

UNIVERSIDAD TORCUATO DI TELLA, ESCUELA DE NEGOCIOS

MASTER IN MANAGEMENT + ANALYTICS

Gestión eficiente de pequeños comerciantes en redes de e-commerce: problemas, modelos y optimización

Tesis de Maestría

Alumno: Juan Pablo Santos

Mentor: Juan José Miranda Bront

Julio 2020

Gestión eficiente de pequeños comerciantes en redes de e-commerce: problemas, modelos y optimización

Abstract

El presente trabajo aborda un problema de toma de decisión dentro del diseño de la red logística en el contexto de un e-commerce muy importante de Latinoamérica. Si bien estas plataformas tradicionalmente se focalizan en grandes vendedores, este trabajo se centra en explorar el segmento denominado long-tail para la gestión logística de las ventas de pequeños comerciantes mediante centros de consolidación; usualmente tercerizados a través de comercios alternativos con capacidad ociosa. Partiendo de datos reales provistos por la empresa, se realizaron distintos ejercicios con el fin de responder preguntas relevantes de negocio relacionadas con el diseño y la operación de la red de este segmento particular. Las decisiones son formuladas mediante modelos de programación lineal entera, en particular relacionados con problemas de asignación generalizada (GAPs) y abordadas utilizando algoritmos específicos para su resolución. Los resultados obtenidos muestran una mejora sustancial comparada con el procedimiento actual de la plataforma. Las soluciones encontradas muestran reducciones en las distancias recorridas del orden del 50%, implicando una mejora tanto en los costos como en la experiencia de los actores involucrados.

Efficient management of small merchants in e-commerce networks: problems, models and optimization

Abstract

This paper addresses a decision-making problem within the design of the logistics network in the context of a very important e-commerce in Latin America. Although these platforms traditionally focus on large sellers, this work focuses on exploring the segment called long-tail for the logistics management of small merchant sales through consolidation centers, usually outsourced through alternative businesses with idle capacity. Based on real data provided by the company, different exercises were carried out in order to answer relevant business questions related to the design and operation of the network for this particular segment. Decisions are formulated using integer linear programming models, particularly related to generalized allocation problems (GAPs), and addressed using specific algorithms for their resolution. The results obtained show a substantial improvement compared to the current platform procedure. The solutions found show reductions in distances traveled of the order of 50%, implying an improvement both in costs and in the experience of the actors involved.

Alumno: Juan Pablo Santos

Introducción	5
Motivación	5
Literatura previa	6
Contexto	7
Aparición de una nueva tendencia: incorporación de los newbie sellers a la red	11
Objetivo de investigación	13
Situación actual: ¿cómo es el proceso de asignación seller-place que hace el e-commerce?	14
Los datos	15
Los sellers	16
Los places	17
La matriz de distancia	18
Solución “original del e-commerce”	19
Los modelos	26
Notación de parámetros y variables	27
Modelo (i): Minimizando la distancia recorrida	27
Modelo (ii): Maximizando la cantidad de paquetes que entran a la red	28
Modelo (iii): ¿Qué places conviene abrir y qué sellers incorporamos a la red?	30
En busca de las soluciones a los problemas	31
Solución al modelo (i)	32
Solución al modelo (ii)	39
Solución al modelo (iii)	41
Detalle computacional de los modelos	45
Conclusiones y posibles extensiones	46
Resumen de los resultados	46
Posibles extensiones	47
Extensiones vinculadas herramientas y procesos	47
Extensiones vinculadas a los datos	47
Extensiones vinculadas a los modelos	48
Bibliografía	50
Bibliografía académica	50
Bibliografía no académica	51
Anexo 1: Solución del e-commerce	52
Anexo 2: Scripts	72

Introducción

Motivación

Un problema clásico en investigación operativa y la toma de decisiones consiste en decidir la ubicación óptima de instalaciones (*facility location*, según su nombre en inglés) a fin de minimizar costos operativos garantizando un determinado nivel de servicio. Estos problemas surgen naturalmente en diferentes contextos reales, como por ejemplo en el diseño de la red logística cadena de suministro o la determinación de puestos de asistencia en el diseño de estrategias de contingencia ante emergencias. Como afirman Ross y Soland (1977) una variedad de problemas de *facility-location* y *location-allocation* pueden resolverse con problemas de asignación generalizados (GAP's por sus siglas en inglés). Sin embargo, a pesar de que estos temas en la literatura académica se estudian desde hace más de 50 años, empresas que nacieron en un mundo online y tuvieron (por su evolución natural) que acercarse al mundo físico, suelen ignorar estas cuestiones y resolverlas de manera manual evitando llegar a soluciones óptimas o más eficientes.

Esta investigación nace de un problema muy simple: asignar actores que generan tráfico en una red a aquellos nodos concentradores que les son más cercanos, procurando no colapsar su capacidad. A pesar de su simpleza, en este trabajo veremos el caso de una empresa de *e-commerce* líder en Latinoamérica, cuya metodología actual subyace en un proceso manual, con herramientas básicas y reglas que, si bien se basan en el sentido común, no son las más eficientes y menos aún las más escalables.

Partimos este trabajo estudiando su *modus operandi* en la resolución de un problema de asignación. Un analista con una planilla de cálculo, determina de forma manual la distancia entre los actores y los nodos y asigna el más cercano a cada actor. Variables como la capacidad eran completamente ignoradas o tenidas en cuenta de manera parcial según la subjetividad de quien realizaba el análisis.

Nuestro objetivo es analizar métodos alternativos para abordar la toma de decisión, buscando una forma más eficiente y escalable de realizar la tarea. A través del modelado de tres problemas del tipo GAPs con distintas características y usando programación lineal entera, buscando contestar preguntas de negocio muy concretas y utilizando paquetes de softwares de optimización muy conocidos en el mercado (SCIP y CPLEX), mostramos que la solución aplicada en la práctica es mejorable tanto desde la calidad de la solución como desde la metodología utilizada. Gracias a los modelos planteados se logra reducir a la mitad la distancia mínima total a recorrer por estos actores mediante una mejor utilización de la capacidad de la red y generando modificaciones a las asignaciones hechas por el *e-commerce* inferiores al 25%.

Alumno: Juan Pablo Santos

Por sus tiempos de respuesta y simplicidad, las herramientas desarrolladas pueden ser fácilmente aplicables por la plataforma para probar diferentes variantes y seguir haciendo crecer su red de manera óptima. Esto, sin lugar a dudas, es el mayor aporte del presente trabajo.

En la siguiente sección, abordaremos más en detalle trabajos académicos de otros autores que tienen un espíritu similar a éste. Luego, ahondaremos un poco más en el contexto en el cual se enmarca esta investigación, describiremos la aparición de una nueva tendencia en el mundo del *e-commerce*, plantaremos el objetivo puntual que perseguimos con este trabajo, cómo es el *status quo*, los datos, modelos y soluciones que encontramos y finalmente se concluirá y se propondrán posibles extensiones.

Literatura previa

El foco de esta sección no se centra en el modelado de los problemas de asignación generalizada ni en métodos de resolución, sino en su utilización en la vida real. Para ello, tomaremos como base el trabajo realizado por Oncan (2007). Enunciamos también algunos ejemplos relevantes que, si bien no utilizan GAPs, representan aplicaciones concretas a problemas reales mediante técnicas de optimización en línea como las utilizadas en este trabajo.

Como se afirma en su *trabajo*, hay seis grandes tipos de aplicaciones de GAPs: (1) Organizaciones de agendas (*Scheduling Problems*) (2) Transporte y ruteo (3) Telecomunicaciones y servicios esenciales (4) Planificación de producción (5) *Facility Location* y (6) *Supply Chain*.

En lo que respecta a Organizaciones de agendas (1), las aplicaciones más típicas involucran el ordenamiento o la asignación de una fuerza de trabajo (ya sean personas o máquinas) a la realización de una tarea de acuerdo a sus habilidades, necesidades y/o esfuerzos requeridos por el *decision maker*. Un ejemplo de este tipo de trabajo puede ser la organización de un fixture deportivo como la investigación realizada por Durán et al. (2007) en la que se reorganiza la agenda de la liga chilena de fútbol buscando optimizar las necesidades señaladas por los organizadores.

En lo que respecta a Transporte y ruteo (2), las aplicaciones son muy variadas y dependen de la industria. En relación a logística, existen numerosos usos vinculados a la optimización de rutas de retiro de productos de usuarios vendedores o entregas de *last mile* a usuarios compradores. Un caso de éxito conocido en la literatura es el de Holland et al. (2017) que según las estimaciones, cuando la empresa UPS implementó un nuevo sistema de ruteo (ORION), pudo generar ahorros de entre 300-400 millones de dólares anuales y reducir la huella de carbono en 100,000 toneladas anuales. La utilización de GAPs en relación a este problema puede observarse en una etapa previa a la abordada por el

Alumno: Juan Pablo Santos

trabajo, a fin de determinar el flujo de envíos a los respectivos centros de consolidación intermedios para luego ser distribuidos a su destino final.

En lo referido a Telecomunicaciones y servicios esenciales (3) y *Facility Location* (5) las aplicaciones están vinculadas a las mejoras en el servicio y reducción de costos. Un ejemplo de aplicación en este rubro puede ser el de Erkut et al. (2000) donde la empresa TransAlta Utilities (TAU) de Canadá, implementó los resultados de algoritmos de *facility location* y definió cerrar dos centros logísticos y abrir otros tres en otra ubicación con el objetivo de mejorar su operación y atención a sus clientes.

En cuanto a Planificación de producción (4) las aplicaciones típicas corresponden a problemas de *lot sizing*. Por ejemplo, Viveros y Salazar (2010) presentaron un modelo para la planificación de la producción en un sistema multiproducto con líneas de producción en la fabricación de medios de acero para la molienda de minerales. El modelo entrega una tasa de producción por línea, la utilización de la capacidad instalada, el balanceo de inventario de productos y materias primas y la planificación de sus adquisiciones en un horizonte anual dividido en períodos mensuales.

En lo que respecta a Supply Chain (6), el trabajo de Higgins (1999), es un buen ejemplo cercano al presente trabajo. El autor realiza un modelo de programación lineal entera que busca optimizar las fechas de la cosecha y la duración del ciclo de cultivo maximizando los ingresos netos para una región en particular. El modelo es una extensión de GAPs, usa más de 500 mil variables enteras y está sujeto a las restricciones de la tierra total asignada, la molienda de los molinos y la capacidad de transporte geográfico. La resolución se logra a través de una heurística por la enorme cantidad de variables usando una forma de búsqueda local que incorpora ideas de *tabu search*.

El presente trabajo toca al menos dos de las tipificaciones señaladas por Oncan: Facility Location (5), y Supply Chain (6).

Finalmente, siguiendo el análisis de la literatura hecho por M.T. Melo et. al. (2009) y ReVelle et. al. (2005) podemos enmarcar los problemas que se describirán más adelante como problemas de múltiples instalaciones de ubicación plana con necesidad de ser ubicadas y de asignar clientes en las mismas. Las soluciones no sólo requieren la ubicación de estas instalaciones sino también la asignación de los clientes a dichas instalaciones.

A continuación se brinda al lector un contexto general de la evolución del *e-commerce* en Latinoamérica que servirá para enmarcar el problema, presentar a los actores involucrados y ayudar a un mejor entendimiento del desafío en cuestión.

Contexto

La historia del *e-commerce* en América Latina fue evolucionando a lo largo de los años. Pasó de un esquema de subastas (similar a lo que hacía Ebay en Estados Unidos) donde lo

Alumno: Juan Pablo Santos

que el vendedor publicaba se lo llevaba el mejor postor (e.g. DeRemate a comienzos del nuevo milenio) a plataformas mucho más complejas, cada vez están más integradas verticalmente con diferentes eslabones que colaboran para que el usuario final reciba su compra. Si bien en un comienzo se limitaban a que un comprador y un vendedor entren en contacto y concreten una transacción iniciada de manera online, hoy en día estas plataformas fueron evolucionando y ofrecen herramientas que permiten garantizar la concreción exitosa de una transacción. Estas mejoras pueden dividirse en dos tipos:

1. Mejoras en las formas de pago.
2. Mejoras en las formas de envío.

En lo que respecta a la forma de pago, hace ya algunos años, nacieron las *plataformas de pagos* que permiten al usuario comprador poder pagar su compra con múltiples medios: online (tarjetas de crédito, débito, criptomoneda o dinero en cuenta) y offline (metodologías que requieren que el usuario vaya a un lugar físico y que con dinero en efectivo pague esa transacción). Gracias a este servicio, la interacción en la plataforma se despersonaliza y se evitan posibles fraudes. Adicionalmente, el usuario comprador tiene la posibilidad de particionar el pago de su compra en cuotas (con o sin interés dependiendo de mercado y del contexto) como también pedir créditos de manera online usando como *collateral* su comportamiento en la plataforma (e.g. su reputación, nivel de lealtad, cantidad de compras en un período de tiempo, etc). Es tal el crecimiento de estas formas de pago que hoy en día trascendieron al mundo físico y es posible encontrar muchísimo comercios en varios países de Latinoamérica como Argentina, Brasil y México que permiten que los compradores paguen a través de aplicaciones como si se tratara de billeteras virtuales donde el dinero físico y las tarjetas plásticas dejan de tener un rol relevante [1]. Es importante mencionar de todos modos, que este fenómeno no nació en Latinoamérica, el mercado donde esta modalidad ya forma parte de la cotidianidad de las personas es China a través de la aplicación WeChat principalmente donde los usuarios ahorran, pagan y se relacionan con otros usuarios.

En lo que respecta a los envíos, identificamos dos actores principales que interactúan en la plataforma. Por un lado tenemos a los denominados *sellers*, que son aquellos encargados de disponibilizar la oferta. En términos prácticos, un *seller* representa un comercio que busca vender y distribuir su producto a través de la plataforma. Como contraparte, tenemos a los denominados *buyers*, que son usuarios que consumen la oferta provista por los *sellers*. En la práctica, los *buyers* son esencialmente los usuarios de la plataforma que buscan comprar y, eventualmente, recibir el producto en su domicilio o en un punto de *pick up* (lugares donde pueden retirar los productos que compraron, ya sea porque una persona se los entrega como en la oficina de un correo postal o porque quedan en *lockers* estratégicamente ubicados de acuerdo a sus hábitos y preferencias).

Los *sellers* fueron transicionando del mundo físico al mundo online y no estaban acostumbrados a lidiar con problemas logísticos. Su operatoria tradicional consistía en que el *buyer* fuera a su local físico, vea la oferta disponible y de acuerdo con sus preferencias,

Alumno: Juan Pablo Santos

defina si compraba o no. Al tener ahora *buyers* en diferentes puntos geográficos que acceden a su oferta de manera online, ingeniárselas para hacer llegar los productos a lugares recónditos fue algo que los superó completamente.

Dada esta falta de conocimiento logístico, las plataformas de *e-commerce*, decidieron introducirse en el mundo de las entregas de productos buscando hacer la experiencia de venta del usuario *seller* más fácil y, que de esta manera, pudiera concentrarse en lo que mejor hace: vender y traer la oferta más atractiva para los *buyers*.

En una primera iteración, las plataformas buscaron hacer acuerdos con grandes operadores logísticos conocidos como *commercial carriers* (e.g. UPS, FedEx, DHL, Correios, Correo Argentino, etc). Mientras que el *e-commerce* garantiza *envíos* (paquetes físicos que debían moverse desde el *seller* al *buyer* principalmente) en una cantidad considerablemente alta, el *commercial carrier* garantiza a los usuarios de la plataforma dos cosas: (i) un tiempo de entrega razonable para una combinación de orígenes y destinos, (ii) un costo de envío menor teniendo en cuenta el volumen transaccionado en la plataforma.

En esta primera versión, cuando el *seller* hace una venta en la plataforma, se le disponibiliza una etiqueta que se adhiere al paquete y tiene que llevarlo a alguna oficina de despacho del *commercial carrier* donde hace del *drop-off* del producto.

Sin embargo, a medida que el *e-commerce* empieza a ganar penetración y el mundo físico es sustituido por el mundo online, esta operatoria deja de ser eficiente. Para un vendedor que hace más de veinte ventas en promedio por día, las implicancias operativas que significa acercarse diariamente a la oficina de un *commercial carrier* son excesivamente costosas. Las colas de espera en las horas pico, las oficinas abarrotadas de paquetes empiezan a reflejarse en altos tiempos de *handling time*¹ que repercuten negativamente en el tiempo total de entrega de los paquetes al *buyer*. Es importante aclarar que veinte ventas por día no es un número arbitrario. Tiene dos motivos principales:

(1) Teniendo en cuenta el valor promedio de venta en Latinoamérica, cuando hacemos referencia a ventas diarias en promedio superiores a veinte, dejamos de hablar de un *seller* esporádico y empezamos a hablar de un gran vendedor cuyo *core business* muy probablemente sea la venta online a través del canal *e-commerce*. Asumiendo que el ticket promedio de un *e-commerce* es de treinta dólares [2]², veinte ventas por día implica una facturación mensual de 18,000 dólares al mes. Esto en casi todos los países de la región es considerado una pequeña o mediana empresa [3].

(2) Si bien se menciona más abajo, clientes con más de veinte ventas por día, llegan al *breakeven* que justifica que un camión los visite para recolectar paquetes. Es decir, el costo de pasar a hacer una colecta por el domicilio de ese vendedor tiene sentido económico

¹ *Handling time* es el tiempo que transcurre desde que un *buyer* hace una compra hasta que el *seller* entrega este paquete al *commercial carrier*.

² Para calcular el ticket promedio se tomó el Quarterly Business Review de Q12020 de la empresa B2W Digital Company, y se tomó el cociente entre Total GMV (facturación total) y el assortment (ítems).

Alumno: Juan Pablo Santos

cuando se buscan más de veinte ventas por día ya que el esquema de pago suele ser por paquete recolectado.

Continuando con la evolución del *e-commerce*, fue así como aparece el servicio de colectas que consiste básicamente en una solución logística que visita con una frecuencia diaria y en un horario determinado el domicilio de los *sellers* y retira de allí todos los paquetes que éste le entrega. Posteriormente, estos paquetes van a un *cross docking center* (propio del *e-commerce* o tercerizado) donde se elige la mejor³ transportadora para hacer el envío final al destino (conocido como *last mile* o última milla).

En paralelo a estos modelos y viendo lo que hacía Amazon en EEUU, surge una tercera modalidad: *fulfillment*. El modelo consiste en varias etapas: *inbound*, almacenaje de stock, *waving*, *picking*, *packing* y *outbound*. En el proceso de *inbound*, el *seller* entrega su stock al *e-commerce*; a través de un sistema de agendamiento elige un día y horario en la semana y va a la ubicación del centro de almacenaje y le entrega a la plataforma su stock en consignación. Luego la plataforma se encarga del almacenaje de stock; dependiendo el país y la complejidad de la operación, puede tratarse de un almacenaje caótico o que tenga en cuenta la rotación de los productos. Cuando sucede una venta, se diseñan *ráfagas* de búsquedas que realizan el proceso de *picking* y llevan los productos comprados por los *buyers* a las estaciones de *empaquetado*, *previo a la distribución*. Finalmente, los paquetes se separan por destinos y transportadoras y pasan al proceso de *outbound* donde un proveedor logístico lleva estos productos a su centro de distribución desde el cual se entregará al *buyer*.

Para el *e-commerce* esto tiene una funcionalidad estratégica no menor: es una *lock-in strategy*; todo el stock que el *seller* le da en consignación, sólo puede ser vendido en la plataforma en cuestión. Adicionalmente, la existencia de penalidades asociadas a la no rotación de este inventario, aseguran al *e-commerce* que el *seller* está entregando su mejor oferta.

Finalmente, es importante mencionar que el *seller* tiene incentivos para entregar su stock al *e-commerce* por dos motivos:

(1) Su costo de almacenaje y procesamiento de pedidos (etiquetado y empaquetado) pasa a ser mucho más bajo o casi nulo ya que depende en su totalidad del *e-commerce*. Si bien en muchos modelos existe un costo que el sitio cobra al *seller*, el mismo es despreciable comparado a hacerlo por su propia cuenta.

(2) Las entregas desde el *fulfillment center* son más rápidas ya que el tiempo desde que se hace una compra hasta que la transportadora tiene el paquete en su poder para hacer la entrega final sucede en horas, pues el stock está consolidado en un sólo lugar y no en depósitos de miles de *sellers*. A su vez, menores tiempos de entrega están asociados a mayores ventas. Los *buyers* tienen preferencias muy exigentes: quieren el producto lo más

³ Los criterios para elegir la mejor transportadora dependían de tres variables principalmente: tiempos de entrega, costo y nivel de cumplimiento del *service level agreement* contratado.

Alumno: Juan Pablo Santos

rápido posible y si no encuentran este servicio prefieren ir ellos a una tienda física y realizar la transacción por sus propios medios.

