arcos entrantes a } v) \end{aligned}$$

Para cada $f = (u,v) \in I_v$ definimos el conjunto de arcos salientes de v que son incompatibles con f:

$$\rho_f = \{(v,w) \in A_R \colon (u,v,w) \in F\}$$

Sea $K = (V_K, A_K, \omega_K)$ el digrafo completo de la zona, para cada arco $e = (v,w) \in A_R$ con $v \in V_K$ y para cada $f = (w,v) \in A_R$ con $v \in V_K$ definimos:

$$\begin{aligned} P_e &= \{u \in V_K \colon \gamma_{vu} \text{ comienza con } e\} \\ Q_f &= \{u \in V_K \colon \gamma_{uv} \text{ termina con } f\} \end{aligned}$$

Además, sea $v \in V_K$ y $f = (u,v) \in I_v$ consideramos el conjunto de arcos salientes de v cuyo ángulo con f determina un giro:

$$G_f = \{e = (v,w) \in O_v - \rho_f \colon (u,v,w) \text{ tiene un giro en } R\}$$

Si una zona tiene más de un sector, resulta razonable esperar que en un recorrido eficiente los sectores sean recorridos en cierto orden. Para hallarlo, proponemos un procedimiento heurístico para acelerar la solución del modelo.

En la Figura 20 mostramos el ejemplo de una zona que tiene tres sectores, enumerados del 1 al 3. El punto 0 corresponde al vértice de donde salen los camiones y el punto 4 al de llegada. La heurística que describiremos a continuación busca estimar cuál de las seis posibles secuencias de recorridos de sectores es la más eficiente.

Sean $R_q = (V_R, A_R, \omega_q)$ el digrafo de ruteo del turno q, Z una zona con n sectores $S_1, \ldots, S_n$ ($n \geq 2, S_i \subset V_R$) que se recorre en el turno q, consideramos el digrafo completo $G = (V, A, \zeta)$ donde:

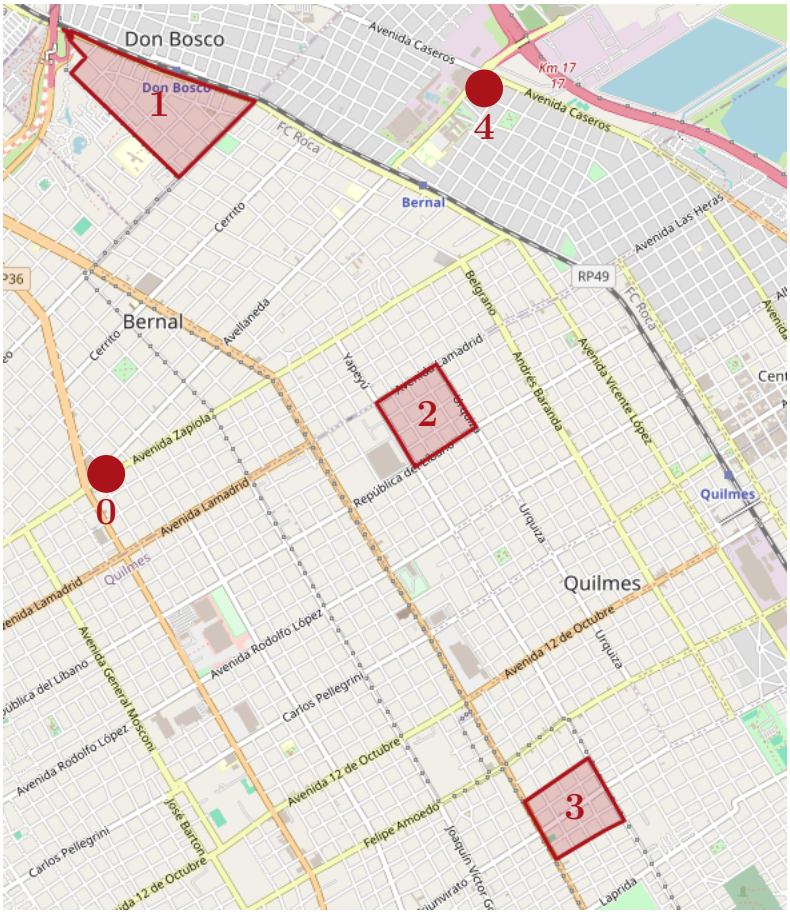


Figura 20: Una zona con tres sectores. Los puntos 0 y 4 representan al punto de partida de los camiones y al punto de llegada, respectivamente.

- $V = \{0, 1, \ldots, n+1\}$, cada vértice $i \in V$ representa al sector S_i de Z, el vértice 0 representa al vértice de partida de los camiones s y el vértice $n+1$ al correspondiente al vértice de llegada t.
- $A = \{(i,j) \in V \times V : i \neq j\}$
- $\zeta(i,j) = \begin{cases} \frac{\sum_{v \in S_j} \omega_q(\gamma_{sv}^{(q)})}{|S_j|} & \text{si } i = 0, 1 \leq j \leq n \quad \text{(Media de distancias desde } s \text{ a } S_j\text{)} \\ \frac{\sum_{(u,v) \in S_i \times S_j} \omega_q(\gamma_{uv}^{(q)})}{|S_i||S_j|} & \text{si } 1 \leq i, j \leq n \quad \text{(Media de distancias desde } S_i \text{ a } S_j\text{)} \\ \frac{\sum_{v \in S_i} \omega_q(\gamma_{vt}^{(q)})}{|S_i|} & \text{si } 1 \leq i \leq n, j = n+1 \quad \text{(Media de distancias desde } S_i \text{ a } t\text{)} \\ 0 & \text{si } i = 0, j = n+1 \\ +\infty & \text{c.c.} \end{cases}$

En G resolvemos una instancia del ATSP con el modelo MILP de Desrochers y Laporte [35], tomando 0 como vértice inicial. El resultado representa un orden de recorrido de los sectores de Z. Sea $c = (k_0, \ldots, k_{n+2})$ el ciclo hamiltoniano de costo mínimo obtenido a partir de la solución

del ATSP en T, por la definición de ζ es claro que $k_0 = k_{n+2} = 0 = s$ y $k_{n+1} = n + 1 = t$. Por lo tanto, el orden de recorrido de los sectores viene dado por $\hat{c} = (k_1, \ldots, k_n)$. En nuestro ejemplo, si $c = (0, 3, 2, 1, 4, 0)$, tenemos que $\hat{c} = (3, 2, 1)$.

Definimos, entonces, las siguientes constantes para $1 \leq i, j \leq n$:

$$w_{ij} = \begin{cases} 1 & \text{si } i \text{ precede directamente a } j \text{ en } \hat{c} \\ 0 & \text{c.c.} \end{cases}$$

Ahora estamos en condiciones de formular el modelo MILP a través del cual hallamos rutas eficientes para la zona Z. En la Tabla 6 describimos sus conjuntos, constantes y variables.

Notación	Descripción
Conjuntos y constantes	
V_K	conjunto de vértices de K
V_Z	conjunto de vértices que representan a las esquinas de Z ($V_Z = V_K - \{s, t\}$)
F	conjunto de caminos prohibidos
τ	umbral de achique.
$\mathcal{L}$	$\{e \in A_P \cup E_P : \omega_P(e) > \tau$ y los extremos de e están en $Z\}$ el conjunto de cuadras de Z que superan el umbral de achique τ.
L_e	para cada $e \in \mathcal{L}$, $(u, v) \in A_K$ pertenece a L_e si y solo si alguno de los arcos que representan a e en R es parte de γ_{uv}.
O_v	$\{(v, u) \in A_R : u \in N_R^{out}(v)\}$ (arcos salientes de $v \in V_K$)
I_v	$\{(u, v) \in A_R : u \in N_R^{in}(v)\}$ (arcos entrantes a $v \in V_K$)
P_e	sea $e = (v, w)$, $P_e = \{u \in V_K : \gamma_{[v,u]}$ comienza con $e\}$
Q_f	sea $f = (w, v)$, $\{u \in V_K : \gamma_{uv}$ termina con $f\}$
ρ_f	sea $f = (u, v)$, $\rho_f = \{e = (v, w) \in O_v : (u, v, w) \in F\}$
G_f	sea $f = (u, v)$, $G_f = \{e = (v, w) \in O_v - \rho_f : (u, v, w)$ tiene un giro en $R\}$
p	penalidad por giro.
n	$\lvert V_K \rvert$
ζ	cantidad de sectores de Z
S_i	conjunto de vértices del i-ésimo sector de Z ($S_i \subset V_Z$)
w_{ij}	indica si el sector i precede al sector j de Z (si Z tiene más de un sector)
g_{uv}	cantidad de giros en γ_{uv}
Variables	
x_{uv}	vale 1 si y solo si γ_{uv} forma parte del recorrido ($x_{uv} \in \{0, 1\}$).
y_v	indica la posicion de v en el recorrido ($y_v \in \mathbb{R}_{\geq 0}$).
z_v	indica si en la union de caminos en v se incurre en un giro ($z_v \in \{0, 1\}$)

Tabla 6: Conjuntos, constantes y variables modelo de ruteo

$$\text{mín} \quad \sum_{u \in V_K} \sum_{\substack{v \in V_K \\ u \neq v}} (\omega_K(u,v) + p \cdot g_{uv}) x_{uv} + p \cdot \sum_{v \in V_K - \{s,t\}} z_v \tag{28}$$

s.a:

$$\sum_{u \in V_K} x_{uv} = 1 \qquad v \in V_K \tag{29}$$

$$\sum_{v \in V_K} x_{uv} = 1 \qquad u \in V_K \tag{30}$$

$$y_u - y_v + (n-1)x_{uv} + (n-3)x_{vu} \leq n-2 \qquad u, v \in V_K - \{s\} \tag{31}$$

$$-y_u + (n-3)x_{us} + \sum_{\substack{v \in V_K - \{s\} \\ v \neq u}} x_{vu} \leq -1 \qquad u \in V_K - \{s\} \tag{32}$$

$$y_u + (n-3)x_{su} + \sum_{\substack{v \in V_K - \{s\} \\ v \neq u}} x_{vu} \leq n-1 \qquad u \in V_K - \{s\} \tag{33}$$

$$\sum_{(u,v) \in L_e} x_{uv} \geq 1 \qquad e \in \mathcal{L} \tag{34}$$

$$x_{ts} = 1 \tag{35}$$

$$\sum_{e \in \rho_f} \sum_{w \in P_e} x_{vw} \leq 1 - \sum_{w \in Q_f} x_{wv} \qquad v \in V_Z,\ f \in I_v \tag{36}$$

$$\sum_{e \in G_f} \sum_{w \in P_e} x_{vw} + \sum_{w \in Q_f} x_{wv} \leq z_v + 1 \qquad v \in V_K - \{s,t\},\ f \in I_v \tag{37}$$

$$\sum_{u \in S_i} \sum_{v \in S_j} x_{uv} = w_{ij} \qquad 1 \leq i, j \leq \zeta, i \neq j \tag{38}$$

$$x_{uv} \in \{0,1\} \qquad (u,v) \in A_K$$

$$y_u \in \mathbb{R}_{\geq 0} \qquad u \in V_K$$

$$z_v \in \{0,1\} \qquad v \in V_K$$

La función objetivo (28) minimiza la longitud de la ruta que visita cada vértice en K exactamente una vez, penalizando la cantidad de giros en los caminos entre los vértices y en su concatenación. Las restricciones (29)–(35) son las mismas que las del modelo del Enfoque $F_4 \neq \emptyset$ y garantizan que el camión visite todos los vértices de V_K, que transite al menos una vez por todas las cuadras largas y que el recorrido comience en s. El conjunto de restricciones (36) evita que en v se de una concatenación de caminos que incurra en una violación de normas de tránsito: para cada f arista entrante a v en R, los caminos γ_{uv} que terminan en f no pueden ser concatenados con caminos γ_{vw} cuya primera arista conforme un camino prohibido al ser precedida por f. Este conjunto de restricciones cumple la misma función que (21) en el modelo del Enfoque $F_4 \neq \emptyset$ pero, como mostraremos más adelante, permite una relajación lineal más ajustada. Por su parte, (37) vincula las variables z con las variables x: si en v la concatenación de caminos da lugar a un giro, z_v vale 1. Finalmente, si Z tiene más de un sector, las restricciones (38) aseguran que se respete el orden obtenido a través de la heurística.

Concatenando los caminos de costo penalizado mínimo en el orden indicado por una

solución (x^*, y^*, z^*) del modelo, obtenemos un recorrido de recolección para Z. Notar que este recorrido es factible para el MGRPTP y la demostración es análoga a la de la Proposición 15. A partir de la solución del modelo, podemos calcular la *distancia mixta* del recorrido como

$$\sum_{u \in V_K} \sum_{\substack{v \in V_K \\ u \neq v}} \omega_K(u, v) x^*_{uv}$$

La distancia mixta considera la distancia ponderada fuera de la zona de recolección y en el viaje entre sectores, mientras que considera la distancia real de los caminos entre vértices del mismo sector de Z.

Ahora demostraremos que las restricciones (36) dan lugar a una relajación lineal más ajustada.

Proposición 20. Sean $K = (V_K, A_K, \omega_K)$ el digrafo completo de la zona, el conjunto de restricciones para evitar caminos prohibidos de tres vértices en la concatenación de caminos (36) provee una relajación lineal más ajustada que:

$$\sum_{w \in V_{uv}} x_{vw} \leq 1 - x_{uv} \quad v \in V_Z,\ u \in V_K - \{v\} \tag{39}$$

con $V_{uv} = \{w \in V_K - \{u, v\} \colon \gamma_{uv}{}^\frown \gamma_{vw} \text{ tiene un camino prohibido}\}$

Demostración. En primer lugar, veamos que el conjunto de restricciones (36) domina al conjunto (39), es decir, si $x \in \{0,1\}^{|V_K| \times |V_K|}$ es tal que cumple (36), entonces cumple (39). Sean $v \in V_Z$ y $u \in V_K - \{v\}$ queremos ver que vale:

$$\sum_{w \in V_{uv}} x_{vw} \leq 1 - x_{uv}$$

Sea $f \in I_v$ el último arco de γ_{uv}. Por definición, cada $w \in V_{uv}$ es tal que γ_{vw} comienza con un arco en ρ_f, por lo tanto, $V_{uv} \subseteq \cup_{e \in \rho_f} P_e$. Tenemos entonces que:

$$\sum_{w \in V_{uv}} x_{vw} \leq \sum_{e \in \rho_f} \sum_{w \in P_e} x_{vw} \overset{(36)}{\leq} 1 - \sum_{w \in Q_f} x_{wv} = 1 - x_{uv} - \underbrace{\sum_{\substack{w \in Q_f \\ w \neq u}} x_{wv}}_{\geq 0} \leq 1 - x_{uv}$$

Por ende, las restricciones (36) dominan a (39).

Ahora probemos que utilizar (36) no hace infactible ninguna solución entera que sería factible para (39). Notando $M1$ al modelo (28)–(38) y $M2$ al modelo $M1$ reemplazando (36) por (39), quisiéramos ver que si $(x^\star, y^\star, z^\star) \in \{0,1\}^{|V_K| \times |V_K|} \times \mathbb{R}_{\geq 0}^{|V_K|} \times \{0,1\}^{|V_K|}$ es factible para $M2$, entonces es factible para $M1$. Probémoslo con el contrarrecíproco. Supongamos que $(x^\star, y^\star, z^\star)$ no es factible para $M1$. Como la única diferencia entre los dos modelos es el conjunto de restricciones que impiden caminos prohibidos en la concatenación de caminos, entonces existen $v \in V_K, f \in I_v$ que no cumplen (36). Tenemos entonces que:

$$\underbrace{\sum_{e \in \rho_f} \sum_{w \in P_e} x^\star_{vw}}_{\in \{0,1\} \text{ por (30)}} > 1 - \underbrace{\sum_{w \in Q_f} x^\star_{wv}}_{\in \{0,1\} \text{ por (29)}} \Rightarrow \sum_{e \in \rho_f} \sum_{w \in P_e} x^\star_{vw} = 1 \wedge \sum_{w \in Q_f} x^\star_{wv} = 1$$

Como $x^\star \in \{0,1\}^{|V_K|\times|V_K|}$, se deduce que existen $w \in P_e$ para algún $e \in \rho_f$ y $u \in Q_f$ tales que $x^\star_{vw} = x^\star_{uv} = 1$. Notemos que, como $e \in \rho_f$, por definición de ρ_f, $\gamma_{uv}\frown\gamma_{vw}$ tiene un camino prohibido. Entonces, $w \in V_{uv}$ y la correspondiente restricción $M2$ quedaría como:

$$1 = x^\star_{vw} \leq 1 - x^\star_{uv} = 0$$

Entonces $(x^\star, y^\star, z^\star)$ no es factible para $M2$. Podemos concluir que el modelo con cualquiera de los dos conjuntos de restricciones tiene el mismo conjunto de soluciones factibles. □

En el ejemplo de la Figura 16, la relajación lineal del modelo con las restricciones (36) ($M1$) es más ajustada que la del modelo con las restricciones (39) ($M2$). Con penalidad por giro $p = 50$, el valor óptimo de la función objetivo de la relajación lineal de $M1$ es 718,1 mientras que el de $M2$ es 482,711. Más aún, la relajación lineal de $M1$ provee una solución factible (y por lo tanto óptima) para $M1$.

Soluciones iniciales para el modelo MILP

Es razonable esperar que en un recorrido eficiente el camión recolecte los residuos acumulados en una esquina y se desplace a otra que no se encuentre demasiado lejos. Con esta idea en mente, buscamos soluciones iniciales donde cada vértice $v \in V_Z$ solo pueda ser sucedido por t, vértices de otro sector de la zona o vértices w en su sector tales que $w \in N^{(k)}(v) \cap V_Z$, donde $N^{(k)}(v)$ es el k-vecindario[8] de v en R. Por lo general, tomamos $k = 2$ o $k = 3$.

Para esto, sea S_v el conjunto de vértices del sector al que pertenece v, añadimos el siguiente conjunto de restricciones al modelo:

$$\sum_{j \in S_v - N^{(k)}(v)} x_{vw} = 0 \quad v \in V_Z \tag{40}$$

y configuramos un límite de tiempo muy acotado para hallar una solución. Si el modelo modificado logra obtener una solución s, resolvemos el modelo sin (40) utilizando s como solución inicial.

[8] Definición 6

8. Asignación a zonas

Si los equipos de recolección deben recorrer más de una zona por día, la asignación a zonas también influye en la distribución de trabajo. Según lo analizado en nuestros casos de estudio, identificamos dos factores que influyen en la duración de la jornada laboral: la longitud de la ruta y la cantidad de residuos a recolectar. Según lo consultado con las autoridades y lo analizado en los datos disponibles, ambos factores son independientes. Una zona podría tener una ruta no muy extensa, pero si tiene considerable densidad poblacional, genera más desechos y se consume más tiempo para recolectarlos. Por el contrario, una zona podría tener una ruta extensa en una zona con baja densidad poblacional donde no se produzcan tantos residuos. Además, si bien suponemos que cada camión tiene capacidad suficiente para servir a cada zona individualmente, en el caso en el que un camión alcance su capacidad máxima antes de haber completado el recorrido de sus zonas, debe interrumpirlo para viajar a la planta de tratamiento, descargar el camión y volver al municipio para continuar. Esto conlleva una extensión significativa de la jornada laboral para el chofer y los recolectores.

En esta sección abordamos el problema de la asignación de equipos de recolección a zonas en el caso en el que no se dispone de datos para determinar cuantitativamente cuánto influye la distancia recorrida y la cantidad de residuos recolectados. Enfocamos este problema desde la perspectiva de la optimización biobjetivo: hallar una asignación que minimice la disparidad en longitud de recorrido y, al mismo tiempo, que minimice la disparidad en el peso a ser recolectado. Suponemos que a cada equipo se le deben asignar la misma cantidad de zonas. Si los residuos generados por las zonas asignadas a un camión superan su capacidad, suponemos que el camión se dirige a la planta de tratamiento entre el servicio a las zonas. Por ejemplo, si la capacidad del camión son 10 Tn. y tiene tres zonas asignadas con una generación de residuos de 5, 4 y 6 toneladas respectivamente, consideramos que luego de servir la primera zona, o luego de recolectar la segunda, regresa al Predio para dejar a los recolectores y desde allí se dirige a la planta de procesamiento a descargar el camión. Luego pasa a buscar a los recolectores por el Predio y se dirigen a la tercera zona. A continuación formulamos el problema biobjetivo y en la Tabla 7 presentamos sus conjuntos, parámetros y variables.

$$\text{mín} \quad f_1(x) = \max_{i,k \in T} \Big\{ \sum_{j \in Z} (x_{ij} - x_{kj}) w_j \Big\} \tag{41}$$

$$f_2(x, \theta) = \max_{i,k \in T} \Big\{ \sum_{j \in Z} (x_{ij} - x_{kj}) \ell_j + d(\theta_i - \theta_k) \Big\} \tag{42}$$

$$\text{s.a:} \quad \sum_{j \in Z} x_{ij} = n \qquad i \in T \tag{43}$$

$$\sum_{i \in T} x_{ij} = 1 \qquad j \in Z \tag{44}$$

$$\sum_{j \in Z} x_{ij} w_j \leq c_i \theta_i \qquad i \in T \tag{45}$$

$$\sum_{j \in Z} x_{ij} w_j \geq c_i (\theta_i - 1) \qquad i \in T \tag{46}$$

$$x_{ij} \in \{0, 1\} \qquad i \in T, j \in Z$$

$$\theta_i \in \mathbb{Z}_{\geq 0} \qquad i \in T$$

Notación	Descripción
Conjuntos	
T	el conjunto de equipos de recolección.
Z	el conjunto de zonas.
Parámetros	
c_i	la capacidad de carga del camión del equipo de recolección $i \in T$.
w_j	el peso de los residuos generados por la zona $j \in Z$.
ℓ_j	la longitud del recorrido de la zona $j \in Z$.
n	cantidad de zonas que deben asignarse a cada equipo de recolección.
d	distancia del viaje de ida y vuelta desde el Predio a la planta de tratamiento.
Variables	
x_{ij}	vale 1 si y solo si el equipo de recolección $i \in T$ es asignado a la zona $j \in Z$ ($x_{ij} \in \{0, 1\}$)
θ_i	cantidad de descargas en la planta de tratamiento del camión $i \in T$ ($\theta_i \in \mathbb{Z}_{\geq 0}$)

Tabla 7: Conjuntos, parámetros y variables del modelo de asignación

(41) expresa el objetivo de minimizar la máxima disparidad de carga y (42) el objetivo de minimizar la máxima disparidad de longitud de recorrido, penalizando los viajes adicionales a la planta de tratamiento. Con respecto a las restricciones del problema, (43) asegura que a cada camión sean asignadas exactamente n zonas , (44) garantiza que a cada zona le sea asignado exactamente un camión y (45)-(46) modelan que el camión realice la cantidad mínima de viajes necesarios a la planta de tratamiento según la cantidad de residuos generada por las zonas asignadas. Es trivial que la condición necesaria para la factibilidad del modelo es que $|Z| = n|T|$.

Es posible que no exista una solución que cumpla ambos objetivos simultáneamente [43]. Por lo tanto, se trata de hallar un conjunto de soluciones conocido como soluciones eficientes [124]. Sea X el espacio de soluciones factibles del problema, se define $f(x) = (f_1(x), f_2(x))$. Dados $u, v \in \mathbb{R}^2$, se dice que u domina a v si $(u_1 \leq v_1 \wedge u_2 < v_2) \vee (u_1 < v_1 \wedge u_2 \leq v_2)$ y se nota como $u \succ v$. Sea $\hat{x} \in X$, se dice que $\hat{x}$ es una solución eficiente si no existe $x \in X$ tal que $f(x) \succ f(\hat{x})$. En ese caso, se dice que $f(\hat{x})$ es un punto no dominado o Pareto-óptimo. Se denomina Frente de Pareto al conjunto de todos los puntos no dominados.

Para aproximar el Frente de Pareto empleamos el *Epsilon-Constraint Method* [62], que consiste en resolver el problema de optimización considerando solo uno de los objetivos e incluir el resto como restricciones del problema. En el caso de la optimización biobjetivo de minimización, se busca $\min_{x \in X}\{f_1(x)\}$ sujeto a que $f_2(x) \leq \varepsilon$. Modificando sistemáticamente el valor de ε se puede hallar el Frente de Pareto [73]. Se comienza resolviendo el problema con un único objetivo $\min_{x \in X} f_1(x)$. Sea x^1 su solución, a continuación se resuelve el problema $\min_{x \in X}\{f_1(x) | f_2(x) \leq f_2(x^1) - \delta\}$, y así sucesivamente, hasta que el problema resulte infactible. Finalmente, se resuelve $\min_{x \in X} f_2(x)$.

Seguidamente, presentamos la formulación del modelo MILP, que utilizamos para aplicar el Epsilon-Constraint Method.

$$
\begin{aligned}
\text{mín} \quad & y && && && (47)\\
\text{s.a:} \quad & \sum_{j\in Z} x_{ij} && = && n && i \in T && (48)\\
& \sum_{i\in T} x_{ij} && = && 1 && j \in Z && (49)\\
& \sum_{j\in Z} x_{ij} w_j && \leq && c_i\theta_i && i \in T && (50)\\
& \sum_{j\in Z} x_{ij} w_j && \geq && c_i(\theta_i - 1) && i \in T && (51)\\
& \sum_{j\in Z} (x_{ij} - x_{kj}) w_j && \leq && y && i,k \in T && (52)\\
& \sum_{j\in Z} (x_{ij} - x_{kj})\ell_j + d(\theta_i - \theta_k) && \leq && \varepsilon && i,k \in T && (53)\\
& x_{ij} && \in && \{0,1\} && i \in T, j \in Z\\
& \theta_i && \in && \mathbb{Z}_{\geq 0} && i \in T\\
& y && \in && \mathbb{R}
\end{aligned}
$$

La variable y se utiliza para acotar superiormente la diferencia de peso total recolectado entre cada par de equipos (52). La restricción (53) es la del Epsilon-Constraint Method. Es importante notar que si se pueden hallar $\alpha, \beta \in \mathbb{R}$ para cuantificar la influencia de la cantidad de residuos recolectados y la distancia recorrida en la duración de la jornada laboral mediante una expresión lineal $\alpha w_j + \beta \ell_j$, la asignación puede llevarse a cabo con el siguiente modelo MILP, sin necesidad de utilizar el Epsilon-Constraint Method:

$$
\begin{aligned}
\text{mín} \quad & y && && && (54)\\
\text{s.a:} \quad & \sum_{j\in Z} x_{ij} && = && n && i \in T && (55)\\
& \sum_{i\in T} x_{ij} && = && 1 && j \in Z && (56)\\
& \sum_{j\in Z} x_{ij} w_j && \leq && c_i\theta_i && i \in T && (57)\\
& \sum_{j\in Z} x_{ij} w_j && \geq && c_i(\theta_i - 1) && i \in T && (58)\\
& \sum_{j\in Z} (x_{ij} - x_{kj})(\alpha w_j + \beta \ell_j) + d(\theta_i - \theta_k) && \leq && y && i,k \in T && (59)\\
& x_{ij} && \in && \{0,1\} && i \in T, j \in Z\\
& \theta_i && \in && \mathbb{Z}_{\geq 0} && i \in T\\
& y && \in && \mathbb{R}
\end{aligned}
$$

9. Aplicación a caso de estudio: Berazategui

En la Tabla 8 se encuentra un breve resumen de las características de la recolección en el municipio y de los objetivos del proyecto. Como la jornada laboral de los recolectores y de los choferes no es la misma, consideramos necesario definir distintos criterios para medir la carga de trabajo de cada uno. Con tal motivo, mantuvimos entrevistas con supervisores y choferes. Por un lado, cada grupo de recolectores utiliza el metraje como medida de cantidad de trabajo. Recordemos que deben recorrer todas las cuadras de la zona asignada, por lo tanto, la cantidad de trabajo de un grupo de recolectores viene dada por la suma de la longitud de las cuadras de la zona que tiene asignada. Por otro lado, según la experiencia de los choferes, la duración de su jornada laboral se ve afectada por dos factores independientes entre sí: la distancia recorrida por el camión y la cantidad de residuos recolectados. En el área incluida en el estudio de caso, se presta servicio alrededor de 15000 cuadras agrupadas en 42 zonas todos los días.

Características y objetivos del proyecto en Berazategui	
Características	▪ Los camiones parten del predio de Higiene Urbana (HU). ▪ Cada chofer recorre dos zonas por día. ▪ Cada equipo de recolectores a pie recorre una zona por día. ▪ El chofer descarga el camión al final de su jornada laboral. ▪ Se pueden achicar las cuadras de longitud menor o igual a 130 mts.
Objetivos	▪ Diseñar una nueva zonificación cuyas zonas estuvieran compuestas por manzanas adyacentes y, en lo posible, tuvieran forma cuadrada. ▪ Definir rutas de recolección eficientes en cuanto a la cantidad de kilómetros recorridos. ▪ Disminuir la disparidad en la distribución del trabajo entre recolectores y entre choferes.

Tabla 8: Resumen de las características y los objetivos del caso de estudio de Berazategui

La diferencia en la cuantificación de carga laboral nos llevó a desarrollar un enfoque de solución que dividiera el problema en tres etapas, resueltas secuencialmente:

1. Zonificación: utilizamos la heurística desarrollada en la Sección 5 para construir una zonificación que balancee el metraje entre las zonas y así distribuir el trabajo entre los recolectores a pie. Como objetivo secundario, se busca que las zonas tengan forma cuadrada.