Aparición de una nueva tendencia: incorporación de los *newbie sellers* a la red

Si bien estas plataformas tradicionalmente se concentran en grandes vendedores, la oportunidad en el segmento *long-tail* resulta cada vez más atractiva por dos motivos principalmente:

- (1) Oferta: el sector presenta una variedad de productos únicos. Las manualidades, obras de carácter artístico, personalización de los productos, etcétera, sólo puede venir de este segmento no masificado. Además, permite a estas plataformas posicionarse en un lugar bastante particular ya que se vuelven habilitadores para que pequeñas y medianas empresas puedan animarse a emprender con un costo de entrada bastante bajo: abrir una cuenta y pagar una comisión por cada venta que realizan.
- (2) Diversificación: la oferta de estas plataformas no puede concentrarse en un grupo pequeño de *sellers*. Si esto sucede, su capacidad de negociación es mayor y genera una asincronía de poder que puede ser muy difícil de controlar. Es decir, si por ejemplo diez *sellers* de los diez mil que hay en una plataforma imaginaria concentran el 90% de la facturación, el riesgo que tiene si uno de estos *sellers* decide dejar de operar es enorme. Por lo tanto, el incentivo siempre está en atomizar su oferta en la mayor cantidad de actores posibles.

Ahora la pregunta que claramente surge es cómo incorporamos estos *newbie sellers* a la red. Si bien se probaron muchos modelos y hay varios en ejecución aún, en este trabajo de investigación vamos a concentrarnos en uno en particular que pasaremos a llamar *Places*.

¿En qué consiste? El modelo en sí es bastante simple y toma ciertas ideas del tradicional de *drop-off* descrito anteriormente. Cada vez que el *newbie seller* hace una venta, se le disponibiliza una etiqueta, él tiene la responsabilidad de embalar este paquete y llevarlo a un *Place* o punto de consolidación. A posteriori, un vehículo recolecta los paquetes por cada uno de los puntos y los lleva a un *cross docking center*, consolidando así la carga de un conjunto de *sellers* de escala chica en una única parada.

La gran diferencia entre el modelo de *places* y el modelo de *drop-off* tradicional está en las características que tienen estos puntos de consolidación. Mientras que en el modelo de *drop-off* se trata de oficinas postales de *commercial carriers*, en el modelo de *Places*, un *place* es una tienda tradicional que se encuentra en la ciudad y tiene la particularidad de que su *core business* no es consolidar carga. Ejemplo de un *place* puede ser una librería de

Alumno: Juan Pablo Santos

barrio, un kiosco, una tienda que vende fundas para celulares, etc. Siguen a continuación algunas imágenes de *places*:

Imágenes 1: Ejemplos de Places en Latinoamérica



Ejemplos de *Places* reales: la primera fotografía de arriba a la izquierda, muestra el típico *place*, un local de ventas de accesorios para celulares. La segunda foto de arriba a la derecha, un local de asistencia técnica de celulares. Finalmente, la tercera foto es un *place* que su principal negocio es brindar servicios y soluciones informáticas.

Con el advenimiento del mundo online, estas tiendas empezaron a perder clientes que optan cada vez más por comprar desde sus hogares en lugar de trasladarse; esto genera espacios físicos ociosos que pueden aprovecharse para otras actividades como por ejemplo la consolidación de paquetes. Para el *place* esto es conveniente desde un punto de vista monetario por un doble motivo, por un lado, amortiza el costo que implica tener espacio ocioso porque recibe un ingreso monetario por cada paquete que un *seller* deja en el *place*

Alumno: Juan Pablo Santos

y por el otro lado, recibe tráfico indirecto en su tienda que potencialmente puede monetizar a través de un *cross-selling*.

El modelo de *Places* como punto de drop-off es poco conocido entre las plataformas, la mayoría de los marketplaces que existen en esta región mutaron desde el retail, por lo que, tradicionalmente están acostumbrados a ser ellos quienes venden sus propios productos online. El rol del *seller* es ocupado por la propia plataforma. Dicho esto, es poco usual que los *sellers* tengan que ir a dejar paquetes a puntos de consolidación. Sin embargo, desde el punto de vista de los *buyers*, el modelo de *Places* sí es bastante conocido como una red de puntos de *pick-up*. Frecuentemente los usuarios que no quieren que los productos lleguen a sus hogares, se acercan a estos lugares y desde allí los retiran. Alibaba o más específicamente TaoBao en Hong Kong [4], tiene una red enorme de puntos de *pick-up* que aprovechan los *buyers*; la cercanía, la amplitud horaria de atención y la buena relación con quienes operan estas tiendas físicas son las principales variables que tienen en cuenta al elegir un lugar al cual mandar lo que compraron de manera online. Otro caso de éxito en el mundo es el de Allegro en Polonia. Este marketplace cuenta con una extensa red de *PUDOs (Pick-up and Drop-off) Points* y 52% de los compradores online usan esta metodología para el envío de sus compras [5]. Este valor es realmente muy superior a los que encontramos en Latinoamérica donde la preferencia de los *buyers* por recibir sus compras online en sus domicilios, sigue siendo la opción más elegida.

Objetivo de investigación

El objetivo de este trabajo es generar una asignación óptima de *sellers* y *places*. Vamos a abordar el problema utilizando metodologías que impactan en cómo se toman las decisiones hoy, mejorando la calidad de las mismas, basándonos en datos. Para ello, analizaremos y formalizaremos el proceso de toma de decisión (qué *sellers* a qué *places*) desarrollando una serie de modelos matemáticos e implementando un prototipo que permita abordar el problema de una manera sistemática.

El desafío es cuestión es un problema **real** al que se enfrenta un *e-commerce* en la actualidad. Como explicamos anteriormente, este proceso hoy en día se resuelve de una manera manual donde se calcula la distancia que debe caminar el *seller* al *place* más cercano intentando de no colapsar la capacidad declarada del lugar. La asignación no es la óptima porque provee una solución que tiene el riesgo de colapsar el *place* y por lo tanto ofrecer una alternativa subóptima al *seller*, degradando su experiencia. Esto se debe a que hay ciertas restricciones que por las herramientas utilizadas, no son tenidas en cuenta.

Como resultado de este trabajo hay tres beneficios potenciales para el *e-commerce* y los actores involucrados en el proceso:

Alumno: Juan Pablo Santos

- (1) En primer lugar lograremos sistematizar la decisión lo que permitirá a los *decision makers* probar escenarios, corroborar los resultados con la realidad y evaluar si se deben quitar o sumar restricciones de una forma muy sencilla.
- (2) En segundo lugar, los *sellers* que forman parte de la red probablemente reduzcan el tiempo promedio que destinan a trasladarse para dejar sus paquetes en los *places*.
- (3) En tercer lugar, podremos evitar que *places* se vean colapsados porque como veremos en la siguiente sección, si bien declarativamente las personas encargadas de tomar la decisión en la plataforma afirman que la variable de capacidad sí es tenida en cuenta, esto es una verdad a medias que no se cumple sistemáticamente para todos los casos.

Situación actual: ¿cómo es el proceso de asignación seller-place que hace el e-commerce?

La manera que tiene la plataforma de asignar *sellers* a *places* es completamente manual. Un analista en una planilla de cálculo se hace de la siguiente información:

- *Sellers* (código identificador del *seller*, cantidad de ventas promedio por día, geolocalización de su domicilio i.e. latitud y longitud)
- *Places* (código identificador del *place*, capacidad promedio declarada y geolocalización del lugar físico i.e. latitud y longitud).

Generalmente, las instancias con las cuales se trabaja son manejables fácilmente, es decir, la cantidad de *sellers* oscila los mil y la cantidad de *places* los trescientos o quinientos. Hasta el momento (de acuerdo a lo declarado) nunca hubo una instancia más grande porque la asignación de los *sellers* y apertura de los *places* suele ser por ciudades o barrios dependiendo de la densidad.

Con alguna fórmula de cálculo de distancia (generalmente la más usada es la fórmula Haversine⁴), se determina los kilómetros que separan a un *seller* del *place* más cercano. La distancia es una variable súper importante en este problema: mientras más tenga que desplazarse un *seller* a un *place*, peor es su experiencia. Los costos asociados al traslado corren por cuenta del *seller*, con lo cual, el tiempo destinado a esta parte del proceso puede generar desincentivos a vender en la plataforma y por lo tanto un riesgo al *e-commerce* de perder esta valiosa oferta.

⁴ Para el cálculo de esta distancia, véase la siguiente página web: Haversine formula (2 de julio de 2020). En Wikipedia.

https://en.wikipedia.org/wiki/Haversine_formula

Alumno: Juan Pablo Santos

Como se mencionó anteriormente, la variable capacidad declarativamente es tenida en cuenta como parte del proceso, pero de forma parcial. Existe un preconceito por el lado del *e-commerce* respecto a que los *places* suelen subestimar su capacidad real o que tienden a declarar un valor mucho menor al real. Con lo cual, violar la capacidad del lugar no es considerado como algo grave o como una restricción para hacer asignaciones para los *decision makers* de la plataforma siempre y cuando la asignación no supere un 50% de la capacidad declarada por el *place*. Adicionalmente, existe dentro del *e-commerce* la idea de que si un *place* colapsa, gestionar un salvataje, es decir, una colecta de emergencia que vaya al lugar y desagote de paquetes del punto, es algo fácil de gestionar y una buena opción en caso de darse una contingencia.

Es importante mencionar una diferencia sutil, pero no menor, de este trabajo de investigación en comparación con el operar normal del *e-commerce*. El proceso de asignación en la plataforma se hace por tandas semanales. En el momento en el cual se realizó el relevamiento del presente trabajo, las tandas nunca eran de más de veinte *sellers* a la vez. Una vez que los *sellers* eran designados, se iniciaba el contacto efectivo y se les informaba de su nuevo *modus operandi* y luego de dos semanas comenzaba a operar teniendo que ir a despachar de manera mandatoria al lugar asignado. Las tandas iban alternando barrios o ciudades intentando poco a poco llegar a todos los *sellers* deseables para participar en este nuevo flujo logístico.

Como veremos más adelante, el dataset con el que se cuenta no tiene una apertura por tandas, contamos con *sellers*, *places* y la asignación hecha por el *e-commerce*, es decir, la foto final. De todos modos, dado que los *decision makers* se manifestaron abiertos a la opción de reasignar *sellers* si los resultados encontrados difieren de los actuales, el presente trabajo tiene validez aunque no replique exactamente igual cómo la plataforma llegó a esta asignación compartida.

Los datos

Se compartió un set de datos de *sellers* y *places* que representa el tamaño total de la red al momento de iniciar el trabajo. Por lo mencionado anteriormente, trabajaremos con la red en su totalidad y no de manera secuenciada y por tandas. Esto nos aleja un poco de cómo fue la sucesión de los hechos, pero nos permite buscar optimizar la red como un todo. Recordemos que la red no tiene por qué ser estática, es decir, hay apertura por el lado de los *decision makers* del *e-commerce* de reasignar *sellers* a *places*. Esto es así ya que no hay en sí un gran costo asociado a cambiarle al *seller* su lugar de despacho: se prepara una comunicación a los *places* comentando cuáles son los nuevos *sellers* que irán a despachar y una comunicación a los *sellers* mostrando a qué nuevo *places* deberán ir a dejar sus productos. Si bien es probable que al comienzo los *sellers* continúen yendo a su

Alumno: Juan Pablo Santos

lugar de despacho habitual, la plataforma puede determinar un día D a partir del cual el *place* a nivel sistémico deja de recibir paquetes de ciertos *sellers* lo cual lo fuerza a ir a despachar al nuevo lugar que se le asignó.

Profundizando en los datos, contamos con información real de 1036 *sellers* y 310 *places* de una de las empresas de *e-commerce* más grandes de Latinoamérica. En lo que sigue de la sección haremos una descripción estadística de los mismos. Empezamos por los *sellers*, luego por los *places*, la matriz de distancia y finalizamos geolocalizando los datos. Esto nos permitirá adentrarnos en la solución manual al problema que implementó el *e-commerce*.

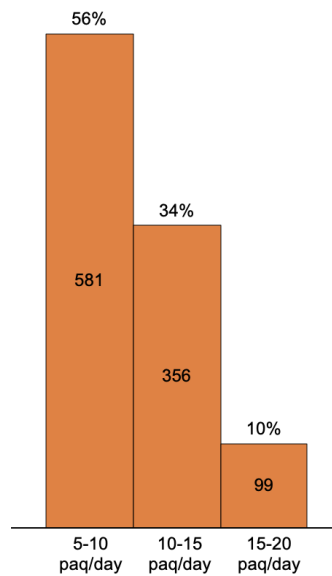
Los sellers

El *seller* típico de la muestra realiza en promedio 10 envíos al día y el desvío estándar es de 3 paquetes, esto nos demuestra que no hay una alta dispersión en cuanto a las características de estos actores.

El Gráfico 1 nos muestra un histograma de cómo está compuesta esta muestra de *sellers* teniendo en cuenta tres rangos de envíos promedio por día. Como se puede observar, la gran mayoría de los *sellers* hace entre 5-10 envíos promedio diario, no hay *sellers* que hagan menos de 5 paquetes en esta muestra y todos ellos hacen menos de 20 por lo que son considerados “pequeños *sellers*” aunque no esporádicos (i.e. menos de 5 paquetes por día).

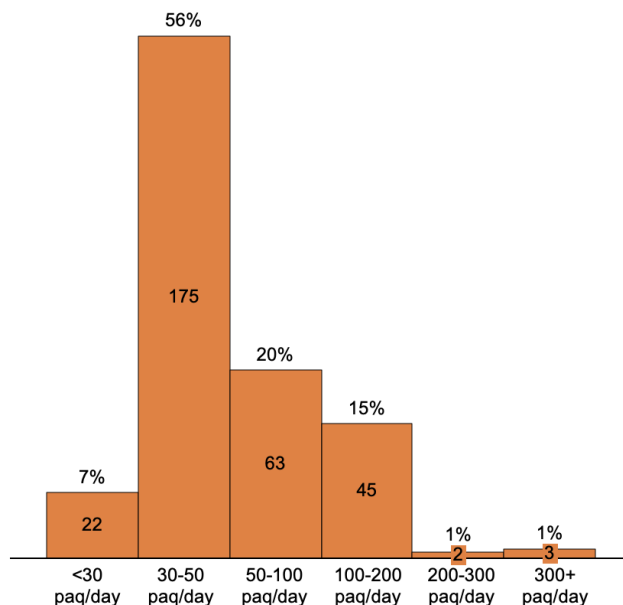
Alumno: Juan Pablo Santos

Gráfico 1: Cantidad de *sellers* por rango de envíos promedio por día



Los *places*

En lo que respecta a los *places*, el *place* promedio tiene una capacidad de 62 paquetes por día, sin embargo, el desvío estándar es de 162 paquetes diarios; esto se debe a la gran disparidad de capacidades existentes entre los *places* como se puede apreciar en el Gráfico 2.

Gráfico 2: Cantidad de *places* por rango de capacidad


Cuando se le preguntó al *e-commerce* por esta disparidad de capacidades, lo que se comentó fue que si bien el típico *place* es una tienda de barrio como locales de repuestos o accesorios para celulares, librerías, kioscos, almacenes, etc., se incorporaron también *places* más grandes como pequeños depósitos o hubs propios del *e-commerce* o de sus socios estratégicos, con la idea de robustecer a la red en aquellos centros urbanos donde había una gran concentración de *sellers*. Estos espacios de mayor tamaño suelen ser locales rentados donde anteriormente funcionaba una mediana empresa y que fueron transformados en centros logísticos de despacho.

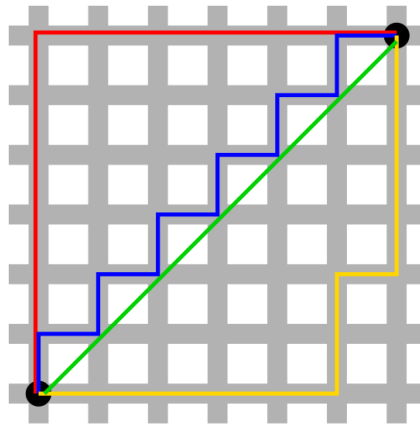
La matriz de distancia

Entre los datos compartidos por el *e-commerce*, contamos con la geolocalización de la totalidad de *sellers* y *places*. Esta información nos permite calcular la distancia entre todos los *sellers* a todos los *places*. Para proceder con este cálculo se analizan diferentes medidas de distancias, pero se concluye que la más adecuada era la *Manhattan Distance* [6] dado que el comportamiento esperado de los *sellers* es que se dirijan a los puntos a pie o en su defecto en un automóvil. La fórmula de la distancia Manhattan permite calcular desplazamientos en cuadrículas, lo cual simula muy bien cómo son los traslados que debe

Alumno: Juan Pablo Santos

hacer un seller en una topología de ciudad como es el caso de la geografía de los datos compartidos. Sigue a continuación un ejemplo⁵ extraído de Wikipedia a modo ilustrativo:

Ejemplo ilustrativo 1: Comportamiento de la distancia Manhattan



Distancia Manhattan versus distancia Euclidiana: Las líneas roja, azul y amarilla tienen la misma longitud (distancia Manhattan) mientras que la línea verde (distancia Euclidiana) tiene longitud menor, y es el camino más corto.

Volviendo a nuestro dataset, teniendo en cuenta que tenemos 1,036 *sellers* y 310 *places*, la matriz de distancia cuenta con 321,160 distancias, con una media de 30 kilómetros y una mediana de 15 kilómetros.

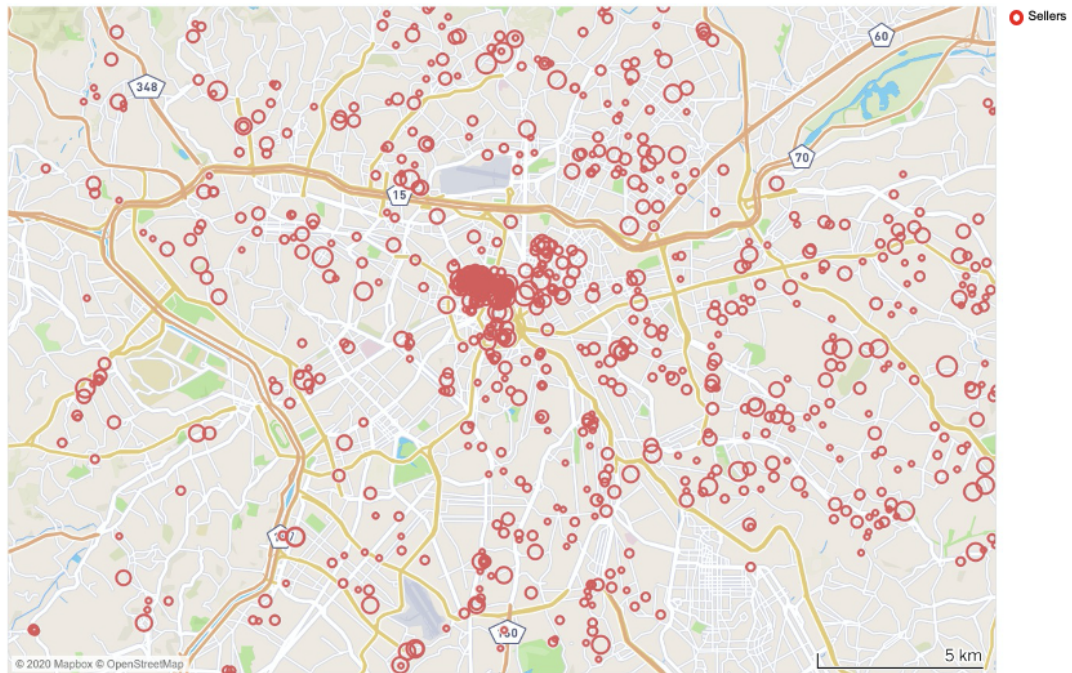
Solución “original del e-commerce”

Antes de avanzar con la solución manual que planteó el *e-commerce*, buscamos entender visualmente cómo lucen los datos. Para ello, geolocalizamos en un mapa tanto *sellers* como *places* obteniendo los siguientes resultados:

⁵ La imagen se extrajo de Wikipedia. URL:
https://upload.wikimedia.org/wikipedia/commons/0/08/Manhattan_distance.svg

Alumno: Juan Pablo Santos

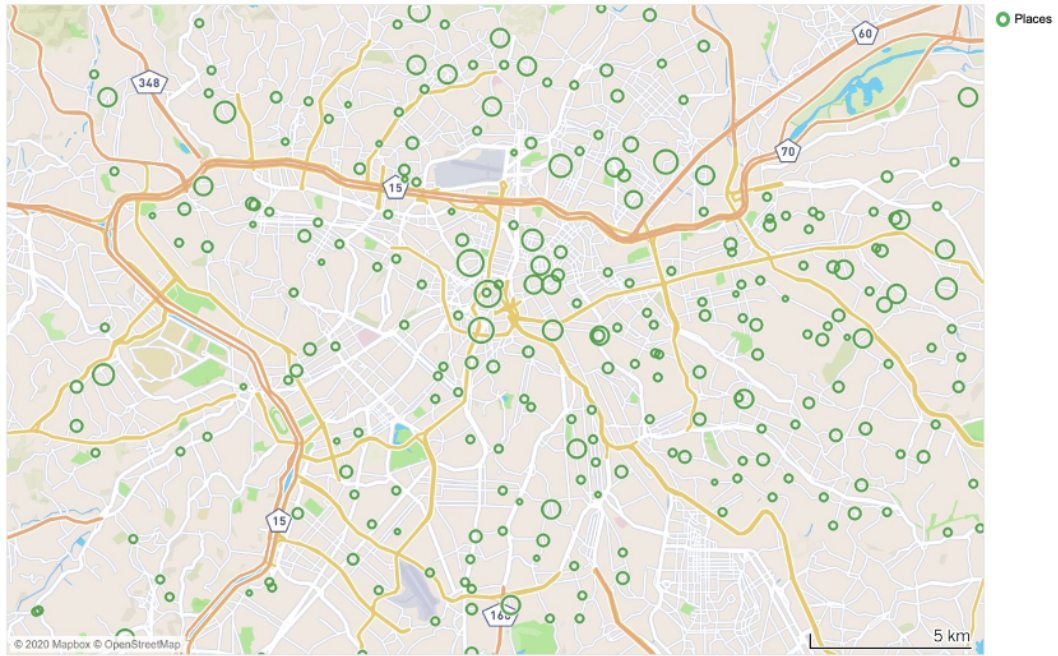
Mapa 1: Visualización de *sellers*. Tamaño proporcional a la cantidad de envíos