2. Ruteo: por la presencia de giros prohibidos con cuatro vértices, utilizamos el Enfoque $F_4 \neq \emptyset$ (Sección 7.1) para hallar soluciones eficientes al MGRPTP.

3. Asignación de choferes a zonas: a partir de los resultados de las etapas anteriores, para cada zona podemos estimar la cantidad de residuos que genera y conocemos la longitud de una ruta eficiente. Utilizamos el enfoque propuesto en la Sección 8 para asignarle dos zonas a cada chofer.

A continuación presentamos los resultados obtenidos en cada una de esas etapas. Empleamos Python 3.7 para la implementación y los modelos MILP se resolvieron utilizando CPLEX 12.10 y su API para Python. Los tiempos de ejecución corresponden a una computadora con procesador Intel Core i5 2.60GHz con 16 GB de RAM.

Los resultados son constrastados con la situación en Berazategui antes del comienzo del proyecto. Más específicamente, comparamos el resultado de zonificación con la que estaba en uso en marzo de 2020; los resultados de la distancia promedio de la ruta de los camiones con los determinados usando datos GPS para el período de julio de 2019 a febrero de 2020; y los resultados de la asignación de camión a zona con los que estaban vigentes en marzo de 2020.

9.1. Zonificación

Desde Higiene Urbana nos indicaron que deseaban mantener algunas de las zonas con las que contaban y definir manualmente otras. Esto se debía a que en ciertos lugares de la ciudad la recolección de residuos tenía una organización especial que no les resultaba conveniente modificar: el barrio El Pato, el barrio Juan María Gutiérrez, los barrios cerrados y algunos sectores céntricos.

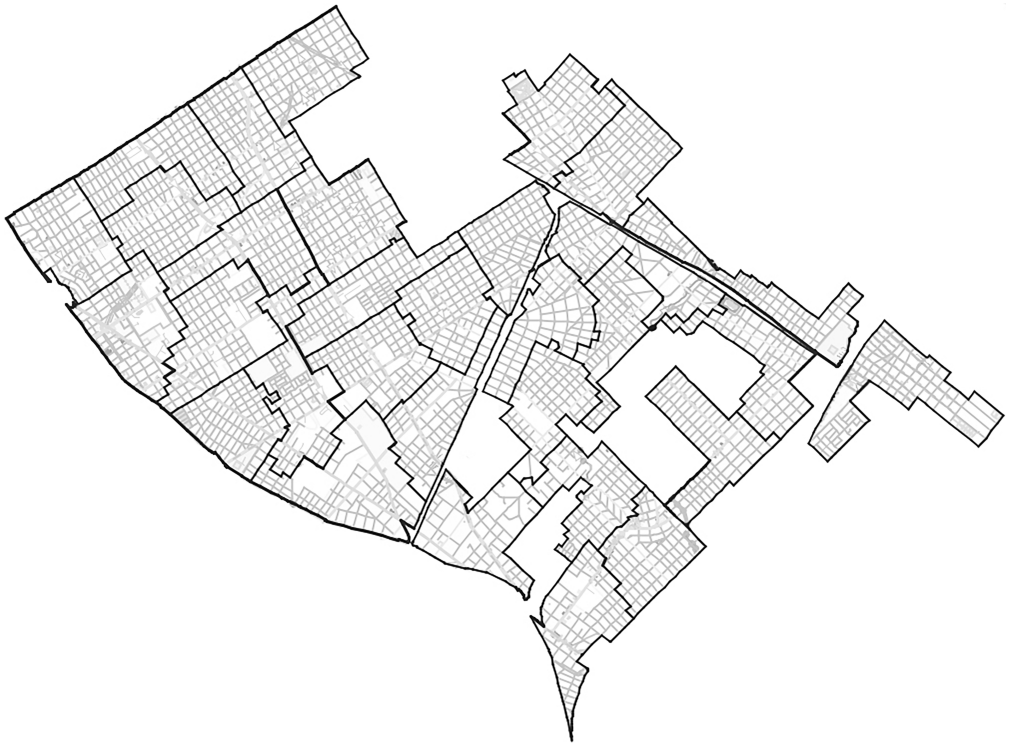
Por otro lado, para la heurística de zonificación, consideramos las manzanas que cumplieran las siguientes restricciones. En primer lugar, una manzana no debe ser atravesada por vías de ferrocarril ni por carriles de una autopista; de esta manera, se busca evitar demoras en el recorrido debido a desvíos en la ruta del camión recolector. En segundo lugar, una manzana debe tener un perímetro menor o igual a 2200 metros, debido a la existencia de espacios no urbanizados que no forman parte de ninguna zona de recolección. Ese límite, obtenido a partir de pruebas empíricas, permite excluirlos.

Descartar las manzanas que no cumplían con esas características desconectó al grafo de manzanas. Por lo tanto, aplicamos la heurística a cada una de sus componentes conexas, previamente calculándoles el metraje de recolección y, con el asesoramiento de Higiene Urbana, determinando en cuántas zonas dividirla.

Con respecto al parámetro λ para la medida de cuadratura S_λ, tras diversas pruebas experimentales, $\lambda = 0{,}75$ resultó el valor más adecuado para realizar el proceso de zonificación en el plano de Berazategui buscando que las zonas se asemejen lo más posible a cuadrados. Con el asesoramiento de Higiene Urbana, determinamos la cantidad mínima deseable de metraje de una zona como $m_{MIN} = 3150$. Por otra parte, utilizamos $\alpha = \frac{1}{5}\pi$ para definir los α-vértices de la Fase II de la heurística.

Las zonas que obtuvimos después de aplicar las dos fases de la heurística se muestran en la Figura 21. La Tabla 9 muestra los metrajes mínimos y máximos de recolección y sus respectivas desviaciones estándar para las zonas diseñadas manualmente anteriormente en el municipio, junto con los valores correspondientes que obtuvimos en la primera y la segunda fase de la heurística.

Como se puede apreciar en la tabla, la heurística generó un mejor balance del metraje, que varía en un rango mucho más estrecho que en el caso del sistema manual. En efecto, el desvío estándar se reduce en un 75,8 % y la diferencia entre el máximo y el mínimo de metraje, en un 83,4 %. La Fase II mejoró la cuadratura de las zonas, al tiempo que introdujo solo



(a)



(b)

Figura 21: (a) Zonificación resultante de aplicar la heurística a las áreas del plano que se deseaban rezonificar. (b) Zonificación resultante de aplicar la Fase II a la zonificación de (a), que afectaron a las manzanas sobreadas en gris.

	Manual	Fase I	Fase II
Desvío estándar	7,84 km	1,82 km	1,89 km
Máximo	46,15 km	26,62 km	27,13 km
Mínimo	5,03 km	20,53 km	20,34 km

Tabla 9: Comparación de metraje entre la zonificación manual y las obtenidas con ambas fases de la heurística

un ligero deterioro en la variación de la distancia en comparación con la primera fase. Esto pareciera corroborar que los objetivos de equilibrio y cuadratura compiten entre sí, como suele ocurrir en problemas multiobjetivo. Además de lograr una zonificación con una distribución más equitativa de la carga de trabajo entre los recolectores a pie, todas las zonas generadas son conexas mientras que en la zonificación manual seis zonas estaban divididas en dos o más sectores.

Según las indicaciones de Higiene Urbana, implementamos algunas modificaciones puntuales sobre algunas zonas con el objetivo de mejorar su cuadratura y considerando aspectos demográficos. Estos cambios causaron la desconexión de algunas zonas, principalmente por estar atravesadas por una vía de ferrocarril. De esta manera, obtuvimos la nueva zonificación del municipio y seguimos con el diseño de las rutas.

9.2. Ruteo

Como los camiones deben dar servicio a dos zonas diariamente, tomamos en cuenta la capacidad en las asignaciones de camión a zona. En los casos en los que el total de desechos del día completo exceda la capacidad, asumimos que el conductor hace el viaje adicional a la planta de tratamiento luego de completar la primera zona y antes de comenzar la segunda.

Diseñamos las rutas para todas las zonas, excepto El Pato y los barrios cerrados. Asesoramiento de Higiene Urbana mediante, establecimos el umbral de achique en 130 metros. Configuramos el solver para la resolución del modelo MILP de la siguiente manera. Si después de 45 minutos el solver aún no había encontrado una solución óptima, durante los 45 minutos siguientes aceptaríamos una solución con un *gap* de optimalidad menor a 1 %. Si agotaba ese límite de tiempo sin éxito, extendemos su tiempo de ejecución en 90 minutos configurándolo para mejorar la cota inferior del valor óptimo, dado que observamos que después de los primeros 90 minutos el solver generalmente no encontraba mejores soluciones.

De las 38 zonas para las que se diseñaron rutas, se encontraron óptimas para 20 con tiempos de solución promedio de 11 minutos y 25 segundos. Para otras 17 zonas, nuestro enfoque de solución identificó rutas con un gap de optimalidad de no más de 1 % y un tiempo de solución promedio de 87 minutos. En estos últimos casos, el gap de optimalidad promedio fue de 0,7 %, mientras que el gap absoluto promedió 172,54 m. El tamaño de estas instancias varió entre 52 y 195 vértices, con un promedio de 139. El tiempo de solución para la única zona restante, que cubría el distrito de Juan María Gutiérrez, fue el doble de largo dado que tiene 308 vértices, más del doble del promedio para todas las demás. En este caso, ampliamos el gap de optimalidad aceptable a 5 % y el solver encontró una solución después de 164 minutos

con un gap de optimalidad de 3,31 % y un gap absoluto de 1,61 km.

Las rutas definidas manualmente, utilizadas antes del proyecto, tenían una longitud promedio de 62,87 km. Nuestro enfoque de solución disminuyó esa cifra a 48,63 km., logrando una reducción de 22,6 %. Dado que las rutas fueron diseñadas para una flota de 20 camiones de recolección, esta mejora se traduce en 284, 79 kilómetros menos recorridos diariamente en todo el sistema. En el servicio de recolección de seis días a la semana de Berazategui, la reducción en la longitud total de la ruta de los camiones sobre una base anual totalizaría 88854,48 km, lo que se traduce en ahorros de alrededor de USD 25000 anuales[9] en combustible y en una disminución en el desgaste de los vehículos. Nótese también que la distancia total de recolección en las zonas para las que diseñamos rutas es de aproximadamente 884 km., mientras que la longitud total de la ruta de los camiones (es decir, excluyendo el viaje hacia y desde el edificio de Higiene Urbana) es de 649 km. Esto implica que achique ahorra al menos 234 km. en el viaje total de los camiones por zona.

A modo de ejemplo, las rutas que obtuvimos para dos zonas se muestran en la Figura 22, que ilustra cómo el achique reduce la cantidad de cuadras recorridas por los vehículos. Para simplificar los diagramas, no se muestra el recorrido hacia y desde el edificio de Higiene Urbana.



Figura 22: Las filas superior e inferior ilustran una ruta eficiente (línea negra) diseñada por el enfoque propuesto para una zona (sombreada en gris) Para mayor claridad, los diagramas de la columna izquierda muestran solo una parte de las rutas (el punto representa la posición del camión) y los diagramas de la columna derecha muestran las rutas completas.

[9]Según el precio del combustible en septiembre de 2021

9.3. Asignación a zonas

La última etapa del trabajo consistió en asignar cada uno de los camiones a las zonas de recolección, con el objetivo de equilibrar la duración de la jornada laboral de los choferes. Recordemos que según el esquema de recolección de Berazategui, a cada chofer le corresponden dos zonas y la secuencia diaria suele ser la siguiente: el camión parte de Higiene Urbana con el equipo de recolectores del turno mañana, recorre la primera zona, regresa al predio de Higiene Urbana para cambiar el equipo de recolectores, recorre la segunda zona, visita Higiene Urbana para dejar a los recolectores, viaja a CEAMSE a descargar los residuos y finalmente el camión es regresado a Higiene Urbana. Los últimos dos pasos de la secuencia son comunes a todos a los camiones recolectores y, por lo tanto, no son tenidos en cuenta al momento de calcular la longitud total de su recorrido. Se considera que la longitud del recorrido de una zona viene dada por la longitud de la ruta obtenida en la subsección anterior.

Para esta etapa necesitamos estimar la cantidad de residuos que generaba cada zona diseñada por nuestro método. Con tal fin, aproximamos la cantidad de residuos generados por cada manzana del plano. El primer paso fue cuantificar cuántas toneladas generaba cada zona original. Para esto, implementamos una rutina que examina el recorrido diario de cada camión obtenido a partir de los datos GPS y reconoce qué zonas recolectó. Esta información luego fue procesada junto con la cantidad de residuos descargados y con los datos demográficos proporcionados por la municipalidad para estimar los residuos generados por cada zona original del municipio. Luego, para cada zona, se repartía la cantidad de residuos estimada entre sus manzanas, ponderando según cuánta gente habitaba en cada una, la cantidad de comercios, establecimientos públicos, etc.

A partir del cálculo aproximado de la cantidad de residuos que genera cada zona y la longitud del recorrido que le corresponde, se asignaron dos zonas a cada camión, de manera tal que la carga laboral de los choferes sea lo más balanceada posible. Aplicamos el enfoque propuesto en la Sección 8: el Epsilon-Constraint Method junto al modelo MILP (47)–(53). El valor inicial de ε fue dado por la diferencia máxima de longitud de ruta de la asignación que minimizó la diferencia en el peso de la carga de desechos mientras ignoraba el otro objetivo. En cada iteración, este parámetro disminuyó en $\delta = 250$ m. hasta que el problema se volvió infactible. Finalmente, calculamos la solución tomando como único objetivo la diferencia en la longitud de la ruta. El resultado aproximó el frente de Pareto para la optimización bi-objetivo.

El epsilon-constraint method identificó un total de 24 soluciones Pareto-óptimas. En Figura 23 se muestra la aproximación del frente de Pareto. Antes de comenzar con esta etapa del proyecto, a partir de los resultados de las dos etapas anteriores, el municipio elaboró una asignación manual de camiones a zonas, la cual es incluida en Figura 23 a modo de comparación. Debido a que se pretendía mantener las asignaciones existentes para los barrios El Pato y Juan María Gutiérrez, la zona céntrica y los barrios cerrados, estos fueron excluidos de esta etapa. En la figura también se destaca la asignación definitiva elegida por el municipio. La decisión final de cuál elegir fue producto de la percepción y el conocimiento del problema de parte de los responsables operativos de la recolección.

En la situación previa a nuestro trabajo, la diferencia máxima de peso recolectado entre los camiones era de aproximadamente 6070 kg. y la disparidad máxima de kilómetros recorridos era de aproximadamente 44,5 km. En la asignación de nuestra propuesta elegida por

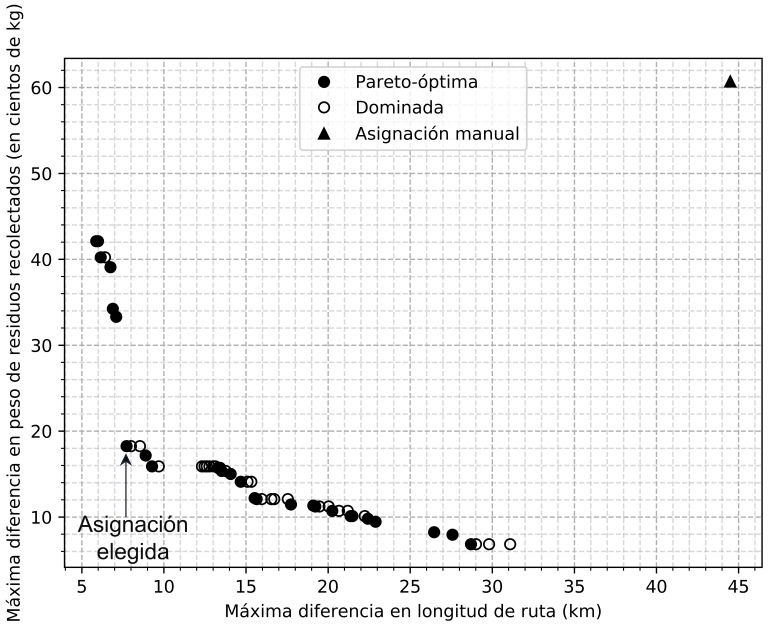


Figura 23: Aproximación del frente de Pareto.

la Municipalidad, dichas diferencias son de 1825 kg. y 7,72 km., respectivamente. Uno de los factores que contribuyó a mejorar el balance de trabajo entre los choferes fue el rediseño de las zonas de recolección que permitió disminuir las zonas no conexas. Por otro lado, el proceso de asignación permitió corregir la situación en la que ambas zonas de algunos choferes estaban geográficamente cerca de Higiene Urbana mientras que las de otros se encontraban más alejadas.

La zonificación fue entregada a las autoridades de Higiene Urbana en un plano interactivo y la ruta diseñada para cada zona fue presentada en tres formatos: un archivo `html` con la animación de la ruta sobre el plano del municipio, que permite una visualización sencilla y rápida; una hoja de ruta tradicional con la secuencia de indicaciones de giro y un archivo `gpx` para ser utilizado en la aplicación OsmAnd[10], que proporciona una guía GPS incluso si el usuario se encuentra offline.

Las autoridades de Higiene Urbana estuvieron satisfechas con los resultados obtenidos y los implementaron a partir de abril de 2020. Según nos comunicaron meses después, hubo mucha resistencia inicial por parte de los trabajadores, acostumbrados al antiguo sistema que había estado vigente durante muchos años. Además, algunos de ellos tenían la impresión de que las áreas de las zonas eran proporcionales a la distancia de recolección y, por lo tanto, sentían que la carga de trabajo estaba distribuida injustamente. Sin embargo, el Departamento pudo demostrar que la nueva distribución era de hecho más justa que antes, dado que, entre otras cosas, la cantidad de residuos recolectados, como lo demostraban los comprobantes de descarga de CEAMSE, era similar entre todos los camiones. Esto, junto con algunas modificaciones manuales menores a las zonas y a las rutas realizadas por el Departamento después de consultar

[10] https://osmand.net/

con los supervisores del sistema, poco a poco superó las objeciones de los trabajadores.

Por otra parte, como se puede apreciar en la Figura 22, las rutas que resultaron de nuestro modelo tienen demasiados giros, dificultando la organización del achique y, sobre todo, las maniobras que debe realizar el chofer. Por lo tanto, esta característica también resultó un obstáculo para la implementación. En el desarrollo de un enfoque de solución para el partido de Quilmes tuvimos en cuenta los aspectos que podrían mejorarse para facilitar la implementación de los resultados obtenidos.

10. Aplicación a caso de estudio: Quilmes

En la Tabla 10 recordamos las características y los objetivos de la optimización de la recolección en Quilmes. La diferencia más notoria con Berazategui es que tanto los recolectores como los choferes recorren una zona por día, por lo cual no es necesario llevar adelante un proceso de asignación de equipos a zonas. Además, dado que contamos con los datos necesarios, definimos una medida de carga laboral a partir de la longitud de la ruta y de la cantidad de residuos recolectados para abordar simultáneamente el problema de zonificación y de ruteo.

Características y objetivos del proyecto en Quilmes	
Características	▪ Los camiones parten del predio de la Secretaría de Ambiente y Gestión Integral de Residuos (GIRSU). ▪ Cada equipo de recolección (chofer y recolectores a pie) recorre una zona por día. ▪ El equipo de recolección descarga el camión al final de su jornada laboral en alguna de las dos plantas de procesamiento. ▪ Se pueden achicar las cuadras de longitud menor o igual a 140 mts.
Objetivos	▪ Modificar la zonificación actual para disminuir la disparidad en la distribución del trabajo entre equipos de recolección. ▪ Definir rutas de recolección eficientes en cuanto a la cantidad de kilómetros recorridos.

Tabla 10: Resumen de las características y los objetivos del caso de estudio de Quilmes

Por otra parte, tomamos en cuenta los aprendizajes que nos dejó el proyecto anterior. Como la resistencia al cambio puede comprometer la implementación de los resultados, enfocamos el problema desde la perspectiva de mejorar la recolección realizando cambios a las zonas vigentes, en vez de elaborar una zonificación completamente nueva. Para esto, utilizamos los métodos de la Sección 6. Con respecto al ruteo, incluimos la penalización por giros con el fin de simplificar las rutas para los choferes. Como en el caso de Quilmes todos los giros prohibidos consistían en exactamente tres vértices, utilizamos el Enfoque $F_4 = \emptyset$ para el ruteo (Sección 7.2).

Antes de presentar los resultados obtenidos, definimos la función que mide la carga laboral de una zona.

10.1. Medida de carga laboral

Como definimos en la Sección 6, consideramos que la disparidad en la distribución laboral puede disminuir si se achica la brecha entre las zonas con carga laboral máxima y mínima. Notamos Φ a la función que mide la carga laboral de una zona. A partir de lo informado por las autoridades de GIRSU, y tal como ocurría en el caso de Berazategui, observamos que dos factores de gran influencia en la duración de la jornada laboral son la distancia recorrida y la cantidad de residuos recolectados. Para este caso de estudio, consideramos la distancia mixta en vez de la distancia real.

La distancia mixta de una zona Z viene dada por una ruta eficiente obtenida con el Enfoque $F_4 = \emptyset$. Para estimar la cantidad de residuos generada por cada manzana del municipio, utilizamos los datos del Precenso de Viviendas de 2020 [74]. Identificamos las manzanas pertenecientes a cada radio censal y distribuimos equitativamente entre ellas la cantidad de habitantes. Luego, se distribuye la cantidad de residuos promedio generada por cada zona entre sus manzanas según su proporción de habitantes con respecto a la cantidad total de habitantes en la zona. De esta manera, para aproximar la cantidad de residuos generada por cada zona basta sumar los residuos generados por cada una de sus manzanas. A continuación, definimos la función de carga laboral para una zona.

Definición 50 (Función de carga laboral)**.** Sean Z una zona, $W(Z)$ la suma de la cantidad de residuos generados por cada una de sus manzanas y $D(Z)$ la distancia mixta de un recorrido eficiente para Z, definimos la función de carga laboral como:

$$\Phi(Z) = \beta_1 D(Z) + \beta_2 W(Z) \quad \beta_1, \beta_2 \in \mathbb{R}$$

A partir de la información proporcionada por la municipalidad, armamos un dataset con 1970 datos sobre tres aspectos de la recolección diaria de las zonas: distancia mixta en kilómetros, cantidad de residuos en toneladas y tiempo de recorrido. Realizamos una regresión lineal sin término independiente para aproximar el tiempo (en horas) en función de la distancia mixta (en km.) y la cantidad de residuos (en Tn.). Los valores obtenidos fueron $\beta_1 = 0{,}0931$ y $\beta_2 = 0{,}194$. Con respecto a las medidas de error, la raíz del Error Cuadrático Medio es 36,88 minutos mientras que el Error Absoluto Medio es 28,83 minutos.

10.2. Resultados computacionales

En esta subsección describiremos cómo determinamos los parámetros (Tabla 11) de nuestro enfoque de solución para el caso de estudio de Quilmes y mostraremos los resultados de su aplicación. Compararemos las tres métodos basados en las metaheurísticas de Steepest-Ascent Hill Climbing (SAHC), Simulated Annealing (SA) y Variable Neighbourhood Search (VNS) propuestos en la Sección 6. Con respecto a la información disponible, contamos con los datos de GPS de los camiones para los meses de julio y agosto de 2023, la cantidad de residuos desde mayo hasta agosto de 2023 descriminados por zona y por camión.

Nuestra estrategia de solución fue implementada en Python 3.10 y ejecutada en el Centro de Computación de Alto Rendimiento ($CeCAR$[11]). Para resolver los modelos MILP, utilizamos CPLEX 22.11.

Parámetros

A partir de un análisis empírico de las rutas de los camiones y el intercambio con las autoridades de GIRSU, consideramos que $\alpha = \frac{1}{5}\pi$ es un valor apropiado para el ángulo que determina si existe un giro entre dos arcos del grafo de ruteo y que $\ell = 140$ es la longitud máxima de una cuadra que puede achicarse.

Con respecto a los coeficientes de la distancia ponderada, utilizamos los datos de GPS de diez camiones para hallar los valores de $v_{(t,c)}$ y $\hat{v}_{(t,c)}$ para cada turno t y tipo de calle c.

[11] `https://cecar.fcen.uba.ar/`

Parámetro	Descripción
ℓ	el umbral de achique
α	ángulo que sirve como argumento para determinar si existe un giro entre dos arcos del digrafo de ruteo
p	la penalización por realizar un giro, utilizada en el cálculo de caminos de costo mínimo s.c.p. y en el modelo MILP de ruteo.
$v_{(t,c)}$	la velocidad media en calles de tipo c en el turno t durante la recolección, utilizada en los coeficientes para la distancia ponderada
$\hat{v}_{(t,c)}$	la velocidad media en calles de tipo c durante el turno t fuera de la zona de recolección, utilizada en los coeficientes para la distancia ponderada
β_1, β_2	los coeficientes para la definición de Φ, la función que mide la carga laboral de una zona (Definición 50)
T_0	temperatura inicial para el método de Simulated Annealing
δ	parámetro que regula la disminución de la temperatura para el método basado en Simulated Annealing

Tabla 11: Resumen de parámetros del enfoque de solución.

Construimos un dataset con 764227 datos correspondientes a segmentos recorridos por los camiones y sus correspondientes características de interés: velocidad, tipo de calle, turno en el que fue recorrido y si se encontraba dentro de la zona de recolección. Tomamos el promedio para cada tipo de calle en cada turno, dentro y fuera de la zona de recolección, y determinamos los coeficientes para la distancia ponderada calculando $\frac{v_{(t,c)}}{\hat{v}_{(t,c)}}$. En la Tabla 12 mostramos el resultado de este cociente para cada turno y tipo de calle. Como en el caso de estudio ninguna autopista formaba parte de una zona de recolección, el coeficiente fue calculado utilizando la media de velocidad de recolección en avenidas como divisor.

	Mañana	Tarde	Noche
Residencial	0,568	0,52	0,608
Avenida	0,465	0,541	0,489
Autopista	0,2	0,196	0,205

Tabla 12: Cociente $\frac{v_{(t,c)}}{\hat{v}_{(t,c)}}$ para cada tipo de calle en cada turno.

La inclusión de la penalidad de giro p en el procedimiento de ruteo transforma la búsqueda de rutas eficientes en un problema multiobjetivo: minimizar la distancia mixta y minimizar la cantidad de giros. Por lo tanto, es esperable que el tiempo de resolución del modelo MILP aumente. Teniendo esto en cuenta, nos interesó analizar cuánto empeora la distancia mixta y cuánto aumenta el tiempo de resolución a medida que crece p. Para un conjunto de zonas, buscamos recorridos eficientes para valores de p en el intervalo $(0, 100]$. Comparamos la distancia mixta, la cantidad de giros y el tiempo de resolución de los resultados obtenidos con las soluciones donde $p = 0$. También, para cada zona, calculamos la cantidad mínima de giros utilizando el mismo MILP que para ruteo, sin los coeficientes de distancia mixta en la función objetivo. En la Figura 24 mostramos resultados de la comparación para dos zonas

del conjunto. Los gráficos de la fila superior muestran la proporción de distancia mixta y de cantidad de giros del recorrido dentro de la zona con respecto a la solución con $p = 0$; la línea punteada indica cuánto debería decrecer la cantidad de giros para alcanzar su mínimo. En la fila inferior, se muestra el tiempo de resolución del modelo MILP para cada valor de p. Basándonos en los resultados obtenidos, decidimos utilizar $p = 50$, pues la cantidad de giros se reduce considerablemente sin comprometer demasiado la distancia mixta y con un impacto moderado en el tiempo de solución. En la Figura 25 mostramos la comparación de la ruta obtenida con $p = 0$ y con $p = 50$ para una zona, donde se evidencia que añadir la penalización a los giros efectivamente simplifica la ruta.

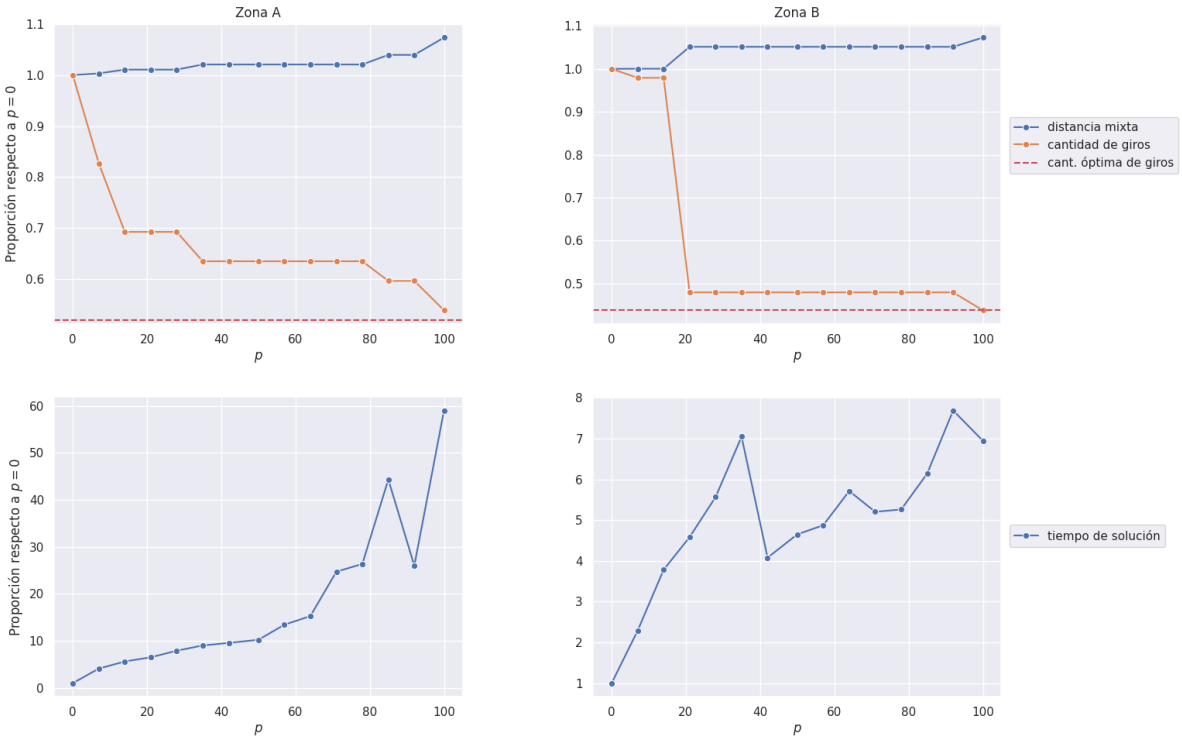


Figura 24: Comparación de cantidad de giros, distancia mixta y tiempo de ejecución entre distintos valores de p para dos zonas.

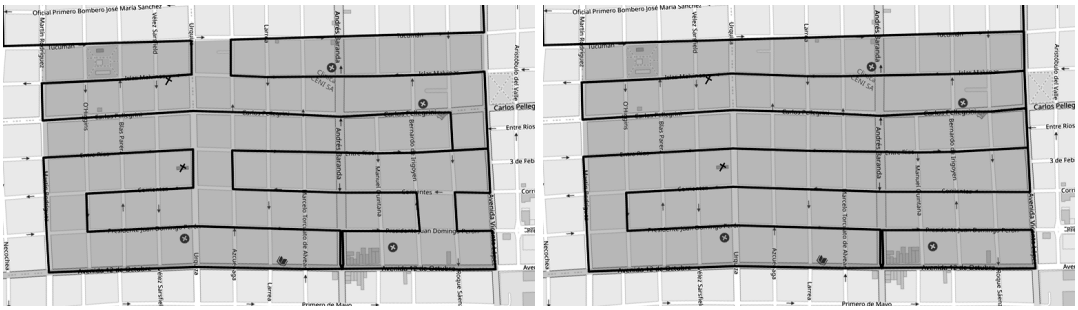
Finalmente, determinamos los parámetros para el método basado en Simulated Annealing. Al comienzo del procedimiento, quisiéramos que la probabilidad de aceptar un movimiento que empeore el valor de la función objetivo $\mathcal{F}$ (movimiento *uphill*) sea de 0,8. Por otra parte, recordar que en cada iteración dicha probabilidad está dada por:

$$e^{\frac{-\Delta F}{T}}$$

A partir de esto, podemos proponer un valor para T_0:

$$T_0 = -\frac{\overline{\Delta F}}{\ln(0{,}8)}$$

donde $\overline{\Delta F}$ es el promedio de los ΔF para movimientos uphill registrados en la ejecución del



(a) (b)

Figura 25: Comparación de rutas obtenidas para una zona con $p = 0$ (a) y $p = 50$ (b). La ruta con $p = 0$ tiene 31 giros en la zona, mientras que la de $p = 50$ tiene 16 y es 68 m. mas larga.

método basado en Steepest-Ascent Hill Climbing. En nuestra experimentación, obtuvimos el valor $T_0 = 0{,}38$.

Para elegir el parámetro de disminución de temperatura, establecimos que luego de 50 iteraciones quisiéramos que la probabilidad de aceptar un movimiento uphill fuera de alrededor de 0,01. Por lo tanto, propusimos:

$$\delta = \sqrt[50]{\frac{-\overline{\Delta F}}{T_0 \ln(0{,}01)}}$$

En nuestro caso, $\delta = 0{,}94$.

Resultados

Para aplicar los métodos propuestos en la Sección 6, en primer lugar debimos calcular el valor de la función objetivo F en la zonificación inicial $\mathcal{Z}_0$, lo cual implicó calcular una ruta eficiente para cada de sus zonas. Para esto, aplicamos el Enfoque $F_4 = \emptyset$ (Sección 7). El grafo mixto del plano tiene 57507 vértices, 68464 aristas y 28395 arcos. En su correspondiente grafo de ruteo calculamos el camino de costo penalizado mínimo sin giros prohibidos desde el punto de partida de los camiones (GIRSU) hacia todas las esquinas del municipio y desde ellas hasta el punto de llegada de los camiones (GIRSU, pasando por la planta de tratamiento) para cada turno (mañana, tarde y noche) considerando la distancia ponderada. A la zonificación con las rutas optimizadas la notaremos $\mathcal{Z}'_0$.

El límite de tiempo para la resolución del modelo MILP de ruteo fue de 30 minutos y le indicamos al solver que enfatice la aplicación de heurísticas para obtener buenas soluciones, configurando en 5 el parámetro `parameters.emphasis.mip`. En las Tablas 13 y 14 describimos características de cada una de las zonas y de la resolución del modelo. Recordar que $|V_Z|$ nota la cantidad de esquinas de la zona, $|\hat{A}|$ y $|\hat{E}|$ la cantidad de arcos y aristas, respectivamente, que deben recorrerse obligatoriamente y $|F_3^{(Z)}|$ la cantidad de restricciones de tránsito dentro de la zona. En las tablas también se indican el número de sectores de cada zona, el tiempo de resolución del modelo en segundos y el gap de optimalidad para los casos en los que no se halló solución óptima. Se pudieron resolver a optimalidad 72 de las 84 zonas iniciales, y el tiempo promedio de solución de todas las zonas fue de 389 segundos. En promedio, el gap

absoluto de las rutas de las zonas que no pudieron resolverse a optimalidad fue de 513,96 m. Comparamos las rutas eficientes obtenidas con la situación previa al proyecto para 82 de las 84 zonas, pues no contamos con datos de GPS para dos de ellas. Las rutas halladas disminuyen en un 10,93 % (557,09 km.) la distancia total diaria recorrida por la flota respecto al promedio de la situación previa. En cuanto a la distancia mixta, la reducción es del 17,6 %.

Para comparar el impacto de la optimización de las rutas en el valor de la función objetivo para la zonificación inicial, tuvimos en cuenta 78 de las 84 zonas, pues para seis de ellas faltaban datos de GPS o de cantidad de residuos recolectados que imposibilitaban calcular el valor de Φ. En la situación previa al proyecto, el valor de F se situaba en 2,24. Con las rutas optimizadas, este valor disminuye a 1,81, logrando una reducción del 19,03 %. Más aún, el valor máximo de Φ disminuye un 16,16 %, mostrando el impacto que tiene el diseño de rutas eficientes no solo en la distribución de trabajo, sino también en la reducción de la duración de la jornada laboral.

De igual manera calculamos las rutas eficientes para las 84 zonas sin penalidad por giro (es decir, con $p = 0$) para analizar más profundamente su efecto en el desempeño del modelo. En la Tabla 15 mostramos la comparación de algunas características de las soluciones obtenidas. Como puede apreciarse, las diferencias en el tiempo de solución y en la cantidad de giros son significativas, pero la distancia mixta del recorrido no varía demasiado. Por esta razón, en los métodos iterativos, decidimos calcular las rutas eficientes con $p = 0$ y con una tolerancia de gap de optimalidad del 5 %. Cada tres iteraciones del método y al satisfacerse algún criterio de parada, llevamos adelante una etapa de *corrección*, donde recalculamos las rutas eficientes para cada zona modificada durante esas tres iteraciones con $p = 50$ y tolerancia de gap de optimalidad de 1 %. En el Algoritmo 17 mostramos el pseudocódigo general para adaptar esta idea a los métodos de la Sección 6.

Algoritmo 17: Adaptación de los métodos iterativos propuestos en la Sección 6 para calcular las rutas con $p = 0$ y recalcularlas con $p = 50$ cada 3 iteraciones o al satisfacer algún criterio de parada.

```
it ← 1
t ← 0
Inicializar cooldown^(e) y cooldown^(c)
zonas_modificadas ← {}
mientras it ≤ iter_max y t < tiempo_limite hacer
    Modificar 𝒵 según indique el método, usando p = 0 para el ruteo
    Agregar las zonas modificadas en la iteracion a zonas_modificadas
    it ← it + 1
    t ← tiempo trascurrido
    si it ≡ 0 (mód 3) o se cumple algún criterio de parada entonces
        para cada Z ∈ zonas_modificadas hacer
            Calcular una ruta para Z utilizando p = 50.
        zonas_modificadas ← ∅
devolver 𝒵
```

Ejecutamos los tres métodos iterativos sobre $\mathcal{Z}_0'$ añadiendo la etapa de corrección cada

zona	$\lvert V_Z \rvert$	$\lvert \hat{A} \rvert$	$\lvert \hat{E} \rvert$	$F_3^{(Z)}$	sectores	tiempo (s.)	gap
1	70	1	8	200	1	180	-
2	52	5	8	119	1	1.82	-
3	66	8	2	123	1	5.2	-
4	117	19	9	168	1	290	-
5	130	11	2	96	1	8.44	-
6	111	7	5	128	1	10.4	-
7	109	0	2	102	1	1.23	-
8	98	0	0	250	1	58.7	-
9	129	0	6	357	1	127	-
10	101	0	5	306	1	53.7	-
11	107	4	1	110	1	5.61	-
12	102	13	4	158	1	2.81	-
13	135	0	6	381	2	386	-
14	120	1	8	381	1	128	-
15	107	2	1	166	1	7.44	-
16	77	0	0	208	1	10.4	-
17	116	12	4	131	1	17.1	-
18	100	2	25	332	1	114	-
19	68	0	60	224	1	173	-
20	131	3	3	176	1	653	-
21	69	0	14	220	1	8.23	-
22	97	0	1	258	1	6.2	-
23	77	38	20	170	1	1.68	-
24	116	17	50	340	1	374	-
25	105	3	40	315	1	335	-
26	110	6	4	236	1	93.6	-
27	159	3	18	541	1	1160	-
28	134	8	48	427	1	1800	0.39 %
29	91	0	11	298	1	29.6	-
30	79	0	10	246	2	32.2	-
31	151	0	30	462	1	1800	1.03 %
32	70	0	6	218	1	7.12	-
33	87	0	9	258	1	347	-
34	85	0	26	258	1	18.4	-
35	130	10	4	257	1	405	-
36	94	3	2	138	1	1.51	-
37	136	22	3	158	1	13.8	-
38	96	0	0	240	1	13.8	-
39	61	0	4	176	2	2.8	-
40	102	14	30	279	1	1800	0.02 %
41	118	3	16	367	2	112	-
42	104	0	0	115	1	2.23	-

Tabla 13: Detalles de los resultados obtenidos de haber aplicado el Enfoque $F_4 = \emptyset$ a las zonas de la zonificación inicial.

zona	$\|V_Z\|$	$\|\hat{A}\|$	$\|\hat{E}\|$	$F_3^{(Z)}$	sectores	tiempo (s.)	gap
44	154	0	8	456	3	1800	0.39 %
45	84	2	11	195	1	47	-
46	105	0	0	162	3	2.6	-
47	121	0	3	314	1	12.4	-
48	129	1	4	282	1	322	-
49	86	0	13	278	1	19.2	-
50	123	2	3	239	1	6.3	-
51	149	3	25	402	1	1800	2.99 %
52	138	1	6	457	1	1800	1.22 %
53	84	0	50	252	1	262	-
54	121	0	8	402	1	1050	-
55	97	0	5	343	1	154	-
56	108	1	10	283	1	1800	0.76 %
57	120	10	28	357	1	1800	0.52 %
58	64	20	36	172	1	5.09	-
59	96	5	45	316	1	1800	1.45 %
60	81	0	1	262	1	506	-
61	82	1	8	238	1	44.1	-
62	81	0	13	266	1	9.7	-
63	134	0	4	341	3	264	-
64	134	12	19	292	1	786	-
65	82	4	15	215	2	6.9	-
66	113	1	12	385	1	73.8	-
67	87	0	5	270	1	585	-
68	86	8	3	83	1	2.67	-
69	111	0	1	169	1	52	-
70	145	2	26	458	1	1800	0.99 %
71	77	1	7	198	3	41.7	-
72	152	4	33	392	1	1800	0.48 %
73	64	1	4	195	1	3.59	-
74	135	3	7	367	1	1800	0.34 %
75	100	0	0	99	1	2.62	-
76	138	4	11	361	2	395	-
77	92	2	2	177	1	20.7	-
78	106	0	11	343	1	827	-
79	102	0	4	348	2	28.4	-
80	121	1	6	290	3	47.9	-
81	42	8	0	120	2	7.91	-
82	120	8	18	347	2	73.7	-
83	81	2	2	207	2	44.4	-
84	104	1	12	394	3	34.9	-

Tabla 14: Detalles de los resultados obtenidos de haber aplicado el Enfoque $F_4 = \emptyset$ a las zonas de la zonificación inicial.

	$p = 0$	$p = 50$	
Resueltas a optimalidad	79	72	(−8,86 %)
Para las zonas resueltas a optimalidad por ambos			
Tiempo de solución promedio	41,92 s.	152,56 s.	(263,93 %)
Distancia mixta promedio	26,31 km.	26,66 km.	(0,76 %)
Cantidad de giros promedio	88,28	60,77	(−31,16 %)
Para las zonas no resueltas a optimalidad por ninguno			
Gap de optimalidad promedio	0,35 %	0,99 %	

Tabla 15: Comparación entre las rutas obtenidas para $p = 0$ y $p = 50$. Entre paréntesis se indica la diferencia relativa de $p = 50$ respecto a $p = 0$.

tres iteraciones. Los criterios de parada se establecieron en seis horas para el tiempo límite y en 100 para la cantidad máxima de iteraciones. El tiempo límite para la resolución del modelo MILP para hallar una ruta eficiente fue de 15 minutos y también configuramos al solver para enfatizar la aplicación de heurísticas.

Como los métodos basados en VNS y SA tienen elementos estocásticos, los corrimos con cinco semillas aleatorias para comparar su rendimiento. Las zonificaciones finales obtenidas por VNS presentaron un valor de $\mathcal{F}$ promedio de 0,92, mientras que para las de SA el valor promedio fue de 1,03. En pro de la claridad, para comparar los métodos utilizaremos los resultadoos obtenidos con una de las semillas. En la Figura 26 mostramos la evolución del valor de $\mathcal{F}$ para cada uno de los métodos iterativos. Cada punto representa el valor de $\mathcal{F}$ luego de la etapa de corrección. Otra posible medida para la distribución de trabajo es el desvío estándar de los valores de Φ de las zonas que componen la zonificación. La Figura 27 ilustra la evolución del desvío estándar de la carga laboral, mostrando que efectivamente minimizar $\mathcal{F}$ ayuda a balancear la distribución de trabajo. En este caso, cada punto se corresponde al desvío estándar de los valores de Φ luego de una etapa de corección.

En la Tabla 16 comparamos las características de las soluciones obtenidas con los tres métodos. Mostramos su valor en la función objetivo y la diferencia relativa respecto al valor de $\mathcal{F}$ en $\mathcal{Z}_0$, el desvío estándar de los valores de Φ y el tiempo requerido para obtener la solución. Como nos interesa que las soluciones puedan ser aplicadas, la tabla también muestra la cantidad de zonas modificadas respecto a $\mathcal{Z}_0$. Incluimos también el cociente entre la disminución en $\mathcal{F}$ y la cantidad de zonas modificadas, que representa el promedio de cuánto impacta en $\mathcal{F}$ la modificación de una zona. Finalmente, mostramos la diferencia relativa de la distancia diaria total con respecto a $\mathcal{Z}_0$.

El método basado en SAHC se detuvo luego de 15 interaciones al no hallar ninguna zonificación vecina que mejorara el valor de F durante la expansión ni durante la contracción. Tanto SA como VNS agotaron el tiempo límite, alcanzando a completar 54 y 13 iteraciones, respectivamente. Por su parte, SAHC logró el mejor promedio de disminución de $\mathcal{F}$ por zona modificada, el menor tiempo de cómputo y una mejora considerable respecto a la situación previa al proyecto. A cambio de transformar una mayor cantidad de zonas, SA y VNS logran reducir el valor de $\mathcal{F}$ por debajo de la mitad del de $\mathcal{Z}_0$. Sin embargo, en este caso SA presenta

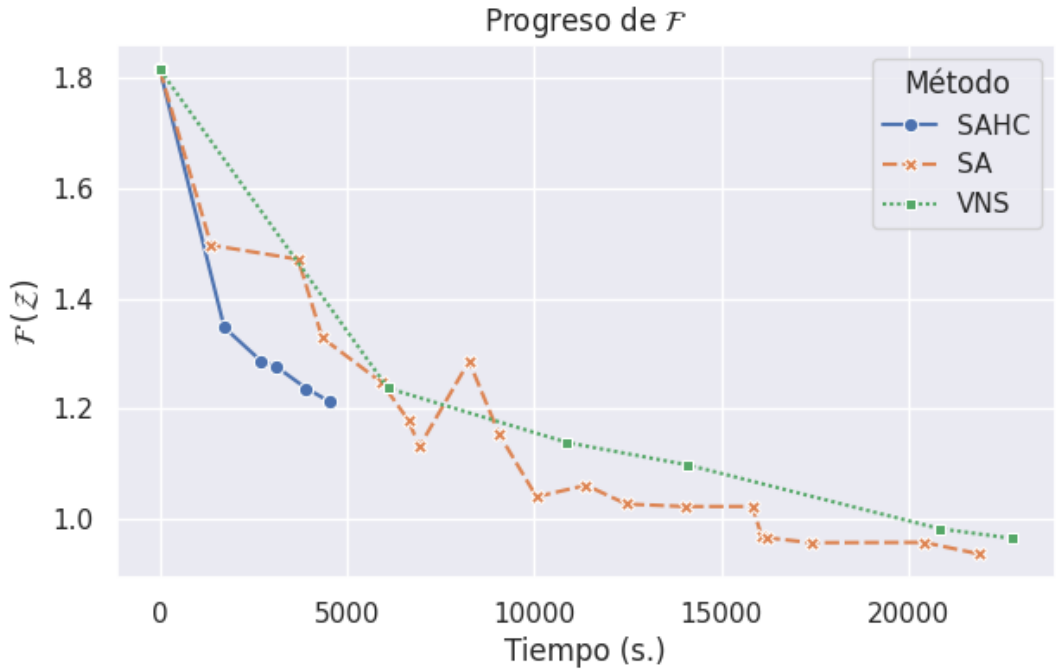


Figura 26: Evolución del valor de $\mathcal{F}$ para los tres métodos iterativos. Cada punto representa el valor de $\mathcal{F}$ luego de una etapa de corrección.



Figura 27: Evolución del desvío estándar de los valores de Φ para los tres métodos iterativos. Cada punto se corresponde al desvío estándar luego de una etapa de corección.

	SAHC	SA	VNS
Valor en $\mathcal{F}$	1.21	0.94	0.97
Valor en $\mathcal{F}$ respecto a ${\mathcal{Z}_0}^{(1)}$	-45.9 %	-58.2 %	-56.9 %
Desvío estándar de Φ	0.344	0.322	0.317
Tiempo	1:15:34	6:02:36	6:19:20
Zonas modificadas	25	49	43
Disminución promedio de $\mathcal{F}$ por modificación	0.024	0.018	0.02
Distancia total recorrida respecto a ${\mathcal{Z}_0}^{(2)}$	-10.9 %	-10.6 %	-10.7 %

(1) Omitiendo las cinco zonas de las cuales faltan datos para calcular Φ

(2) Omitiendo las dos zonas para las cuales no tenemos datos de GPS.

Tabla 16: Comparación de resultados obtenidos con cada uno de los métodos.

el peor promedio de disminución de $\mathcal{F}$ por modificación de zona. Esto probablemente esté relacionado con que SA admite movimientos desfavorables durante las primeras iteraciones, modificando más zonas de las que son necesarias para hallar una zonificación con mejor distribución de trabajo. Como, al menos para esta semilla, VNS logra sortear los mínimos locales cercanos a la solución inicial, la aceptación de movimientos desfavorables por parte de SA no brinda ningún beneficio en este sentido. Por esta razón, a partir de la Tabla 16, podemos concluir que el método más apropiado para este experimento depende a qué aspecto se le dé prioridad: la facilidad de implementación o la minimización de la carga laboral. Si se prioriza la primera, SAHC resulta ser el más apropiado, pues se modifica alrededor de un cuarto de las zonas al mismo tiempo que se distribuye más equitativamente la carga laboral. Si el foco está puesto en disminuir el valor de $\mathcal{F}$, VNS o SA permiten alcanzar muy buenos resultados a cambio de modificar alrededor de la mitad de las zonas.

Un aspecto positivo de los métodos iterativos es que a lo largo de su procedimiento hallaron zonificaciones que mejoran la distribución de la carga laboral. Por lo tanto, la municipalidad podría decidir aplicar alguna de las soluciones intermedias si considera que es más fácil de llevar a la práctica, y aun así mejorarían las condiciones de trabajo para los recolectores. A partir de las soluciones intermedias también podría obtenerse una sucesión de cambios paulatinos que permitieran transicionar desde la zonificación en vigencia a la solución propuesta por el método.

En lo que respecta a la distancia total recorrida por la flota de vehículos, la disminución relativa respecto a $\mathcal{Z}_0$ de las soluciones de los tres métodos se mantuvo alrededor de la de $\mathcal{Z}'_0$. Esto representa una disminución diaria de, aproximadamente, 167 litros de combustible que se traducen en un ahorro anual de hasta USD 47600[12]. Más aún, los costos de mantenimiento de los vehículos también se reducen y se emiten 448 kg. menos de CO_2. Por lo tanto, los tres métodos proponen soluciones que no solo mejoran la distribución de trabajo, sino que también rebajan costos y emisiones de gases.

En la Figura 28 comparamos la zonificación $\mathcal{Z}_0$ con las obtenidas con los tres métodos iterativos. En pro de la claridad, para cada método solo ilustramos las zonas que han sido

[12]Según el precio de combustible de junio 2025.

modificadas. El color de cada zona se corresponde con la variación de su carga laboral respecto a $\mathcal{Z}_0$: mientras más rojo, más disminuyo y, mientras más verde, más aumentó. Podemos observar que, en general, la forma geométrica de las zonas modificadas no ha cambiado demasiado respecto a $\mathcal{Z}_0$. Esto demuestra la ventaja de modificar la zona teniendo en cuenta sus lados en vez de considerar sus manzanas individualmente. También podría adaptarse la Fase II del enfoque de creación de zonas (Sección 5.2) para reducir la cantidad de vértices del polígono que define la frontera de las zonas.

Los resultados obtenidos de la aplicación están en proceso de ser presentados a las autoridades de GIRSU para la elección del resultado que consideren más conveniente, su posible corrección y eventual implementación. Como hicimos en el proyecto de Berazategui, las zonificaciones se muestran con un plano interactivo y entregamos cada ruta en distintos formatos: una animación que permite visualizarla, una hoja de ruta y un archivo `gpx` que puede utilizarse con la aplicación OsmAnd.

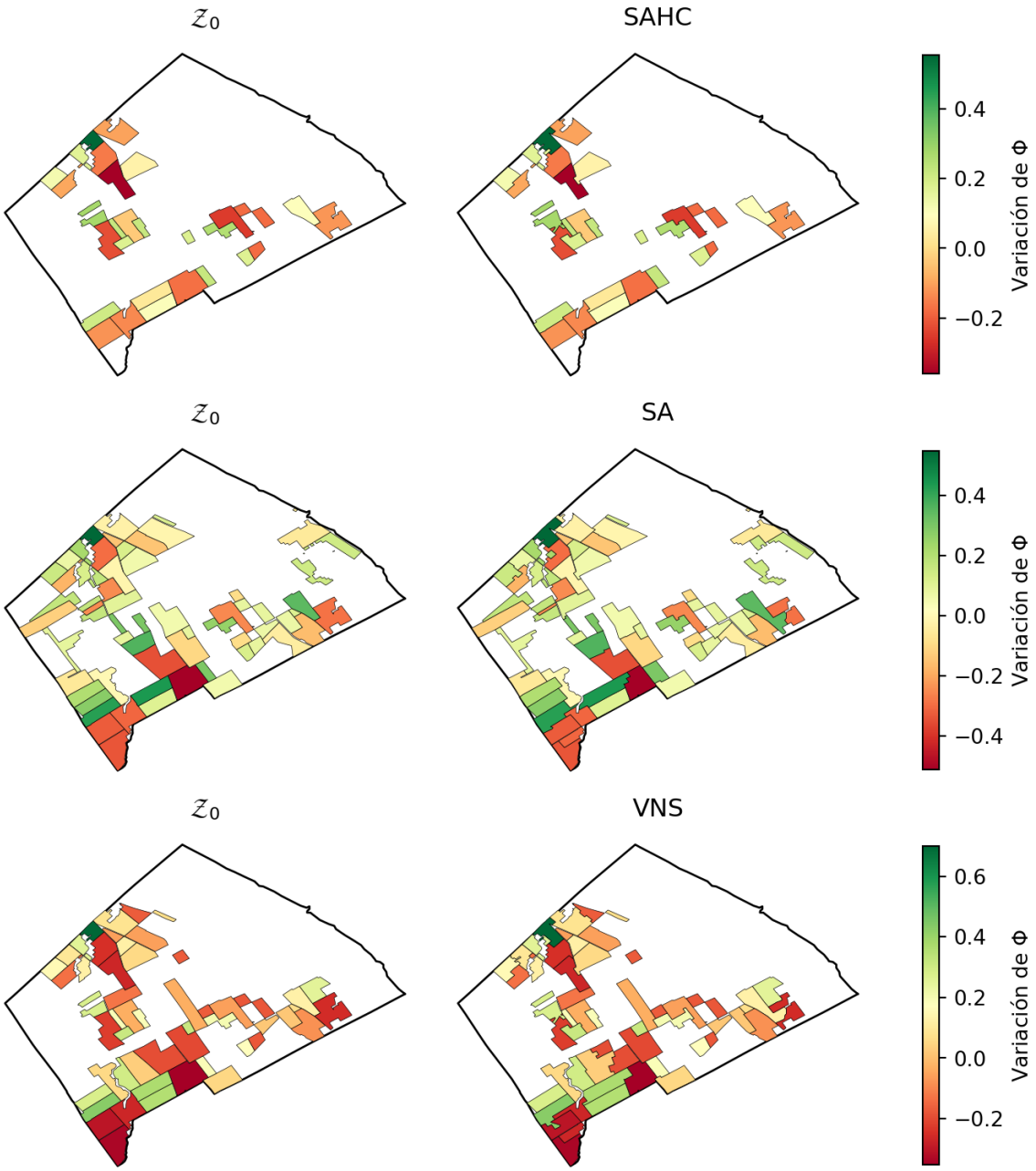


Figura 28: Se muestra cada zona en $\mathcal{Z}_0$ y luego de ser modificada por el correspondiente método. El color corresponde a la diferencia en su valor de Φ: si es rojo, disminuyó después de aplicar el método y, si es verde, aumentó.

11. Discusión, conclusiones y trabajo futuro

Hemos desarrollado un conjunto de herramientas basadas en heurísticas y modelos matemáticos para la optimización de recolección de residuos. Presentamos un enfoque de solución basado en MILP para hallar soluciones eficientes a instancias reales del Mixed General Routing Problem with Turn Penalties. Su desempeño ha sido mejorado mediante heurísticas para hallar soluciones iniciales de buena calidad y para establecer un orden de recorrido para las zonas con más de un sector. A diferencia de las metodologías revisadas en la literatura, el procedimiento que proponemos puede ser aplicado a casos donde los giros prohibidos y/o penalizados pueden ser secuencias de cuatro vértices. Para el caso en el que todos los giros son secuencias de tres vértices, propusimos una descomposición polinomial del digrafo de ruteo a partir de la cual se pueden hallar eficientemnte caminos mínimos sin caminos prohibidos, considerando la penalización por giros.

Con respecto a la zonificación, propusimos dos enfoques de solución: una heurística que genera una nueva zonificación y tres procedimientos que modifican la zonificación vigente. El primero mostró muy buenos resultados en la distribución de trabajo entre los recolectores a pie, disminuyendo en alrededor del 75 % el desvío estándar de la cantidad de kilómetros que deben recorrer diariamente por zona, además de garantizar que no se generan zonas disconexas. En cuanto al trabajo futuro sobre esta metodología, se podría ajustar la métrica de cuadratura para incrementar su sensibilidad en la etapa del procedimiento de zonificación iterativa, de modo que el algoritmo de corrección no tenga que hacer tantos ajustes posteriores. Otra posibilidad es adaptarlo para que, en vez de agregar una manzana por vez a la zona, se expanda por alguno de sus lados y de esta manera generar zonas con forma regular. Asimismo, si la carga laboral viene dada por el metraje, se podría identificar un criterio de parada más adecuado para el crecimiento de la zona que se adapte mejor a un plano con una variación significativa en las dimensiones de las manzanas. En nuestro caso, el uso del metraje promedio de las áreas sin zonificar significó que las primeras zonas generadas fueron algo más grandes que las últimas.