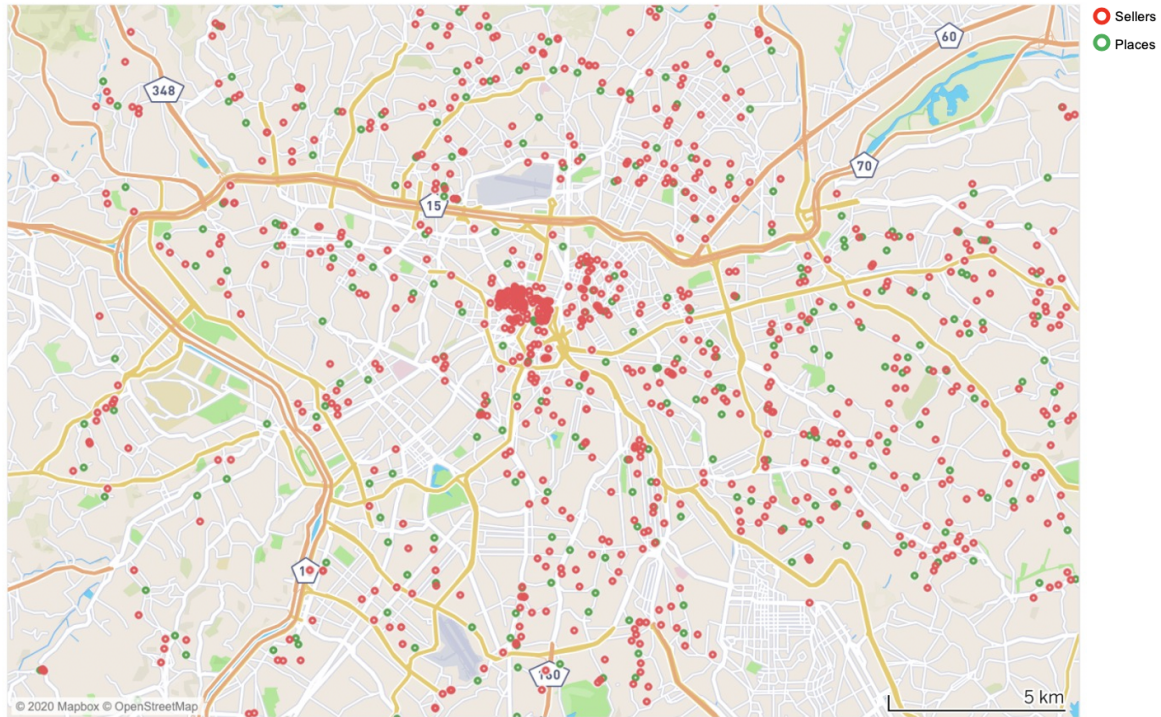
Geolocalización de *sellers*: las burbujas rojas representan la ubicación de los *sellers* dada su latitud y longitud. El tamaño de las burbujas representa la cantidad de envíos promedio por día que hace dicho *seller*.

Alumno: Juan Pablo Santos

Mapa 2: Visualización de *places*. Tamaño proporcional a capacidad



Geolocalización de *places*: las burbujas verdes representan la ubicación de los *places* dada su latitud y longitud. El tamaño de las burbujas representa la capacidad declarada por dicho *place*.

Mapa 3: Visualización de *sellers* y *places*


Geolocalización de *places* y *places*: las burbujas verdes representan la ubicación de los *places* dada su latitud y longitud. Las burbujas rojas representan la ubicación de los *sellers* dada su latitud y longitud.

Como se puede observar en el Mapa 1, no existe una zona caracterizada por *sellers* grandes o pequeños. Hacia el este parece haber una mayor cantidad de *sellers* y en el centro de la imagen hay una gran concentración. Evidentemente esta intuición fue tenida en cuenta al momento de buscar la ubicación de los *places* pues los de mayor capacidad están localizados cerca de las zonas de alta concentración como se puede observar en el Mapa 2 donde los *places* de mayor capacidad están en las zonas noreste .

Si bien no se muestra en el Mapa 1, al realizar el ejercicio anterior, se detectaron *sellers* que se encontraban muy lejos de la zona geográfica a analizar. Al preguntar el motivo a los responsables de la iniciativa, se explicó que podía suceder que algunas geolocalizaciones sufran de *poor quality data*. Dado que el cálculo de la distancia es sumamente importante para el análisis, se procedió a la eliminación de estos registros siempre y cuando la mínima distancia de ese *seller* al *place* más cercano, sea superior a 10 kilómetros⁶.

⁶ Más de diez kilómetros por lo que nos explicó el *e-commerce* implica un traslado obligado en automóvil por carreteras o autopistas que suelen estar muy congestionadas en los horarios promedio de entregas, obligando a destinar un tiempo de traslado superior al buscado por la plataforma.

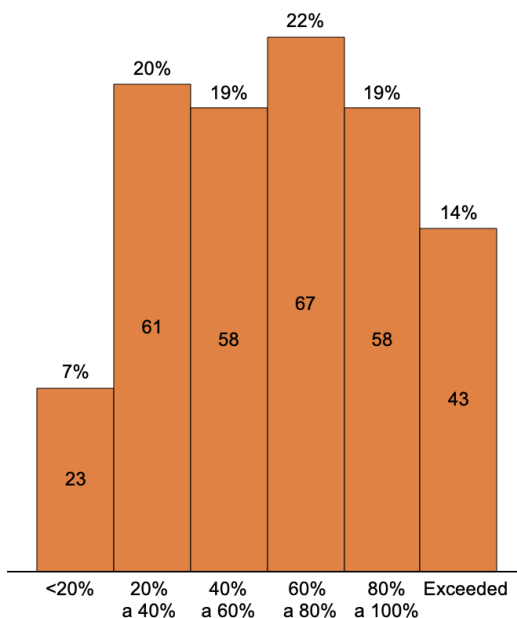
Alumno: Juan Pablo Santos

Se terminaron excluyendo 14 *sellers* cuya distancia media al *place* más cercano era de 1,018 kilómetros. Esto demuestra claramente la existencia de algún error con la información provista.

Ahora que entendemos bien los datos con los que contamos, podemos pasar a la asignación realizada por el *e-commerce*. La misma se encuentra caracterizada en la Tabla 1 del Anexo 1.

Si nos guiamos únicamente por los valores totales de capacidad y envíos promedio por día sumariados, pareciera que la solución no viola ninguna restricción de capacidad: la capacidad total de la red es superior a la cantidad de paquetes que los *sellers* pueden introducir en la misma. Sin embargo, al analizar más en detalle, podemos ver que la capacidad máxima declarada por el *place* es violada para 43 de ellos. El Gráfico 3 muestra la utilización promedio de la capacidad declarada de los *places*. Como se puede ver, el comportamiento es muy parejo entre los rangos 20-40%, 40-60%, 60%-80% y 80%-100%. La utilización promedio de los *places* es de un 64%. Los 43 *places* que tienen asignado volumen que excede su capacidad, están excedidos en un 17% en promedio, siendo el máximo exceso un 44%.

Gráfico 3: Solución e-commerce, histograma de utilización de la capacidad de los *places*



Al reportar esto al *e-commerce*, la solución sugerida fue variar la capacidad en un 50% en todos los casos, y de esta manera se llegó a una solución factible. Se optó por este método *greedy* porque fue el *enfoque* que los *decision makers* de la compañía sugirieron, es decir, negociar con los *places* para que aumenten su capacidad hasta donde fuese

Alumno: Juan Pablo Santos

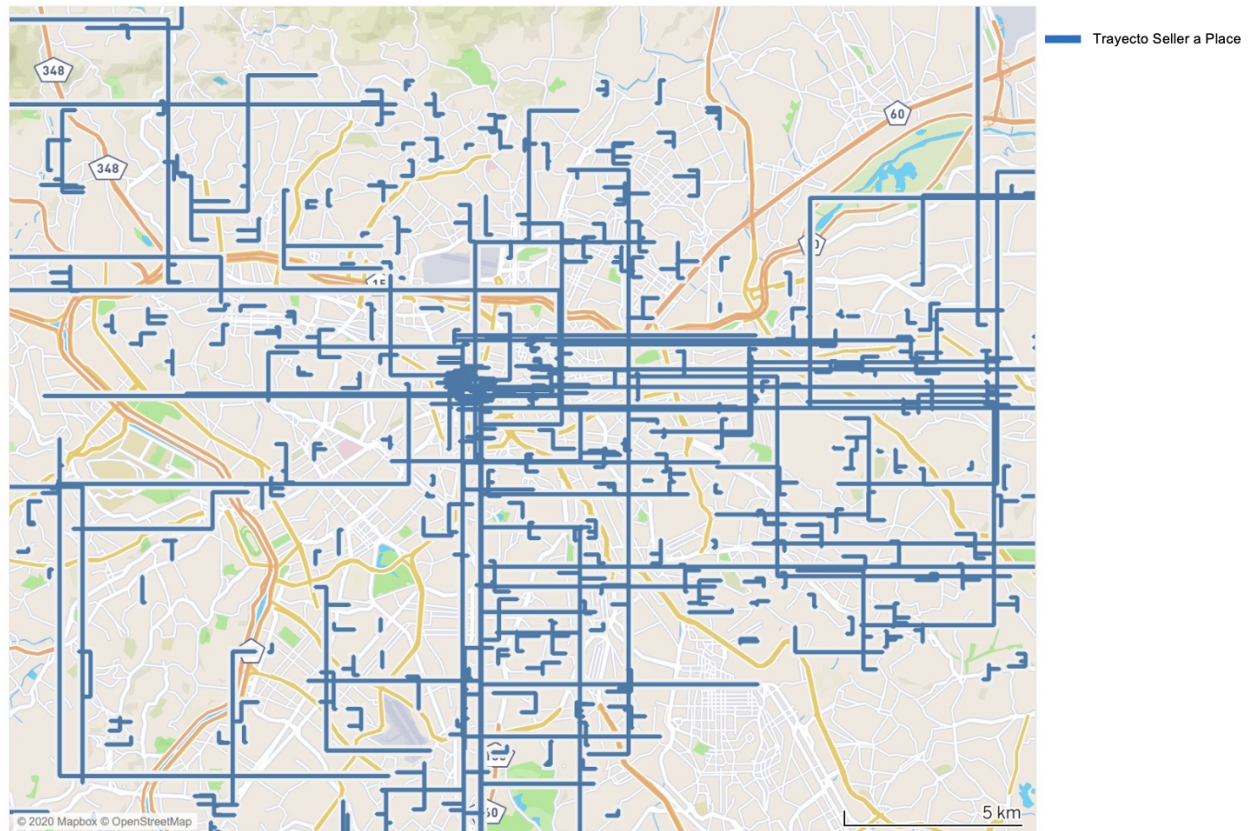
necesario. Para no diferenciar el pedido se haría a todos los *places* por igual bajo el supuesto de que esta capacidad extra es factible de ser negociada porque como se mencionó anteriormente, hay un preconcepto de que los *places* tienden a declarar una capacidad menor a la que realmente poseen.

Sigue a continuación una caracterización de la solución del *e-commerce* en forma resumida como así también una caracterización visual.

Resumen solución e-commerce

# Places	# Sellers	# Shipments	Sum Distance	Total Real Capacity
310	1,022	10,319.43	1,215 kms	19,102

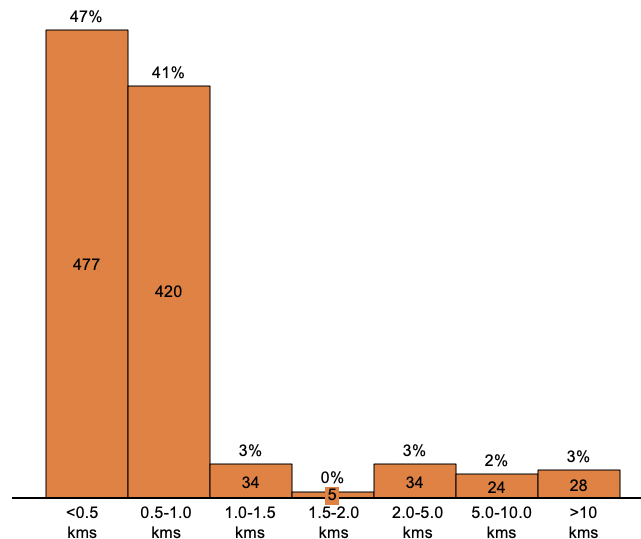
La primera columna de la tabla resumen indica la cantidad de *places* totales usados en la solución de la plataforma (**# Places**), la segunda la cantidad de *sellers* asignados (**# Sellers**), la tercera la cantidad de envíos en promedio (**# Shipments**) que hacen esos *sellers* (al trabajar en promedio, en número se encuentra en decimales), la cuarta columna representa la suma promedio de las distancias recorridas por los *sellers* (**Sum Distance**). Finalmente, la última columna sumaliza la capacidad total de todos los *places* de la red (**Total Real Capacity**).

Mapa 4: Visualización solución e-commerce


Si bien la distancia promedio de la solución es 1.2 kilómetros, hay algunos *sellers* que se tienen que trasladar hasta 33 kilómetros lo cual puede afectar su compromiso con el uso de la plataforma. Los traslados son evidenciados a través de las líneas azules del Mapa 4.

Finalmente, en el Gráfico 4 podemos ver que la calidad de la solución del *e-commerce* en términos de traslados promedios implica que el 88% de los *sellers* tienen que moverse a lo sumo un kilómetro. Sin embargo, hay un grupo de *sellers* cuyo traslado tiene un comportamiento poco usual que deja en evidencia que la estrategia descrita más arriba también tiene otros matices no declarados al momento de describir el procedimiento. No se logró indagar mucho al respecto pero se asume que puede tratarse de algún error humano al momento de hacer la asignación.

Gráfico 4: Histograma de traslados de los sellers a los places de acuerdo a la solución obtenida del e-commerce



Los modelos

Al hablar con los responsables del *e-commerce* en cuestión se detectaron tres preguntas diferentes que intentan responderse sin un orden de prioridad claro. Fue por ello que se tomó la decisión de abordar las tres a pesar de que las funciones objetivo y los resultados derivados no son comparables entre sí.

Las preguntas por responder fueron las siguientes:

- (1) Si buscamos que todos los *sellers* sean asignados a un *place* ¿cuál es la distancia mínima total que deberán recorrer?
- (2) Si queremos maximizar la cantidad de paquetes que ingresan a la red pero asegurando un cierto estándar de experiencia para el *seller* en términos de distancia que deben recorrer ¿cuántos paquetes se incorporarán a la red? ¿qué *sellers* terminarán formando parte de la misma?
- (3) Teniendo en cuenta que recorrer cierta distancia tiene un costo para el *seller* y que abrir un *place* tiene un costo fijo de *instalación* ¿qué *places* sería conveniente abrir, qué *sellers* debería sumar a la red y a qué *places* los debería asignar buscando minimizar el costo?

Alumno: Juan Pablo Santos

Para abordar las respuestas a estas preguntas, en la siguiente sección detallaremos la notación que se usará, luego introduciremos cada uno de los problemas y los modelos asociados.

Notación de parámetros y variables

Sea $S = \{1, \dots, n\}$ el conjunto de *sellers* y $P = \{1, \dots, m\}$ el conjunto de *places*. Notamos con d_{ij} a la distancia (medida en kilómetros) que debe desplazarse un *seller* “ i ” perteneciente a S ” para acceder al *place* “ j ” perteneciente a P ”. Además, cada *seller* “ i ” perteneciente a S ” entrega q_i paquetes promedio por día, y para cada *place* “ j ” perteneciente a P ” conocemos su capacidad diaria de almacenamiento que llamamos c_j . Finalmente, definimos wd como la distancia máxima en kilómetros que la plataforma está dispuesta a permitir recorrer a un *seller* hasta un *place* para dejar sus productos.

Esquemáticamente entonces estos serán los parámetros del problema:

Notación	Descripción
S	Conjunto de <i>sellers</i> en una zona geográfica determinada
P	Conjunto de <i>places</i> en una zona geográfica determinada
wd	Constante en kilómetros que el <i>e-commerce</i> está dispuesto a que un <i>seller</i> camine para ir a despachar sus productos
q_i	Cantidad de paquetes promedio por día que despacha el <i>seller</i> $i \in S$
c_j	Capacidad en paquetes promedio por día que puede recibir el <i>place</i> $j \in P$
d_{ij}	Distancia en kilómetros del <i>seller</i> $i \in S$ al <i>place</i> $j \in P$

Adicionalmente, introduciremos la siguiente variable binaria x_{ij} al problema que tomará valor uno cuando el *seller* i sea asignado al *place* j y cero en caso contrario. Se remarca que no es posible que un *seller* particione su entrega entre múltiples *places*.

Notación	Descripción
x_{ij}	Variable binaria que toma valor uno cuando el <i>seller</i> $i \in S$ es asignado al <i>place</i> $j \in P$ y cero en caso contrario

Modelo (i): Minimizando la distancia recorrida

Claramente la pregunta (1) es la que refleja más fielmente lo que el *e-commerce* viene haciendo hasta este momento de manera manual y es por eso que sólo el modelo (i) es el que usaremos como *baseline* comparativo contra la solución presente de la plataforma.

Alumno: Juan Pablo Santos

Bajo este modelo, el *decision maker* busca minimizar la distancia total que los *sellers* tienen que recorrer para llegar al *place* asegurándose que todos tengan un lugar donde dejar sus productos y que los *places* no colapsen su capacidad⁷.

Formalmente:

$$\text{Min } \sum_{i \in S} \sum_{j \in P} d_{ij} x_{ij}$$

Sujeto a:

1. Todo *seller* tiene asignado un *place* donde dejar sus paquetes:

$$\sum_{j \in P} x_{ij} = 1 \quad \forall i \in S$$

Si bien de acuerdo al estudio previo que hicimos de los datos, sabemos que podemos garantizar que todos los *sellers* pueden ser atendidos, en caso de que la cantidad de paquetes promedio sea mayor a la capacidad de la red en su conjunto, podríamos balancear el problema incorporando un *place* ficticio que captura la demanda desatendida.

2. La suma de los envíos de los *sellers* asignados a cada *place*, no supera la capacidad total del *place*:

$$\sum_{i \in S} q_i x_{ij} \leq c_j \quad \forall j \in P$$

3. La variable x_{ij} es binaria:

$$x_{ij} \in \{0, 1\} \quad \forall i \in S; \forall j \in P$$

Como resultado de este problema sabremos cuál es la distancia total recorrida por los *sellers*, qué *seller* será asignado a qué *place* y cuánto deberá trasladarse cada *seller* dado el *place* que le fue asignado. Como se puede observar, este es un clásico modelo de GAPs.

Modelo (ii): Maximizando la cantidad de paquetes que entran a la red

¿Qué pasa si por un momento dejamos de lado la variable distancia y nos paramos desde una posición mucho más ambiciosa? Sabemos que hay un target específico que queremos incorporar a nuestra red y nos interesa atraer la mayor cantidad de este volumen.

⁷ Tal suposición tiene sentido sí sólo si (1) los *sellers* hacen viajes por separado a los *places*, (2) las características de los *places* (como los tiempos de espera, calidad de atención, etc) son todas iguales y (3) los *sellers* son planificadores racionales (es decir, maximizan la utilidad) con funciones de utilidad que incluyen solo los costos de transporte, que se supone que son proporcionales a las distancias entre el *seller* y el *place*.

Alumno: Juan Pablo Santos

Su oferta es única, nos ayuda a diversificar y no sólo eso sino que también podemos extraer una mayor rentabilidad dado que su poder de negociación es menor. Si nos despreocupamos completamente de la distancia que tienen que recorrer los *sellers*, sabemos que la capacidad disponible existe (al menos en este set de datos), con lo cual, el problema se vuelve trivial. Ahora, esta idea en sí misma es muy *naive*, no podemos pretender traer todos los *sellers* a la red y asumir que si los obligamos a que se trasladen más de cierta distancia lo harán, lo más probable es que algunos optarán por dejar de vender. Es por esto que es importante también contemplar ciertos estándares mínimos de experiencia para los vendedores. ¿Qué define esos estándares? No tenemos una respuesta cuantitativa pero, de todos modos, sabemos que a menor distancia de traslado, menor es el costo para el vendedor y por lo tanto mejor su experiencia. Este modelo busca explorar el impacto de imponer condiciones en esta dirección y evaluar su impacto en la solución del problema.