Los procedimientos para la modificación de la zonificación en uso también fueron exitosos en mejorar la distribución laboral: se logró reducir la disparidad en hasta un 58,2 %. Los métodos propuestos ofrecen una variedad de elección en el espectro de intercambio entre cantidad de modificaciones y la reducción de la disparidad de la carga laboral, además de permitir elegir cualquier solución intermedia. El método basado en Steepest-Ascent Hill Climbing otorgaba una solución con pocos cambios en poco tiempo y una moderada mejora en la distribución laboral; mientras que los métodos basados en Variable Neighbourhood Search y Simulated Annealing requerían más tiempo, brindando una solución con una marcada mejora en la distribución de trabajo pero con un mayor número de modificaciones con respecto a la situación inicial. Estos métodos ponen el foco en no modificar radicalmente la zonificación y así simplificar su implementación. En el futuro, se podría desarrollar una variante de estos métodos que asistieran a crear nuevas zonas considerando la zonificación vigente. Esto sería últil, por ejemplo, cuando la municipalidad adquiera más camiones. Por otra parte, también podría adaptarse la Fase II del algoritmo de creación de zonificación para corregir la frontera de las zonas resultantes de SAHC, VNS o SA, mediante la expansión o contracción por alguno de sus lados y limitando el impacto sobre la distribución del trabajo.

En ambos casos de estudio, la metodología desarrollada para hallar rutas eficientes logró disminuir significativamente la distancia recorrida por la flota de vehículos. En el municipio de Berazategui hemos logrado disminuir la distancia promedio de recorrido en un 22,6 % mientras que en Quilmes la reducción se encuentra en 10,93 %. Esto impacta en menor consumo de combustible, emisión de gases, desgaste de los vehículos y menor tiempo requerido para la recolección. Además, en el caso de Quilmes, agregar la penalización por giros permitió obtener rutas más sencillas de ejecutar sin aumentar demasiado la distancia total recorrida, simplificando así su implementación. El concepto de distancia ponderada también permitió reflejar mejor la diferencia que existe entre transitar distintos tipos de calle fuera de la zona de recolección durante los distintos turnos. Si se contara con datos de tránsito, se podría llevar adelante un análisis estadístico más profundo para incluirlos en la definición de los coeficientes para cada cuadra.

Por otra parte, nos interesa analizar los factores que influyen en el rendimiento de los modelos MILP para hallar rutas eficientes, más particularmente el del modelo (28)–(38) descripto en la Sección 7.2. Para esto, medimos su desempeño en las zonas de Quilmes y de Berazategui, con la siguientes consideraciones:

1. No hay penalización a giros permitidos y utilizamos la distancia real, no la mixta.
2. Para las zonas de Berazategui, añadimos las restricciones (24), para evitar caminos prohibidos de cuatro vértices en la concatenación de caminos mínimos.
3. Configuración del solver: el tiempo límite de resolución del modelo es de 90 minutos y se configura en 5 el parámetro `parameters.emphasis.mip` para que enfatice el uso de heurísticas.

Ejecutamos en el CeCAR el modelo para el total de las 122 zonas. En la Figura 29 mostramos el tiempo de ejecución en función de distintas características de las zonas: la cantidad de esquinas ($|V_Z|$), sectores, calles de un sentido ($|A_Z|$), calles de un sentido que deben ser recorridas obligatoriamente ($|\hat{A}|$), calles de doble sentido ($|E_Z|$), calles de doble sentido que deben ser recorridas obligatoriamente ($|\hat{E}|$), calles que deben ser recorridas obligatoriamente ($|\hat{A}| + |\hat{E}|$) y caminos prohibidos de tres vértices ($|F_3|$). En primer lugar observamos que los gráficos para $|E_Z|$ y $|F_3|$ son similares pues, en general, cada arista tiene asociados dos giros prohibidos que se corresponden a girar en U en cada una de sus esquinas. No se observa una relación evidente entre alguna de las características con el tiempo de resolución del modelo, aunque se podría conjeturar que mayores valores de $|E_Z|$ y de $|\hat{E}|$ incrementan el tiempo necesario para hallar una solución óptima.

Por lo tanto, otra línea de trabajo futuro es realizar un análisis de sensibilidad más meticuloso para revelar la causa de la diferencia en el desempeño del modelo. Relacionado a esto, también se podrían investigar técnicas para mejorar el rendimiento del modelo MILP, por ejemplo reformulándolo desde el enfoque de generación de columnas. Hemos intentado incluir cortes hallados en la literatura del ATSP, como los propuestos por Campuzano et al. [20], pero no hemos obtenido mejoras en el rendimiento del modelo. Suponemos que en parte se debe a que, al estar implementados en Python, los algoritmos de separabilidad no se ejecutan lo suficientemente rápido y, por ende, no mejoran (o incluso ralentizan) la resolución del modelo. Si el modelo tuviera mejor performance, podría adaptarse para ser utilizado por las autoridades municipales para redefinir las rutas en caso de que alguna de sus calles se vuelva

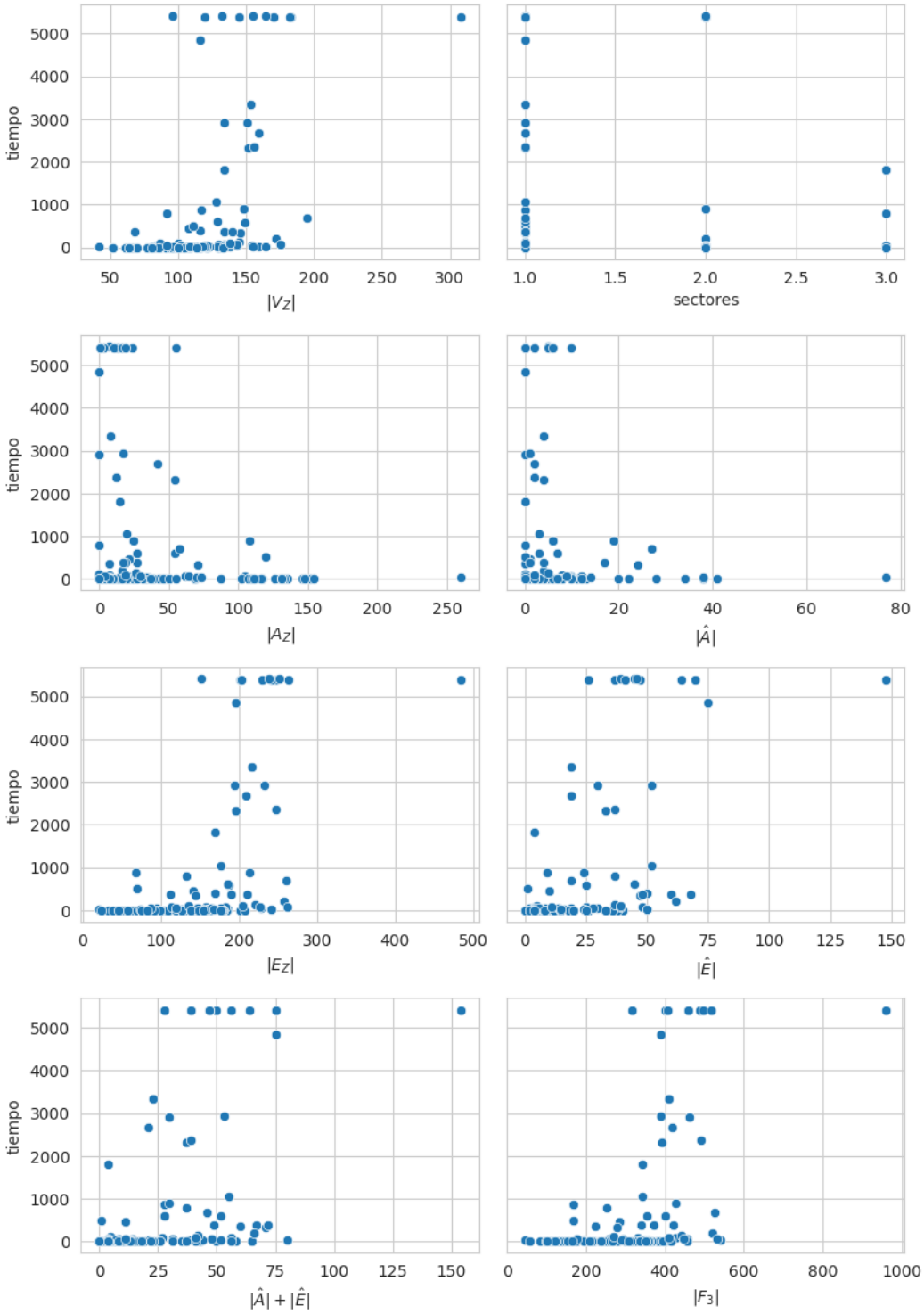


Figura 29: Tiempo de ejecución del modelo MILP de ruteo en función de distintas características de las zonas de recolección de Quilmes y Berazategui

instransitable (por ejemplo, debido a obras). Adicionalmente, otra parte del trabajo futuro consiste en comparar nuestro enfoque para resolver el MGRPTP con el propuesto por Soler et al. ([115]). Esta comparación posicionaría más claramente las ventajas, limitaciones y aportes del método que hemos desarrollado.

Es importante remarcar que, si bien tuvimos en cuenta características de los casos de estudio, las metodologías que desarrollamos son fácilmente generalizables a otras ciudades del país. Independientemente del achique, la estrategia para hallar rutas eficientes puede aplicarse a cualquier sistema de recolección en la que los camiones deban visitar esquinas y cuadras, o, en otras palabras, que el ruteo pueda modelarse dentro de MGRPTP. Los enfoques para generar una zonificación o modificar la que se encuentre vigente también son adaptables a cualquier sistema de recolección cuyas zonas puedan modelarse como un conjunto de manzanas y las heurísticas presentan diversos parámetros que permiten ajustarlas a distintas aplicaciones. Por estas razones, en esta tesis hemos elaborado herramientas para la optimización de recolección de residuos que pueden ser adaptadas y utilizadas por municipios para reducir el gasto del servicio y de su impacto en el ambiente y al mismo tiempo mejorar la calidad de trabajo de los recolectores mediante una distribución más justa de la carga laboral.

Capítulo II

Planificación de la producción

1. Introducción

Este capítulo está dedicado al trabajo que llevamos a cabo para resolver una instancia real del *Unrelated Parallel Machine Problem* (UPMP) utilizando como caso de estudio la planta de producción de TAPI South America ubicada en Córdoba, Argentina. El objetivo consistió en desarrollar una herramienta de programación matemática que permitiera optimizar su planificación de producción según criterios que describiremos más adelante. TAPI es un grupo internacional especializado en el diseño y elaboración de cierres sintéticos para bebidas alcohólicas. La empresa cuenta con fábricas en Italia, Argentina, Francia y México, así como también bases comerciales en Alemania y Estados Unidos. Globalmente provee sus productos a más de 5000 clientes ubicados en 70 países alrededor del mundo. La planta ubicada en Córdoba fue inaugurada en el año 2004. Desde entonces, TAPI South America ha provisto de cierres sintéticos para bebidas alcohólicas a clientes argentinos y exporta alrededor del 65 % de su producción a Sudamérica, Estados Unidos, India y Sudáfrica, entre otros, siendo Estados Unidos el principal destino de exportación.

El proceso de elaboración de los cierres consiste esencialmente en inyectar la materia prima dentro de un molde. Algunos productos están compuestos por dos partes: la cabeza y la parte del tapón que es introducida dentro de la botella, denominada *gambo*. Según el tipo del producto, ambas partes pueden ser producidas simultánea o secuencialmente. En el último caso, se dice que el gambo es coinyectado a la cabeza. Los cierres pueden ser clasificados en cuatro categorías según el proceso de producción:

- tapones de vino: de forma cilíndrica con diámetros y longitudes variables. Requiere la utilización de un sólo tipo de molde.
- tapón *monoblocco*: inyectado en un solo proceso, con moldes que pueden variar en geometría de la cabeza y diámetro del gambo. Requiere la utilización simultánea de dos moldes distintos.
- gambos (producto intermedio) que son de diversos diámetros y longitudes y que se usan para pegar en cabezas de madera (T-wood) o cabezas de plástico rígido (para productos bicolor).
- tapón T-Wood: se fabrica coinyectando el gambo en la cabeza de madera que tiene un diseño especial que bloquea el "gambo" de manera que no pueda separarse a pesar de que no tiene pegamento.
- tapón bicolor: primero se inyecta una cabeza de plástico sintético que posteriormente, a través de un robot, es incorporada en el molde donde se inyecta el gambo, que queda fundido a la cabeza plástica. Requiere que primero se utilice un molde para elaborar la cabeza de plástico y luego otro molde para la coinyección del gambo. Ambas partes deben ser fabricadas secuencialmente y la coinyección no necesariamente ocurre en la misma máquina en la que fue elaborada la cabeza.

En la Figura 30 se muestran ejemplos de productos de cada una de las categorías. De esta manera, cada trabajo está caracterizado por su molde (o moldes), el procedimiento de elaboración (tapón de vino, monoblocco, T-Wood o bicolor) y su color (o colores, en el caso de los productos bicolor). Al momento de realizar este proyecto, Tapi South America fabricaba

el 80 % de sus tapas de madera en una segunda planta industrial dedicada a tal fin. El resto lo compraba a productores independientes en Córdoba. En este trabajo sólo incluimos el procedimiento de coinyección del gambo a la cabeza para los productos T-Wood, no la producción de las cabezas de madera.



Figura 30: Categorías de productos según el proceso de producción. Arriba a la izquierda, tapón de vino. Arriba a la derecha, tapón monoblocco, donde se indica el gambo y la cabeza. Abajo a la izquierda, tapón T-Wood. Abajo a la derecha, tapón bicolor.

Desde TAPI South America se pusieron en contacto con nuestro grupo del Instituto de Cálculo para desarrollar una herramienta computacional que les asistiera en la toma de decisiones al momento de planificar la producción de la fábrica durante el periodo de, a lo sumo, un mes. La herramienta desarrollada también les permitiría evaluar el impacto que tendría en la producción la adquisición de insumos (nueva maquinaria, mayor cantidad de moldes, etc.). También, la herramienta sería de utilidad para efectuar cambios a corto plazo ante un cambio en las necesidades comerciales: un nuevo trabajo que se requiere con urgencia o alguna cancelación de un trabajo que ya estaba programado, por ejemplo. Previamente a este proyecto, el personal encargado de programar la producción utilizaba Excel para elaborar un diagrama de Gantt en base a las órdenes con fecha de entrega dentro del horizonte de planificación, teniendo en cuenta las numerosas restricciones de la manufacturación. Entre otras complicaciones, esta situación causaba que hubieran órdenes entregadas luego de la fecha pactada y que surgieran problemas y desorganización en los inventarios de materias primas y de stock de productos elaborados. La empresa deseaba contar con un software que pudiera proveer una planificación de producción que requiriera la menor cantidad de ajustes manuales posteriores y que tuviera en cuenta las restricciones específicas de la manufacturación en la fábrica. Debido a la duración del periodo de planificación, resultaba aceptable que el software pudiera correr durante varias horas para elaborar una planificación horaria en favor de valores satisfactorios en los objetivos.

Con las características de la producción de cierres y con los datos que nos proveyeron, estudiamos este problema como una aplicación al mundo real del *Unrelated Parallel Machine Problem* (UPMP). El UPMP consiste en distribuir y programar la realización de n trabajos en un conjunto de m máquinas que se asumen distintas unas de otras, es decir, que el tiempo

de procesamiento del trabajo j depende de a qué máquina es asignado ([50]).

El resto del capítulo se organiza de la siguiente manera. En la próxima sección describimos el problema a resolver, detallando las características que permiten modelarlo como un UPMP. En la Sección 3 llevamos a cabo una revisión de la literatura y precisamos cuál es el aporte de este trabajo. En la Sección 4 describimos nuestro enfoque de solución, el cual consiste en dividir al problema en tres etapas. En la Sección 5, mostramos los experimentos computacionales que llevamos a cabo en instancias que generamos para probar la escalabilidad de nuestra estrategia de resolución. Finalmente, en la Sección 6 compartimos las conclusiones y algunos lineamientos para trabajo futuro.

2. Descripción del Problema

En esta sección describimos las características del proceso de manufactura de los cierres. Como hemos mencionado anteriormente, cada cierre es elaborado mediante la inyección de materia prima en un molde. Cada máquina tiene cierta cantidad de grupos inyectores. En cada grupo inyector puede colocarse a lo sumo un juego de un tipo de molde; por esta razón, de aquí en adelante diremos que un grupo inyector tiene asignado un molde si está equipado con uno de sus juegos. Una máquina puede operar con algunos grupos inyectores vacíos. La velocidad de procesamiento de un trabajo en una máquina depende de la velocidad de sus grupos inyectores, su procedimiento de elaboración y del molde. Como la velocidad de producción de los grupos inyectores no es igual para todas las máquinas, se trata de un UPMP. Debido a una cuestión de configuración, ciertos moldes no pueden compartir la máquina con otros, a menos que se trate de una máquina con grupos inyectores no solidarios. Sin embargo, la mayoría de los moldes sí pueden compartir máquina con otros, por lo cual una máquina puede estar procesando más de un trabajo al mismo tiempo. Además, cada molde tiene un conjunto de máquinas con el que es compatible. De esta manera, el problema presenta lo que se denominan *machine elegibility constraints*. También puede suceder que ciertas máquinas se utilicen exclusivamente para la etapa de coinyección de los productos bicolor y T-Wood.

El cambio de moldes en una máquina requiere que todos sus grupos inyectores se detengan. El modelo de la máquina y la cantidad de grupos inyectores a los que hay que cambiarles el molde determinan la longitud de la parada. Este tipo de restricciones es conocida como *setup time constraints*. El cambio de moldes sólo puede ocurrir durante el horario administrativo y, por disponibilidad del personal, sólo se pueden cambiar moldes en una máquina a la vez. Asimismo, por cuestiones de organización, es preferible evitar demasiados cambios de molde en un mismo día.

Cada trabajo tiene una fecha de entrega y no necesariamente debe ser completado en una misma máquina, es decir, su proceso puede interrumpirse y retomarse más tarde en la misma máquina o en otra. Ambas características se corresponden a la presencia de *deadline constraints* y de *preemption*, respectivamente. Por otro lado, la fábrica cuenta con una cantidad limitada de juegos de moldes de cada tipo. Además al producir la cabeza de los productos bicolor es necesario utilizar un juego de *gambeto*, mientras que para la coinyección en productos bicolor y T-Wood es necesario utilizar un juego de portacabezas. El número de juegos de gambeto y portacabezas también es limitado. Por lo tanto, el problema presenta *resource constraints*. Como los productos bicolor requieren la producción secuencial de la cabeza y la coinyección del gambo, se deben considerar *precedence constraints*.

Con motivo de mantener un stock de seguridad de los productos, para cada trabajo la empresa asigna una sobreproducción admisible, es decir, un exceso sobre la demanda. Ese límite no es una restricción dura del problema, pero es deseable que se respete para no tener demasiados productos en stock y aprovechar mejor la materia prima. De aquí en adelante llamaremos sobreproducción no deseable a la cantidad de cierres que se produzcan por encima del límite impuesto por la sobreproducción admisible.

Con respecto al criterio de optimalidad, se trata de un problema multiobjetivo. La empresa solicitó que la máxima prioridad de la planificación horaria sea la minimización de la demora en el cumplimiento de los trabajos. Sea C_j el tiempo de terminación del j-ésimo trabajo y d_j el

tiempo límite de entrega, se define la demora del j-ésimo trabajo como $T_j = \text{máx}(C_j - d_j, 0)$. En general todos los trabajos tendrían la misma prioridad por lo que este objetivo se traduce como minimización de la demora total $Tt = \sum T_j$, pero podría haber situaciones en las que algunos trabajos tuvieran más prioridad que otros y en ese caso se formularía como minimización de la demora ponderada total $Twt = \sum w_j T_j$ donde w_j es la prioridad del trabajo j. En segundo lugar, la planificación horaria debería minimizar la cantidad de sobreproducción no deseable. Finalmente, una característica que debería buscar la elaboración de la planificación es, en la medida de lo posible, reducir el tiempo de inactividad de las máquinas, considerando como tal al tiempo muerto entre periodos de producción en los que la máquina se encuentra detenida pero no a causa de un cambio de moldes. De esta forma, se busca garantizar la entrega puntual de los pedidos, reducir el desperdicio de recursos y aumentar la eficiencia de las operaciones.

Según el esquema de clasificación $\alpha|\beta|\gamma$ propuesto por Rinnooy Kan [109] y ampliado por Graham et al.[58], donde α se corresponde al diseño de procesamiento, β a las restricciones y γ a los objetivos, podemos categorizar a nuestro problema como:

$$R|pmtn, res, prec, sep\ dep, M_j, \bar{d}, M_k\ non-bott|Twt, overproduction, idle_time$$

con algunas restricciones adicionales. En la Tabla 17 describimos cada componente de esta notación.

Notación	Descripción
Diseño de procesamiento	
R	instancia del UPMP.
Restricciones	
pmtn	*preemption.*
res	*resource constraints.*
prec	*precedence constraints.*
sep dep	*setup time constraints.*
M_j	*machine elegibility constraints.*
$\bar{d}$	*deadline constraints.*
$M_k\ non-bott$	*una máquina puede estar procesando más de un trabajo simultáneamente.*
Objetivos	
Twt	demora ponderada total.
overprod	sobreproducción no admisible.
idle_time	tiempo de inactividad de las máquinas.

Tabla 17: Notación de la definición del tipo de problema, sus restricciones y sus objetivos.

3. Estado del Arte

En su definición más amplia, el *scheduling* es un campo de la Investigación Operativa que estudia la asignación eficiente de recursos a la realización de tareas para optimizar uno o más criterios. Como define Pinedo [103], es un proceso de toma de decisiones que se aplica a una gran variedad de ámbitos en la teoría y en la práctica, entre los que se encuentra la planificación de la producción de una fábrica. En este contexto, una buena planificación permite utilizar de manera más eficiente la maquinaria, reducir los costos de manufactura y disminuir la demora en la entrega de productos a los clientes ([91, 69, 103]). En la literatura, los problemas de planificación de producción son clasificados de acuerdo a la disposición de las máquinas en el proceso de producción y las características de la manufactura de los productos. Según las categorías generales descriptas por Ruiz [112], en un problema de UPMP los trabajos deben ser procesados por una máquina de entre las m disponibles y la velocidad con la que se completa un trabajo depende de a qué máquina se le asigne. La aplicación del *parallel machine scheduling problem* no se limita al ámbito de la manufactura, sino que también tiene aplicaciones a *multiprocessor task scheduling* ([128]) y *scheduling* de equipamiento ([54]), por nombrar algunas.

Si bien todavía no se conoce la complejidad de todos los problemas de UPM con objetivo de minimizar demora total, la literatura indica que varios de ellos son $\mathcal{NP}-$Hard [16]. Por ejemplo, Du y Leung [38] demuestran que el problema $1||Tt$ (una máquina, sin restricciones y minimizando demora total) es $\mathcal{NP}-$Hard. A partir de este resultado, pudo demostrarse que $1|pmtn|Tt$ también es $\mathcal{NP}-$Hard [16]. Además, los problemas relacionados a situaciones reales presentan más restricciones que contribuyen a complicarlos aún más [112]. Por esta razón, como se puede apreciar en los surveys [25, 96, 121] existe una vasta literatura que trata los casos particulares de este problema desde distintos enfoques: algoritmos exactos, métodos de aproximación y heurísticas.

Lenstra et al. [87] desarrollaron un algoritmo polinomial de 2-aproximación basado en programación lineal (LP) que construye una planificación horaria para el problema con objetivo de minimizar el tiempo de finalización de todos los trabajos (*makespan*) ($R||C_{\text{máx}}$). Además, propusieron un esquema de aproximación polinomial para el caso en el que la cantidad de máquinas está fijo y probaron que ningún algoritmo polinomial de aproximación puede garantizar una solución que esté a un factor menor que 3/2 del óptimo. Mokotoff y Chrétienne [97] presentan un algoritmo exacto y un algoritmo de aproximación que emplean un esquema de planos de corte a partir de la identificación de desigualdades válidas.

Guinet [60] encaró el problema con *setup times constraints* y objetivo de minimizar la demora promedio utilizando teoría de grafos y planteando modelos de programación lineal entera mixta (MILP). Avalos-Rosales et al. [7] desarrollaron un modelo MILP más eficiente que permitía resolver instancias de hasta 60 trabajos en 8 máquinas. Más tarde, Fanjul-Peyro et al. [48] lograron formular un modelo MILP que resolviera instancias de hasta 400 trabajos. Por otro lado, los mismos autores [47] presentaron dos modelos MILP que consideran el UPMP con restricciones dinámicas de recursos para resolver instancias de hasta 16 trabajos y 6 máquinas y una metaheurística que permite resolver eficientemente instancias de hasta 30 trabajos y 6 máquinas. Fanjul-Peyro [46] desarrolló un modelo MILP que permite resolver instancias pequeñas del problema con restricciones de *setup times* y *resource constraints* con

objetivo de minimizar el *makespan* ($R|sep\ dep, res|C_{max}$). También desarrolla un algoritmo exacto de tres fases que permite resolver eficientemente instancias de hasta 400 trabajos.

Las *dispatching rules* (DR) son un tipo especial de heurísticas aplicadas a problemas de scheduling. Zhang et al. [130] aplican un método de aprendizaje por refuerzo donde el problema es modelado como proceso semi-Markov, cinco DRs son utilizadas como acciones durante la fase de aprendizaje y el objetivo es minimizar la demora ponderada media. Tseng et al. [120] utilizan cinco DRs para atacar el problema con *setup times* y tres objetivos, entre los que se encuentran minimizar la demora ponderada total. Đurasević et al. [122] generan DRs automáticamente a través de *genetic programming* para el UPMP con *release time constraints* (el trabajo j puede comenzar su procesamiento a partir del momento r_j) y cuatro objetivos que incluyen la minimización de la demora ponderada total. Jaklinović et al. [76] adaptan un algoritmo genético (GA) para generar DRs de manera automática para el problema con objetivo de minimización de demora ponderada total y diversas combinaciones de *release time, setup time, machine elegibility* y *precedence constraints*.

Una variedad de metaheurísticas también han servido como enfoque de solución a problemas $R|\beta|Tt$ y $R|\beta|Twt$, donde β son distintos tipos de restricciones que incluyen *machine elegibility, setup times, release times* y *precedence constraints*. *Simulated annealing* es utilizado, entre muchos otros, por Bank y Werner [8] para $R|r_j, D|Twt$ (donde D denota que todos los trabajos comparten la fecha límite de entrega), por Pinheiro et al. [104] para $R|res|Tt$ y por Çakar et al. [19] para $R|r_j, sep\ dep|Tt$, quienes también utilizaron GA. Esta última también fue empleada, entre otros, por Liu [88] para $R|sep\ dep|Tt$. *Ant colony optimisation* es otra de las metaheurísticas que permitieron encarar, para nombrar algunos, problemas como $R||Twt$ ([131]) y $R|s_{ij}, M_j, r_j, sep\ dep|Tt$ ([1]).

Sin embargo, en la literatura existe una brecha de investigación en lo que se refiere a aplicaciones a problemas del mundo real ([112, 121]). En primer lugar, en ese tipo de problemas en general se busca optimizar más de un criterio. Si bien en los últimos años se ha prestado más atención a problemas multiobjetivo desde el plano teórico ([126, 86]) no hay muchos registros de aplicaciones a problemas reales. Además, en la aplicación a problemas reales el conjunto de restricciones es más específico, numeroso y complejo. Por ejemplo, *preemption* o el hecho de que cada máquina pueda procesar más de un trabajo simultáneamente no es algo que aparezca frecuentemente en la literatura. Además de reportar la aplicación a un caso real de manufactura de productos, nuestro trabajo aporta un enfoque de solución basado en un esquema de programación matemática en etapas donde cada una de ellas es resuelta a través de MILP, para una instancia real que presenta una variedad y un número de restricciones mayores a los trabajos que hemos revisado en la bibliografía. Este trabajo también aporta instancias de prueba que pueden servir como experimentos para métodos que resuelvan UPMP, incluso con otros objetivos y menor variedad de restricciones que los que tratamos en este artículo, lo cual también implica una contribución a este campo de investigación, donde las instancias presentadas frecuentemente no son generalizables a otros tipos de problemas [121].

4. Enfoque de solución

Como hemos mencionado en la sección anterior, se ha probado que algunos problemas de UPMP son $\mathcal{NP}-$Hard, presenta alta complejidad computacional, a lo que se le suma la gran cantidad de restricciones propias de nuestro caso de estudio. Por estas razones, nuestro enfoque de solución descompone el problema en subproblemas más simples, una técnica común para permite obtener soluciones eficientes en un tiempo razonable [50]. En nuestro caso el problema es resuelto en tres etapas y cada una emplea un modelo MILP. La primera etapa fue pensada teniendo en cuenta dos características. En primer lugar, la máquina debe detenerse por un cambio de molde, no necesariamente al completar un trabajo. Es decir, podría completarse un trabajo y continuar con otro que utilizara el mismo molde sin necesidad de detener la máquina, pues esto sólo implicaría cambiar el color de la materia prima. En segundo lugar, cada máquina puede procesar más de un trabajo simultáneamente. Por estas razones, en la primera etapa decidimos no considerar los trabajos, sino sus moldes. De esta manera, la primera etapa del enfoque de solución discretiza el tiempo en intervalos y, en cada uno de ellos, decide cuántos moldes de cada tipo utiliza cada máquina. Al discretizar el tiempo, se sobreestiman los tiempos de parada debido al cambio de moldes, por lo tanto en la segunda etapa son corregidos. Finalmente, en la tercera etapa, para cada molde, se decide el color con el que será elaborado, diferenciando así el producto de cada trabajo.

En la Figura 31 mostramos un ejemplo de cómo funciona nuestro enfoque de solución mediante tres diagramas de Gantt, uno para cada etapa. En el ejemplo, el horizonte de planificación es de una semana. Cada color le corresponde a un tipo de molde diferente y, en la tercera figura, los patrones indican distintos colores para el mismo molde. En cada diagrama, *INY* se refiere a la máquina inyectora y la letra se refiere al grupo inyector. Así, por ejemplo, *INY 01 - A* es el grupo inyector A de la máquina 1.

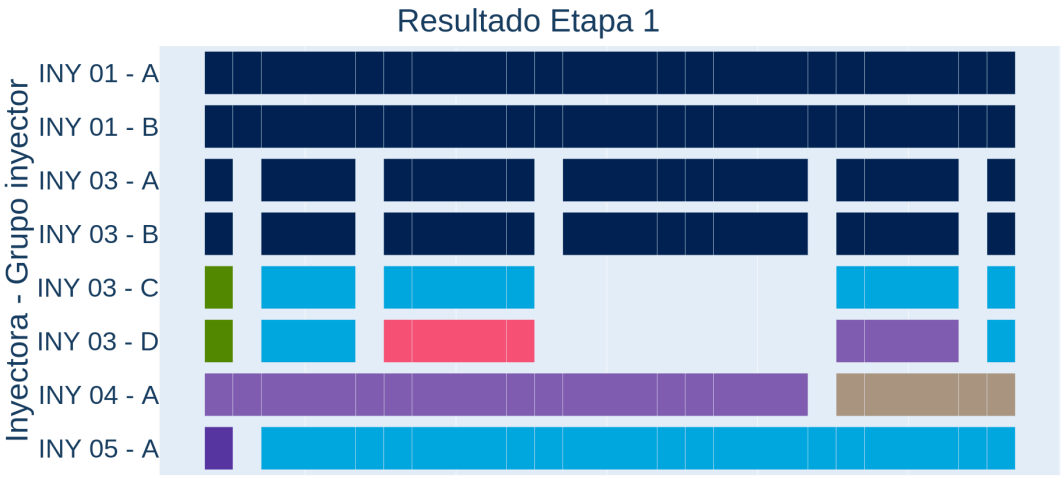
4.1. Etapa 1 - Asignación de moldes

En esta etapa introducimos los tres objetivos de la planificación horaria en su orden de prioridad descendiente: minimizar demora ponderada total, la sobreproducción no deseable y el tiempo de inactividad de las máquinas. Como el orden de prioridad entre estos objetivos es muy marcado y no se acepta que el valor de una función objetivo empeore para lograr una mejora en alguna función objetivo de menor prioridad, lo modelamos como un problema de optimización lexicográfica [4]. Sea S el conjunto de soluciones factibles y sean $f_1(x), f_2(x), \ldots, f_n(x)$ las funciones objetivo ordenadas en orden decreciente de prioridad, el método lexicográfico consiste en resolver n problemas de optimización donde para cada $i = 1, \ldots, n$:

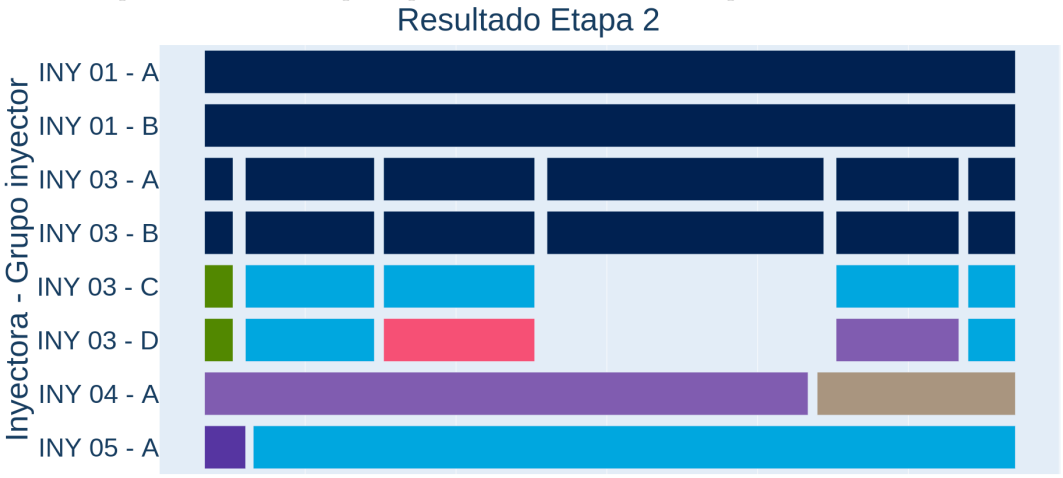
$$\begin{aligned} \text{mín} \quad & f_i(x) \\ \text{sujeto a:} \quad & x \in S \\ & f_j(x) \leq f_j(x_j^*) \quad j = 1, \ldots, i-1 \end{aligned}$$

donde $f_j(x_j^*)$ es el valor que toma la función objetivo en la solución hallada para el j-ésimo problema de optimización. La ventaja de este método es que garantiza que la solución hallada es Pareto óptima.

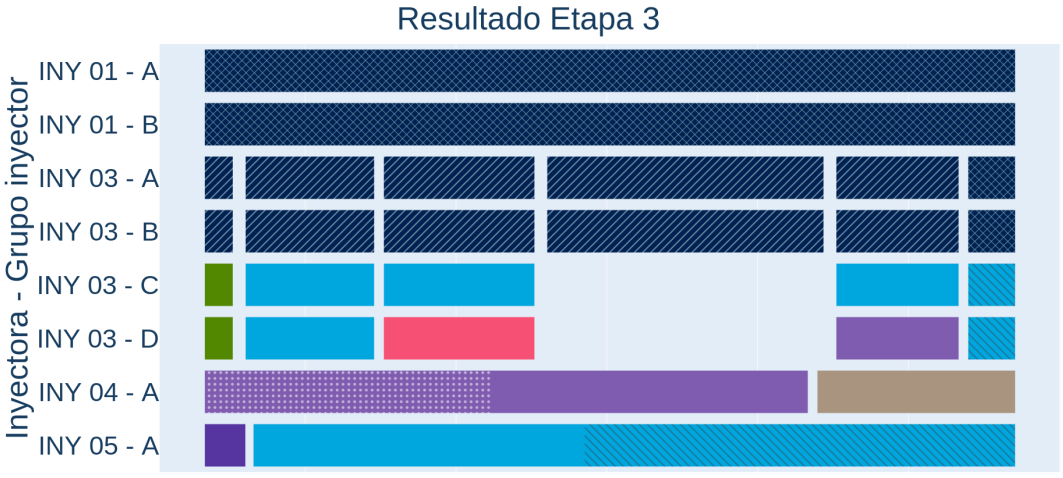
El primer paso de esta etapa consiste en discretizar el horizonte de planificación. Consideramos que la jornada comienza con el horario administrativo y termina con el comienzo del



(a) La Etapa 1 discretiza el horizonte de planificación y determina cuántos juegos de moldes de cada tipo utiliza cada máquina para cada intervalo de tiempo.



(b) La Etapa 2 determina cuándo se realizan los cambios de molde durante los intervalos indicados por la Etapa 1.



(c) Para cada tipo de molde, la Etapa 3 determina cuándo se realizan los cambios de color. Cada patrón representa un color para el correspondiente molde.

Figura 31

horario administrativo del día siguiente. La franja del horario administrativo es segmentada en intervalos de igual duración, que debe ser mayor que el máximo tiempo requerido para el cambio de moldes de las máquinas. Por ejemplo, si el horario administrativo tiene una duración de nueve horas y el máximo tiempo posible de cambio de una máquina es de dos horas y media, entonces puede dividirse en tres intervalos de tres horas o dos intervalos de cuatro horas y media. El horario no administrativo de una jornada no es segmentado pues no pueden haber cambios de moldes durante ese periodo. En la Figura 32 se muestra un ejemplo de discretización de dos jornadas.

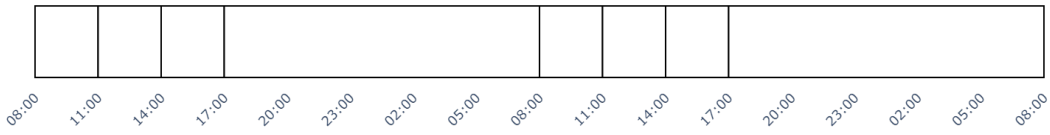


Figura 32: Ejemplo de la discretización de tiempo de dos jornadas. El horario administrativo (de 8 a 17hs.) es particionado en tres intervalos de tres horas.

A continuación, con el objetivo de contabilizar correctamente la cantidad de moldes de cada tipo que están en uso durante cada intervalo temporal, introducimos el término de *molde hermano* para referirnos a los moldes involucrados en los productos monoblocco y los moldes de los gambos que son coinyectados a las cabezas de los productos bicolor y T-Wood. Cada producto monoblocco requiere la utilización de dos moldes durante el mismo intervalo temporal. Por eso, para cada producto de este tipo creamos un molde ficticio que será hermano de los dos moldes que deben ser utilizados simultáneamente. Por ejemplo, si se deben utilizar los moldes M05 y M40 de manera simultánea, consideramos que en su lugar utilizará un molde ficticio M05+40, que es hermano de M05 y de M40, pero M05 y M40 no son hermanos entre sí. Por otra parte, como la velocidad de coinyección de un gambo a la cabeza de un producto T-Wood o bicolor es menor que su velocidad de inyección, consideramos un molde hermano por cada cabeza de producto T-Wood o bicolor a los que el gambo deba ser coinyectado. Por ejemplo, si se deben fabricar dos productos T-Wood, uno con cabeza de madera de tipo H01 y el otro con cabeza de madera de tipo H02, que requieren un gambo cuyo molde es M06, consideramos los moldes ficticios M06xH01 y M06xH02, que son moldes hermanos entre sí, hermanos de M06 y de los demás hermanos de M06. Ocurre una situación análoga para los gambos de los productos bicolor. Para cada molde no ficticio k, notaremos como S_k al conjunto de moldes hermanos de k y consideraremos que $k \in S_k$. En los ejemplos anteriores, $S_{M05} = \{M05, M05{+}40\}$, $S_{M40} = \{M40, M05{+}40\}$ y $S_{M06} = \{M06, M06\text{x}H01, M06\text{x}H02\}$.

Para cada molde k definimos un vector NP_k con la demanda acumulada de los trabajos que utilizan exclusivamente al molde k, en orden cronológico según su fecha de entrega, restando, si hubiese, el stock de los correspondientes productos. Por otro lado, para cada k también definimos la sobreproducción admisible como SPA_k, que resulta de la suma de la sobreproducción admisible para cada trabajo que utiliza exclusivamente a k. Por ejemplo, si M04 es utilizado para los trabajos, en orden cronológico, j_1, j_2, j_3 con demanda de $100, 200$ y 300 cierres respectivamente y del producto de j_2 se tienen 40 unidades en stock, entonces $NP_{M04} = (100, 260, 560)$. Si la respectiva sobreproducción admisible para j_1, j_2, j_3 es $20, 30$ y 10 cierres, entonces $SPA_{M04} = 60$.

Si bien se debe tener en cuenta que hay un límite para la cantidad de cambios de moldes que se puede llevar a cabo simultáneamente en distintas máquinas, puede ocurrir que la suma del máximo tiempo de parada posible de dos máquinas sea menor que la duración de un intervalo del horario administrativo. En ese caso, se permite que esas máquinas puedan cambiar de molde simultáneamente en ese intervalo y en la etapa 2 se decidirá el orden de parada. Sin embargo, para limitar la cantidad de cambios de molde diarios, se establece un límite a la cantidad de cambios por intervalo. Por otra parte, diremos que el intervalo t es un intervalo de inactividad para la máquina i si durante el intervalo t la máquina se encuentra detenida, no está realizando un cambio de moldes y en algún intervalo posterior vuelve a producir. En la Tabla 18 presentamos la notación y descripción cada uno de los conjuntos y parámetros del modelo MILP para resolver la primera etapa y en la Tabla 19 detallamos las variables.

A continuación presentamos el modelo de minimización lexicográfica:

$$\text{lex mín} \sum_{k\in K}\sum_{j\in N_k} p_{kj}\mu_{kj} \tag{60}$$

$$\sum_{\substack{k\in K\\ SPA_k\neq 0}} \frac{z_k}{SPA_k} + \sum_{\substack{k\in K\\ SPA_k=0}} z_k \tag{61}$$

$$\sum_{i\in I}\sum_{t\in T} r_{it} \tag{62}$$

sujeto a:

$$\sum_{k\in K} x_{ikt} \leq \sigma_i(1-y_{it}) \qquad i\in I,\ t\in T \tag{63}$$

$$\sum_{t\in T}\sum_{k\in K-K^{(i)}} x_{ikt} = 0 \qquad i\in I \tag{64}$$

$$\sum_{c\in S_k}\sum_{i\in I} \eta_{ict} + x_{ict} \leq CM_k \qquad k\in K,\ t\in T_{\geq}1 \tag{65}$$

$$\sum_{k\in K_{HD}}\sum_{i\in I} \eta_{ikt} + x_{ikt} \leq GMB \qquad t\in T_{\geq}1 \tag{66}$$

$$\sum_{k\in K_{CI}}\sum_{i\in I} \eta_{ikt} + x_{ikt} \leq HDS \qquad t\in T_{\geq}1 \tag{67}$$

$$x_{ikt} \leq \theta_{ikt}\sigma_i \qquad i\in I,\ k\in K,\ t\in T \tag{68}$$

$$\theta_{ikt} \leq x_{ikt} \qquad i\in I,\ k\in K,\ t\in T \tag{69}$$

$$\sigma_i\rho_{it} \leq \sigma_i - \sum_{k\in K}\theta_{ikt} \qquad i\in I,\ t\in T \tag{70}$$

$$1-\rho_{it} \leq \sum_{k\in K}\theta_{ikt} \qquad i\in I,\ t\in T \tag{71}$$

$$\eta_{ikt} \leq x_{ik(t-1)} \qquad i\in I,\ k\in K,\ t\in T_{\geq 1} \tag{72}$$

$$\eta_{ikt} \geq -\sigma_i(1-\rho_{it}) + x_{ik(t-1)} \qquad i\in I,\ k\in K,\ t\in T_{\geq 1} \tag{73}$$

$$\eta_{ikt} \leq \sigma_i\rho_{it} \qquad i\in I,\ k\in K,\ t\in T_{\geq 1} \tag{74}$$

Notación	Descripción
Conjuntos	
I	Conjunto de máquinas
I_{NJ}	Conjunto de máquinas con grupos inyectores no solidarios ($I_{NJ} \subset I$)
T	Conjunto de índices de intervalos temporales
$T_{[a,b]}$	$\{t \in T : a \leq t \leq b\}$
T_{NA}	Conjunto de índices de intervalos temporales fuera del horarios administrativo ($T_{NA} \subset T$)
K	Conjunto de tipos de moldes
K_{HD}	Conjunto de moldes que son cabeza de un producto bicolor ($K_{HD} \subset K$)
B_k	Conjunto de moldes coinyectados a una cabeza elaborada con el molde k para productos bicolor ($k \in K_{HD}$)
S_k	Conjunto de moldes hermanos de k ($S_k \subset K$)
$K^{(i)}$	Conjunto de moldes compatibles con la máquina i ($K^{(i)} \subset K$)
K_{NC}	Conjunto de moldes que no pueden compartir la máquina con otros moldes ($K_{NC} \subset K$)
Parámetros	
CM_k	Cantidad de juegos del molde k ($CM_k \in \mathbb{N}$)
σ_i	Cantidad de grupos inyectores de la máquina i ($\sigma_i \in \mathbb{N}$)
V_{ik}	$\text{mín}(\sigma_i, CM_k) \quad (V_{ik} \in \mathbb{N})$
τ	Índice del último intervalo temporal
g_t	Cantidad de intervalos del horario administrativo que suceden al intervalo t: $g_t = \lvert T \rvert - (t+1) - \lvert T_{NA} \cap T_{[t+1,\tau]} \rvert$ ($g_t \in \mathbb{Z}_{\geq 0}$)
O_k	Cantidad de stock para el molde $k \in K_{HD}$ ($O_k \in \mathbb{R}_{\geq 0}$)
$CMAX$	Cantidad máxima de cambios de moldes en un intervalo de tiempo ($CMAX \in \mathbb{Z}_{\geq 0}$)
$start_{ik}$	Cantidad de grupos inyectores de la máquina i que están utilizando el molde k en el setup inicial.
P_i	Tiempo máximo de parada por cambio de moldes de la máquina i, en minutos
dur	Duración en minutos de los intervalos de tiempo del horario administrativo
φ_{ikt}	Cantidad de cierres del molde k que son producidos en el intervalo t por un grupo inyector de la máquina i
$NP_k^{(j)}$	Demanda acumulada de los primeros j trabajos del molde k
N_k	Índices de los trabajos que requieren exclusivamente al molde k, ordenados por fecha de entrega
d_{kj}	Índice del intervalo temporal correspondiente a la fecha de entrega del j-ésimo trabajo del molde k
SPA_k	Sobreproducción admisible acumulada para el molde k
p_{kj}	prioridad del j-ésimo trabajo del molde k ($p_{kj} \in [0{,}05, 1]$)
GMB	cantidad de gambettos
HDS	cantidad de portacabezas

Tabla 18: Conjuntos y parámetros del modelo MILP del Etapa 1

Notación	Descripción
x_{ikt}	cuántos grupos inyectores de la máquina i utilizan el molde k en el intervalo t, ($x_{ikt} \in \mathbb{Z}_{\geq 0}$, $i \in I, k \in K, t \in T$)
θ_{ikt}	indica si la máquina i utiliza el molde k en el intervalo temporal t ($\theta_{ikt} \in \{0,1\}$, $i \in I, k \in K, t \in T$)
y_{it}	indica si hay un cambio de moldes en la máquina i en el intervalo temporal t ($y_{it} \in \{0,1\}$, $i \in I, t \in T$)
ρ_{it}	indica si la máquina i está detenida en el intervalo temporal t, ($\rho_{it} \in \{0,1\}$, $i \in I, t \in T$)
η_{ikt}	cuántos grupos inyectores de la máquina i está utilizando el molde k en el intervalo $t-1$ si i está detenida en el intervalo t ($\eta_{ikt} \in \mathbb{Z}_{\geq 0}$, $i \in I, k \in K, t \in T$)
w_{kjt}	indica si la demanda del j-ésimo trabajo del molde k está satisfecha al comienzo del intervalo t ($w_{kjt} \in \{0,1\}$, $k \in K, j \in N_k, t \in T$)
μ_{kj}	cuantifica la demora del j-ésimo trabajo del molde k en términos de intervalos temporales ($\mu_{kj} \in \mathbb{R}$, $k \in K, j \in N_k$)
z_k	Cuantifica la sobreproducción no deseable del molde k ($z_k \in \mathbb{R}_{\geq 0}$, $k \in K$)
r_{it}	Indica si la máquina i está inactiva durante el intervalo temporal t ($r_{it} \in \{0,1\}$)

Tabla 19: Variables de decisión del MILP de la Etapa 1

$$\sum_{\ell \in K-\{k\}} \theta_{i\ell t} \leq \sigma_i(1-\theta_{ikt}) \qquad k \in K_{NC},\ i \in I_{NJ},\ t \in T \tag{75}$$

$$x_{ik(t-1)} - x_{ikt} \leq V_{ik} y_{it} \qquad i \in I,\ k \in K,\ t \in T_{\geq 1} \tag{76}$$

$$x_{ikt} - x_{ik(t-1)} \leq V_{ik} y_{i(t-1)} \qquad i \in I,\ k \in K,\ t \in T_{\geq 1} \tag{77}$$

$$\sum_{i \in I} P_i y_{it} \leq dur \qquad t \in T_{\geq} 1 \tag{78}$$

$$\sum_{i \in I} y_{it} \leq CMAX \qquad t \in T_{\geq} 1 \tag{79}$$

$$\sum_{i \in I} y_{it} = 0 \qquad t \in T_{NA} \tag{80}$$

$$x_{ik0} = start_{ik} \qquad i \in I,\ k \in K \tag{81}$$

$$\sum_{s=0}^{t} \sum_{i \in I} \sum_{\ell \in B_k} \varphi_{i\ell s} x_{i\ell s} \leq O_k + \sum_{s=0}^{t-1} \sum_{i \in I} \varphi_{iks} x_{iks} \qquad k \in K_{HD},\ t \in T_{\geq 1} \tag{82}$$

$$\sum_{s=0}^{t-1} \sum_{i \in I} \varphi_{iks} x_{iks} \geq NP_k^{(j)} w_{kjt} \qquad k \in K,\ j \in N_k,\ t \in T_{[d_{kj}+1, \tau-1]} \tag{83}$$

$$w_{k,j,\tau} = 1 \qquad k \in K,\ j \in N_k \tag{84}$$

$$\mu_{kj} \geq t(1-w_{kjt}) - d_{kj} \qquad k \in K,\ j \in N_k,\ t \in T_{[d_{kj}+1, \tau-1]} \tag{85}$$

$$\sum_{t \in T} \sum_{i \in I} \varphi_{ikt} x_{ikt} \leq SPA_k + z_k \qquad k \in K \tag{86}$$

$$g_t(\rho_{it} - 1 - y_{it}) + \sum_{s=t+1}^{\tau} y_{is} \leq g_t r_{it} \qquad i \in I,\ t \in T \tag{87}$$

$$x_{ikt}, \eta_{ikt} \in \mathbb{Z}_{\geq 0} \qquad i \in I,\ k \in K,\ t \in T$$

$$\theta_{ikt} \in \{0, 1\} \qquad i \in I,\ k \in K,\ t \in T$$

$$y_{it}, \rho_{it} \in \{0, 1\} \qquad i \in I,\ t \in T$$

$$w_{kjt} \in \{0, 1\} \qquad k \in K,\ j \in N_k,\ t \in T$$

$$\mu_{kj} \in \mathbb{R}_{\geq 0} \qquad k \in K,\ j \in N_k$$

$$z_k \in \mathbb{R}_{\geq 0} \qquad k \in K$$

$$r_{it} \in \{0, 1\} \qquad i \in I,\ t \in T$$

(60)–(62) plantean los objetivos de la minimización lexicográfica. (60) expresa la demora ponderada total; (61) contabiliza la cantidad de sobreproducción no deseable en relación a la sobreproducción admisible y (62) la suma de los tiempos en los que terminó de trabajar cada máquina para reducir los tiempos de inactividad. (63) asegura que se respete la cantidad de grupos inyectores de cada máquina y que la misma no produzca si está ocurriendo un cambio de moldes. (64) garantiza que se respete la compatibilidad molde-máquina. (65)–(67) indica que se debe respetar la cantidad de juegos de moldes, de gambetos y de portacabezas disponibles, respectivamente. La presencia de las variables η en (65)–(67) es esencial para establecer que un molde no está automáticamente disponible cuando la máquina se detiene por un cambio de moldes, sino que hay que considerarlo como no disponible durante ese intervalo de tiempo. De lo contrario, podría ocurrir una situación como la que se ilustra en la Figura 33 que no es factible en la práctica si sólo se cuenta con un juego de molde M01. (68) y (69) modelan que una máquina no está utilizando el molde k si y sólo si ningún grupo inyector está equipado con un juego de ese molde. (70) y (71) garantizan que la máquina está detenida si y sólo si no está elaborando productos de ningún molde. (72)–(74) modelan que $\eta_{ikt} = x_{ik(t-1)}$ si y sólo si la máquina i se detuvo a tiempo t, de lo contrario $\eta_{ikt} = 0$. (75) asegura que se cumple que los moldes en K_{NC} no compartan máquinas con grupos inyectores solidarios con otros moldes. (76) y (77) declaran que si en el intervalo temporal actual ni en el anterior hubo cambio de moldes, entonces la cantidad de moldes de cada tipo en la máquina i sigue siendo la misma. (78) y (79) garantizan que la duración de los cambios de molde en un intervalo temporal no excedan dicho periodo y que se respete la cantidad máxima de cambios de moldes, respectivamente. (80) prohíbe que se realicen cambios de molde fuera del horario administrativo. (81) impone que se respete el setup inicial de las máquinas. (82) controla la producción de gambos de los productos bicolor: hasta el intervalo de tiempo t, la cantidad de producida de gambos que serán inyectados a una cabeza de molde k están acotada por lo producido del molde k hasta el intervalo anterior más su stock. (83) indica que cuando $w_{kjt} = 1$, la demanda del j-ésimo trabajo del molde k está satisfecha. (84) obliga que la demanda de todos los trabajos estén satisfechas al finalizar el horizonte de planificación. (85) acota inferiormente a la demora del j-ésimo trabajo del molde k. (86) acota inferiormente la sobreproducción no deseable del molde k a lo largo de todo el horizonte de producción. (87) modela que t es un intervalo de inactividad para la máquina i si la máquina i está detenida en el intervalo t, no está realizando un cambio de moldes y en algún intervalo posterior se la volverá a equipar con algún molde. Notar que, por definición, $\sum_{s=t+1}^{\tau} y_{is} \leq g_t$.

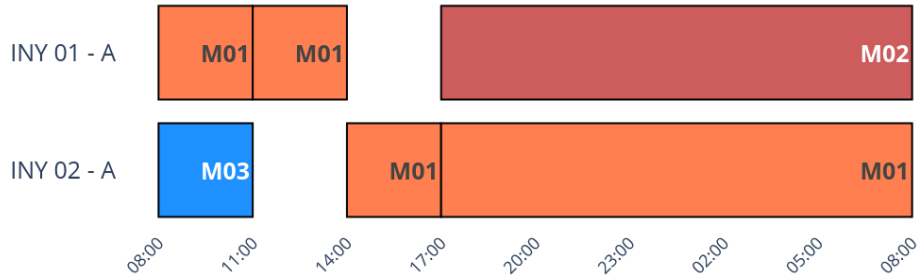


Figura 33: Ejemplo de lo que podría ocurrir si no se consideraran las variables η, con dos máquinas (un grupo inyector cada una) y un juego de molde M01. En el intervalo de tiempo 14-17hs. el molde M01 no está disponible pues todavía se está retirando de la máquina 1. Sin embargo ya está asignado en la máquina 2.

También incluimos las siguientes desigualdades como planos de corte para resolver el modelo con la segunda función objetivo:

$$x_{ikt} \leq x_{ik(t+1)} + \eta_{ik(t+1)} \quad i \in I,\ k \in K^{(i)},\ t \in T_{NA} - \{\tau\} \tag{88}$$

$$x_{ikt} \geq x_{ik(t+1)} + \eta_{ik(t+1)} \quad i \in I,\ k \in K^{(i)},\ t \in T_{NA} - \{\tau\} \tag{89}$$

$$y_{it} \geq \rho_{it} - \rho_{i(t-1)} \quad i \in I,\ t \in T - \{\tau\} \tag{90}$$

Proposición 21. Las desigualdades (88)–(90) son desigualdades válidas.

Demostración. Observemos que las desigualdades (88) y (89) modelan la siguiente igualdad:

$$x_{ikt} = x_{ik(t+1)} + \eta_{ik(t+1)} \tag{91}$$

por lo tanto, basta probar que cualquier solución del modelo cumple con (91). Sean $i \in I$, $k \in K^{(i)}$, $t \in T_{NA} - \{\tau\}$, por (80) tenemos que $y_{it} = 0$. Para cualquier solución del modelo, se tiene que $y_{i(t+1)} = 0$ o $y_{i(t+1)} = 1$. Veamos que en ambos casos se cumple la igualdad:

$$y_{i(t+1)} = 0 \xrightarrow{(76)(77)} x_{ikt} = x_{ik(t+1)} \xrightarrow{(72)\text{–}(74)} \eta_{ik(t+1)} = 0 \Rightarrow x_{ikt} = x_{ik(t+1)} + \eta_{ik(t+1)}$$

$$y_{i(t+1)} = 1 \xrightarrow{(63)} x_{ik(t+1)} = 0 \xrightarrow{(72)\text{–}(74)} \eta_{ik(t+1)} = x_{ikt} \Rightarrow x_{ikt} = x_{ik(t+1)} + \eta_{ik(t+1)}$$

Como la igualdad se respeta para cualesquiera $i \in I$, $k \in K^{(i)}$, $t \in T_{NA} - \{\tau\}$, se deduce que (91) se cumple para toda solución del modelo.