En definitiva, bajo este modelo, el *decision maker* busca maximizar el tamaño de la red intentando incorporar la mayor cantidad de paquetes posible, pero guardando ciertos estándares mínimos de experiencia de cara a los *sellers* en términos de la distancia que deben recorrer (*wd*) al momento de dejar sus paquetes.

Formalmente:

$$\text{Max} \sum_{i \in S} \sum_{j \in P} q_i x_{ij}$$

Sujeto a:

1. Todo *seller* puede ser asignado a un *place* donde dejar sus paquetes:

$$\sum_{j \in P} x_{ij} \leq 1 \quad \forall i \in S$$

2. La suma de los envíos de los *sellers* asignados a cada *place*, no superan la capacidad total del *place*:

$$\sum_{i \in S} q_i x_{ij} \leq c_j \quad \forall j \in P$$

3. La distancia que debe caminar un *seller* a un *places* no será superior a un valor target objetivo:

$$x_{ij} \times d_{ij} \leq wd \quad \forall i \in S; \forall j \in P$$

Vale mencionar que esta restricción podría ser evitada realizando un pre-procesamiento de la instancia. Los softwares con los que trabajaremos realizan esta tarea, con lo cual, mantenemos la restricción ya que ayuda a un mejor entendimiento del problema.

4. La variable x_{ij} es binaria:

$$x_{ij} \in \{0, 1\} \quad \forall i \in S; \forall j \in P$$

Alumno: Juan Pablo Santos

Como resultado de este problema sabremos cuál es la cantidad de paquetes máxima que el modelo incorpora a la red sujeto a un umbral máximo de distancia recorrida que nos permitirá garantizar cierta experiencia al *seller*.

Modelo (iii): ¿Qué places conviene abrir y qué sellers incorporamos a la red?

Imaginemos por un segundo que la red de *sellers* y *places* aún no existe, tenemos un conjunto de potenciales lugares donde abrir los *places* y un conjunto de *sellers* que nos gustaría traer a la red. Con esta motivación hay dos preguntas triviales que surgen:

1. ¿Cuánto sale abrir un *place*?
2. ¿Qué costo tiene para el *seller* ir a dejar sus ventas al *place*?

Independientemente de la respuesta, claramente lo que se busca es minimizar estos costos, abrir la menor cantidad de *places* asignando todos los *sellers* y minimizar el costo de traslado.

Ahora dado que la red existe, ¿por qué este problema es interesante de todos modos? Porque la red actual de la plataforma se encuentra en un estado inicial y en un solo país. Sin embargo, esta idea podría expandirse a otros estados/provincias u otros países también. Por lo tanto, contar con la formalización de este modelo, podría funcionar para potenciales nuevas aperturas en otras zonas geográficas.

En definitiva, bajo este modelo, el *decision maker* cuenta con un pool de *places* que puede decidir si abrir o no teniendo en cuenta que la apertura tiene un costo. Adicionalmente, también contempla que el traslado del *seller* desde su domicilio al *place* posee un costo monetario. Su objetivo entonces es minimizar los costos totales.

Introduciremos la siguientes variable y parámetros al problema:

Notación	Descripción
z_s	Parámetro que representa el costo monetario que asume el <i>seller</i> al trasladarse desde su domicilio al <i>place</i>
z_p	Parámetro que representa el costo monetario que implica abrir un <i>place</i>

Adicionalmente, introduciremos la siguiente variable al problema:

Notación	Descripción
y_j	Variable binaria que toma valor uno cuando el <i>place</i> $j \in P$ será abierto y cero en caso contrario

Alumno: Juan Pablo Santos

Formalmente:

$$\text{Min } \sum_{i \in S} \sum_{j \in P} d_{ij} x_{ij} z_s + \sum_{j \in P} y_j z_p$$

Sujeto a:

1. Todo *seller* tiene asignado un *place* donde dejar sus paquetes:

$$\sum_{j \in P} x_{ij} = 1 \quad \forall i \in S$$

Como se explicó en el modelo (i), esta restricción la podemos poner de esta manera ya que sabemos que nuestra base de datos lo permite. En caso contrario, podríamos agregar un *place* ficticio con una capacidad lo suficientemente grande como para poder asignar los envíos que nos permitan balancear el problema.

2. Ligamos las variables x_{ij} a la apertura de los *places*, es decir, si no tiene un *seller* asignado, no tiene sentido que se abra el *place*:

$$x_{ij} \leq y_j \quad \forall i \in S, \forall j \in P$$

3. La suma de los envíos de los *sellers* asignados a cada *place*, no superan la capacidad total del *place*:

$$\sum_{i \in S} q_i x_{ij} \leq c_j \quad \forall j \in P$$

Otra forma de representar lo mismo sería la siguiente:

$$\sum_{i \in S} q_i x_{ij} \leq c_j y_j \quad \forall j \in P$$

De esta manera, si el *place* no es elegido, forzamos a que ningún *seller* tenga que ir a despachar allí.

4. Las variables x_{ij} y y_j son binarias y los parámetros sólo toman valores positivos (esto último son *inputs* que no se incorporan al modelo):

$$x_{ij} \in \{0, 1\} \quad \forall i \in S; \forall j \in P$$

En busca de las soluciones a los problemas

En esta sección nos focalizamos en la solución a los modelos planteados con especial atención en el modelo (i) dado que es el que mejor refleja la solución planteada desde el *e-commerce*.

Para la búsqueda de las soluciones usaremos el software SCIP⁸ para el modelo (i) y (ii) y CPLEX para el modelo (iii). Los scripts correspondientes quedarán como anexos.

⁸ Para más información véase: <https://scip.zib.de/>

Solución al modelo (i)

Recordemos que el modelo (i) lo que busca es minimizar la distancia total que recorren los *sellers* a los *places* asignando todos los *sellers* a algún lugar y usando los *places* disponibles (puede que no sean todos). Se harán dos versiones del modelo, una contemplando las capacidades declaradas y otra aumentando esas capacidades en un 50% siguiendo la solución *greedy* recomendada por los *decision makers* de la compañía.

Sigue a continuación los resultados de la solución del modelo (i) con capacidades reales:

Resumen solución problema (i.a) versión capacidades reales

# Places	# Sellers	# Shipments	Sum Distance	Sum Real Capacity
299	1,022	10,319.43	599.21 kms	18,592

La primera columna (# Places) representa la cantidad de *places* que terminó usando el modelo. La segunda (# Sellers) muestra la cantidad de *sellers* asignados, en este caso son todos dado que fue una restricción que le impusimos al modelo. La tercera columna (# Shipments) contiene la cantidad de envíos de los *sellers* asignados. La cuarta (Sum Shipments) la suma de distancia total recorrida por los *sellers* y la última (Sum Real Capacity) la capacidad total de los *places* que el modelo determinó que usaría.

En lo que respecta a la performance del algoritmo, el tiempo total de resolución fue de 91.41 segundos, el problema cuenta con 316,820 variables y 1,332 restricciones y se logró llegar a 14 soluciones factibles y un óptimo⁹.

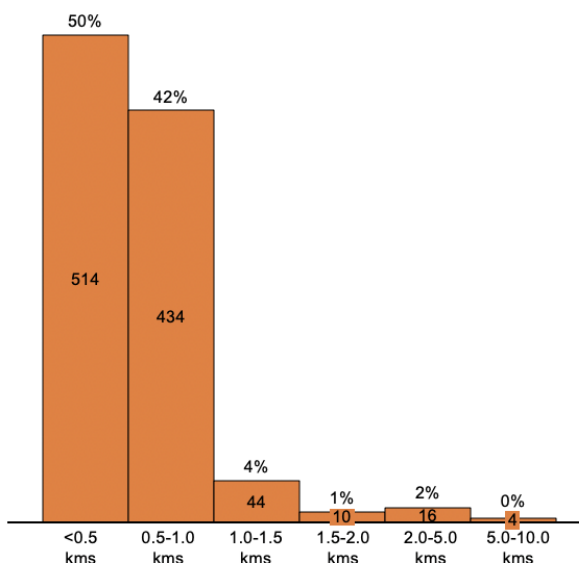
Como se puede observar, la solución encontrada por el algoritmo aún sin expandir capacidades es superadora a la solución del *e-commerce*. La distancia total recorrida es menos de la mitad, no requiere la utilización de todos los *places*, los 1,022 *sellers* forman parte de la red por lo que todos los envíos son incorporados. El *seller* que más se traslada tiene que moverse 9.18 kilómetros y el promedio de desplazamiento es de 600 metros con

⁹ Se testeó el modelo con 10 instancias benchmark, 5 generadas con muestras aleatorias equivalentes al 10% de la instancia total (10 *sellers* y 3 *places*) llegando también a resultados óptimos aunque en algunos casos teniendo que balancear el problema pues la capacidad de los *places* no llegaba a suplir la cantidad de paquetes que los *sellers* pretendían introducir en la red. Adicionalmente, se testeó el modelo con cinco instancias GAP presentes en el siguiente sitio <http://astarte.csr.unibo.it/gapdata/gapinstances.html> (id's: bogota, elba, ForliCesenaGAP, galway y GAP40x5a), llegando también el óptimo. Se incorporan estos chequeos de robustez en el Anexo III.

Alumno: Juan Pablo Santos

un desvío de 622 metros. Adicionalmente, como se puede apreciar en el histograma del gráfico 4, el 92% de los *sellers* se traslada menos de un kilómetro.

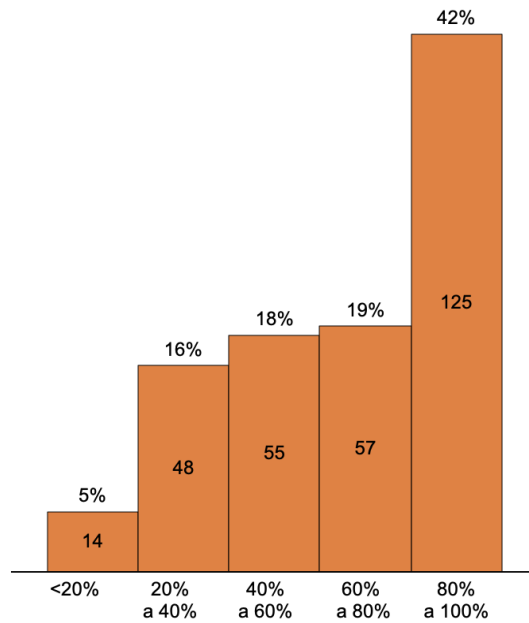
Gráfico 5: Histograma de traslados de los *sellers* a los *places* de acuerdo a la solución (1.a)



En cuanto a la utilización de la capacidad de los *places*, se observa que un 42% tiene una utilización de su capacidad entre 80%-100% muy superior a la solución manual del *e-commerce*. A su vez, el comportamiento es similar entre los rangos 20%-40%, 40%-60% y 60%-80%. Por otro lado, la ocupación promedio es del 66%. Podemos afirmar entonces que aún asignando todos los *sellers*, hay capacidad ociosa suficiente como para incorporar a la red más usuarios. Bajo esta perspectiva es cuestionable la decisión del *e-commerce* de demandar más capacidad a los *places*. Esta solución demuestra que se puede hacer un mejor uso de la capacidad, minimizar la distancia recorrida por los *sellers* y no tener necesidad de destinar un costo extra a más capacidad.

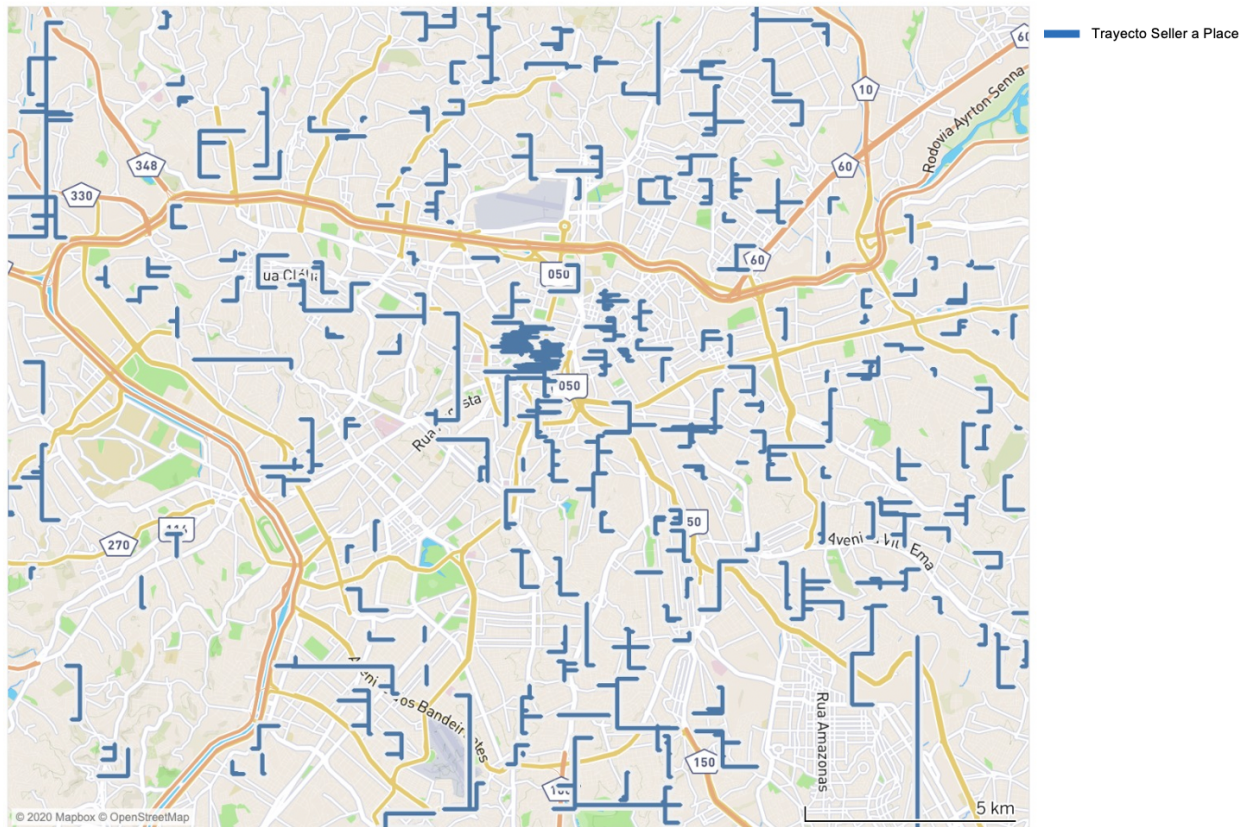
Alumno: Juan Pablo Santos

Gráfico 6: Solución modelo (1.a), histograma de utilización de la capacidad de los *places*



Alumno: Juan Pablo Santos

Mapa 3: Visualización solución modelo (i.a)



El Mapa 3 visualiza la solución; como se puede apreciar, versus la solución del *e-commerce*, los traslados son más cortos en la gran mayoría de los casos.

Al expandir las capacidades de los *places* en un 50%, como era de esperar, el modelo hace uso de esto y logramos optimizar aún más la distancia total como se puede ver en la siguiente tabla resumen:

Resumen solución problema (i.b) versión capacidades extendidas

# Places	# Sellers	# Shipments	Sum Distance	Sum Real Capacity
294	1,022	10,319.43	558.82 kms	27,678

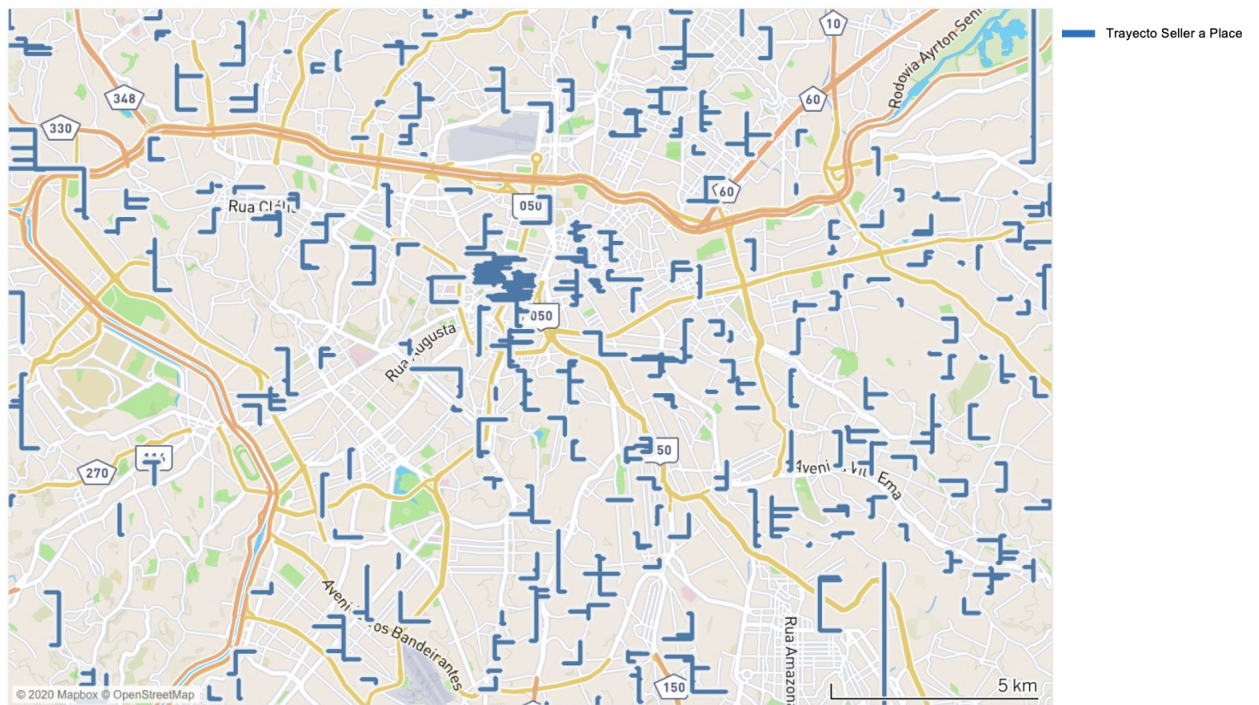
En lo que respecta a la performance del algoritmo, el tiempo total de resolución fue de 24.35 segundos, el problema cuenta con 316,820 variables y 1.332 restricciones y se logró llegar a 9 soluciones factibles y un óptimo.

Alumno: Juan Pablo Santos

Si bien se observa una mejora en la distancia total, la misma es solo de un 6.74% comparada con el modelo (i.a) que no contempla negociaciones en las capacidades totales. Esta comparación sería la más “justa” versus la solución a la que llegaron los *decision makers* del *e-commerce*. De todos modos, el modelo anterior, nos permite darle menos importancia a esta opción ya que la mejora en distancia lograda con el modelo sin amplificación de capacidades es lo suficientemente importante como para evitar destinar esfuerzos comerciales en negociaciones con todos los *places* de la red.

Al igual que con el modelo anterior, llevamos la solución a un mapa que luce bastante similar al del modelo 1.a.

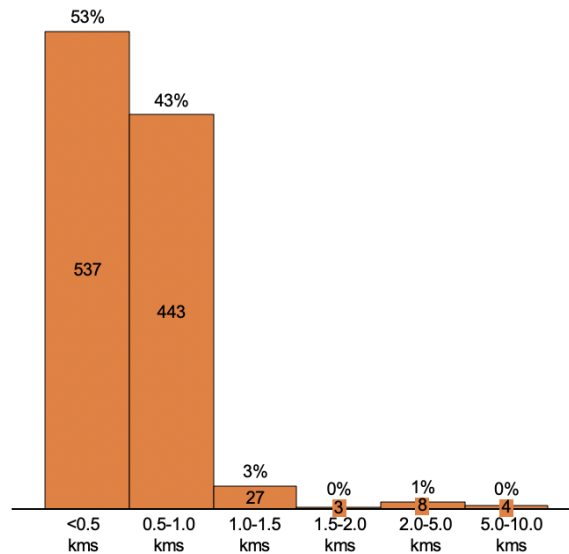
Mapa 4: Visualización solución modelo (i.b)



Algo muy similar sucede con el histograma de traslados aunque el número de *sellers* que se trasladan al menos un kilómetro mejora a 96%.

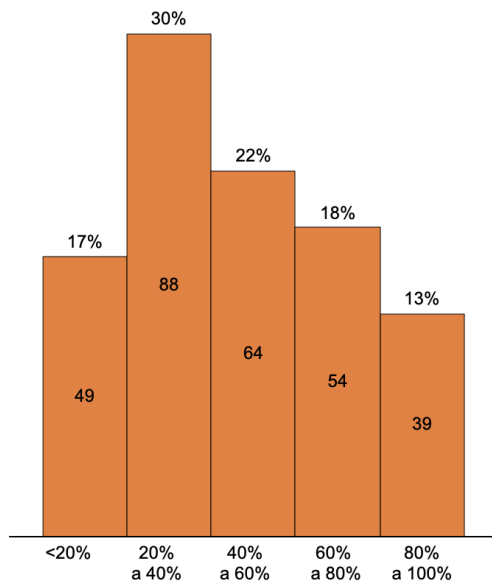
Alumno: Juan Pablo Santos

Gráfico 7: Histograma de traslados de los *sellers* a los *places* de acuerdo a la solución del modelo (1.b)



En cuanto a la capacidad de los *places*, se observa una clara subutilización ya que un 30% tiene una utilización de su capacidad extendida entre 20%-40%. A su vez, el comportamiento es similar entre los rangos menores a 20%, 40%-60% y 60%-80% y solo un 13% cae en el rango entre 80%-100%. Por otro lado, la ocupación promedio es sólo del 47%. Estos números refuerzan aún más que realizar esfuerzos comerciales para ampliar la capacidad declarada de los *places* no parece ser una buena estrategia.

Gráfico 8: Solución modelo (1.b), histograma de utilización de la capacidad de los *places*



Finalmente, se hizo un análisis para entender cuán distinta es la asignación de ambas versiones del modelo versus la solución del *e-commerce*. La idea es entender cómo cambiaría la topología de la red si la plataforma decidiera aplicar nuestro algoritmo. Los resultados fueron los siguientes:

Resumen diferencia en asignaciones *e-commerce* vs modelo (1.a) y (1.b)

# Asignaciones	# Matches <i>e-commerce</i> vs modelo (1.a)	% Matches <i>e-commerce</i> vs modelo (1.a)	# Matches <i>e-commerce</i> vs modelo (1.b)	% Matches <i>e-commerce</i> vs modelo (1.b)
1022	773	75.64%	798	78.08%

Como se puede observar, al comparar la asignación hecha por el *e-commerce* versus el modelo (1.a), hay una coincidencia en el 75.64% de las asignaciones *seller-place* mientras que en el modelo (1.b) este número aumenta a 78.08%. Se deduce de estos resultados que aplicar la asignación del algoritmo implica cambios para el *e-commerce* pero que llevarlos a cabo, generaría una mejora sustancial en la experiencia de esos *sellers* y una reducción a la mitad (o más) de la distancia total recorrida. Si bien sabemos que la reasignación tiene un costo no muy significativo asociado pero un costo al fin, es bueno saber que no implica cambiar radicalmente la topología actual de la red y que con pequeños ajustes, la mejora conjunta no es despreciable.

Alumno: Juan Pablo Santos

Se concluye la sección con un cuadro resumen de las dos versiones del modelo en comparación con la solución del *e-commerce* donde se puede ver la cantidad de *places* usados por cada modelo, los *sellers* asignados y su volumen, la distancia total recorrida y la suma de la capacidad de los *places* usados.

Cuadro comparativo de las soluciones vs la solución del *e-commerce*

	# Places	# Sellers	# Shipments	Sum Distance	Sum Capacity
<i>e-commerce</i>	310	1,022	10,319.43	1,215 kms	19,102
(1.a)	299	=	=	599.21 kms	18,592
(1.b)	294	=	=	558.82 kms	27,678

Solución al modelo (ii)

Como se describió anteriormente, el objetivo del modelo (ii) es maximizar la cantidad de paquetes que ingresan a la red procurando garantizar cierta experiencia de cara a los *sellers*. Esta “experiencia” vamos a garantizarla limitando la cantidad de metros que un *seller* tiene que trasladarse (*wd* en el modelo). Esto claramente genera un *trade-off* entre aumentar el tamaño de la red y asegurar una propuesta de valor que el *seller* esté dispuesto a adoptar. Para entender mejor esta disyuntiva, probaremos diferentes valores mínimos de traslado: 300 metros, 600 metros, 1.2 kilómetros y 10 kilómetros. Podemos ver los resultados en la siguiente tabla resumen:

Resumen solución modelo (ii)

Valor máx de traslado (<i>wd</i>)	300 metros		600 metros		1200 metros		10000 metros	
	# cant.	% del total	# cant.	% del total	# cant.	% del total	# cant.	% del total
Sellers	253	25%	657	64%	985	96%	1022	100%
Places	151	49%	238	77%	274	88%	260	84%
Shipments	2601.59	25%	6634.35	64%	9898.11	96%	10319.43	100%

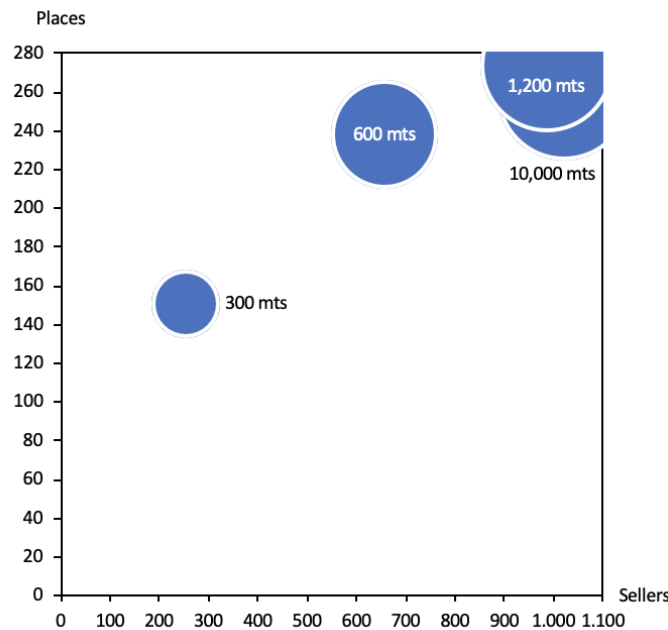
La primera fila representa los diferentes valores de *walking distance* (*wd*) que se fueron probando y están ordenados en forma ascendente hasta llegar al valor máximo de traslado. La primera columna (# cant.) en cada escenario describe en sus filas la cantidad de *sellers* que el modelo atrae a la red, la cantidad de *places* y la cantidad de envíos. La segunda (%)

Alumno: Juan Pablo Santos

del total) enumera en sus filas el porcentaje que representan esos *sellers* designados, el porcentaje de los *places* sobre el total y la cantidad de envíos sobre el total potencial.

Como se puede observar, mientras más relajamos la restricción de la distancia, más *sellers* captamos a la red y por lo tanto, mayor es el porcentaje sobre el volumen total que logramos incorporar. Con una restricción de 1200 metros llegamos al 96% de los *sellers* y del volumen usando el 88% de los *places* disponibles. Si abrir un *place* tuviera algún tipo de costo asociado tener en cuenta este hallazgo puede ser muy importante. Gráficamente luce de la siguiente manera:

Gráfico 9: Variación sobre la variable *walking distance* (*wd*) y su impacto en la captación de volumen



El eje de las x's representa la cantidad de *sellers* que capta cada variación del modelo, el eje de las y's la cantidad de *places* que utiliza y el tamaño de la burbuja es la cantidad de volumen que cada variación traería a la red.

En lo que respecta a la distancia promedio recorrida, para la variante de 300 metros es de 196 metros, para la de 600 metros es de 371 metros, para la de 1,200 metros es de 678 metros y para la de 10,000 metros es de 5.88 kilómetros. Esto de algún modo demuestra que al maximizar la cantidad de paquetes que nos gustaría que ingresen a la red, cuidar la experiencia de los *sellers* es importante.

Algo que llama la atención es que la cantidad de *places* que usa el modelo al modificar la restricción de la distancia recorrida no es lineal. En el caso de 1,200 metros, la cantidad de *places* usados es mayor al caso de 10,000 metros. Al entrar más en detalle poniendo el

Alumno: Juan Pablo Santos

foco principalmente en las variantes de 1,200 y 10,000 metros notamos que el modelo no selecciona los mismos *places*. Hay una coincidencia en 230 *places* pero hay una divergencia en los restantes. Esto está asociado a que dado que en el modelo de 10,000 metros nos preocupa menos la experiencia de los *sellers*, puede a su vez, ocupar menos *places*.

En lo que respecta a la performance del algoritmo para las cuatro variaciones del parámetro *walking distance*, el tiempo total de resolución fue de 9.12 segundos, para el caso de 300 metros fue de 9.45 segundos para el caso de 600 metros, 9.50 segundos para el caso de 1200 metros y 37.05 segundos para el caso de 10000 metros. Para las cuatro variantes del problema, la cantidad de variables fue de 316,820 y la cantidad de restricciones fue de 318,152. Remarcamos que muchas de estas variables posiblemente han sido pre-procesadas por el paquete de optimización.

Un hallazgo importante de este modelo es que el *gap* en cantidad de *sellers* y *shipments* que se incorporan a la red de la variante 1,200 metros versus la variante 10,000 metros (variante total) es sólo del 4%. Podemos interpretar de este resultado que la red en términos de experiencia es robusta y que, quizás, hay una oportunidad de incorporar nuevos *places* más cercanos a aquellos *sellers* que el modelo de 1,200 metros dejó afuera.

Solución al modelo (iii)

Recordemos que el objetivo del modelo (iii) es elegir qué *places* formarán parte de la red teniendo en cuenta que la incorporación de los mismos tiene un costo asociado y también, qué *sellers* conviene sumar a la red contemplando que su traslado al *place* implica un costo. En definitiva, se trata de un problema que puede enmarcarse dentro de lo que Ross y Soland (1977) llaman *facility location*. Se buscará entonces minimizar el costo total vinculado a la existencia de la red, garantizando que todos los *sellers* formen parte de la misma.

Para modelar el costo de desplazamiento del *seller* al *place* (z_s) usaremos la distancia existente entre estos dos puntos llevada a un valor monetario equivalente (e.g. 3 kilómetros de desplazamiento, equivalen a 3 unidades monetarias de costo). Se asume entonces que el costo es lineal y que a mayor distancia mayor es el mismo. Los *decision makers* del *e-commerce* encontraron este *proxy* razonable y representativo.

En lo que respecta al costo de apertura de un *place* (z_p) por temas de confidencialidad esta información no fue entregada. De todos modos, se mencionó que este parámetro depende en gran medida de la capacidad de cada *place*: los más grandes tienen un costo de apertura mayor. Esto se debe a que el espacio físico que se tiene que destinar para recibir los paquetes es más grande, la cantidad de personas que se debe instruir en el uso del aplicativo que recibe los paquetes es mayor, como así también las medidas de seguridad asociadas al resguardo de la mercadería. Por esto, la variable elegida para

Alumno: Juan Pablo Santos

representar el costo de apertura fue la capacidad declarada del *place* llevada a un valor monetario (e.g. 300 paquetes de capacidad, equivale a 300 unidades monetarias de costo).

Para la resolución del problema en una primera instancia se intentó usar el *software* SCIP Op Suite 7.0.0 Darwin. Luego de mantener el algoritmo corriendo por más de 10 horas, el *gap* versus el óptimo nunca bajó del 15%. Fue por esta razón que se recurrió a IBM ILOG CPLEX Optimization Studio Developer Edition 12.8 (5725-A06). Si bien no se llegó al óptimo, luego de 113 minutos, el *gap* resultante fue de 2.37%.

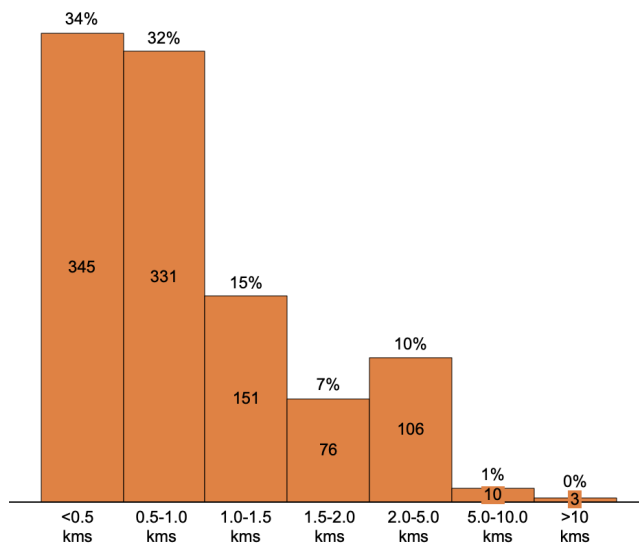
En cuanto a la solución no óptima, sigue a continuación una tabla resumen:

Resumen solución problema (iii)

# Places	# Sellers	# Shipments	Best Solution ¹⁰	Gap	Sum Distance	Capacity Allocated	% of total potential capacity
201	1,022	10,319.43	11,324.84	2.37%	1,438 kms	10,525	55.1%

Como se puede observar, de los 310 *places* disponibles, el modelo sólo usa 202 para asignar a todos los *sellers* (i.e. un 55% de la capacidad potencial). Esto genera un impacto directo en la distancia que tienen que recorrer algunos *sellers* como se puede ver en el histograma del gráfico 10 donde sólo el 66% tiene que desplazarse un kilómetro o menos. La distancia se parece (incluso excede) a la solución original manual, pero se usan sólo 201 *places* en lugar de 310.

Gráfico 10: Histograma de traslados de los *sellers* a los *places* modelo (iii)



¹⁰ Recordemos que este valor se encuentra en unidades monetarias

Alumno: Juan Pablo Santos

Una interpretación posible de estos resultados es que el modelo está priorizando abrir la menor cantidad de *places* relegando experiencia de *sellers* ya que el costo de abrir uno es mucho mayor comparado a que los *sellers* se trasladen más kilómetros. Esta hipótesis se reafirma al analizar la utilización de los *places* seleccionados por el modelo. La utilización promedio es del 97% con un desvío estándar de 3 puntos porcentuales. Otra manera de ver esto mismo es analizando la capacidad total de los *places* seleccionados; los 10,319.43 envíos por día promedio que hacen estos *sellers* son asignados a *places* con una capacidad total de 10,525.

Para entender cómo varía la cantidad de *places* utilizados por el modelo, se hizo un ejercicio multiplicando por diez el costo por kilómetro de traslado de los *sellers*. Esto quiere decir que si el *seller* se tiene que trasladar un kilómetro, este traslado se incorpora al experimento con un costo de diez unidades. Sigue a continuación los resultados obtenidos:

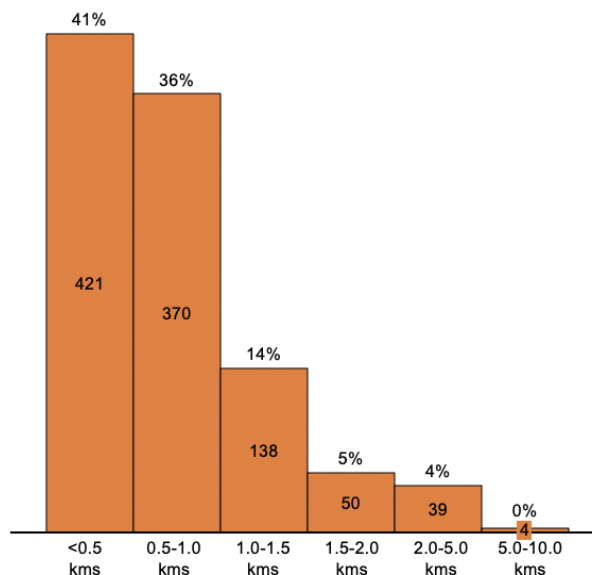
Resumen solución problema (iii) multiplicando por diez el costo de traslado

# Places	# Sellers	# Shipments	Best Solution	Gap	Sum Distance	Allocated Capacity	% of total potential capacity
218	1,022	10,319.43	18,985.64	0.45%	788.57 kms	11,185	58.6%

Para la resolución del problema se usó directamente el *software* CPLEX. Si bien no se llegó al óptimo, luego de 34 minutos, el *gap* resultante fue de 0.45%, mucho mejor que el modelo anterior en cuanto a *performance*.

Siguiendo con la interpretación de los resultados obtenidos, como se puede observar, de los 310 *places* disponibles, el modelo usa 218 (i.e. un ~70% de los *places* pero un 58.6% de la capacidad total de la red); 16 *places* más que el anterior. Esto genera un impacto directo en la distancia total a recorrer que se reduce en un 45% versus el anterior modelo. Si bien es trivial porque es una restricción que pusimos, se comprueba que nuevamente todos los *sellers* son asignados.

Gráfico 11: Histograma de traslados de los *sellers* a los *places* modelo (iii) multiplicando por 10 el costo de traslado



El histograma de traslados del gráfico 11, nos muestra que el 77% de los *sellers* deben desplazarse al menos un kilómetro, 11 puntos porcentuales más que la anterior versión del modelo. Esto podríamos interpretarlo como que el costo del traslado en el modelo anterior era muy “barato” y que al incrementarse, el modelo prioriza más la experiencia de los *sellers*.

Finalmente, en lo que respecta a la ocupación promedio, la misma es de un 91%, 6 puntos porcentuales menos que el modelo anterior. El desvío estándar también aumenta considerablemente a 11 puntos porcentuales y nos encontramos con *places* donde sólo se utiliza un 33% de su capacidad declarada.

Si bien los resultados de este modelo son meramente ejercicios pues no contamos con datos reales sino con proxies, queda en evidencia que la magnitud relativa de los costos influye fuertemente en la toma de decisión. Formalmente podríamos modelar este hallazgo incorporando a la función objetivo del modelo una ponderación de los costos de la siguiente manera:

$$\text{Min } a \sum_{i \in S} \sum_{j \in P} d_{ij} x_{ij} z_s + (1 - a) \sum_{j \in P} y_j z_p$$

donde a representa la importancia relativa que tiene el costo de traslado de los *sellers* sobre el costo de abrir nuevos *places*.

Al igual que con el modelo (i), concluimos esta sección con un cuadro resumen de ambas variantes del modelo:

Alumno: Juan Pablo Santos

Cuadro comparativo de las dos variantes del modelo (iii)

	# Places Used	# Shipments	Best Solution	Gap	Sum Distance	Allocated Capacity	% of total potential capacity
Modelo (iii)	201	10,319.43	11,324.84	2.37%	1,438 kms	10,525	55.1%
Modelo (iii) x10	218	10,319.43	18,985.64	0.45%	788.57 kms	11,185	58.6%
Vs	+8.46%	-	-	-1.92pp	-45.16%	6.27%	-3.5pp

Detalle computacional de los modelos

El objetivo de esta sección es realizar un compendio técnico de los modelos, detallando cómo los mismos fueron desarrollados, cuáles son sus tiempos de respuesta y los *softwares* usados.

Sigue a continuación una tabla resumen con esta información:

Cuadro resumen con detalles técnicos de los modelos

	Software	Optimal Solution	Solving Time
Modelo (i.a)	SCIP	SI	91.41 secs
Modelo (i.b)	SCIP	SI	24.35 secs
Modelo (ii) 300 m	SCIP	SI	9.12 secs
Modelo (ii) 600 m	SCIP	SI	9.45 secs
Modelo (ii) 1.2 km	SCIP	SI	9.50 secs
Modelo (ii) 10 km	SCIP	SI	37.05 secs
Modelo (iii)	CPLEX	NO	113 mins
Modelo (iii) x10 wd	CPLEX	NO	34 mins

En todos los casos primero se intentó resolver el problema con SCIP Op Suite 7.0.0 Darwin y cuando no se llegó a un buen resultado, vía ZIMPL se pasó el modelo a lenguaje legible por IBM ILOG CPLEX Optimization Studio Developer Edition 12.8 (5725-A06).

Los modelos se ejecutaron en la terminal de comandos pues se tratan sólo de 8 instancias y no era el objetivo de esta investigación realizar simulaciones sino buscar respuestas a inquietudes de negocio manifestadas por los *decision makers* de la plataforma con los mismos datos. Es importante mencionar también que todos los algoritmos se

Alumno: Juan Pablo Santos

ejecutaron en la misma máquina MacBook Pro (13-inch, 2019, Two Thunderbolt 3 ports) procesador 1.7 GHz Quad-Core Intel Core i7 y memoria de 16 GB 2133 MHz LPDDR3.

Los scripts en lenguaje SCIP quedan como anexo al final del presente trabajo.

Conclusiones y posibles extensiones

Resumen de los resultados

El presente trabajo presenta tres modelos que buscan responder a diferentes necesidades de negocio de un *e-commerce* muy importante en latinoamérica. Se introducen dos actores: los *sellers* y los *places*. Los *sellers* son vendedores con ciertas características que exponen oferta en la plataforma. Los *places* son lugares físicos donde estos *sellers* deben ir a dejar las ventas que hacen a través del *e-commerce*. Una vez que el *place* recibe estas ventas, vía diferentes canales logísticos, los paquetes llegan al usuario comprador.

El primer modelo busca minimizar la distancia total de traslado de los *sellers* a los *places* asegurando que todos los *sellers* sean asignados a un *place* controlado por la capacidad declarada de los *places*. Luego se expande esa capacidad en un 50%. Ambos resultados se comparan contra la asignación manual hecha por la plataforma demostrando que el modelo en sus dos variantes logra reducir significativamente la distancia total recorrida obteniendo una utilización más eficiente de la red. Se observa entonces que la potencial negociación por el lado del *e-commerce* para aumentar la capacidad de los *places* no es necesaria pues la capacidad original de la red es suficiente.

El segundo modelo tiene un enfoque diferente, busca maximizar la cantidad de paquetes que ingresan en la red pero controlado por la distancia que tiene que trasladarse un *seller*. Se prueban diferentes umbrales de distancia máxima recorrida por cada *seller*. Mientras mayor es el umbral, mayor es la cantidad de *sellers* y paquetes que ingresan en la red.

El tercer modelo aborda un problema de naturaleza distinta. Al cuantificar económicamente el traslado de los *sellers* y la apertura de los *places*, no sólo busca responder qué *sellers* deben ir a qué *places* sino también qué *places* conviene abrir. Si bien no se llega al óptimo en este tercer modelo, el resultado ofrece una interesante intuición de negocio sobre el problema. Se incorpora también un experimento que nos permite entender la importancia relativa de los costos sobre la experiencia de los *sellers* y se sugiere una manera de incorporar esto al modelo original.