La desigualdad (90) modela que, si una máquina no está detenida durante el intervalo temporal $(t-1)$ y está detenida durante el intervalo t, entonces en el intervalo t se realiza un cambio de moldes. Supongamos que no es una desigualdad válida, es decir, que existe s una solución del modelo que no cumple (90). Notamos $(x^*, y^*, \theta^*, \rho^*)$ a los valores que toman las variables en s. Dado que las variables y y ρ sin binarias, que s no cumpla (90) es equivalente

a que existan $\hat{\imath} \in I$, $\hat{t} \in T - \{\tau\}$ tales que el lado izquierdo de la desigualdad valga 0 y el derecho valga 1. Esto ocurre si y solo si $y^*_{\hat{\imath}\hat{t}} = \rho^*_{\hat{\imath}(\hat{t}-1)} = 0$ y $\rho^*_{\hat{\imath}\hat{t}} = 1$. Tenemos entonces que:

$$\rho^*_{\hat{\imath}(\hat{t}-1)} = 0 \xRightarrow{(71)} \exists\, \hat{k} \in K \text{ tal que } \theta^*_{\hat{\imath}\hat{k}(\hat{t}-1)} = 1 \xRightarrow{(69)} x^*_{\hat{\imath}\hat{k}(\hat{t}-1)} \geq 1 \xRightarrow{(76), y^*_{\hat{\imath}\hat{t}}=0} x^*_{\hat{\imath}\hat{k}(\hat{t}-1)} \leq x^*_{\hat{\imath}\hat{k}\hat{t}}$$
$$\xRightarrow{(68)} \theta^*_{\hat{\imath}\hat{k}\hat{t}} = 1 \xRightarrow{(70)} \rho^*_{\hat{\imath}\hat{t}} = 0 \quad \text{Abs!}$$

El absurdo provino de suponer que existe una solución del modelo que no cumple (90). Por lo tanto, es una desigualdad válida. □

La solución de la optimización lexicográfica determina cuántos juegos de cada tipo de molde están en uso en cada máquina durante cada intervalo de tiempo. La factibilidad del modelo está fuertemente ligada a la compatibilidad de los parámetros. Por ejemplo, el modelo puede resultar infactible si la demanda de cierto tipo de molde excede la capacidad productiva de la fábrica para el horizonte de planificación. Por esta razón, en la implementación computacional incluimos rutinas que analizan los datos de entrada e informan sobre errores e incompatibilidad entre parámetros que causarían que el modelo fuera infactible.

4.2. Etapa 2 - Ajuste de los tiempos de parada

En la primera etapa se sobreestima el tiempo que la máquina debe detenerse por cambio de moldes, pues los intervalos de tiempo son más largos que el máximo entre todas las posibles duraciones de parada de las máquinas. Recordemos que la duración de la parada está determinada no sólo por la máquina, sino también por la cantidad de grupos inyectores a los que hay que cambiarle el molde. Por otro lado, sólo se pueden cambiar moldes en una máquina a la vez.

Si dos máquinas cambian de molde durante el intervalo t, puede ocurrir que deban llevarse a cabo en un orden determinado. Llamaremos I_1 e I_2 a las máquinas que cambian molde durante cierto intervalo de tiempo t. Pueden ocurrir tres situaciones: el orden de parada no es importante, una máquina debe parar antes que la otra o las dos deben detenerse simultáneamente. La descripción de cada una de ellas se encuentra a continuación, y es acompañada por los ejemplos de la Figura 34, donde hay dos máquinas que poseen un grupo inyector cada una y donde sólo hay un juego de cada tipo de molde.

1. Orden libre: los moldes a introducir en cada una de las máquinas no dependen de los moldes que salen de la otra, ya sea porque son de distinto tipo (Figura 34a) o porque hay otros juegos disponibles.

2. Orden fijo: el juego de moldes que se saca de I_1 debe ser introducido en I_2, pero el juego que sale de I_2 no debe ser necesariamente introducido en I_1; por lo tanto I_1 debe detenerse primero (Figura 34b). Análogo para el caso inverso.

3. Parada simultánea: el juego de moldes que se saca de I_1 debe ser introducido en I_2, y el molde que sale de I_2 debe ser introducido en I_1. En este caso, ambas máquinas se detienen por la suma del tiempo de parada de ambas. (Figura 34c).

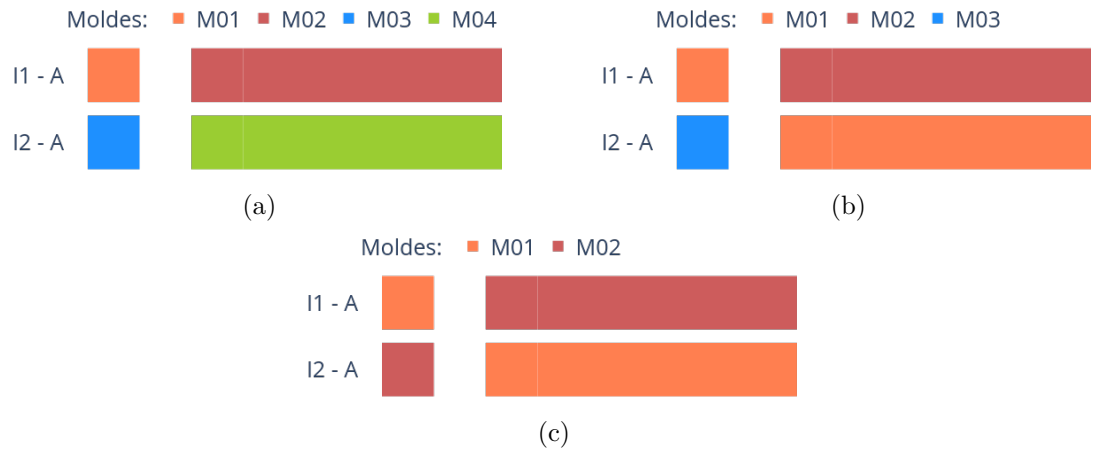


Figura 34: Ejemplos de las tres situaciones posibles cuando dos máquinas cambian de molde en el mismo intervalo temporal. (a) cualquier orden de parada es factible. (b) I_1 debe parar primero para liberar M01. (c) deben detenerse simultáneamente

Formalicemos estas tres situaciones para los casos en los que m máquinas $I_1, \ldots, I_m$ deben realizar un cambio de moldes durante el mismo intervalo de tiempo. Definimos el digrafo de precedencia $G = (V, A)$ con $V = \{1, \ldots, m\}$ y A tal que $(k, h) \in A$ si y solo si al menos un juego de moldes que se saca de I_k debe ser introducido a I_h. En la Figura 35 mostramos un ejemplo en el que ocho inyectoras deben cambiar moldes durante el mismo intervalo de tiempo.

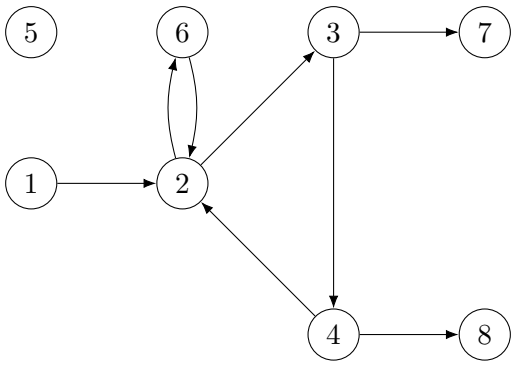


Figura 35: Ejemplo de digrafo utilizado para determinar el orden de parada de ocho máquinas que deben cambiar moldes durante el mismo intervalo de tiempo.

La situación 3 (Parada simultánea) ocurre para las máquinas que formen parte de la misma componente fuertemente conexa de G. En el ejemplo, el conjunto de inyectoras $\{I_2, I_3, I_4, I_6\}$ deben detenerse simultáneamente por la cadena de requerimientos de moldes que las vinculan. En este caso consideraremos que el tiempo de parada se corresponde a la suma del tiempo necesario para cambiar los moldes de cada máquina. Por ejemplo, si para cambiar los moldes de I_2, I_3 e I_4 se requiere media hora para cada uno y es necesaria una hora para el cambio de I_6, entonces la duración de la parada para las cuatro inyectoras es de dos horas y media. La situación 1 (Orden libre) entre dos máquinas es equivalente a que no exista un camino en

G que las conecte. En el ejemplo, I_1 e I_5 pueden detenerse en cualquier orden. La situación 2 (Orden fijo) entre I_j e I_h es equivalente a que exista un camino desde j hasta h en G y que no exista un camino desde h a j (de lo contrario, pertenecerían a la misma componente fuertemente conexa y sus paradas deberían ser simultáneas). En el ejemplo, se deben cambiar los moldes de I_1 antes que los de I_7. Notar que cada par de inyectoras I_k, I_h se encuentra en exactamente una de estas tres situaciones. Si pertenecen a la misma componente fuertemente conexa, se detienen al mismo tiempo. De lo contrario, no existe un camino entre ellas en G (orden libre) o k es alcanzable desde h o h es alcanzable desde k en G, pero no ambas (orden fijo).

El modelo MILP que implementamos en esta etapa determina cuándo comienza cada cambio de molde en cada máquina y decide el orden en que se detienen las máquinas que cambian moldes en el mismo intervalo de tiempo. El objetivo es minimizar la cantidad de sobreproducción no deseable de cada molde, pues el ajuste de los tiempos de parada no provoca retraso en el cumplimiento de las fechas de entrega ni aumenta el tiempo de inactividad de las máquinas obtenido por la Etapa 1. En la Tabla 20 detallamos los conjuntos y parámetros del modelo MILP de la etapa 2 y en la Tabla 21, sus variables.

El modelo MILP2 queda planteado de la siguiente manera:

$$\text{mín} \sum_{\substack{k\in K\\ SPA_k\neq 0}} \frac{z_k}{SPA_k} + \sum_{\substack{k\in K\\ SPA_k=0}} z_k \tag{92}$$

sujeto a:

$$x_{ij} \geq \ell_{ij} \qquad i\in I,\ j\in J_i \tag{93}$$

$$x_{ij} + \tau_{ij} \leq u_{ij} \qquad i\in I,\ j\in J_i \tag{94}$$

$$\sum_{i\in I}\left(\sum_{j=2}^{|J_i|-1}(C_{kij}\varphi_{ik}(x_{ij}-(x_{i(j-1)}+\tau_{i(j-1)}))+ C_{ki1}\varphi_{ik}x_{i1} + C_{ki|J_i|}\varphi_{ik}(t_f-(x_{i|J_i|}+\tau_{i|J_i|}))\right) \leq SPA_k + z_k \qquad k\in K \tag{95}$$

$$x_{hj_2} \geq x_{ij_1} + \tau_{ij_1} \qquad (i,h,j_1,j_2)\in\mathcal{O} \tag{96}$$

$$x_{hj_2} + \tau_{hj_2} - t_f\theta_{ihj_1j_2} \leq x_{ij_1} \qquad (i,h,j_1,j_2)\in\mathcal{F} \tag{97}$$

$$x_{ij_1} + \tau_{ij_1} - t_f(1-\theta_{ihj_1j_2}) \leq x_{hj_2} \qquad (i,h,j_1,j_2)\in\mathcal{F} \tag{98}$$

$$x_{ij_1} = x_{hj_2} \qquad (i,h,j_1,j_2)\in\mathcal{S} \tag{99}$$

$$x_{ij} \in \mathbb{R}_{\geq 0} \qquad i\in I,\ J\in J_i$$

$$z_k \in \mathbb{R}_{\geq 0} \qquad k\in K$$

$$\theta_{ihj_1j_2} \in \{0,1\} \qquad (i,h,j_1,j_2)\in\mathcal{F}$$

El objetivo (92) es minimizar la sobreproducción no admisible. Las restricciones (93) y (94) aseguran que cada parada ocurra dentro del intervalo indicado por la solución de la Etapa 1. El lado izquierdo de (95) cuantifica la producción total de molde k, que sirve de cota inferior para la sobreproducción no deseable. (96) garantiza que se respete la disposición de las paradas con orden fijo, mientras que (97)–(98) no permiten que las paradas con orden libre se superpongan. Finalmente, (99) asegura la simultaneidad de las paradas que deben ocurrir al mismo tiempo.

Notación	Descripción
Conjuntos	
I	Conjunto de máquinas
J_i	Conjunto de índices de paradas de la máquina i
K	Conjunto de moldes
$\mathcal{F}$	(i, h, j_1, j_2) tal que $i, h \in I$, $j_1 \in J_i$, $j_2 \in J_h$ pertenece a $\mathcal{F}$ si y sólo si la j_1-ésima parada de la máquina i y la j_2-ésima parada de la máquina k ocurren durante el mismo intervalo temporal pero no es necesario que una ocurra antes que la otra. (Orden libre)
$\mathcal{O}$	(i, h, j_1, j_2) tal que $i, h \in I$, $j_1 \in J_i$, $j_2 \in J_h$ pertenece a $\mathcal{O}$ si y sólo si la j_1-ésima parada de la máquina i y la j_2-ésima parada de la máquina h ocurren durante el mismo intervalo temporal y j_1 debe ocurrir antes que j_2. (Orden fijo).
$\mathcal{S}$	(i, h, j_1, j_2) tal que $i, h \in I$, $j_1 \in J_i$, $j_2 \in J_h$ pertenece a $\mathcal{S}$ si y sólo si la j_1-ésima parada de la máquina i y la j_2-ésima parada de la máquina h ocurren durante el mismo intervalo de tiempo y j_1 debe ocurrir al mismo tiempo que j_2. (Parada simultánea)
Parámetros	
t_0	tiempo inicial del periodo de planificación horaria
t_f	longitud del periodo de planificación horaria, en horas ($t_f \in \mathbb{R}_{\geq 0}$)
φ_{ik}	Cantidad de cierres del molde k que son producidos por hora por un grupo inyector de la máquina i
SPA_k	Sobreproducción admisible para el molde k
τ_{ij}	Duración, en horas, de la j-ésima parada de la máquina i ($\tau_{ij} \in \mathbb{R}_{\geq 0}$)
ℓ_{ij}	Comienzo del intervalo temporal donde ocurre la j-ésima parada de la máquina i , en horas desde t_0 ($\ell_{ij} \in \mathbb{R}_{\geq 0}$)
u_{ij}	Final del intervalo temporal donde ocurre la j-ésima parada de la máquina i, en horas desde t_0 ($u_{ij} \in \mathbb{R}_{\geq 0}$)
C_{kij}	Cantidad de grupos inyectores utilizando el molde k en la máquina i antes de su j-ésima parada ($C_{kij} \in \mathbb{Z}_{\geq 0}$)

Tabla 20: Conjuntos y parámetros del modelo MILP de la Etapa 2

Notación	Descripción
x_{ij}:	cuándo comienza la j-ésima parada de la máquina i, en horas desde t_0 ($x_{ij} \in \mathbb{R}_{\geq 0}$)
z_k:	cantidad de sobreproducción no deseable del molde k ($z_k \in \mathbb{R}_{\geq 0}$)
$\theta_{ihj_1j_2}$:	para $(i, h, j_1, j_2) \in \mathcal{F}$, $\theta_{ihj_1j_2}$ vale 1 si y solo si la j_1-ésima parada de la máquina i ocurre durante el mismo intervalo temporal que la j_2-ésima parada de la máquina h y la máquina i se detiene primero ($\theta_{ihj_1j_2} \in \{0, 1\}$)

Tabla 21: Variables de decisión del modelo MILP de la Etapa 2

A continuación, probaremos que MILP2 siempre es factible. Para esto, recordamos la definición de condensación de un grafo dirigido y de orden topológico y un resultado conocido para digrafos acíclicos.

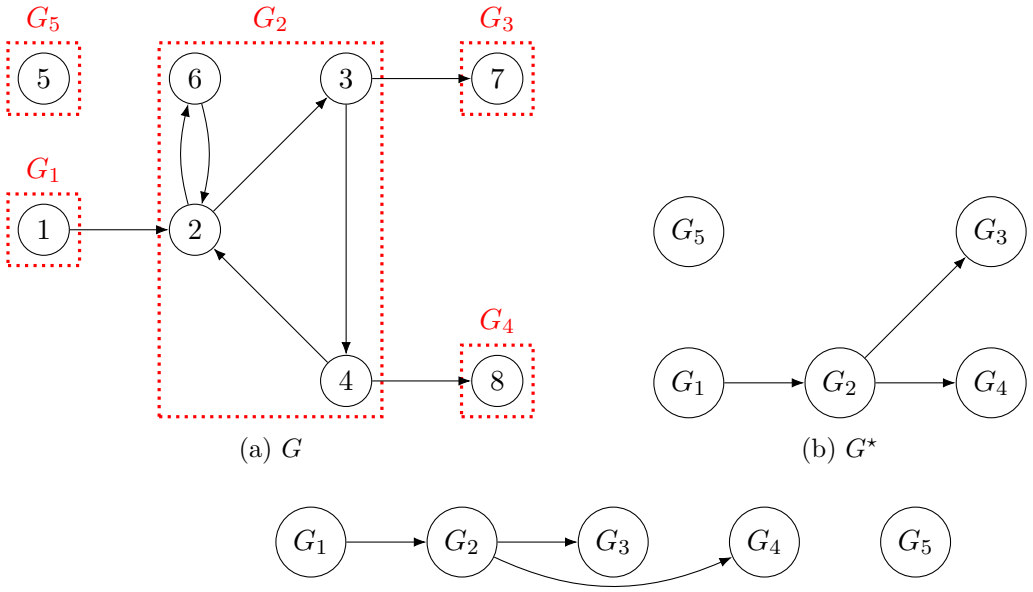
Definición 51 (Condensación de un digrafo)**.** Sea $G = (A, V)$ un digrafo cuyas componentes fuertemente conexas son $G_1, \ldots, G_k$, la *condensación* de G es un digrafo $G^\star$ cuyos vértices son las componentes fuertemente conexas de G y donde un arco une G_i con G_j si y solo si un vértice de G_i es adyacente a uno de G_j en G.

Notar que la condensación de un digrafo G es un digrafo acíclico (DAG, por sus siglas en inglés).

Definición 52 (Orden topológico)**.** Sea G un digrafo, un orden topológico de G es un ordenamiento de los vértices tal que para cada arco (u, v), u precede a v en el orden.

Proposición 22. Sea G un digrafo, G tiene al menos un orden topológico si y solo si G es un DAG.

En la Figura 36 mostramos un ejemplo con las componentes fuertemente conexas del grafo de la Figura 35, su condensación y la representación de un orden topológico de ella.



(a) G (b) $G^\star$

(c) Representación de un orden topológico de $G^\star$

Figura 36: Las componentes fuertemente conexas del grafo G del ejemplo de la Figura 35 (a), su condensación (b) y un orden topológico de su condensación (c).

Proposición 23. El modelo MILP de la Etapa 2 siempre es factible.

Demostración. Construyamos una solución $(x^\star, \theta^\star, z^\star)$ para MILP2. Sin pérdida de generalidad supondremos que hay un único intervalo temporal durante el cual se detienen algunas máquinas. Si hubiera más de uno, el procedimiento de construcción de una solución es análogo al que mostraremos a continuación. Supongamos que este intervalo se extiende desde la hora

ℓ hasta la hora u del horizonte de planificación y que durante este intervalo m máquinas cambian de molde.

Elaboramos el digrafo de precedencia de paradas G. Sea $G^\star$ la condensación de G, $G^\star$ es un DAG y por lo tanto admite un orden topológico. Etiquetamos $G_1, \ldots, G_n$ a las componentes conexas de G según ese orden topológico en $G^\star$. Notamos V_{G_p} al conjunto de vértices de G_p. A continuación, definimos:

(A) $x^\star_{i1} = \ell$ para cada $i \in V_{G_1}$.

(B) $x^\star_{i1} = x^\star_{h1} + \tau_{h1}$ para cada $i \in V_{G_p}$ con $h \in V_{G_{p-1}}$, $2 \leq p \leq n$.

(C) para cada $(i, j, 1, 1) \in \mathcal{F}$:

$$\theta^\star_{(i,h,1,1)} = \begin{cases} 1 & \text{si } x^\star_{i1} \leq x^\star_{h1} \\ 0 & \text{c.c.} \end{cases}$$

(D) $z^\star_k = \text{máx}(LHS - SPA_k, 0)$ para cada molde k con LHS el lado izquierdo de la correspondiente desigualdad (95).

Probemos que $(x^\star, \theta^\star, z^\star)$ es solución de MILP2. Por las restricciones (78) de MILP1, se tiene que la suma de los tiempos de parada de las m máquinas son mejores o iguales que la duración del intervalo temporal $u - \ell$. Como en $x^\star$ la parada de las máquinas está ordenada según el orden topológico de las componentes fuertemente conexas a las que pertencen, por (A) y (B) se deduce que se cumplen (93) y (94). Por (D), $z^\star$ cumple las restricciones (95).

Tomemos dos máquinas i y h que deben detenerse durante este intervalo temporal. Si $(i, h, 1, 1) \in \mathcal{O}$, entonces la componente fuertemente conexa a la que pertenece i precede a la que pertenece j en el orden topológico de $G^\star$. Por (B), vale la restricción (96). Si $(i, h, 1, 1) \in \mathcal{F}$, por (C) se cumplen las restricciones (97) y (98). Finalmente, si $(i, h, 1, 1) \in \mathcal{S}$, pertenecen a la misma componente fuertemente conexa. Por lo tanto, por ítems (A) o (B), sus paradas comienzan en el mismo momento y se cumple (99).

Por lo tanto, $(x^\star, \theta^\star, z^\star)$ es una solución factible para MILP2. Si hubiera más de un intervalo temporal en los que ocurren cambios de moldes, el mismo razonamiento puede llevarse a cabo en cada uno de ellos para construir una solución factible. □

4.3. Etapa 3 - Determinación del color para cada molde

La solución obtenida luego de las primeras dos etapas sirve como planificación horaria para la utilización de los moldes, garantizando que se satisfagan las restricciones operativas y de procedimientos de producción. La Etapa 3 determina cuándo deben llevarse a cabo los cambios de color que diferencian a los productos de cada molde. En esta instancia, se resuelve un modelo MILP por separado para cada molde que participe en la elaboración de cierres de vino, monoblocco y T-Wood. Generalizamos el modelo para los productos bicolor, debido a la dependencia que existe entre la producción de la cabeza y la coinyección de los gambos.

Por esa razón, presentaremos el modelo que denominamos MILP3 para el caso de los productos bicolor y luego explicaremos cómo se aplica a los modelos de los demás productos. En esta etapa, nombraremos a los colores como Cx donde x es un número de uno o dos dígitos, notaremos con I al conjunto de índices de combinaciones molde-color y con I_k al conjunto de

combinaciones en I asociadas con el molde k. Notaremos con F_i a la vinculación del conjunto de combinaciones molde-color de gambos a una combinación molde-color i de la cabeza.

Para mayor claridad, acompañaremos la explicación con un ejemplo, en el cual los productos a elaborar tienen las características que se detallan en la Tabla 22. En el ejemplo, $I = \{1, 2, 3, 4, 5, 6, 7\}$, M01 es el molde de la cabeza y {M02, M03, M04} son los moldes que van coinyectados a ella. Según la tabla, las combinaciones correspondientes al molde M03 son la combinación 3 y la combinación 4, por lo tanto $I_{M03} = \{3, 4\}$. Por otro lado, notar que las producciones de M02-C10 (combinación 3), M03-C05 (combinación 4) y M03-C06 (combinación 5) están vinculadas con la producción de M01-C01 (combinación 1), por ende, $F_1 = \{3, 4, 5\}$.

Producto	Molde de la cabeza	Color	Indice combinación	Molde del gambo	Color	Índice combinación
Producto 1	M01	C01	1	M02	C10	3
Producto 2	M01	C01	1	M03	C05	4
Producto 3	M01	C01	1	M03	C06	5
Producto 4	M01	C02	2	M04	C08	6
Producto 5	M01	C02	2	M04	C09	7

Tabla 22: Características de los productos del ejemplo e índices de las combinaciones molde-color

Un aspecto esencial para la resolución de esta etapa es la definición de los segmentos de producción. Un segmento de producción es un intervalo temporal en el cual un molde está asignado en una máquina, abarcando todos sus grupos inyectores en los que está equipado. Cada segmento de producción está delimitado por un cambio de moldes o por una fecha de entrega. En el caso de los productos bicolor, otro criterio para la finalización de un segmento es el comienzo del segmento de otro molde luego de un cambio de moldes. Notamos como T al conjunto de los instantes en el tiempo t que marcan la división de segmentos debido a una fecha de entrega o debido al comienzo de un segmento luego de un cambio de moldes. Notaremos $\mathcal{S}^{(k)}$ a los segmentos correspondientes al molde k y diremos que un segmento S_n precede a S_m si ocurren en la misma máquina y el final de S_n coincide con el comienzo de S_m.

En la Figura 37 mostramos la planificación horaria resultante de las primeras dos etapas para nuestro ejemplo, restringida a los moldes de interés (M01–M04) y cómo quedan determinados los segmentos de producción. Según las definiciones del párrafo anterior, por ejemplo, $\mathcal{S}^{(M02)} = \{S_9, S_{12}, S_{15}, S_{18}\}$ y S_{12} precede a S_{15}.

Para cada molde k, cada segmento en $\mathcal{S}^{(k)}$ es particionado en fragmentos y en cada fragmento se utilizará exactamente una combinación molde-color. La cantidad de fragmentos viene dada por la cantidad de combinaciones asociadas a ese molde, es decir, $|I_k|$. Por ejemplo, $S_{10} \in \mathcal{S}^{(M04)}$ es particionado en dos fragmentos pues $|I_{M04}| = 2$. En cada segmento debe utilizarse al menos una combinación de I_k, pero no es necesario que se utilicen todas. Por lo tanto, todos los fragmentos, salvo el primero, pueden tener una duración nula. La resolución de MILP3 decidirá la duración de cada fragmento, minimizando la cantidad de cambios de

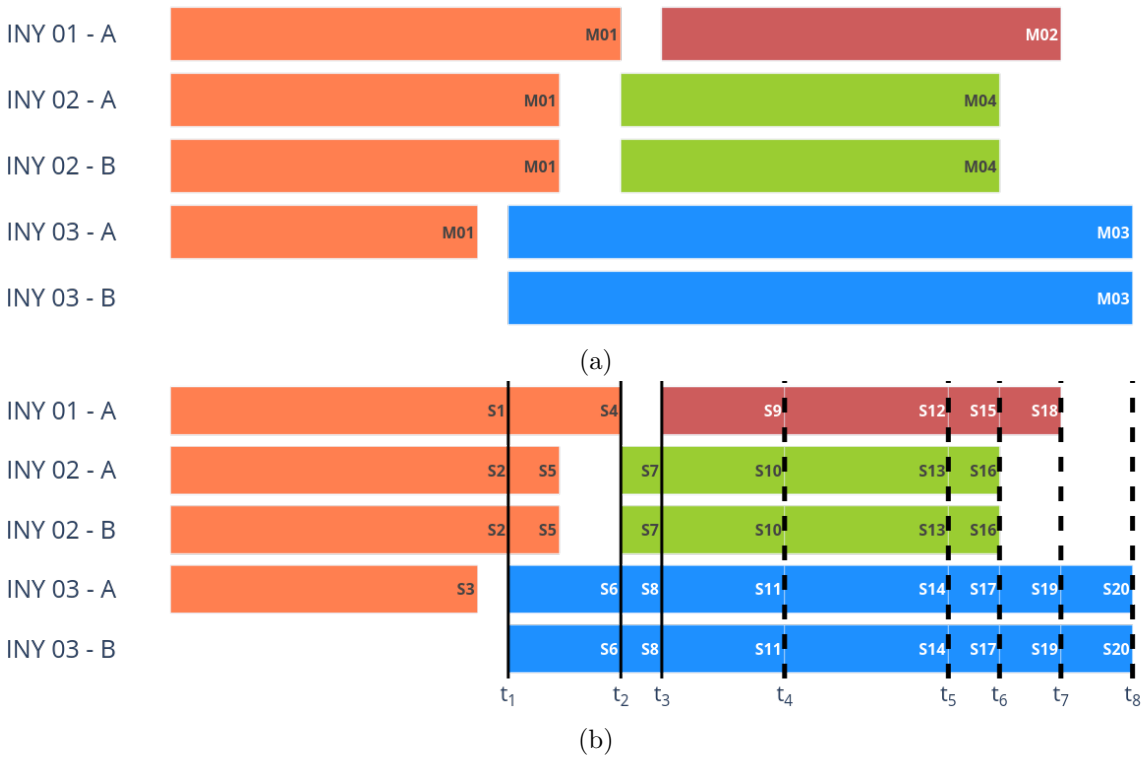


Figura 37: (a) Planificación obtenida luego de las dos primera etapas, restringida a los moldes M01–M04 y (b) sus correspondientes segmentos de producción. Las líneas punteadas corresponden a las fechas de entrega de los productos de la Tabla 22

color y garantizando que se cumpla la demanda de cada producto. En Tabla 23 describimos los demás conjuntos de MILP3.

La Tabla 24 muestra la descripción de las variables de MILP3. Para minimizar la cantidad de cambios de color, no sólo es necesario conocer cuántas combinaciones se utilizan en cada segmento, sino también si, para cada $(s, s') \in D$ ocurre un cambio entre el final de s y el comienzo de s'. Por esta razón incluimos las variables z y w.