Salvo el tercer modelo, los tiempos de respuesta de los algoritmos fueron más que razonables lo cual es importante para permitir una aplicación de los mismos por el lado del *e-commerce*.

Alumno: Juan Pablo Santos

Se sugiere a los *decision makers* del *e-commerce* tres cambios en su proceso actual: (1) Reasignar un porcentaje de los *sellers* actuales para mejorar la distancia total recorrida, la potencial mejora de un 50% es significativa y el costo de ejecutar esto es bajo. (2) Cambiar la metodología actual e incorporar el modelo (i) o (ii) a su proceso de asignación rutinario de nuevos *sellers* a la red. (3) Considerar el modelo (iii) en el proceso de diseño de la red para incorporar los *places* más costo-efectivos.

Posibles extensiones

Respecto a las extensiones que se pueden hacer a los modelos, dividiremos esta sección en tres: la metodología, los datos y los modelos.

Extensiones vinculadas herramientas y procesos

La interfaz de ejecución de los modelos en el presente trabajo tiene muchas oportunidades de mejora. Los mismos fueron ejecutados en la terminal de comandos de un computador exportando los resultados obtenidos a una planilla de cálculo. Esto claramente deja en evidencia que la experiencia del usuario no fue priorizada en absoluto pudiendo generar una barrera enorme al momento de buscar la adopción de esta solución. Desarrollar una *user interface* donde el usuario suba la información, se le pregunte qué modelo quiere ejecutar, qué restricción y de qué manera quiere que sean tenidas en cuenta y con qué parámetros es sumamente importante. Lo mismo aplica para la forma en la que se le expone la solución encontrada. Recordemos que el usuario que se encarga de llevar esto a la realidad cumple un rol comercial y operacional dentro de la compañía, con lo cual, no está familiarizado con un modelo de optimización lineal ni nada por el estilo: él o ella conoce del problema y quiere una solución que le permita tomar decisiones, probar asignaciones y seguir expandiendo este modelo de negocio a otras regiones, cuestiones técnicas no le interesan en absoluto.

Adicionalmente, es importante tener en cuenta que los resultados del modelo (i) sugieren realizar reasignaciones de *sellers* a *places* con el objetivo de minimizar la distancia total recorrida. Si bien el porcentaje que sugiere reasignar el modelo no es alto y según lo declarado por el *e-commerce* el costo no es elevado, puede pasar que al probar con un set de datos diferentes, las reasignaciones sí sean significativas. Una posible extensión al modelo sería incorporar esto como una restricción, es decir, de lo que existe actualmente sólo reasignar una parte evitando de esta forma una mayor fricción en la implementación optimizada y buscando quizá priorizar a aquellos *sellers* donde el resultado de la reasignación genere una mejora sustancial versus su situación actual.

Extensiones vinculadas a los datos

Si bien la información compartida por el *e-commerce* es más que razonable, hay tres mejoras en la calidad de la información que podrían ayudar a una mayor precisión de los modelos.

En primer lugar, hablaremos de las ventas promedio por día de los *sellers*. Dado que los *places* y los *sellers* operan en días laborables, los fines de semana generan una acumulación de ventas que implican que las entregas que los *sellers* hacen en los *places* los días lunes son más elevadas versus el resto de los días de la semana. Teniendo eso en cuenta, puede que haya algunos *places* que los días lunes no den abasto con su capacidad declarada dada la asignación hecha por los modelos que contemplan ventas promedio por día y no la variabilidad semanal. Una manera de soslayar este efecto sería incorporar al modelo el desvío estándar o considerar el día que el *seller* despacha más paquetes para asegurarnos que nunca estará colapsado el punto de despacho. Lamentablemente, al no contar con esta información, no se pudo incorporar la mejora. De todos modos, los responsables del *e-commerce* dieron a conocer en varias ocasiones que los *places* suelen declarar una menor capacidad de la que realmente tienen, con lo cual, si bien la debilidad del modelo persiste podría no ser tan relevante al menos para el modelo 1.a.

En segundo lugar hablaremos de la matriz de distancia. Recordemos que para elaborar la misma, se tuvo en cuenta la latitud y la longitud de los *sellers* y *places* y vía la norma Manhattan se calculó la distancia. Podríamos haber hecho este cálculo aún más preciso usando por ejemplo la API de GoogleMaps y asumir que el *seller* realizaría esta distancia caminando o en automóvil. Dado que la cantidad de consultas implicaba hacer un uso pago de la API, optamos por el *proxy* de la distancia Manhattan siendo quizá más imprecisos de lo que podríamos haber sido. Otra alternativa podría haber sido usar la API de OpenStreetMap en su uso no comercial.

En tercer lugar, en el modelo (iii), se realizaron dos supuestos que podrían ser fácilmente evitables si se hubiera contado con más información. Monetizar el costo de traslado de un *seller* a un *place* asignándole un valor económico a la distancia y hacer algo similar con el costo de apertura de un *place* puede ser totalmente cuestionable. Los resultados obtenidos podrían ser muy diferentes si se incorporaran los verdaderos costos como se demostró con el experimento realizado. De todos modos, desde un punto de vista teórico el modelo es el correcto.

Finalmente vale la pena mencionar quizá no como una mejora pero sí como un chequeo de robustez de los modelos que sería muy conveniente validar los resultados con otros datos existentes en otras regiones geográficas para corroborar la validez obtenida comparando con la asignación de la plataforma y la *performance* de los algoritmos. Es bastante evidente que las instancias más grandes podrían degradar exponencialmente los tiempos de resolución.

Extensiones vinculadas a los modelos

El modelo (i), en sus dos versiones, asume en algún sentido que el lugar más conveniente en el cual el *seller* debe dejar sus productos es el *place* que permite que la distancia total recorrida sea la menor sujeto a las capacidades existentes de los *places* de la red. De todos modos, puede que el *seller* tenga cierta preferencia en donde dejar sus paquetes y quizás ese lugar no es el *place* seleccionado por el modelo. Capturar esta preferencia del vendedor enriquece el ejercicio y su experiencia. Recordemos que estamos modelando traslados en vecindarios donde muy probablemente quienes viven allí se conocen y la afinidad que tienen entre ellos varía.

El modelo (ii) por como está planteado por las restricción de *walking distance*, deja de lado *sellers* de la red priorizando la cantidad de paquetes que aportan. La realidad es que cada *seller* trae una oferta diferente de productos y no es igual dejar afuera a un *seller* que vende productos que ya se encuentran en la plataforma y en forma masiva como por ejemplo celulares, a dejar afuera a un *seller* que fabrica productos únicos y personalizados como por ejemplo retratos, obras de arte, manualidades, etc. Tener en cuenta las características de la oferta que incorpora el *seller* a la plataforma es otra variable que podría ser considerada para que la decisión se vea enriquecida con otras variables del negocio. Esta información de carácter más cualitativo puede ser difícil de cuantificar para el negocio, de todos modos, el volumen o la facturación promedio por *seller* de alguna forma captura esta información.

El modelo (iii), define qué puntos de despacho conviene abrir de un *pool* de *places* existentes. Puede que *places* de mucha capacidad (y por lo tanto muy costosos) sean dejados de lado en ciertos barrios implicando que los *sellers* tengan que trasladarse más para poder dejar sus productos. Para minimizar esta fricción se podría incorporar al modelo una cantidad de *places* mínima por barrio que permita a la plataforma garantizar cierta experiencia a los *sellers* y poder afirmar que en todos los barrios hay al menos un *place* disponible. Algo similar a esto también podría incorporarse en el modelo (ii) donde no todos los *places* son tenidos en cuenta. Es decir, maximizar el volumen que se incorpora a la red controlando la experiencia de los *sellers* pero no sólo con la variable *walking distance* sino también garantizando que un set de *places* sean tenidos en cuenta obligatoriamente.

Finalmente, todos los modelos asumen que el conjunto de *places* y *sellers* es estático, esto si bien es un supuesto razonable para los *places* no lo es para los *sellers*. La estacionalidad de su venta y su madurez durante el ciclo de vida en la plataforma puede llevar a que los mismos crezcan y ameriten recibir una colecta en sus domicilios o que nuevos *sellers* nazcan y sea oportuno incorporarlos a la red. Si bien este tema es muy relevante, trasciende a los objetivos del presente trabajo.

Bibliografía

Bibliografía académica

Duran, Guillermo & Guajardo, Mario & Sauré, Denis & Souyris, Sebastian & Weintraub, Andres & Wolf, Rodrigo. (2007). Scheduling the Chilean Soccer League by Integer Programming. *Interfaces*. 37. 539-552. 10.1287/inte.1070.0318.

Erhan Erkut, Tony Myroon, and Kevin Strangway. 2000. TransAlta Redesigns Its Service-Delivery Network. *Interfaces* 30, 2 (March 2000), 54–69.

Higgins, A.J. (1999) Optimizing cane supply decisions within a sugar mill region. *Journal of Scheduling*, 2: 229–244.

Holland, C., Levis, J., Nuggehalli, R., Santilli, B., & Winters, J. (2017). UPS Optimizes Delivery Routes. *Interfaces*, 47, 8-23.

Melo M.T., Nickel S., Saldanha-da-Gama F. (2009). Facility location and supply chain management , A review, *European Journal of Operational Research*, Volume 196, Issue 2, Pages 401-412, ISSN 0377-2217, <https://doi.org/10.1016/j.ejor.2008.05.007>.

Oncan, Temel. (2007). A Survey of the Generalized Assignment Problem and Its Applications. *Infor*. 45. 123-141. 10.3138/infor.45.3.123.

ReVelle C.S. , Eiselt H.A. (2005). Location analysis: A synthesis and survey, *European Journal of Operational Research*, Volume 165, Issue 1, Pages 1-19, ISSN 0377-2217, <https://doi.org/10.1016/j.ejor.2003.11.032>.

Ross, G.T. and Soland, R.M. (1975) A branch and bound approach for the generalized assignment problem. *Mathematical Programming*, 8: 91–105.

Ross, G.T. and Soland, R.M. (1977) Modeling facility location problems as generalized assignment problems. *Management Science*, 24: 345–357.

Ross, G.T. and Zoltners, A.A. (1979) Weighted assignment models and their application. *Management Science*, 25 (7): 683–696.

Alumno: Juan Pablo Santos

Ross, G.T. and Soland, R.M. (1980) A multicriteria approach to the location of public facilities. *European Journal of Operational Research*, 4 (5): 307–321.

Viveros, R. y Salazar, E. (2010) Modelo de planificación de producción para un sistema multiproducto con múltiples líneas de producción. *Revista Ingeniera de Sistemas*, Volumen XXIV (Junio 2010), 89-102.

Bibliografía no académica

[1] Ayelén Actis (lunes 11 de mayo de 2020) El mundo de las billeteras virtuales: ¿sabías que ya existen más de 20 en el país? (la interoperabilidad, el siguiente paso). *Infonegocios*. <https://infonegocios.info/nota-principal/el-mundo-de-las-billeteras-virtuales-sabias-que-ya-existen-mas-de-20-en-el-pais-la-interoperabilidad-el-siguiente-paso>

[2] B2W Digital Q1 2020 Quarterly Business Review (jueves 7 de mayo de 2020) <https://static.b2wdigital.com/upload/releasesderesultados/00003387.pdf>

[3] Lucio Tavora (viernes 27 de marzo de 2020) Gobierno de Brasil anuncia programa para preservar empleo de pymes http://spanish.xinhuanet.com/2020-03/28/c_138924016.htm

[4] Editor (22 de agosto de 2014). Taobao launches self pick-up service in HK. *Retail Asia*. https://retailinasia.com/in-shops/e-commerce/online_retailing/taobao-launches-self-pick-service-hk/

[5] Marek Różycki and Prof. Arkadiusz Kawa (3 de septiembre de 2019). PUDOs and lockers – the new power in the Polish last mile. *Parcel and Postal Technology International*. <https://www.parcelandpostaltechnologyinternational.com/analysis/pudos-and-lockers-the-new-power-in-the-polish-last-mile.html>

[6] Cálculo de la Manhattan Distance https://es.wikipedia.org/wiki/Geometr%C3%ADa_del_taxista

Anexo 1: Solución del e-commerce

Tabla 1: Caracterización de la solución provista por el e-commerce

ID	Place	Capacity Place	# Sellers	# Shipments	Sum Distance	Real Capacity	Extended Capacity
1		200	8	76.09	24.1954	ok	ok
25		30	2	29	1.32502	ok	ok
26		20	2	16.27	0.91628	ok	ok
28		30	3	22.91	1.40958	ok	ok
29		50	5	53.73	1.69319	Problem	ok
35		20	2	13.13	0.95092	ok	ok
39		30	1	8.05	0.7095	ok	ok
41		20	2	18.23	1.39876	ok	ok
44		20	2	15.63	0.964	ok	ok
45		50	6	50.18	1.41	Problem	ok
46		50	5	44.45	1.78176	ok	ok
48		20	2	25.23	5.7613	Problem	ok
52		30	2	12.37	1.58064	ok	ok
55		30	1	8.95	2.9563	ok	ok
56		30	2	20.82	0.2548	ok	ok
59		30	2	20.46	1.19904	ok	ok
62		30	3	26.78	0.44823	ok	ok
63		30	2	24.72	1.0766	ok	ok
65		20	2	14.82	0.4145	ok	ok
67		20	2	16.5	0.44568	ok	ok
70		30	3	34.68	6.14757	Problem	ok
78		40	3	21.77	2.25654	ok	ok
79		20	2	19.87	0.88936	ok	ok
99		30	3	19.95	1.16578	ok	ok
9880		30	2	21.6	1.66548	ok	ok
9939		30	1	11.64	1.24591	ok	ok
9953		50	3	30.86	8.10781	ok	ok
9959		30	2	16.73	0.54548	ok	ok
9960		30	2	16.18	0.94556	ok	ok
9961		30	3	35.36	2.24983	Problem	ok
10369		30	2	18.77	1.1353	ok	ok
10405		40	3	45.32	0.89069	Problem	ok
10665		30	2	26.87	1.34736	ok	ok
10705		30	2	21.6	45.497	ok	ok
10722		30	3	28.96	3.34264	ok	ok
10805		30	2	24.36	0.33194	ok	ok

Alumno: Juan Pablo Santos

10826	30	2	20.32	12.3307	ok	ok
10840	30	2	29.09	0.59372	ok	ok
10910	30	3	30.5	0.87225	Problem	ok
10941	30	2	18.32	0.39898	ok	ok
10949	30	1	15.73	0.75266	ok	ok
10953	30	1	10.95	0.46725	ok	ok
11034	100	5	47.9	1.5646	ok	ok
11232	30	3	25.14	1.11091	ok	ok
11233	30	1	9.14	0.73575	ok	ok
11238	30	2	19.55	1.1195	ok	ok
11240	517	15	147.34	25.7995	ok	ok
11283	30	1	10.32	0.69678	ok	ok
11285	30	3	26.36	0.89853	ok	ok
11311	30	2	20.5	0.40444	ok	ok
11320	30	3	27.96	1.7924	ok	ok
11371	30	2	26.28	1.04218	ok	ok
11386	30	3	25.22	1.16354	ok	ok
11389	30	2	18.5	0.7494	ok	ok
11448	60	7	63.36	4.56693	Problem	ok
11458	20	2	22.68	4.43364	Problem	ok
11482	20	2	16.18	0.54058	ok	ok
11510	30	1	13.18	0.46497	ok	ok
11551	30	2	21	0.5812	ok	ok
11565	20	1	9.82	19.1096	ok	ok
11588	30	3	25.27	0.30671	ok	ok
11590	20	1	14.36	0.15232	ok	ok
11602	30	3	28.95	3.55609	ok	ok
11628	20	2	23.27	1.3669	Problem	ok
11643	20	2	28.18	2.06654	Problem	ok
11648	20	2	18.41	0.5846	ok	ok
11957	20	1	8.36	4.67477	ok	ok
11962	20	2	17.77	0.33444	ok	ok
12026	20	1	15.5	0.46783	ok	ok
12027	25	3	33.64	1.84578	Problem	ok
12189	30	4	31.5	3.17756	Problem	ok
12224	30	3	39.45	2.11492	Problem	ok
12228	70	4	51.91	1.9789	ok	ok
12425	30	2	25.14	0.81382	ok	ok
12453	30	2	15	2.9565	ok	ok
12454	30	2	17.18	1.39012	ok	ok
12519	30	1	10.09	0.61899	ok	ok
12528	30	1	8.23	0.20349	ok	ok
12540	20	2	22.27	0.55722	Problem	ok
12543	30	2	14.14	0.80896	ok	ok
12560	30	1	6.23	0.1843	ok	ok
12570	30	2	21.23	1.1817	ok	ok
12571	30	2	25.91	1.04468	ok	ok

Alumno: Juan Pablo Santos

12574	50	1	7.18	0.80692	ok	ok
12604	30	2	22.95	6.4996	ok	ok
12917	50	2	26.72	15.5509	ok	ok
13280	30	3	21.22	1.58026	ok	ok
13358	30	2	17.5	0.26874	ok	ok
13383	30	2	15.45	1.2015	ok	ok
13386	30	1	8.36	0.24801	ok	ok
13412	30	2	16.14	0.83996	ok	ok
13415	30	1	13.5	0.298	ok	ok
13416	30	3	23.82	1.47462	ok	ok
13443	30	3	38.41	0.86415	Problem	ok
13445	30	1	8.36	0.54625	ok	ok
13453	30	1	10.55	0.41931	ok	ok
13523	50	5	56.22	2.8114	Problem	ok
13556	40	3	30.95	1.0162	ok	ok
13557	30	2	16.41	0.838	ok	ok
13571	30	3	27.86	8.93786	ok	ok
13588	30	2	17.59	0.8487	ok	ok
13652	30	2	22.09	0.46712	ok	ok
13655	100	4	50.95	2.1971	ok	ok
13686	30	3	25.63	2.70866	ok	ok
13690	40	3	25.23	1.21358	ok	ok
13694	30	1	9.27	0.3465	ok	ok
13746	100	3	31.72	1.40071	ok	ok
13750	30	1	13.64	0.54358	ok	ok
13752	40	2	19.31	1.32368	ok	ok
13785	50	3	26.92	8.26634	ok	ok
13790	30	2	20.32	0.5105	ok	ok
13804	30	3	25.18	5.44602	ok	ok
13831	50	3	30.04	1.2698	ok	ok
13836	30	2	15.36	0.65608	ok	ok
13856	30	4	32.23	2.3001	Problem	ok
13857	30	2	21.27	0.67386	ok	ok
13859	30	2	27.82	0.94734	ok	ok
13881	50	4	33.77	1.43486	ok	ok
13883	30	2	16.5	0.60316	ok	ok
13956	50	2	25.23	0.55654	ok	ok
13957	50	4	40.19	0.6548	ok	ok
13982	50	2	12.64	1.25184	ok	ok
13983	60	5	45.04	3.78268	ok	ok
13999	30	2	20.55	1.10374	ok	ok
14023	30	2	26.14	0.98534	ok	ok
14027	30	2	20.13	1.87406	ok	ok
14029	30	2	18.82	1.9407	ok	ok
14030	40	2	18.73	1.0247	ok	ok
14036	50	2	17.09	1.1316	ok	ok
14041	30	2	16.32	0.7887	ok	ok

Alumno: Juan Pablo Santos

14043	50	5	45.77	2.52043	ok	ok
14044	50	4	38.59	1.5958	ok	ok
14066	60	2	22.36	0.33606	ok	ok
14068	30	2	14.27	0.92256	ok	ok
14071	100	5	46.4	20.4525	ok	ok
14081	60	2	22.59	1.38026	ok	ok
14102	30	3	23.13	1.18418	ok	ok
14103	100	1	7	0.557	ok	ok
14104	40	1	10.77	3.34579	ok	ok
14106	30	1	8.77	0.29561	ok	ok
14107	50	3	28.5	1.29528	ok	ok
14114	30	2	18.18	0.5757	ok	ok
14160	30	2	15.41	0.78876	ok	ok
14163	30	3	36.72	1.76909	Problem	ok
14165	30	1	9.18	4.4681	ok	ok
14196	30	3	27.92	1.3566	ok	ok
14197	30	2	14.95	1.15716	ok	ok
14199	20	1	15.14	0.66579	ok	ok
14203	50	3	31.45	1.23753	ok	ok
14206	50	6	67.83	27.6607	Problem	ok
14216	30	1	7.36	3.0507	ok	ok
14245	30	3	26.59	0.4882	ok	ok
14251	60	8	79.99	14.9802	Problem	ok
14252	30	2	32.86	1.45622	Problem	ok
14253	30	1	9.77	1.1477	ok	ok
14254	40	5	44.23	2.02849	Problem	ok
14272	30	4	34.91	2.3409	Problem	ok
14287	30	4	39.86	1.4979	Problem	ok
14290	30	3	27.27	1.70761	ok	ok
14291	30	2	15.91	0.4534	ok	ok
14293	30	5	43.26	2.1415	Problem	ok
14303	30	3	24.99	0.90854	ok	ok
14332	30	4	32.91	2.42384	Problem	ok
14334	40	1	9.23	0.36616	ok	ok
14336	30	1	9.27	0.31439	ok	ok
14339	50	4	36.46	1.944	ok	ok
14393	30	1	16.77	0.88699	ok	ok
14394	30	1	7.77	0.72138	ok	ok
14399	50	5	37.27	3.20267	ok	ok
14401	40	3	33.73	3.4346	ok	ok
14437	30	2	25.95	0.99	ok	ok
14438	30	3	26.64	0.9283	ok	ok
14474	50	2	18.82	1.54214	ok	ok
14476	40	3	32.81	5.60225	ok	ok
14477	30	2	16.23	1.09616	ok	ok
14479	35	2	30.13	1.08704	ok	ok
14481	50	5	39.55	2.31472	ok	ok

Alumno: Juan Pablo Santos

14505	30	2	16	1.5976	ok	ok
14529	50	5	51.54	3.07961	Problem	ok
14530	30	2	21.13	0.88638	ok	ok
14532	30	4	39.05	0.8898	Problem	ok
14562	30	2	17.63	0.7745	ok	ok
14564	30	2	27.5	0.70916	ok	ok
14572	30	1	7.86	6.47746	ok	ok
14574	30	3	24.87	1.34077	ok	ok
14584	50	2	29.27	5.07292	ok	ok
14592	30	3	36.27	1.00671	Problem	ok
14621	100	8	84.96	3.44756	ok	ok
14629	100	1	9.86	1.14561	ok	ok
14654	60	4	47.64	1.1857	ok	ok
14655	60	2	15.72	1.03978	ok	ok
14656	90	4	54.82	2.1667	ok	ok
14658	50	1	7.91	1.13184	ok	ok
14659	100	4	30.42	2.3219	ok	ok
14660	250	5	50.31	2.73193	ok	ok
14663	35	2	23.46	0.71596	ok	ok
14695	30	2	14.96	1.22232	ok	ok
14697	30	3	26	1.56199	ok	ok
14699	100	3	21.04	0.85833	ok	ok
14721	40	3	29.78	0.48478	ok	ok
14724	30	2	18.13	1.75132	ok	ok
14733	30	3	30.96	2.1402	Problem	ok
14741	40	3	22.6	6.1851	ok	ok
14764	70	1	7.45	2.99679	ok	ok
14779	100	5	46.54	14.8599	ok	ok
14787	50	3	31.73	1.6267	ok	ok
14797	30	2	27.68	0.55672	ok	ok
14799	30	3	26.59	2.25732	ok	ok
14802	50	3	25.32	1.76483	ok	ok
14803	100	1	13.55	1.0397	ok	ok
14869	30	2	19.41	1.0319	ok	ok
14870	100	3	25.04	1.42419	ok	ok
14874	100	2	16.04	1.94236	ok	ok
14914	30	3	30.37	1.24643	Problem	ok
14929	30	1	13.68	0.70064	ok	ok
14937	20	3	24.5	14.9565	Problem	ok
14949	30	2	13.1	0.7896	ok	ok
14950	100	4	36.19	2.13304	ok	ok
14951	30	3	28.73	15.8051	ok	ok
14952	30	3	32.6	9.7603	Problem	ok
14953	150	7	66.37	3.10836	ok	ok
14955	30	2	26.28	10.9177	ok	ok
14958	30	3	37.31	1.9953	Problem	ok
14959	70	4	45.23	1.86214	ok	ok

Alumno: Juan Pablo Santos

14969	50	2	17.59	0.88166	ok	ok
14997	100	2	13.05	0.9147	ok	ok
14998	150	9	116.09	2.40941	ok	ok
15003	100	3	19.27	1.90935	ok	ok
15012	80	1	7.27	0.42602	ok	ok
15013	40	3	21.5	1.9113	ok	ok
15015	50	2	16.45	1.2132	ok	ok
15016	100	3	31.36	1.25198	ok	ok
15035	100	1	10.36	0.98288	ok	ok
15036	30	2	15.5	1.5836	ok	ok
15041	60	4	52.01	1.97226	ok	ok
15044	100	10	103.46	3.85096	Problem	ok
15049	100	4	39.96	1.7213	ok	ok
15060	30	2	22.09	0.64542	ok	ok
15139	100	4	36.64	1.95444	ok	ok
15158	120	2	18.18	1.4616	ok	ok
15160	100	1	8.73	0.91306	ok	ok
15255	100	8	90.55	6.5667	ok	ok
15257	100	4	55.13	3.2528	ok	ok
15261	30	3	32	1.83859	Problem	ok
15271	40	5	44.55	3.57065	Problem	ok
15276	150	2	17.86	0.67512	ok	ok
15324	100	3	25.68	1.22721	ok	ok
15325	100	1	6.27	0.45591	ok	ok
15354	30	3	29.32	0.68276	ok	ok
15355	100	9	94.76	32.0629	ok	ok
15357	50	2	19.73	0.8097	ok	ok
15368	30	2	22.5	1.35214	ok	ok
15370	30	1	10.77	0.32708	ok	ok
15371	50	1	12.41	0.04809	ok	ok
15386	50	3	34.23	21.7636	ok	ok
15406	30	2	18.09	0.55512	ok	ok
15407	50	1	13.68	1.09279	ok	ok
15408	100	6	55.67	5.08258	ok	ok
15410	30	1	7.45	0.09444	ok	ok
15414	30	5	38.27	3.13835	Problem	ok
15477	40	3	32.05	1.07039	ok	ok
15481	2000	110	1164.27	88.7872	ok	ok
15482	2000	84	866.82	194.023	ok	ok
15491	50	1	8.86	7.2649	ok	ok
15494	30	2	23.96	14.1407	ok	ok
15498	30	1	11.91	0.44288	ok	ok
15530	50	3	32.09	18.2143	ok	ok
15546	40	3	40.22	1.47239	Problem	ok
15563	50	5	44.08	7.17642	ok	ok
15588	150	4	40.86	4.23686	ok	ok
15593	30	2	20.19	1.46844	ok	ok

Alumno: Juan Pablo Santos

15596	30	2	24.22	2.3038	ok	ok
15630	100	1	11.32	0.74744	ok	ok
15636	40	3	27.96	1.22033	ok	ok
15637	40	5	47.13	7.59791	Problem	ok
15652	50	3	32	1.66996	ok	ok
15654	50	2	18.23	1.46158	ok	ok
15698	150	5	51.72	1.80489	ok	ok
15780	60	3	31.64	8.82293	ok	ok
15781	50	3	36.22	0.60214	ok	ok
15784	50	2	21.41	7.415	ok	ok
15804	100	1	10.77	0.79627	ok	ok
15806	150	2	25.41	0.5503	ok	ok
15809	30	1	8.32	0.2899	ok	ok
15817	50	1	13.23	0.18928	ok	ok
15825	30	2	20.68	0.64356	ok	ok
15834	50	1	9.95	0.38107	ok	ok
15853	60	3	32.04	1.37308	ok	ok
15857	100	1	13.09	12.969	ok	ok
15858	100	3	23.87	1.80291	ok	ok
15924	50	1	12.91	0.19063	ok	ok
15935	30	1	10.82	0.73199	ok	ok
15947	100	9	98.38	37.4831	ok	ok
15987	30	2	18.04	0.2185	ok	ok
16000	40	1	10.86	0.15733	ok	ok
16022	30	2	28.73	18.3958	ok	ok
16035	30	3	42.24	3.45456	Problem	ok
16036	100	1	10.5	0.51986	ok	ok
16061	40	1	13.23	0.75588	ok	ok
16081	100	3	32.99	5.55035	ok	ok
16091	50	2	23.91	1.18678	ok	ok
16097	30	2	21.19	3.22542	ok	ok
16161	50	2	18.04	1.86126	ok	ok
16174	50	2	14.63	1.73768	ok	ok
16179	100	5	59.87	11.6184	ok	ok
16186	30	3	36.32	4.75997	Problem	ok
16250	50	2	20.41	1.79202	ok	ok
16318	120	7	85.6	5.90304	ok	ok
16734	100	2	14.18	1.17144	ok	ok
16803	30	2	27.27	0.15802	ok	ok
Total	19102	1022	10319.4	1215.42		

Alumno: Juan Pablo Santos

Tabla 2: Caracterización de la solución Problema (i.a)

ID Place	Capacity Place	# Sellers	# Shipments	Sum Distance	Real Capacity
1	200	4	43.86	2.3273	ok
25	30	2	29	1.32502	ok
26	20	2	16.27	0.91628	ok
28	30	2	14.55	0.9843	ok
29	50	4	44.41	1.54824	ok
35	20	2	13.13	0.95092	ok
39	30	1	8.05	0.7095	ok
41	20	2	19.05	1.49458	ok
44	20	2	15.63	0.964	ok
45	50	5	43.91	1.1555	ok
46	50	5	44.45	1.78176	ok
52	30	3	24.01	2.29061	ok
55	30	1	17.14	1.3385	ok
56	30	2	20.82	0.2548	ok
59	30	2	20.46	1.19904	ok
62	30	3	29.73	0.50863	ok
63	30	2	24.72	1.0766	ok
65	20	2	14.82	0.4145	ok
67	20	2	18.77	0.5177	ok
70	30	1	14.95	2.45529	ok
78	40	4	35.36	4.52164	ok
79	20	2	19.87	0.88936	ok
99	30	4	26.22	1.73496	ok
9880	30	2	21.6	1.66548	ok
9953	50	2	23.77	1.3868	ok
9959	30	2	16.73	0.54548	ok
9960	30	2	16.18	0.94556	ok
9961	30	4	27.87	0.72698	ok
10369	30	3	29.77	1.4451	ok
10405	40	3	39.64	1.24029	ok
10665	30	2	26.87	1.34736	ok
10705	30	4	28.45	1.95334	ok
10722	30	3	28.1	1.76864	ok
10805	30	3	29.27	2.25226	ok
10826	30	3	29.46	3.34177	ok
10840	30	2	19.18	0.9027	ok
10910	30	2	16.5	0.5815	ok
10941	30	3	27.27	1.73026	ok
10949	30	2	22.14	1.77036	ok
10953	30	1	10.95	0.46725	ok
11034	100	5	48.94	1.78168	ok
11232	30	4	29.59	1.86704	ok

Alumno: Juan Pablo Santos

11233	30	1	9.14	0.73575	ok
11238	30	2	19.55	1.1195	ok
11240	517	15	150.98	7.05662	ok
11283	30	2	24.46	2.08254	ok
11285	30	3	26.36	0.89853	ok
11311	30	1	12	0.19557	ok
11320	30	3	27.96	1.7924	ok
11371	30	2	27.37	0.82948	ok
11386	30	3	25.13	0.95364	ok
11389	30	3	26.45	1.67834	ok
11448	60	5	47.77	3.08231	ok
11458	20	1	12.73	0.25476	ok
11482	20	2	16.18	0.54058	ok
11510	30	2	24.36	0.99294	ok
11551	30	1	8.45	0.07387	ok
11565	20	1	7.77	0.21632	ok
11588	30	1	8.27	0.12837	ok
11590	20	1	14.36	0.15232	ok
11602	30	2	22.04	1.52372	ok
11628	20	2	19	0.9442	ok
11643	20	1	16.59	1.19285	ok
11648	20	2	18.41	0.5846	ok
11957	20	1	11.77	0.42833	ok
11962	20	2	17.77	0.33444	ok
12026	20	1	15.5	0.46783	ok
12027	25	2	19.64	1.1693	ok
12189	30	4	29.36	4.4446	ok
12224	30	2	25.54	1.41774	ok
12228	70	2	28.05	0.8242	ok
12425	30	3	25.55	1.42419	ok
12453	30	2	20.5	0.4166	ok
12454	30	2	17.18	1.39012	ok
12528	30	1	8.23	0.20349	ok
12540	20	1	9.86	0.16038	ok
12543	30	2	19.87	1.04748	ok
12560	30	1	6.23	0.1843	ok
12570	30	2	27.32	1.3667	ok
12571	30	2	25.91	1.04468	ok
12574	50	1	7.18	0.80692	ok
13280	30	3	21.22	1.58026	ok
13358	30	3	29.86	0.77262	ok
13383	30	2	23.13	1.5471	ok
13386	30	1	8.36	0.24801	ok
13412	30	3	27.14	2.15141	ok
13415	30	2	27.32	1.3344	ok
13416	30	3	24.09	1.60676	ok
13443	30	2	21.96	0.58654	ok

Alumno: Juan Pablo Santos

13445	30	2	20.31	1.55536	ok
13453	30	2	20.14	1.52388	ok
13523	50	3	27.41	1.9407	ok
13556	40	3	30.95	1.0162	ok
13557	30	2	16.41	0.838	ok
13571	30	2	19.54	1.32244	ok
13588	30	3	24.86	1.08664	ok
13652	30	2	17.59	1.02922	ok
13655	100	5	63.27	2.44536	ok
13686	30	1	6.32	0.80812	ok
13690	40	4	32.59	1.72854	ok
13694	30	2	14.78	1.1125	ok
13746	100	3	31.72	1.40071	ok
13750	30	2	24.19	1.0621	ok
13752	40	2	19.31	1.32368	ok
13790	30	2	20.32	0.5105	ok
13804	30	3	29.41	2.29382	ok
13831	50	3	30.04	1.2698	ok
13836	30	3	28.68	1.52501	ok
13856	30	3	26.14	1.70229	ok
13857	30	2	21.27	0.67386	ok
13859	30	2	27.82	0.94734	ok
13881	50	5	41.13	1.76799	ok
13883	30	3	28.05	9.79073	ok
13956	50	4	47.73	2.4751	ok
13957	50	4	40.19	0.6548	ok
13982	50	2	12.64	1.25184	ok
13983	60	6	56.99	4.1013	ok
13999	30	2	17.28	2.06692	ok
14023	30	2	26.14	0.98534	ok
14027	30	2	29.18	2.89732	ok
14029	30	2	22.36	1.37916	ok
14030	40	4	38.55	2.29208	ok
14036	50	1	7.23	0.25719	ok
14041	30	3	27.09	1.24514	ok
14043	50	5	45.77	2.52043	ok
14044	50	5	48.45	2.2621	ok
14066	60	3	34.5	1.00386	ok
14068	30	3	22.36	1.87749	ok
14071	100	1	6.68	0.59474	ok
14081	60	3	34.5	3.53808	ok
14102	30	3	23.13	1.18418	ok
14103	100	4	42.63	5.997	ok
14106	30	2	17	1.31842	ok
14107	50	4	47.72	3.71522	ok
14114	30	3	24.54	0.98586	ok
14160	30	3	22.32	1.73518	ok

Alumno: Juan Pablo Santos

14163	30	2	20.54	1.09716	ok
14165	30	2	15.59	0.52762	ok
14196	30	2	26.32	0.75578	ok
14197	30	3	25.72	1.71814	ok
14199	20	1	15.14	0.66579	ok
14203	50	4	40.72	1.43608	ok
14206	50	2	19.81	1.10196	ok
14216	30	1	7.18	6.6787	ok
14245	30	3	26.59	0.4882	ok
14251	60	5	55.09	1.72163	ok
14252	30	1	14.41	0.61646	ok
14253	30	2	28.91	3.35336	ok
14254	40	4	36.55	1.92932	ok
14272	30	4	28.28	2.19786	ok
14287	30	3	27.5	1.08789	ok
14290	30	3	27.27	1.70761	ok
14291	30	3	25.77	2.02966	ok
14293	30	3	26.9	1.27004	ok
14303	30	3	24.99	0.90854	ok
14332	30	3	26	1.60582	ok
14334	40	1	9.23	0.36616	ok
14336	30	1	9.27	0.31439	ok
14339	50	5	45.6	3.29475	ok
14393	30	2	25.13	1.2715	ok
14394	30	1	7.77	0.72138	ok
14399	50	6	44	3.78568	ok
14401	40	2	22.73	1.6554	ok
14437	30	3	28.41	1.64464	ok
14438	30	3	26.64	0.9283	ok
14474	50	2	18.82	1.54214	ok
14476	40	4	38.86	2.66366	ok
14477	30	2	22.5	1.21456	ok
14479	35	2	30.13	1.08704	ok
14481	50	5	41.64	2.54402	ok
14505	30	3	26.05	2.57147	ok
14529	50	4	44.81	2.4611	ok
14530	30	3	28.49	1.57224	ok
14532	30	3	29.14	0.63026	ok
14562	30	2	17.63	0.7745	ok
14564	30	2	27.96	0.97686	ok
14572	30	1	13.73	0.65244	ok
14574	30	1	11.32	0.35427	ok
14584	50	2	23.91	0.7284	ok
14592	30	3	28.36	0.55371	ok
14621	100	9	92.69	3.88789	ok
14654	60	4	47.64	1.1857	ok
14655	60	2	22.68	0.43558	ok

Alumno: Juan Pablo Santos

14656	90	4	54.82	2.1667	ok
14658	50	3	32.23	5.55794	ok
14659	100	5	38.33	2.76843	ok
14660	250	5	50.31	2.73193	ok
14663	35	1	12.41	0.12807	ok
14695	30	3	29.82	3.66654	ok
14697	30	2	18.18	1.06348	ok
14699	100	5	46.82	2.18075	ok
14721	40	4	36.19	0.96418	ok
14724	30	1	13.45	0.50462	ok
14733	30	2	18.14	1.4152	ok
14741	40	3	28.19	8.7267	ok
14764	70	2	13.55	0.57968	ok
14779	100	4	40.36	1.41694	ok
14787	50	4	48.81	2.07816	ok
14797	30	2	27.68	0.55672	ok
14799	30	3	30	2.29168	ok
14802	50	4	35.64	2.47566	ok
14803	100	1	13.55	1.0397	ok
14869	30	3	26.59	1.06434	ok
14870	100	4	32.27	2.09962	ok
14874	100	4	39.9	2.6376	ok
14914	30	2	16.14	0.56094	ok
14929	30	1	13.68	0.70064	ok
14937	20	2	16.68	1.7722	ok
14949	30	3	23.51	1.57415	ok
14950	100	6	68.27	3.09296	ok
14952	30	2	23.1	1.5577	ok
14953	150	9	83.27	4.79728	ok
14955	30	2	23.82	1.53972	ok
14958	30	2	25.95	1.2682	ok
14959	70	6	64.22	2.8762	ok
14969	50	2	17.59	0.88166	ok
14997	100	3	23.82	1.85871	ok
14998	150	10	125.59	3.0278	ok
15003	100	3	19.27	1.90935	ok
15012	80	1	7.27	0.42602	ok
15013	40	4	31.41	2.961	ok
15015	50	2	20.64	1.07322	ok
15016	100	4	43.72	2.44548	ok
15035	100	1	10.36	0.98288	ok
15036	30	2	15.5	1.5836	ok
15041	60	4	52.01	1.97226	ok
15044	100	8	95.68	2.96384	ok
15049	100	5	49.69	2.2797	ok
15060	30	3	27.59	1.38854	ok
15139	100	6	54.77	3.47858	ok

Alumno: Juan Pablo Santos

15158	120	4	35.18	3.27016	ok
15160	100	2	21.23	1.71374	ok
15255	100	7	70.28	1.64923	ok
15257	100	4	53.64	2.2992	ok
15261	30	2	23	0.7716	ok
15271	40	4	31.28	2.67312	ok
15276	150	6	42.95	6.15674	ok
15324	100	9	82.06	3.46419	ok
15325	100	2	13.36	0.88874	ok
15354	30	3	29.32	0.68276	ok
15355	100	9	96.52	6.75953	ok
15357	50	4	40.82	3.43894	ok
15368	30	1	10.82	0.53021	ok
15370	30	2	19.27	0.46618	ok
15371	50	1	12.41	0.04809	ok
15386	50	3	26.68	1.69452	ok
15406	30	3	29.5	1.11828	ok
15407	50	1	13.68	1.09279	ok
15408	100	8	82.22	14.0641	ok
15410	30	2	17.54	0.46444	ok
15414	30	3	25.86	1.81395	ok
15477	40	3	32.05	1.07039	ok
15481	2000	104	1102.91	48.1185	ok
15482	2000	68	714.09	39.5644	ok
15491	50	1	11.05	0.52888	ok
15498	30	1	11.91	0.44288	ok
15530	50	3	47.81	2.47713	ok
15546	40	3	35.86	2.46719	ok
15563	50	4	39.77	3.89686	ok
15588	150	4	50.31	4.11284	ok
15593	30	2	20.19	1.46844	ok
15596	30	2	20.91	1.2475	ok
15630	100	3	40.45	6.16464	ok
15636	40	3	27.96	1.22033	ok
15637	40	4	37.73	2.72534	ok
15652	50	4	45.09	6.10546	ok
15654	50	3	28.68	2.1422	ok
15698	150	8	87.95	4.132	ok
15780	60	1	12.18	0.49791	ok
15781	50	4	44.49	1.19856	ok
15784	50	1	9.95	0.62546	ok
15804	100	1	10.77	0.79627	ok
15806	150	4	44.87	2.83882	ok
15809	30	1	8.32	0.2899	ok
15817	50	2	26.37	1.50362	ok
15825	30	3	28.45	1.89473	ok
15834	50	1	9.95	0.38107	ok