Formulamos MILP 3 de la siguiente manera:

$$\text{mín} \sum_{k \in K} \sum_{i \in I_k} \sum_{p=2}^{|I_k|} \sum_{s \in \mathcal{S}^{(k)}} y_{ips}^{(k)} + \sum_{k \in K} \sum_{s \in D_k^{(B)}} w_s^{(k)} \tag{100}$$

sujeto a:

$$\sum_{i \in I_k} y_{i1s}^{(k)} \geq 1 \qquad s \in \mathcal{S}^{(k)},\ k \in K \tag{101}$$

$$\sum_{i \in I_k} y_{ips}^{(k)} \leq 1 \qquad s \in \mathcal{S}^{(k)},\ p \in P_k,\ k \in K \tag{102}$$

$$\sum_{i \in I_k} \sum_{p \in P} x_{ips}^{(k)} \leq L_s \qquad s \in \mathcal{S}^{(k)},\ k \in K \tag{103}$$

Notación	Descripción
Conjuntos	
K	Conjunto de moldes (correspondientes a la cabeza y a los moldes coinyectados)
I	Conjunto de combinaciones molde-color
I_k	Conjunto de combinaciones molde-color con molde k
J_i	Conjunto de índices de los trabajos de la combinación i
P_k	Conjunto de índices de los fragmentos de cada segmento del molde k ($P = 1, \ldots, \lvert I_k \rvert$)
$\mathcal{S}$	Conjunto de segmentos
$\mathcal{S}^{(k)}$	Conjunto de segmentos del molde k
$\mathcal{S}_{ij}$	Conjunto de segmentos anteriores a la j-ésima deadline de la combinación i ($S_{ij} \subset S$)
D_k	Conjunto de tuplas de segmentos del molde k, (s, s') tal que s precede a s'
$D_k^{(B)}$	Conjunto de segmentos del molde k que preceden a otro segmento.
T	conjunto de los instantes en el tiempo t que dividen segmentos debido a una fecha de entrega o al comienzo de un segmento luego de un cambio de moldes
$\mathcal{S}_{\leq t}^{(k)}$	conjunto de segmentos del molde k que terminan a lo sumo a tiempo t
Parámetros	
L_s	Duración del segmento s en horas
φ_s	Cantidad de cierres que son producidos por hora en el segmento s
NP_{ij}	Demanda del j-ésima trabajo de la combinación molde-color i
h	Molde de la cabeza
$stock_i$	Stock de la combinación molde-color i ($i \in I_h$)
M	$\lvert I \rvert + 1$

Tabla 23: Parámetros del modelo MILP de la Etapa 3

Notación	Descripción
$x_{ips}^{(k)}$	duración del fragmento p del segmento s que produce combinación i para el molde k ($x_{ips}^{(k)} \in \mathbb{R}_{\geq 0}$)
$y_{ips}^{(k)}$	indica si la combinación i para el molde k es producida durante el fragmento p del segmento s ($y_{ips}^{(k)} \in \{0, 1\}$)
$z_s^{(k)}$	última combinación producida en el segmento s, con $s \in D_k^{(B)}$ ($z_s^{(k)} \in \mathbb{Z}_{\geq 0}$), $\forall\ s \in D^{(B)}$
$w_s^{(k)}$	indica si la última combinación del segmento s no coincide con la primera de s', donde $(s, s') \in D_k$ ($w_s^{(k)} \in \{0, 1\}$)

Tabla 24: Variables del modelo MILP de la Etapa 3

$$x_{ips}^{(k)} \leq L_s y_{ips}^{(k)} \quad i \in I_k,\ p \in P_k,\ s \in \mathcal{S}^{(k)},\ k \in K \tag{104}$$

$$\sum_{i \in I_k} y_{ips}^{(k)} \leq \sum_{i \in I_k} y_{i(p-1)s}^{(k)} \quad p \in P_k, p \geq 2\ ,\ s \in \mathcal{S}^{(k)},\ k \in K \tag{105}$$

$$\sum_{s \in S_{ij}} \sum_{p \in P_k} \varphi_s x_{ips}^{(k)} \geq NP_{ij} \quad i \in I_k,\ j \in J_i,\ k \in K - \{h\} \tag{106}$$

$$\sum_{i \in I_k} i y_{ips}^{(k)} - M \sum_{r=p+1}^{|I_k|} \left(\sum_{i \in I_k} y_{irs}^{(k)} \right) \leq z_s^{(k)} \quad s \in D_k^{(B)},\ p \in P_k,\ k \in K \tag{107}$$

$$\sum_{i \in I_k} i y_{ips}^{(k)} + M \left(1 - \sum_{i \in I_k} y_{ips}^{(k)} \right) + M \sum_{r=p+1}^{|I_k|} \left(\sum_{i \in I_k} y_{irs}^{(k)} \right) \geq z_s^{(k)} \quad s \in D_k^{(B)},\ p \in P_k,\ k \in K \tag{108}$$

$$z_s^{(k)} + M w_s^{(k)} \geq \sum_{i \in I_k} i y_{i1s'}^{(k)} \quad (s, s') \in D_k,\ k \in K \tag{109}$$

$$z_s^{(k)} - M w_s^{(k)} \leq \sum_{i \in I_k} i y_{i1s'}^{(k)} \quad (s, s') \in D_k,\ k \in K \tag{110}$$

$$\left(\sum_{k \in K - \{h\}} \sum_{\ell \in I_k \cap F_i} \sum_{p \in P_k} \sum_{s \in \mathcal{S}_{\leq t}^{(k)}} \varphi_s x_{\ell ps}^{(k)} \right) - \left(\sum_{p \in P_h} \sum_{s \in \mathcal{S}_{\leq t}^{(h)}} \varphi_s x_{ips}^{(h)} \right) \leq O_i \quad i \in I_h,\ t \in T \tag{111}$$

$$x_{ips}^{(k)} \in \mathbb{R}_{\geq 0} \quad k \in K,\ i \in I_k,\ p \in P_k,\ s \in \mathcal{S}_k$$

$$y_{ips}^{(k)} \in \{0, 1\} \quad k \in K,\ i \in I_k,\ p \in P_k,\ s \in \mathcal{S}_k$$

$$z_s^{(k)} \in \mathbb{Z}_{\geq 0} \quad k \in K,\ s \in D_k^{(B)}$$

$$w_s^{(k)} \in \{0, 1\} \quad k \in K,\ s \in D_k^{(B)}$$

El objetivo (100) es minimizar los cambios de combinación molde-color. Las restricciones (101) garantizan que el primer fragmento debe producir alguna combinación y las restricciones (102) aseguran que por fragmento se puede producir a lo sumo una combinación. Las restricciones (103) afirman que la suma de la duración de los fragmentos no debe exceder a la del segmento. Las restricciones (104) imponen que si en un fragmento no se elabora combinación i, entonces la duración de ese fragmento para la combinación i debe ser nula. Las restricciones (105) rompen simetría al requerir que en un fragmento se pueda producir alguna combinación sólo si se produce en el anterior. Las restricciones (106) garantizan que la demanda de todos los productos sea satisfecha en su fecha de entrega. Como explicaremos a continuación, las restricciones (107) y (108) identifican cuál es la última combinación del segmento s. Si en cierto fragmento se produce la combinación i_0 y los fragmentos que lo suce-

den tienen duración nula, entonces los términos con M se anulan y, por ende, $z_s^{(k)} = i_0$. De lo contrario, si ese fragmento tiene duración nula o si alguno de los fragmentos que lo suceden no tiene duración nula, no se impone ninguna nueva restricción a $z_s^{(k)}$. Las restricciones (109) y (110) indican si la última combinación de un segmento coincide con la primera combinación del segmento que la sucede. Finalmente, para cada $t \in T$, las restricciones (111) limita la producción de las combinaciones de gambos en F_i a la cantidad disponible de i, la combinación de la cabeza a la que están vinculados. La aplicación de este modelo a los moldes de productos no bicolor es directa: en esos casos $K = \{k\}$, en (106) se reemplaza $k \in K - \{h\}$ por $k \in K$ y se elimina (111).

Con la solución de resolver MILP3 para cada uno de los moldes, se obtiene la planificación de producción para la planta.

5. Experimentación computacional

En esta sección mostramos los resultados obtenidos de aplicar nuestro método en una serie de instancias con diferentes características. Los procedimientos de lectura de datos, su procesamiento y el postprocesamiento de la solución fueron programados en Python 3.7 y el solver utilizado para resolver los modelos MILP fue CPLEX 12.10. Llevamos a cabo la experimentación computacional en una máquina virtual con 4 núcleos y procesador Intel Xeon Gold 3.1 GHz, 16 GB de RAM y sistema operativo Ubuntu 20.04.5 LTS.

En todas las instancias fijamos el límite de tiempo para la Etapa 1 en 6 horas. El 50 % de ese tiempo es el límite para resolver la minimización de la demora, la mitad del tiempo restante es asignada a minimizar la sobreproducción no deseable y el tiempo restante para la minimización del tiempo de inactividad de las máquinas. Dado que los modelos de la Etapa 2 y de la Etapa 3 son pequeños y se resuelven en pocos segundos, no fue necesario establecer un límite de tiempo.

También modificamos algunos de los parámetros del solver para el modelo de la Etapa 1. En general, encuentra rápidamente muy buenas cotas para la minimización de la demora. Por esta razón, para promover el hallazgo de soluciones, configuramos el solver para que enfatice factibilidad (`parameters.emphasis.mip=1`). A partir de la experimentación, para minimizar el tiempo de inactividad de las máquinas elegir el simplex primal como algoritmo para resolver la relajación lineal (`parameters.mip.strategy.subalgorithm=1`) y configurar el énfasis del solver en encontrar soluciones de buena calidad (`parameters.emphasis.mip=4`) resultó mejorar significativamente el desempeño con respecto a los parámetros por defecto.

5.1. Instancias

Generamos una variedad de instancias que sirvieran para probar nuestro enfoque de solución en contextos semejantes a los de la aplicación en la planta de producción. Están disponibles online[13] con el fin de aportar al repertorio de instancias de prueba para UPMP que permitan considerar conjuntos de restricciones tan diversos como los que han aparecido en este trabajo. Por una cuestión de confidencialidad, los datos de las instancias no coinciden con los datos reales de la empresa pero mantienen la relación en el orden de magnitud.

Naturalmente, uno de los aspectos que resultan de interés para medir el desempeño de nuestro enfoque de solución es la cantidad de trabajos. Por esta razón, elaboramos cinco conjuntos básicos de trabajos con sus fechas de entrega. Tabla 25 muestra la cantidad de trabajos totales (n) y de cada tipo (cierre de vino, monoblocco, T-Wood y bicolor) para cada conjunto de trabajos. En promedio, el 16 % de los trabajos tienen fecha de entrega durante la primera semana, el 31 % durante la segunda semana, el 27 % durante la tercera y el 26 % restante durante la cuarta semana.

Otros aspectos del problema influyen en el desempeño de nuestro enfoque de solución son los siguientes:

1. cantidad de máquinas
2. cantidad de trabajos sin prioridad máxima

[13] `https://osf.io/qb6ds/?view_only=7be46ac724734df89cecbc1d69da6845`

Conjunto	n	Cierres de vino	T-Wood	Monoblocco	Bicolor
1	26	13	2	9	2
2	35	13	1	1	20
3	65	38	9	9	9
4	95	59	19	13	4
5	100	57	16	20	7

Tabla 25: Conjunto de trabajos a partir de los cuales se construyen las instancias

3. duración de cada intervalo del horario administrativo en la Etapa 1
4. cantidad de máquinas dedicadas exclusivamente a coinyectar

Todas las instancias generadas tienen algunas características en común: la cantidad de juegos de cada tipo de molde, el horario administrativo (8 a 17 hs), los modelos de las máquinas (que determinan la cantidad de grupos inyectores y su velocidad de producción), la compatibilidad molde-máquina y las máquinas con grupos inyectores no solidarios.

La primera característica que nos interesó estudiar fue cómo influye la cantidad de máquinas en el tiempo de cómputo y en la calidad del resultado. Las instancias generadas a partir de los conjuntos de trabajos fueron denominadas `instance_i_Mx` donde `i` se corresponde al número del conjunto de trabajos (Tabla 25) y `x` a la cantidad de máquinas. Por ejemplo `instance_2_M05` es la instancia con los trabajos del conjunto 2 resuelta con 5 máquinas. Para cada conjunto de trabajos, generamos instancias con 5, 10 y 15 máquinas. Utilizamos las instancias `instance_i_Px` para analizar cómo influye la variabilidad en la prioridad de los trabajos. En este caso `x` denota el porcentaje de trabajos que son elegidos al azar y asignados una prioridad aleatoria en el intervalo $(0{,}05\,, 1)$, donde 1 es la máxima prioridad y le corresponde a los trabajos no elegidos. Los valores de x son 0, 10, 25, 50 y 100.

Por otra parte, las instancias `instance_i_Hx` se corresponden al estudio del impacto de una discretización más fina del horario administrativo. En este contexto, `x` es la duración de cada intervalo del horario administrativo en minutos, y toma los valores 180 y 270. Finalmente, las instancias `instance_i_Cx` permiten examinar si resulta beneficioso dedicar máquinas exclusivamente a coinyectar. En este marco, `x` indica la cantidad de máquinas dedicadas exclusivamente a coinyectar y toma los valores 0, 3 y 5. El valor 0 significa que cualquier máquina puede coinyectar y que sólo se considera la compatibilidad molde-máquina del molde a coinyectar. En las instancias de cada uno de los cuatro aspectos de nuestro estudio, los valores de los otros tres se mantienen fijos. La Tabla 26 detalla dichos valores. Notar que ciertas instancias con distinta nomenclatura son esencialmente las mismas. Por ejemplo, `instance_1_M10` es igual que `instance_1_H270`. A pesar de la redundancia, esta notación facilitará la interpretación de los resultados.

5.2. Resultados y discusión

A continuación discutiremos los resultados obtenidos de aplicar nuestro enfoque a las instancias que generamos. La Etapa 2 es resuelta en un promedio de 0,11 segundos, siendo 0,19 segundos el máximo, correspondiente a `instance_4_C0`. Por su parte, la Etapa 3 es

Instancias	Aspecto a analizar	Aspecto 1	Aspecto 2	Aspecto 3	Aspecto 4
`instance_i_Mx`	1	`x`	0 %	270 mins.	3 si `x`> 5, 1 si `x`= 5
`instance_i_Px`	2	10	`x` %	270 mins.	3
`instance_i_Hx`	3	10	0 %	`x` mins.	3
`instance_i_Cx`	4	15	0 %	270 mins.	`x`

Tabla 26: Valores de cada uno de los aspectos de estudio según la instancia. Para el Aspecto 1: `x` $\in \{5, 10, 15\}$. Para el Aspecto 2, `x`$\in \{0, 10, 25, 50, 100\}$. Para el Aspecto 3, `x`$\in \{180, 270\}$. Para el Aspecto 4, `x`$\in \{0, 3, 5\}$.

resuelta en un promedio de 1,34 segundos, siendo 6,58 segundos el máximo, alcanzado por `instance_2_C5`. Como la resolución de ambas etapas no consumen una cantidad significativa de tiempo, nos enfocaremos en analizar el desempeño del modelo de la Etapa 1. La Tabla 27 y la Tabla 28 muestran el gap de optimalidad, el mejor valor obtenido de la función objetivo y el tiempo consumido en cada objetivo de la minimización lexicográfica de la Etapa 1 para cada instancia.

A continuación compartimos nuestras observaciones principales a partir de los resultados de la experimentación con respecto al impacto de cada una de las cuatro características en el desempeño del modelo y en los párrafos siguientes profundizaremos en cada una de ellas:

- *Número de máquinas*: mayor cantidad de máquinas aumentan el tamaño y la complejidad del modelo. Cuando la demanda excede la capacidad de producción, deben agregarse más días al horizonte de planificación para permitir factibilidad. Esto afecta notoriamente el desempeño del modelo, incluso para instancias con menos máquinas.
- *Cantidad de trabajos sin máxima prioridad*: la variabilidad en la prioridad de los trabajos no impactó significativamente en la minimización de la demora ponderada total. Sin embargo, en todos los casos, las instancias con la variabilidad máxima (100 %) tuvieron un menor tiempo de solución con la primera función objetivo que los instancias con variabilidad mínima (0 %).
- *Duración de los intervalos del horario administrativo*: intervalos más cortos aumentan la complejidad del problema. Es importante notar que las instancias con intervalos de 180 minutos se desempeñaron mejor en la minimización de sobreproducción.
- *Cantidad de máquinas dedicadas a co-inyección*: la falta de máquinas dedicadas exclusivamente a co-inyectar dificulta el hallazgo de soluciones eficientes.

Con respecto a la primera característica que nos interesa estudiar, podemos observar que la cantidad de máquinas tiene una influencia importante en el desempeño del modelo. Naturalmente, a mayor cantidad de máquinas, mayor es el tamaño del problema y por ende es más difícil de resolver. Por ejemplo, `instance_1_M15` requirió cinco veces más de tiempo para ser resuelta que `instance_1_M10`. Efectivamente, un incremento de 10 a 15 máquinas significó un aumento de 36,96 % en la cantidad de variables y de 47,17 % en la cantidad de restricciones. También resultan interesantes las instancias con 5 máquinas. Salvo para `instance_1_M05`, la

capacidad de producción no podía cubrir la demanda en el horizonte de planificación, por lo que fue necesario añadir hasta ocho días para que el modelo resultara factible. Esto impactó directamente en el tamaño del problema, lo cual se refleja en el alto gap de optimalidad para la primera función objetivo con respecto a las instacias con 10 máquinas. Este fenómeno parece indicar que en los casos donde la demanda excede la capacidad de producción de la planta, sería más conveniente acortar el horizonte de planificación para obtener soluciones de mejor calidad en menos tiempo.

Con respecto al porcentaje de trabajos sin prioridad máxima, no pareció afectar significativamente el desempeño de la minimización de la demora ponderada total para los conjuntos de trabajos 1, 2 y 5. En cambio, para los conjuntos 3 y 4, observamos variabilidad en los tiempos de solución para la primera función objetivo. Es interesante notar que para el conjunto de trabajos 3, `instance_3_P025` fue la que menos tiempo llevó ser resuelta por un amplio margen. A través de un análisis de las prioridades de los trabajos, observamos que la mayoría de los trabajos con prioridad menor que 1 corresponden a productos monoblocco y tapones de vino. En `instance_3_P050`, se reduce la prioridad de varios de los trabajos correspondientes a productos T-Wood y, aunque el valor óptimo de demora ponderada total es igual al de `instance_3_P025`, el tiempo de solución fue mucho mayor. En un análisis futuro, sería interesante estudiar como el tipo de producto de los trabajos de menor prioridad impacta en el desempeño del modelo. Por último, notar que para las instancias `P100` se resolvió más rápidamente el modelo para la primera función objetivo que para las instancias `P000`. Esto podría deberse a que la alta variabilidad en la prioridad de los trabajos contribuye a reducir la simetría del modelo.

Disminuir la duración de los intervalos de tiempo en los que se particiona el horario administrativo también aumentó significativamente el tamaño del problema. Al comparar las dos discretizaciones (180 y 270 minutos), debemos tener en cuenta que los valores de demora ponderada total y de tiempo de inactividad de las máquinas están expresados en cantidad de tiempos de intervalos. Por ejemplo, en `instance_3_H270` hay una demora total de cuatro horas y media (un intervalo temporal) mientras que en `instance_3_H180` la demora total es de seis horas (dos intervalos temporales). Lo que podemos afirmar es que encontrar la solución óptima para cada paso de la minimización lexicográfica fue más costoso para la discretización más fina debido al tamaño del problema. Sin embargo, es interesante que las instancias `H180` hallaron mejores soluciones para la sobreproducción, incluso si el gap de optimalidad es mayor que el de la instancia correspondiente con discretización de 270 minutos, como ocurre con `instance_2_H180`. Comparando los valores de la sobreproducción no deseable entre las dos discretizaciones, la más fina permitió hallar soluciones hasta 88 % mejores, e incluso en `instance_1_H180`, el modelo halló una solución sin sobreproducción no deseable. Por lo tanto, el usuario podría decidir qué partición utilizar de acuerdo al tiempo disponible: una discretización más fina podría permitir menor demora total y menor sobreproducción no admisible pero el tiempo que consume es considerablemente mayor al de una discretización más gruesa.

Finalmente, en relación con la cuarta característica, observamos que cuando ninguna máquina está asignada exclusivamente a coinyectar, el modelo tiene mayor dificultad para hallar soluciones eficientes. De hecho, para `instance_2_C0` el solver no pudo hallar una solución factible en las tres horas destinadas para minimizar la primera función objetivo. En tales

casos, el espacio de soluciones es mucho mayor debido a la relajación de las restricciones de compatibilidad entre moldes y máquinas. Cuando hay máquinas destinadas exclusivamente a coinyectar, la compatibilidad de los moldes ficticios es la intersección entre dichas máquinas y la compatibilidad del molde original. Esto indica entonces que asignar máquinas a coinyectar es beneficioso para el desempeño del modelo. Esta conclusión también es apoyada por el resultado de las instancias con 5 máquinas dedicadas exclusivamente a esa tarea: en general, no sólo se reduce el tiempo de minimización de la demora total sino también los valores de sobreproducción no admisible son menores.

Analizando el desempeño de cada etapa de la minimizción lexicográfica, observamos que, en general, es posible obtener soluciones óptimas en términos de demora ponderada total en tiempo aceptable, excepto en casos donde la demanda excede considerablemente la capacidad de producción de la planta y cuando no hay máquinas dedicadas a la coinyección. En esos casos, una posible solución es reducir el tamaño del problema considerando un horizonte de planificación más corto. Con respecto a la minimización de la sobreproducción no deseable, notamos que cuando el valor óptimo no es alcanzado, el gap de optimalidad está generalmente por debajo del 1 %. Por ejemplo, en `instance_5_M10`, la solución con 0,1 % de gap de optimalidad fue encontrada en aproximadamente 14 minutos mientras que en el caso de `instance_3_M10`, la solución con 0,92 % de gap de optimalidad fue encontrada luego de 21 minutos. Esto parece indicar que para obtener soluciones eficientes en menos tiempo sería aconsejable aceptar soluciones con un gap de optimalidad de menos del 1 % en sobreproducción no deseable. Minimizar el tiempo de inactividad de las máquinas fue particularmente desafiante para el conjunto de trabajos 2, que es el conjunto con la mayor cantidad de productos bicolor. El estudio de la existencia de una relación entre la cantidad de productos bicolor y el desempeño del modelo para la tercera función objetivo podría ser parte de trabajo futuro.

También comparamos el resultado del modelo con una planificación horaria llevada a cabo manualmente por el personal de la planta, en una instancia real con un horizonte de planificación de tres semanas. La planificación manual llevó aproximadamente una hora mientras que nuestro enfoque de solución, con una discretización de 270 minutos para el modelo de la Etapa 1, logró obtener una solución eficiente en tres minutos y medio. En la Tabla 29 mostramos la comparación de los tres objetivos de la Etapa 1. Nuestro enfoque de solución permitió que todas las órdenes fueran entregadas a tiempo, que la sobreproducción no deseable se redujera en un 84,5 % respecto a la planificación manual y que el tiempo de inactividad de las máquinas se mantuviera nulo. Como ejemplifica este caso, el uso de la herramienta que desarrollamos podría permitirle al personal probar distintos escenarios en el mismo tiempo que le tomaría elaborar una sola planificación con los métodos manuales.

Elaboramos una interfaz de usuario para que el uso de la herramienta sea más sencillo. En la misma, además de cargar el archivo de Excel con todos los datos de entrada, el usario puede determinar el horizonte de planificación y la discretización del horario administrativo.

Instancia	Demora total			Sobreproducción no admisible			Tiempo de inactividad			Tiempo total
	Gap	Valor Obj.	Tiempo	Gap	Valor Obj.	Tiempo	Gap	Valor Obj.	Tiempo	
instance_1_M05	0,00 %	0,00	0:00:10	0,00 %	0,62	0:00:05	0,00 %	0,00	0:00:06	0:00:21
instance_1_M10	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:10	0,00 %	0,00	0:00:11	0:00:37
instance_1_M15	0,00 %	0,00	0:00:42	0,00 %	4,50	0:00:21	0,00 %	8,00	0:02:06	0:03:09
instance_2_M05[a]	94,46 %	52,00	3:00:00	0,83 %	5,50	1:30:00	0,00 %	0,00	0:03:15	4:33:15
instance_2_M10	0,00 %	0,00	0:00:25	0,00 %	5,75	0:33:04	100,00 %	3,00	5:26:30	5:59:59
instance_2_M15	0,00 %	0,00	0:02:25	0,22 %	5,06	2:58:47	100,00 %	1,00	2:58:47	5:59:59
instance_3_M05[a]	91,53 %	796,00	3:00:00	3,32 %	3,08	1:30:00	0,00 %	0,00	0:02:10	4:32:10
instance_3_M10	0,00 %	1,00	0:53:07	0,92 %	2,77	2:33:26	0,00 %	0,00	0:07:12	3:33:45
instance_3_M15	0,00 %	1,00	0:43:25	0,00 %	3,26	2:29:00	0,00 %	0,00	0:15:18	3:27:43
instance_4_M05[b]	98,92 %	780,00	3:00:00	2,36 %	29,37	1:30:00	0,00 %	0,00	0:01:24	4:31:24
instance_4_M10	0,00 %	0,00	0:01:33	0,00 %	44,71	0:22:33	0,00 %	0,00	0:05:32	0:29:38
instance_4_M15	0,00 %	0,00	0:09:14	0,58 %	35,48	2:55:22	0,00 %	0,00	0:47:19	3:51:55
instance_5_M05[c]	98,58 %	1047,00	3:00:00	0,00 %	2,43	0:26:45	0,00 %	0,00	0:03:34	3:30:19
instance_5_M10	0,00 %	0,00	0:01:35	0,10 %	2,85	2:59:12	0,00 %	0,00	0:06:19	3:07:06
instance_5_M15	0,00 %	0,00	0:12:01	0,06 %	5,17	2:53:59	0,00 %	0,00	0:12:40	3:18:40
instance_1_P000	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:10	0,00 %	0,00	0:00:11	0:00:37
instance_1_P025	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:10	0,00 %	0,00	0:00:10	0:00:36
instance_1_P050	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:09	0,00 %	0,00	0:00:10	0:00:35
instance_1_P100	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:10	0,00 %	0,00	0:00:10	0:00:36
instance_2_P000	0,00 %	0,00	0:00:25	0,00 %	5,75	0:33:04	100,00 %	3,00	5:26:30	5:59:59
instance_2_P025	0,00 %	0,00	0:00:25	0,00 %	5,75	0:32:37	100,00 %	3,00	5:26:56	5:59:58
instance_2_P050	0,00 %	0,00	0:00:25	0,00 %	5,75	0:32:38	100,00 %	2,00	5:26:55	5:59:58
instance_2_P100	0,00 %	0,00	0:00:24	0,00 %	5,75	0:33:02	100,00 %	2,00	5:26:33	5:59:59
instance_3_P000	0,00 %	1,00	0:53:07	0,92 %	2,77	2:33:26	0,00 %	0,00	0:07:12	3:33:45
instance_3_P025	0,00 %	0,82	0:12:36	0,00 %	2,77	1:13:55	0,00 %	0,00	0:03:52	1:30:23
instance_3_P050	0,00 %	0,82	1:53:51	0,00 %	2,77	1:08:42	0,00 %	0,00	0:07:17	3:09:50
instance_3_P100	0,00 %	0,63	0:03:02	0,92 %	2,77	2:58:28	0,00 %	0,00	0:03:21	3:04:51
instance_4_P000	0,00 %	0,00	0:01:33	0,00 %	44,71	0:22:33	0,00 %	0,00	0:05:32	0:29:38
instance_4_P025	0,00 %	0,00	0:03:12	0,00 %	44,71	0:35:27	0,00 %	0,00	0:03:07	0:41:46
instance_4_P050	0,00 %	0,00	0:02:04	0,00 %	44,71	0:26:26	0,00 %	0,00	0:08:29	0:36:59
instance_4_P100	0,00 %	0,00	0:01:51	0,00 %	44,71	0:14:32	0,00 %	0,00	0:38:03	0:54:26
instance_5_P000	0,00 %	0,00	0:01:35	0,10 %	2,85	2:59:12	0,00 %	0,00	0:06:19	3:07:06
instance_5_P025	0,00 %	0,00	0:02:01	0,04 %	2,85	2:58:59	0,00 %	0,00	0:02:20	3:03:20
instance_5_P050	0,00 %	0,00	0:01:33	0,10 %	2,85	2:59:13	0,00 %	0,00	0:06:13	3:06:59
instance_5_P100	0,00 %	0,00	0:01:33	0,04 %	2,85	2:59:13	0,00 %	0,00	0:06:21	3:07:07

[a] El horizonte de planificación fue extendido 8 días para que el modelo resulte factible

[b] El horizonte de planificación fue extendido 6 días para que el modelo resulte factible

[c] El horizonte de planificación fue extendido 7 días para que el modelo resulte factible

Tabla 27: Resultados de la aplicación de nuestro enfoque de solución a las instancias de prueba.

Instancia	Demora total			Sobreproducción no admisible			Tiempo de inactividad			Tiempo total
	Gap	Valor Obj.	Tiempo	Gap	Valor Obj.	Tiempo	Gap	Valor Obj.	Tiempo	
instance_1_H180	0,00 %	0,00	0:00:37	0,00 %	0,00	0:00:19	0,00 %	0,00	0:00:36	0:01:32
instance_1_H270	0,00 %	0,00	0:00:16	0,00 %	0,56	0:00:10	0,00 %	0,00	0:00:11	0:00:37
instance_2_H180	0,00 %	0,00	0:01:25	9,21 %	0,69	2:59:17	0,00 %	3,00	2:35:38	5:36:20
instance_2_H270	0,00 %	0,00	0:00:25	0,00 %	5,75	0:33:04	100,00 %	3,00	5:26:30	5:59:59
instance_3_H180	0,00 %	2,00	2:04:02	65,52 %	1,81	1:57:59	0,00 %	0,00	0:32:02	4:34:03
instance_3_H270	0,00 %	1,00	0:53:07	0,92 %	2,77	2:33:26	0,00 %	0,00	0:07:12	3:33:45
instance_4_H180	0,00 %	0,00	0:31:04	0,15 %	9,38	2:44:27	100,00 %	12,00	2:44:27	5:59:58
instance_4_H270	0,00 %	0,00	0:01:33	0,00 %	44,71	0:22:33	0,00 %	0,00	0:05:32	0:29:38
instance_5_H180	0,00 %	0,00	0:08:10	0,00 %	0,33	2:12:38	100,00 %	17,00	3:39:11	5:59:59
instance_5_H270	0,00 %	0,00	0:01:35	0,10 %	2,85	2:59:12	0,00 %	0,00	0:06:19	3:07:06
instance_1_C0	0,00 %	0,00	0:01:03	0,00 %	3,94	0:00:40	0,00 %	0,00	0:00:52	0:02:35
instance_1_C3	0,00 %	0,00	0:00:42	0,00 %	4,50	0:00:21	0,00 %	8,00	0:02:06	0:03:09
instance_1_C5	0,00 %	0,00	0:00:20	0,00 %	0,56	0:00:15	0,00 %	0,00	0:01:32	0:02:07
instance_2_C0[a]	-	-	-	-	-	-	-	-	-	-
instance_2_C3	0,00 %	0,00	0:02:25	0,22 %	5,06	2:58:47	100,00 %	1,00	2:58:47	5:59:59
instance_2_C5	0,00 %	0,00	0:01:42	0,22 %	5,06	2:59:11	0,00 %	0,00	0:16:53	3:17:46
instance_3_C0	99,91 %	1085,00	3:00:00	0,00 %	2,39	0:45:03	0,00 %	0,00	0:27:12	4:12:15
instance_3_C3	0,00 %	1,00	0:43:25	0,00 %	3,26	2:29:00	0,00 %	0,00	0:15:18	3:27:43
instance_3_C5	0,00 %	5,00	1:36:02	0,78 %	3,26	2:11:58	100,00 %	3,00	2:11:58	5:59:58
instance_4_C0	100,00 %	275,00	3:00:00	9,63 %	20,20	1:30:00	0,00 %	0,00	0:36:20	5:06:20
instance_4_C3	0,00 %	0,00	0:09:14	0,58 %	35,48	2:55:22	0,00 %	0,00	0:47:19	3:51:55
instance_4_C5	0,00 %	0,00	0:20:37	0,00 %	26,34	2:20:36	100,00 %	36,00	3:18:46	5:59:59
instance_5_C0	100,00 %	1460,00	3:00:00	1,14 %	5,23	1:30:00	0,00 %	0,00	0:31:55	5:01:55
instance_5_C3	0,00 %	0,00	0:12:01	0,06 %	5,17	2:53:59	0,00 %	0,00	0:12:40	3:18:40
instance_5_C5	0,00 %	0,00	0:04:28	0,00 %	4,87	0:49:09	0,00 %	0,00	0:18:25	1:12:02

[a] El solver no encontró una solución factible en las 3 horas designadas a la minimización de la demora total.

Tabla 28: Resultados de la aplicación de nuestro enfoque de solución a las instancias de prueba.

	Demora	Sobreproducción	Inactividad
Manual	3	3,511	0
Nuestro enfoque	0	0,541	0
Diferencia relativa	100 %	84,5 %	0 %

Tabla 29: Valores de las tres funciones objetivo de la Etapa 1 para la planificación manual y la propuesta por nuestro enfoque de solución.

6. Conclusiones

Desarrollamos una estrategia de programación matemática de tres etapas basada en MILP con el objetivo de resolver una aplicación al mundo real del UPMP: la planificación de producción para una planta de fabricación de cierres de botellas. Las características del proceso de manufactura hicieron que este problema tuviera una naturaleza más compleja que la encontrada en estudios llevados a cabo a nivel teórico. Por esta razón lo dividimos en tres subproblemas que trataban cada aspecto de la planificación horaria: asignación de moldes a máquinas, ajuste de los tiempos de parada debido a los cambios de moldes y planificación del coloreo de cada tipo de molde.

El primer subproblema resultó ser el más desafiante de resolver debido a que determina la estructura de la planificación horaria. Mediante experimentos computacionales hemos mostrado que el modelo propuesto es capaz de encontrar soluciones eficientes en términos de demora ponderada total, sobreproducción no admisible y tiempo de inactividad de las máquinas en instancias de hasta 100 trabajos, en un lapso de tiempo razonable para la aplicación. También comparamos el desempeño de nuestro enfoque de solución con la planificación manual elaborada por el personal de la planta para una instancia real. La herramienta que desarrollamos permitió encontrar una planificación que supera a la manual, consumiendo mucho menos tiempo. Esto le permitiría al personal de la fábrica, por ejemplo, invertir más tiempo en probar distintos escenarios. Además, les resultaría más sencillo elaborar una nueva planificación en caso de una situación inesperada, como el ingreso de un nuevo trabajo que debe ser terminado con urgencia o el desperfecto de una máquina. Creemos que la identificación de desigualdades válidas para utilizar como cortes en el modelo de minimización lexicográfica para mejorar su desempeño es una parte importante del trabajo futuro. Paralelamente, se podría investigar la aplicación de heurísticas para generar soluciones iniciales para la Etapa 1 o para llevar a cabo la optimización de la tercera función objetivo, que también parece ser desafiante en algunas instancias.

También analizamos cómo diferentes características de los datos de entrada influencian el desempeño del modelo de la primera etapa. Como es esperable, los resultados mostraron que a mayor cantidad de trabajos y de máquinas, más difícil es la obtención de una solución eficiente. Sin embargo, también observamos que, en los casos donde hay una cantidad reducida de máquinas y la demanda de los trabajos supera la capacidad de producción, la extensión del horizonte de planificación para permitir la factibilidad del modelo de la primera etapa también complejiza cosiderablemente el problema. Asimismo, estudiamos cómo la variabilidad en la prioridad de los trabajos afecta el desempeño de la minimización de la demora ponderada total y el *tradeoff* entre tiempo de cómputo y menor sobreproducción no deseable de una discretización más fina del horario administrativo. Observamos que el modelo se desempeña mejor cuando hay máquinas dedicadas exclusivamente a coinyectar, una de las tareas involucradas en la elaboración de productos T-Wood y bicolor. La posibilidad de examinar estos aspectos es otro atributo útil de la herramienta que proponemos, pues permite determinar cómo la adquisición nuevos insumos impactaría la producción. En este sentido, otra vertiente de trabajo futuro podría ser un análisis de sensibilidad más exhaustivo para evaluar la influencia de otras características de los datos de entrada en el desempeño del modelo.

En este trabajo no sólo hemos analizado y propuesto una forma de resolver una instancia

real del UPMP, sino que además hemos generado y puesto a disposición un conjunto de instancias que representan diferentes escenarios de una aplicación real que pueden servir como referencia para futuros estudios sobre UPMPs similares o más generales.

Finalmente, queremos mencionar que la empresa Tapi South America ha expresado su satisfacción con el resultado de nuestros experimentos computacionales y su interés en comenzar a usar la herramienta para la planificación de la producción en la planta.

Referencias

[1] M. Afzalirad y J. Rezaeian. A realistic variant of bi-objective unrelated parallel machine scheduling problem: Nsga-ii and moaco approaches. *Applied Soft Computing*, 50:109–123, 2017.

[2] C. Ahabchane, A. Langevin, y M. Trépanier. The mixed capacitated general routing problem with time-dependent demands. *Networks*, 76:467–484, 2020.

[3] K. Andreev y H. Racke. Balanced Graph Partitioning. *Theory of Computing Systems*, 39:929–939, 2006.

[4] J. S. Arora. Multi-objective optimum design concepts and methods. In J. S. Arora, editor, *Introduction to Optimum Design*, pages 657–679. Academic Press, Boston, 3 edition, 2012.

[5] C.A. Arribas, C.A. Blazquez, y A. Lamas. Urban solid waste collection system using mathematical modelling and tools of geographic information systems. *Waste Management & Research*, 28(4):355–363, 2010.

[6] J-F. Audy, N. Lehoux, S. D'Amours, y M. Rönnqvist. A framework for an efficient implementation of logistics collaborations. *International Transactions in Operational Research*, 19(5):633–657, 2012.

[7] O. Avalos-Rosales, F. Angel-Bello, y A. Alvarez. Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, 76(2):1705–1718, 2015.

[8] J. Bank y F. Werner. Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. *Mathematical and Computer Modelling*, 33(4):363–383, 2001.

[9] E. Beltrami y L. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4:65–94, 1974.

[10] F. Bertero, M. Cerdeiro, G. A. Durán, y N. A. Faillace Mullen. Developing optimization tools for municipal solid waste collection in the argentine city of berazategui. *INFORMS Journal on Applied Analytics*, 53(6):451–464, 2023.

[11] M. Bianchetti, G. Durán, I. Koch, y J. Marenco. Algoritmos de zonificación para el problema de la recolección de residuos urbanos: El caso de estudio de una ciudad argentina. *Revista Ingeniería de Sistemas*, 31:81–110, 2017.

[12] J. Bogner, J.M. Abdelrafie Ahmed, C. Diaz, A. Faaij, Q. Gao, S. Hashimoto, K. Mareckova, R. Pipatti, y T. Zhang. Waste management. In *Climate Change 2007: Mitigation. Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. [B. Metz and O.R. Davidson and P.R. Bosch and R. Dave and L.A. Meyer (eds)], Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA., 2007.

[13] F. Bonomo, G. Durán, F. Larumbe, y J. Marenco. A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. *Waste management & Research*, 30(3):311–24, 2011.

[14] B. D. Bradley, T. Jung, A. Tandon-Verma, B. Khoury, T. C. Y. Chan, y Y. Cheng. Operations research in global health: a scoping review with a focus on the themes of health equity and impact. *Health Research Policy and Systems*, 15, 2017.

[15] G. Braier, G. Durán, J. Marenco, y F. Wesner. An integer programming approach to a real-world recyclable waste collection problem in Argentina. *Waste Management & Research*, 35(5):525–533, 2017.

[16] P. Brucker. *Scheduling Algorithms*. Springer, New York, 5 edition, 2007.

[17] O. Bräysy, E. Martínez, Y. Nagata, y D. Soler. The mixed capacitated general routing problem with turn penalties. *Expert Systems with Applications*, 38(10):12954–12966, 2011.

[18] V. Cacchiani, C. Malandri, L. Mantecchini, y F. Paganelli. A study on the optimal aircraft location for human organ transportation activities. *Transportation Research Procedia*, 30:314–323, 2018.

[19] T. Çakar, R. Köker, y Y. Sarı. Parallel Robot Scheduling to Minimize Mean Tardiness with Unequal Release Date and Precedence Constraints Using a Hybrid Intelligent System. *International Journal of Advanced Robotic Systems*, 9(6):252, 2012.

[20] G. Campuzano, C. Obreque, y M. M. Aguayo. Accelerating the miller–tucker–zemlin model for the asymmetric traveling salesman problem. *Expert Systems with Applications*, 148:113229, 2020.

[21] T. Capelle, C. E. Cortés, M. Gendreau, P. A. Rey, y L. M. Rousseau. A column generation approach for location-routing problems with pickup and delivery. *European Journal of Operational Research*, 272(1):121–131, 2019.

[22] A. Cataldo, JC. Ferrer, J. Miranda, P. A. Rey, y A. Sauré. An integer programming approach to curriculum-based examination timetabling. *Ann Oper Res*, 258:369–393, 2017.

[23] N. Chang y H.Y. Lu. A new approach for long term planning of solid waste management systems using fuzzy global criterion. *Journal of Environmental Science and Health . Part A: Environmental Science and Engineering and Toxicology*, 32(4):1025–1047, 1997.

[24] B. Chen, C. N. Potts, y G. J. Woeginger. *A Review of Machine Scheduling: Complexity, Algorithms and Approximability*, pages 1493–1641. Springer US, 1998.

[25] T.C.E. Cheng y C.C.S. Sin. A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research*, 47(3):271–292, 1990.

[26] J. Christiaens y G. Vanden Berghe. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54(2):417–433, 2020.

[27] M. Constantino, L. Gouveia, M. C. Mourão, y A. C. Nunes. The mixed capacitated arc routing problem with non-overlapping routes. *European Journal of Operational Research*, 244(2):445–456, 2015.

[28] Coordinación Ecológica Área Metropolitana Sociedad del Estado (CEAMSE. Ingreso de residuos a ceamse. `https://www.ceamse.gov.ar/wp-content/uploads/2025/05/planilla-estadistica.xlsx`, 2025. Accedido el 25 de julio de 2025.

[29] A. Corberán, G. Mejía, y J. M. Sanchis. New results on the mixed general routing problem. *Operations Research*, 53(2):363–376, 2005.

[30] Á. Corberán, A. Romero, y J. M. Sanchis. The mixed general routing polyhedron. *Mathematical Programming*, 96:103–137, 01 2003.

[31] M. Cortinhal, M. Mourão, y A. Nunes. Local search heuristics for sectoring routing in a household waste collection context. *European Journal of Operational Research*, 255, 04 2016.

[32] C. D'Aloisio, G. Durán, y N. A. Faillace Mullen. A solution approach to districting modification and route optimization in waste collection. In *Proceedings of the XIII Latin American Algorithms, Graphs, and Optimization Symposium*, 2025. (*To appear*).

[33] S. D'Amours, M. Rönnqvist, y A. Weintraub. Using operational research for supply chain planning in the forest products industry. *INFOR: Information Systems and Operational Research*, 46(4):265–281, 2008.

[34] D. Delle Donne, V. Di Tomaso, y G. Durán. Optimizing Leaf Sweeping and Collection in the Argentine city of Trenque Lauquen. *Waste Management & Research*, 39(2):209–220, 2021.

[35] M. Desrochers y G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

[36] L. Di Puglia Pugliese y F. Guerriero. Dynamic programming approaches to solve the shortest path problem with forbidden paths. *Optimization Methods and Software*, 28(2):221–255, 2013.

[37] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[38] J. Du y J. Y.-T. Leung. Minimizing Total Tardiness on one machine is NP-Hard. *Mathematics of Operations Research*, 15(3):483–495, 1990.

[39] G. Durán, R. Epstein, C. Martinez, y G. A. Zamorano. Quantitative methods for a new configuration of territorial units in a chilean government agency tender process. *Interfaces*, 41(3):263–277, 2011.

[40] G. Durán, Guajardo M., J. Miranda, D. Sauré, S. Souyris, A. Weintraub, y R. Wolf. Scheduling the chilean soccer league by integer programming. *Interfaces*, 6(37):539–552, 2007.

[41] G.A. Durán, M. Durán, N.A. Faillace Mullen, y J. Velásquez. An application of mathematical programming to a real case of the unrelated parallel machine problem. *Ann Oper Res*, 2024. doi: 10.1007/s10479-024-05938-1.

[42] K. Easton, G. Nemhauser, y M. Trick. The traveling tournament problem description and benchmarks. In *Principles and Practice of Constraint Programming — CP 2001*, pages 580–584. Springer Berlin Heidelberg, 2001.

[43] M. Ehrgott y X. Gandibleux. *Multiobjective Combinatorial Optimization — Theory, Methodology, and Applications*, pages 369–444. Springer US, Boston, MA, 2002.

[44] D. Eisenstein y A. Iyer. Garbage Collection in Chicago: A Dynamic Scheduling Model. *Management Science*, 43:922–933, 1997.

[45] R. Epstein, R. Morales, J. Serón, y A. Weintraub. Use of or systems in the chilean forest industries. *Interfaces*, 29(1):7–29, 1999.

[46] L. Fanjul-Peyro. Models and an exact method for the Unrelated Parallel Machine scheduling problem with setups and resources. *Expert Systems with Applications: X*, 5:100022, 2020.

[47] L. Fanjul-Peyro, F. Perea, y R. Ruiz. Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research*, 260(2):482–493, 2017.

[48] L. Fanjul-Peyro, R. Ruiz, y F. Perea. Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. *Computers & Operations Research*, 101:173–182, 2019.

[49] C.A. Floudas y X. Lin. Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Ann Oper Res*, 139:131–162, 2005.

[50] J. M. Framinan, R. Leisten, y R. Ruiz García. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*. Springer, London, 2014.

[51] M. L. Fredman y R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.

[52] M. Frisk, M. Göthe-Lundgren, K. Jörnsten, y M. Rönnqvist. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2):448–458, 2010.

[53] S. I. Gass y A. A. Assad. *An Annotated Timeline of Operations Research: An Informal History*. International Series in Operations Research & Management Science. Springer New York, NY, 2005.

[54] R. Gedik, D. Kalathia, G. Egilmez, y E. Kirac. A constraint programming approach for solving unrelated parallel machine scheduling problem. *Computers & Industrial Engineering*, 121:139–149, 2018.

[55] C. Gicquel, M. Minoux, y Y. Dallery. On the discrete lot-sizing and scheduling problem with sequence-dependent changeover times. *Operations Research Letters*, 37(1):32–36, 2009.

[56] A. Gliesch, M. Ritt, A. H. S. Cruz, y M. C. O. Moreira. A heuristic algorithm for districting problems with similarity constraints. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.

[57] A. Gliesch, M. Ritt, A. H. S. Cruz, y M. C. O. Moreira. A hybrid heuristic for districting problems with routing criteria. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–9, 2020.

[58] R.L. Graham, E.L. Lawler, J.K. Lenstra, y A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In P.L. Hammer, E.L. Johnson, y B.H. Korte, editors, *Discrete Optimization II*, volume 5 of *Annals of Discrete Mathematics*, pages 287–326. Elsevier, [Online], 1979.

[59] M. Guajardo y M. Rönnqvist. A review on cost allocation methods in collaborative transportation. *International Transactions in Operational Research*, 23(3):371–392, 2016.

[60] A. Guinet. Textile Production Systems: a Succession of Non-identical Parallel Processor Shops. *Journal of the Operational Research Society*, 42(8):655–671, 1991.

[61] UN Habitat. Solid waste management in the world's cities. *London: United Nations Human Settlements Programme*, 2010.

[62] Y. Y. Haimes, L. Lasdon, y D. Wismer. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297, 1971.

[63] S. Hanafi, A. Freville, y P. Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR: Information Systems and Operational Research*, 37(3):236–254, 1999.

[64] P. Hansen, N. Mladenović, J. Brimberg, y José A. M. Pérez. *Variable Neighborhood Search*, pages 57–97. Springer International Publishing, Cham, 2019.

[65] J. Herrmann, J-M Proth, y N. Sauer. Heuristics for unrelated machine scheduling with precedence constraints. *European Journal of Operational Research*, 102(3):528–537, 1997.

[66] C. Hess, A.G. Dragomir, K.F. Doerner, y D. Vigo. Waste collection routing: a survey on problems and methods. *Cent Eur J Oper Res*, 32:399–434, 2024.

[67] F. S. Hillier y G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill Education, 10th edition, 2015.

[68] D. Hoornweg y P. Bhada-Tata. *What a Waste : A Global Review of Solid Waste Management*. World Bank, Washington, DC., 2012.

[69] W. J. Hopp y M. L. Spearman. *Factory Physics*. Waveland Press Inc., Illinois, 2011.

[70] SM. Hosseini-Motlagh, M.R.G. Samani, y B. Karimi. Resilient and social health service network design to reduce the effect of covid-19 outbreak. *Ann Oper Res*, 328:903–975, 2023.

[71] G. H. Huang, B. W. Baetz, y G. G. Patry. Trash-Flow Allocation: Planning Under Uncertainty. *Interfaces*, 28(6):36–55, 1998.

[72] C. Hunt. Child waste pickers in india: the occupation and its health risks. *Environment and Urbanization*, 8(2):111–118, 1996.

[73] C.L. Hwang, A.S.M. Masud, S.R. Paidy, y K.S.P. Yoon. *Multiple Objective Decision Making, Methods and Applications: A State-of-the-art Survey*. Lecture notes in economics and mathematical systems. Springer-Verlag, 1979.

[74] Instituto Nacional de Estadística y Censos de la República Argentina (INDEC). Precenso de viviendas. `https://precensodeviviendas.indec.gob.ar/`, 2020. Accedido el 25 de febrero de 2025.

[75] Instituto Nacional de Estadística y Censos de la República Argentina (INDEC). Censo nacional de población, hogares y viviendas 2022. `https://censo.gob.ar/index.php/datos_definitivos/`, 2022. Accedido el 8 de enero de 2025.

[76] K. Jaklinović, M. Đurasević, y D. Jakobović. Designing dispatching rules with genetic programming for the unrelated machines environment with constraints. *Expert Systems with Applications*, 172:114548, 2021.

[77] X. Jin, H. Qin, Z. Zhang, M. Zhou, y J. Wang. Planning of garbage collection service: An arc-routing problem with time-dependent penalty cost. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2692–2705, 2021.

[78] J. Johnes. Operational research in education. *European Journal of Operational Research*, 243(3):683–696, 2015.

[79] J. Kalcsics. Districting problems. In G. Laporte, S. Nickel, y F. Saldanha da Gama, editors, *Location Science*, chapter 23, pages 595–622. Springer International Publishing, 2015.

[80] J. Kalcsics y R. Z. Ríos-Mercado. *Districting Problems*, pages 705–743. Springer International Publishing, Cham, 2019.

[81] G. Kendall, S. Knust, C. C. Ribeiro, y S. Urrutia. Scheduling in sports: An annotated bibliography. *Computers & Operations Research*, 37(1):1–19, 2010.

[82] B. Kim, S. Kim, y S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642, 2006. Part Special Issue: Recent Algorithmic Advances for Arc Routing Problems.

[83] S. Kirkpatrick, C. D. Gelatt, y M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[84] S. Knust. Scheduling non-professional table-tennis leagues. *European Journal of Operational Research*, 200(2):358–367, 2010.

[85] M. Kwan. Graphic programming using odd or even points. *Chinese Mathematics*, 1:273–277, 1962.

[86] D. Lei, Y. Yuan, y J. Cai. An improved artificial bee colony for multi-objective distributed unrelated parallel machine scheduling. *International Journal of Production Research*, 59(17):5259–5271, 2021.

[87] J.K. Lenstra, D. B. Shmoys, y E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259—271, 1990.

[88] C. Liu. A hybrid genetic algorithm to minimize total tardiness for unrelated parallel machine scheduling with precedence constraints. *Mathematical Problems in Engineering*, 2013:1–11, 2013.

[89] A. Lodi, P. Olivier, G. Pesant, y S. Sankaranarayanan. Fairness over time in dynamic resource allocation with an application in healthcare. *Math. Program.*, 203:285–318, 11 2022.

[90] J.W. Male y J.C. Liebman. Districting and routing for solid waste collection. *Journal of the Environmental Engineering Division*, 104(1):1–14, 1978.

[91] K. N. McKay y V.C.S. Wiers. *Practical production control : a survival guide for planners and schedulers*. J. Ross Publishing, Florida, 2004.

[92] Z. Michalewicz y D. B. Fogel. *How to Solve It: Modern Heuristics 2e*. Springer-Verlag, Berlin, Heidelberg, 2010.

[93] Ministerio de Ambiente del Gobierno de la Provincia de Buenos Aires. Guía para la implementación de la gestión integral e inclusiva de residuos. `https://www.ambiente.gba.gob.ar/imagenes/ruec/011_rsu_guia_giirsu_feb23.pdf`, 2023. Accedido el 8 de enero de 2025.

[94] Ministerio de Ambiente y Desarrollo Sostenible de la Nación. Informe del estado del ambiente 2020. `https://www.argentina.gob.ar/sites/default/files/iea_2020_digital.pdf`, 2021. Accedido el 25 de julio de 2025.

[95] N. Mladenović y P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[96] E. Mokotoff. Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research*, 56(2):193–242, 2001.

[97] E. Mokotoff y P. Chrétienne. A cutting plane algorithm for the unrelated parallel machine scheduling problem. *European Journal of Operational Research*, 141(3):515–525, 2002.

[98] M. C. Mourão, A. C. Nunes, y C. Prins. Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196(3):856–868, 2009.

[99] M.C. Mourão y M.T. Almeida. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research*, 121(2):420–434, 2000.

[100] R. Nauss. Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach. *INFORMS Journal on Computing*, 15:249–266, 2003.

[101] G. L. Nemhauser y M. A. Trick. Scheduling a major college basketball conference. *Operations Research*, 1(46):1–8, 1998.

[102] OpenStreetMap contributors. Datos obtenidos de . `https://www.openstreetmap.org`, Accedido en 2019 y en 2024. Página web operativa en 2025.

[103] M. L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, New York, 2012.

[104] J. C. S. N. Pinheiro, J. E. C. Arroyo, y L. B. Fialho. Scheduling unrelated parallel machines with family setups and resource constraints to minimize total tardiness. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 1409—1417, New York, NY, USA, 2020. Association for Computing Machinery.

[105] P. Pourhejazy, D. Zhang, Q. Zhu, F. Wei, y S. Song. Integrated e-waste transportation using capacitated general routing problem with time-window. *Transportation Research Part E: Logistics and Transportation Review*, 145:102169, 2021.

[106] A. Rais y A. Viana. Operations research in healthcare: a survey. *International Transactions in Operational Research*, 18(1):1–31, 2011.

[107] Registro Nacional de las Personas (ReNaPer). Población urbana en Argentina: Evolución y distribución espacial a partir de datos censales. `https://www.argentina.gob.ar/sites/default/files/poblacion_urbana_dnp.pptx_.pdf`, 2020. Accedido el 8 de enero de 2025.

[108] Registro Nacional de las Personas (ReNaPer). Análisis de crecimiento poblacional 2001-2010 y 2010-2022. `https://www.argentina.gob.ar/sites/default/files/2023/12/crecimiento_poblacional_2001-2022.pdf`, 2023. Accedido el 25 de julio de 2025.

[109] A. H. G. Rinnooy Kan. *Machine Scheduling Problems: Classification, complexity and computations*. Springer, New York, 1 edition, 1976.

[110] R. Roberti y P. Toth. Models and Algorithms for the Asymmetric Traveling Salesman Problem: an Experimental Comparison. *Journal of Mechanical Systems for Transportation and Logistics*, 1:113–133, 2012.

[111] P. Rosin y J. Žunić. Measuring Squareness and Orientation of Shapes. *Journal of Mathematical Imaging and Vision*, 39:13–27, 2011.

[112] R. Ruiz. Scheduling heuristics. In R. Martí, P. Panos, y M. G.C. Resende, editors, *Handbook of Heuristics*, pages 1–24. Springer International Publishing, Cham, 2015.

[113] A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127, 1999.

[114] E.A. Silver, D.F. Pyke, y R. Peterson. *Inventory Management and Production Planning and Scheduling*. Wiley, 1998.

[115] D. Soler, E. Martínez, y J. Micó. A transformation for the mixed general routing problem with turn penalties. *J Oper Res Soc*, 59:540–547, 2008.

[116] M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Oper. Res.*, 35:254–265, 1987.

[117] G. Tchobanoglous y F. Kreith. *Handbook of Solid Waste Management*. McGraw-Hill Education, 2 edition, 2002.

[118] J. Teixeira, A. P. Antunes, y J. P. de Sousa. Recyclable waste collection planning—a case study. *European Journal of Operational Research*, 158(3):543–554, 2004.

[119] E. B. Tirkolaee, P. Abbasian, M. Soltani, y S. A. Ghaffarian. Developing an applied algorithm for multi-trip vehicle routing problem with time windows in urban waste collection: A case study. *Waste Management & Research*, 37(1_suppl):4–13, 2019. PMID: 30761957.

[120] L-Y. Tseng, Y-H. Chin, y S-C. Wang. A minimized makespan scheduler with multiple factors for Grid computing systems. *Expert Systems with Applications*, 36(8):11118–11130, 2009.

[121] M. Đurasević y D. Jakobović. Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: a survey. *Artificial Intelligence Review*, 56(2):3181–3289, 2023.

[122] M. Đurasević, D. Jakobović, y K. Knežević. Adaptive scheduling on unrelated machines with genetic programming. *Applied Soft Computing*, 48:419–430, 2016.

[123] G. Van Brummelen. *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press, 2013.

[124] N. Veerapen, G. Ochoa, M. Harman, y E. K. Burke. An Integer Linear Programming approach to the single and bi-objective Next Release Problem. *Information and Software Technology*, 65:1–13, 2015.

[125] S. E. Vergara y G. Tchobanoglous. Municipal solid waste and the environment: A global perspective. *Annual Review of Environment and Resources*, 37(1):277–309, 2012.

[126] M. Wang y G. Pan. A novel imperialist competitive algorithm with multi-elite individuals guidance for multi-object unrelated parallel machine scheduling problem. *IEEE Access*, 7:121223–121235, 2019.

[127] F. Wesner. *Técnicas de programación lineal entera para la optimización de la recolección de residuos reciclables en el Municipio de Morón*. Tesis de licenciatura, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2015.

[128] L. Wu y S. Wang. Exact and heuristic methods to solve the parallel machine scheduling problem with multi-processor tasks. *International Journal of Production Economics*, 201:26–40, 2018.

[129] M. Zambra-Rivera, P.A. Miranda-Gonzalez, y C.A. Blazquez. A multiperiod household waste collection system for a set of rural islands with dynamic transfer port selection. *Optimization and Engineering*, 25:1779–1816, 11 2024.

[130] Z. Zhang, L. Zheng, y M. X. Weng. Dynamic parallel machine scheduling with mean weighted tardiness objective by q-learning. *The International Journal of Advanced Manufacturing Technology*, 34:968–980, 2007.

[131] H. Zhou, Z. Li, y X. Wu. Scheduling unrelated parallel machine to minimize total weighted tardiness using ant colony optimization. In *2007 IEEE International Conference on Automation and Logistics*, pages 132–136, 2007.

[132] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *J Optim Theory Appl*, 45:45–51, 1985.