Alumno: Juan Pablo Santos

15853	60	3	32.04	1.37308	ok
15858	100	4	41.33	2.89558	ok
15924	50	2	26.05	1.58996	ok
15935	30	3	30	1.22271	ok
15947	100	7	74.69	3.08085	ok
15987	30	3	27.4	0.89995	ok
16000	40	1	10.86	0.15733	ok
16022	30	1	7.73	1.51217	ok
16035	30	2	21.14	1.444	ok
16036	100	1	10.5	0.51986	ok
16061	40	2	19.41	5.48362	ok
16081	100	3	32.99	5.55035	ok
16091	50	2	23.91	1.18678	ok
16097	30	3	25.05	0.05504	ok
16161	50	3	29.64	1.46452	ok
16174	50	2	14.18	0.27162	ok
16179	100	3	36.05	0.95801	ok
16186	30	2	29.09	0.12832	ok
16250	50	2	20.41	1.79202	ok
16318	120	6	66.87	4.04352	ok
16734	100	1	7.36	0.5695	ok
16803	30	2	27.27	0.15802	ok
Total	18592	1022	10319.43	599.2116	

Tabla 3: Caracterización de la solución Problema (i.b)

ID Place	New Capacity Place	# Sellers	# Shipments	Sum Distance	Extended Capacity
1	300	2	25.5	0.8364	ok
25	45	2	29	1.32502	ok
26	30	2	16.27	0.91628	ok
28	45	2	14.55	0.9843	ok
29	75	6	61.59	2.50674	ok
35	30	3	28.68	6.5117	ok
39	45	1	8.05	0.7095	ok
41	30	2	19.05	1.49458	ok
44	30	2	15.63	0.964	ok
45	75	6	50.18	1.41	ok
46	75	7	71.18	4.07636	ok
52	45	3	24.01	2.29061	ok
55	45	1	17.14	1.3385	ok
56	45	3	33.46	0.8695	ok
59	45	2	20.46	1.19904	ok
62	45	3	26.78	0.44823	ok
63	45	3	34.95	2.01092	ok
65	30	3	24.59	1.4005	ok
67	30	3	26.27	0.83064	ok

Alumno: Juan Pablo Santos

78	60	3	21.77	2.25654	ok
79	30	3	29.82	1.27883	ok
99	45	3	19.95	1.16578	ok
9880	45	2	21.6	1.66548	ok
9953	75	1	12.18	0.78861	ok
9959	45	2	16.73	0.54548	ok
9960	45	2	16.18	0.94556	ok
9961	45	6	44.46	1.10362	ok
10369	45	2	22.91	0.8047	ok
10405	60	4	54.5	1.53712	ok
10665	45	1	16.23	0.43868	ok
10705	45	6	44.13	2.75834	ok
10722	45	4	35.05	2.42268	ok
10805	45	3	32.22	2.0165	ok
10826	45	3	33.14	2.22165	ok
10840	45	1	8.86	0.32917	ok
10910	45	3	30.5	0.87225	ok
10941	45	4	44	1.99718	ok
10949	45	1	15.73	0.75266	ok
10953	45	1	10.95	0.46725	ok
11034	150	4	33.99	1.00904	ok
11232	45	4	31.32	1.65474	ok
11233	45	1	9.14	0.73575	ok
11238	45	2	19.55	1.1195	ok
11240	775.5	15	150.98	7.05662	ok
11283	45	1	10.32	0.69678	ok
11285	45	4	36.22	2.29682	ok
11311	45	1	12	0.19557	ok
11320	45	3	27.96	1.7924	ok
11371	45	1	19.05	0.25528	ok
11386	45	4	36.54	1.39742	ok
11389	45	2	18.5	0.7494	ok
11448	90	5	47.77	3.08231	ok
11458	30	1	12.73	0.25476	ok
11482	30	3	26.04	1.35467	ok
11510	45	1	11.18	0.52797	ok
11551	45	1	8.45	0.07387	ok
11565	30	1	7.77	0.21632	ok
11590	30	1	14.36	0.15232	ok
11602	45	2	22.04	1.52372	ok
11628	30	1	8.95	0.1154	ok
11643	30	1	11.59	0.87369	ok
11648	30	2	18.41	0.5846	ok
11962	30	2	17.77	0.33444	ok
12026	30	2	27.09	1.0456	ok
12027	37.5	3	33.64	1.84578	ok
12189	45	5	40.36	5.36035	ok

Alumno: Juan Pablo Santos

12224	45	2	20.27	1.15854	ok
12228	105	2	28.05	0.8242	ok
12425	45	4	43.14	2.01178	ok
12453	45	2	20.5	0.4166	ok
12454	45	2	17.18	1.39012	ok
12528	45	1	8.23	0.20349	ok
12540	30	2	22.27	0.55722	ok
12543	45	1	7.05	0.31924	ok
12560	45	1	6.23	0.1843	ok
12570	45	3	40.41	1.91795	ok
12571	45	2	25.91	1.04468	ok
12574	75	1	7.18	0.80692	ok
13280	45	3	21.22	1.58026	ok
13358	45	2	17.5	0.26874	ok
13383	45	2	19.04	0.76978	ok
13386	45	1	8.36	0.24801	ok
13412	45	3	29.32	1.30289	ok
13415	45	1	13.5	0.298	ok
13416	45	4	38.41	2.57422	ok
13443	45	3	38.41	0.86415	ok
13445	45	3	33.17	2.26607	ok
13453	45	1	10.55	0.41931	ok
13523	75	3	27.41	1.9407	ok
13556	60	3	30.95	1.0162	ok
13557	45	2	16.41	0.838	ok
13571	45	1	7.59	0.13102	ok
13588	45	3	24.86	1.08664	ok
13652	45	2	17.59	1.02922	ok
13655	150	4	50.41	1.72342	ok
13686	45	1	6.32	0.80812	ok
13690	60	5	46.41	2.76308	ok
13694	45	2	14.78	1.1125	ok
13746	150	3	31.72	1.40071	ok
13750	45	1	13.64	0.54358	ok
13752	60	2	19.31	1.32368	ok
13790	45	2	20.32	0.5105	ok
13804	45	1	7	0.66994	ok
13831	75	3	30.04	1.2698	ok
13836	45	2	15.36	0.65608	ok
13856	45	4	32.23	2.3001	ok
13857	45	3	31.82	0.93757	ok
13859	45	3	34.73	1.52597	ok
13881	75	6	52.99	2.36748	ok
13883	45	2	16.5	0.60316	ok
13956	75	3	31.55	1.24025	ok
13957	75	6	64.24	2.00396	ok
13982	75	2	12.64	1.25184	ok

Alumno: Juan Pablo Santos

13983	90	6	52.72	3.83608	ok
13999	45	3	30.6	2.62739	ok
14023	45	2	26.14	0.98534	ok
14027	45	3	27.04	2.68155	ok
14029	45	2	22.36	1.37916	ok
14030	60	3	25.5	1.28487	ok
14036	75	1	7.23	0.25719	ok
14041	45	3	27.09	1.24514	ok
14043	75	5	45.77	2.52043	ok
14044	75	5	48.45	2.2621	ok
14066	90	3	34.5	1.00386	ok
14068	45	2	14.27	0.92256	ok
14071	150	1	6.68	0.59474	ok
14081	90	2	22.59	1.38026	ok
14102	45	3	27.77	9.54934	ok
14103	150	2	13.45	1.7182	ok
14106	45	1	8.77	0.29561	ok
14107	75	4	44.45	2.76816	ok
14114	45	3	24.54	0.98586	ok
14160	45	2	15.41	0.78876	ok
14163	45	3	36.72	1.76909	ok
14165	45	2	15.59	0.52762	ok
14196	45	4	41.19	1.93818	ok
14197	45	4	42.31	2.58996	ok
14199	30	1	15.14	0.66579	ok
14203	75	4	40.72	1.43608	ok
14206	75	2	19.81	1.10196	ok
14216	45	1	7.18	6.6787	ok
14245	45	3	26.59	0.4882	ok
14251	90	6	64.59	2.2888	ok
14252	45	2	32.86	1.45622	ok
14254	60	7	59.96	3.60657	ok
14272	45	4	40.6	3.06364	ok
14287	45	4	39.86	1.4979	ok
14290	45	3	27.27	1.70761	ok
14291	45	3	24.23	0.75668	ok
14293	45	4	37.08	1.58332	ok
14303	45	3	24.99	0.90854	ok
14332	45	4	32.91	2.42384	ok
14334	60	1	9.23	0.36616	ok
14336	45	1	9.27	0.31439	ok
14339	75	5	45.6	3.29475	ok
14393	45	2	25.13	1.2715	ok
14399	75	6	44	3.78568	ok
14401	60	2	22.73	1.6554	ok
14437	45	4	43.36	2.2446	ok
14438	45	3	26.64	0.9283	ok

Alumno: Juan Pablo Santos

14474	75	2	18.82	1.54214	ok
14476	60	3	29	1.60441	ok
14477	45	1	8.5	0.35379	ok
14479	52.5	3	45.08	2.568	ok
14481	75	5	41.64	2.54402	ok
14505	45	2	16	1.5976	ok
14529	75	4	44.81	2.4611	ok
14530	45	2	21.13	0.88638	ok
14532	45	4	39.05	0.8898	ok
14562	45	2	17.63	0.7745	ok
14564	45	2	27.5	0.70916	ok
14572	45	1	13.73	0.65244	ok
14574	45	1	11.32	0.35427	ok
14584	75	2	23.91	0.7284	ok
14592	45	3	28.36	0.55371	ok
14621	150	9	92.69	3.88789	ok
14654	90	3	30.91	0.8625	ok
14655	90	2	22.68	0.43558	ok
14656	135	4	54.82	2.1667	ok
14658	75	3	32.23	5.55794	ok
14659	150	5	38.33	2.76843	ok
14660	375	5	50.31	2.73193	ok
14663	52.5	1	12.41	0.12807	ok
14695	45	3	22.73	1.92508	ok
14697	45	3	32.73	1.44081	ok
14699	150	5	46.82	2.18075	ok
14721	60	5	47.96	1.29328	ok
14724	45	1	13.45	0.50462	ok
14733	45	3	30.96	2.1402	ok
14741	60	3	28.19	8.7267	ok
14764	105	2	13.55	0.57968	ok
14779	150	4	40.36	1.41694	ok
14787	75	4	45.18	1.94456	ok
14797	45	2	27.68	0.55672	ok
14799	45	4	41.09	2.86996	ok
14802	75	3	25.32	1.76483	ok
14803	150	1	13.55	1.0397	ok
14869	45	5	44.95	1.89824	ok
14870	150	4	32.27	2.09962	ok
14874	150	4	39.9	2.6376	ok
14914	45	3	30.37	1.24643	ok
14929	45	1	13.68	0.70064	ok
14937	30	2	16.68	1.7722	ok
14949	45	2	13.1	0.7896	ok
14950	150	5	57.91	2.34685	ok
14952	45	2	23.1	1.5577	ok
14953	225	9	83.27	4.79728	ok

Alumno: Juan Pablo Santos

14955	45	2	23.82	1.53972	ok
14958	45	3	37.31	1.9953	ok
14959	105	6	64.22	2.8762	ok
14969	75	2	17.59	0.88166	ok
14997	150	3	23.82	1.85871	ok
14998	225	9	116.09	2.40941	ok
15003	150	3	19.27	1.90935	ok
15012	120	1	7.27	0.42602	ok
15013	60	5	42.05	3.7129	ok
15015	75	2	20.64	1.07322	ok
15016	150	3	31.36	1.25198	ok
15035	150	1	10.36	0.98288	ok
15036	45	2	15.5	1.5836	ok
15041	90	4	52.01	1.97226	ok
15044	150	10	116.96	3.77252	ok
15049	150	5	49.69	2.2797	ok
15060	45	4	37.95	1.83532	ok
15139	150	6	54.77	3.47858	ok
15158	180	2	18.18	1.4616	ok
15160	150	1	8.73	0.91306	ok
15255	150	7	70.28	1.64923	ok
15257	150	3	37.41	1.36105	ok
15261	45	3	30.18	1.66919	ok
15271	60	5	44.55	3.57065	ok
15276	225	3	24.09	1.30111	ok
15324	150	7	60.78	2.63137	ok
15325	150	2	13.36	0.88874	ok
15354	45	3	29.32	0.68276	ok
15355	150	10	118.07	7.36524	ok
15357	75	2	19.73	0.8097	ok
15368	45	1	10.82	0.53021	ok
15370	45	2	19.27	0.46618	ok
15371	75	1	12.41	0.04809	ok
15386	75	3	26.68	1.69452	ok
15406	45	3	26.04	0.98008	ok
15407	75	1	13.68	1.09279	ok
15408	150	5	42.58	2.83159	ok
15410	45	2	17.54	0.46444	ok
15414	45	6	44.72	5.67126	ok
15477	60	3	32.05	1.07039	ok
15481	3000	103	1096	47.01472	ok
15482	3000	62	657.87	34.85524	ok
15491	75	1	11.05	0.52888	ok
15498	45	1	11.91	0.44288	ok
15530	75	2	29.36	0.96246	ok
15546	60	4	50.63	2.01448	ok
15563	75	3	30.72	2.23604	ok

Alumno: Juan Pablo Santos

15588	225	3	37.13	3.06442	ok
15593	45	3	33.28	3.49403	ok
15596	45	2	20.91	1.2475	ok
15630	150	3	40.45	6.16464	ok
15636	60	3	27.96	1.22033	ok
15637	60	4	36.91	2.22674	ok
15652	75	4	45.09	6.10546	ok
15654	75	2	18.23	1.46158	ok
15698	225	8	87.95	4.132	ok
15780	90	1	12.18	0.49791	ok
15781	75	4	44.49	1.19856	ok
15784	75	1	9.95	0.62546	ok
15804	150	1	10.77	0.79627	ok
15806	225	4	44.87	2.83882	ok
15809	45	1	8.32	0.2899	ok
15817	75	2	26.37	1.50362	ok
15825	45	2	20.68	0.64356	ok
15834	75	1	9.95	0.38107	ok
15853	90	3	32.04	1.37308	ok
15858	150	4	41.33	2.89558	ok
15924	75	1	12.91	0.19063	ok
15935	45	3	30	1.22271	ok
15947	150	7	74.69	3.08085	ok
15987	45	3	27.4	0.89995	ok
16000	60	1	10.86	0.15733	ok
16022	45	1	7.73	1.51217	ok
16035	45	3	40.28	2.44792	ok
16036	150	1	10.5	0.51986	ok
16061	60	2	19.41	5.48362	ok
16081	150	3	32.99	5.55035	ok
16091	75	2	23.91	1.18678	ok
16097	45	4	33.32	0.13618	ok
16161	75	2	15.09	1.03896	ok
16174	75	2	14.18	0.27162	ok
16179	150	3	36.05	0.95801	ok
16186	45	3	39.41	0.69469	ok
16250	75	2	20.41	1.79202	ok
16318	180	6	66.87	4.04352	ok
16734	150	1	7.36	0.5695	ok
16803	45	2	27.27	0.15802	ok
Total	27678	1022	10319.43	558.82334	

Anexo 2: Scripts

Modelo 1.a: Software utilizado SCIP

```
# Conjunto de places
set J := { read "places_ml.txt" as "<1n>" };

# Conjunto de sellers a asignar
set I := { read "sellers_ml.txt" as "<1n>" };

# Capacidad de los places
param cap[J] := read "places_ml.txt" as "<1n> 4n";

# Shipments de los sellers
param env[I] := read "sellers_ml.txt" as "<1n> 4n";

# Matriz de distancias entre la oficina i y la central operativa j
param dist[I*J] := read "distance_ml.txt" as "n+";

do print I;
do print J;

# Variables
var x[I*J] binary; # Nos dice el seller i es asignado al place j

# Buscamos minimizar la distancia que recorre el seller
minimize fobj: sum <i,j> in I*J: (dist[i,j] * x[i,j]);

# Cada seller va a dejar los paquetes en un place
subto unplace: forall <i> in I: sum <i,j> in I*J: x[i,j] == 1;

# Los shipments asignados al place no debe superar su capacidad
subto capacidadmax: forall <j> in J: sum <i,j> in I*J: env[i] * x[i,j] <= cap[j];
```

Modelo 1.b: Software utilizado SCIP

```
# Conjunto de places
set J := { read "places_ml.txt" as "<1n>" };

# Conjunto de sellers a asignar
set I := { read "sellers_ml.txt" as "<1n>" };

# Capacidad de los places
param cap[J] := read "places_ml.txt" as "<1n> 4n";

# Shipments de los sellers
param env[I] := read "sellers_ml.txt" as "<1n> 4n";
```

Alumno: Juan Pablo Santos

```
# Matriz de distancias entre la oficina i y la central operativa j
param dist[I*J] := read "distance_ml.txt" as "n+";

do print I;
do print J;

# Variables
var x[I*J] binary; # Nos dice el seller i es asignado al place j

# Buscamos minimizar la distancia que recorre el seller
minimize fobj: sum <i,j> in I*J: (dist[i,j] * x[i,j]);

# Cada seller va a dejar los paquetes en un place
subto unplace: forall <i> in I: sum <i,j> in I*J: x[i,j] == 1;

# Los shipments asignados al place no debe superar su capacidad
subto capacidadmax: forall <j> in J: sum <i,j> in I*J: env[i] * x[i,j] <= cap[j]*1.5;
```

Modelo 2: Software utilizado SCIP

```
# Conjunto de places
set J := { read "places_ml.txt" as "<1n>" };

# Conjunto de sellers a asignar
set I := { read "sellers_ml.txt" as "<1n>" };

# Capacidad de los places
param cap[J] := read "places_ml.txt" as "<1n> 4n";

# Shipments de los sellers
param env[I] := read "sellers_ml.txt" as "<1n> 4n";

# Matriz de distancias entre la oficina i y la central operativa j
param dist[I*J] := read "distance_ml.txt" as "n+";

do print I;
do print J;

# Variables
var x[I*J] binary; # Nos dice el seller i es asignado al place j

# Buscamos maximizar la cantidad de paquetes que entra en la red
maximize fobj: sum <i,j> in I*J: (env[i] * x[i,j]);

# Cada seller va a dejar los paquetes en un place
subto unplace: forall <i> in I: sum <i,j> in I*J: x[i,j] <= 1;

# Los shipments asignados al place no debe superar su capacidad
```

Alumno: Juan Pablo Santos

```
subto capacidadmax: forall <j> in J: sum <i,j> in I*J: env[i] * x[i,j] <= cap[j];
```

```
# La distancia que esta dispuesto a caminar un seller no es superior a 300mts11
```

```
subto wd: forall <i,j> in I*J: dist[i,j] * x[i,j] <= 0.3;
```

Modelo 3: Software utilizado SCIP, se transforma a formato CPLEX vía ZIMPL

```
# Conjunto de places
```

```
set J := { read "places_ml.txt" as "<1n>" };
```

```
# Conjunto de sellers a asignar
```

```
set I := { read "sellers_ml.txt" as "<1n>" };
```

```
# Capacidad de los places
```

```
param cap[J] := read "places_ml.txt" as "<1n> 4n";
```

```
# Shipments de los sellers
```

```
param env[I] := read "sellers_ml.txt" as "<1n> 4n";
```

```
# Matriz de distancias entre la oficina i y la central operativa j
```

```
param dist[I*J] := read "distance_ml.txt" as "n+";
```

```
do print I;
```

```
do print J;
```

```
# Variables
```

```
var x[I*J] binary; # Nos dice el seller i es asignado al place j
```

```
var y[J] binary; # Nos si el place j esta prendido o no
```

```
# Buscamos minimizar la distancia que recorre el seller
```

```
minimize fobj: sum <i,j> in I*J: (dist[i,j] * x[i,j]) + sum <j> in J: (cap[j] * y[j]);
```

```
# Cada seller va a dejar los paquetes en un place
```

```
subto unplace: forall <i> in I: sum <i,j> in I*J: x[i,j] == 1;
```

```
# Ligamos las variables x a las aperturas de los places (y)
```

```
subto ligado: forall <i,j> in I*J: x[i,j] <= y[j];
```

```
# Los shipments asignados al place no debe superar su capacidad
```

```
subto capacidadmax: forall <j> in J: sum <i,j> in I*J: env[i] * x[i,j] <= cap[j] * y[j];
```

¹¹ Los modelos modifican el walking distance pero en estructura son idénticos

Anexo 3: Instancias Benchmark

Resumen solución problema (1.a) testeo vs instancias benchmark

Instancias	# Places usados	# Sellers asignados	# Shipments	Sum Distance	Sum Real Capacity
1	3	10	111.36	79.12 kms	280
2	2	10	103.99	41.5 kms	190
3	3	10	98.59	84.7 kms	120
4	3	9	78.95	36.4 kms	90
5	3	8	89.23	28.6 kms	100
bogota	5	40	1213	6104 kms	1330
elba	5	40	1176	10979 kms	1230
ForliCesenaGAP	5	40	1324	14800 kms	1450
galway	5	40	1070	4524 kms	1175
GAP40x5a	5	40	1107	3439 kms	1215

En las instancias 1 a 3 y en bogota, elba, ForliCesenaGAP, galway y GAP40x5a, el algoritmo asignó a todos los sellers. En las instancias 4 y 5, dado que la cantidad de shipments a asignar era superior a la capacidad de los places se procedió balanceando el problema incorporando un *place* con una capacidad muy alta y ubicado a una distancia muy alejada. Asumimos que si el modelo asignaba a un *seller* a este *place* el mismo quedaba fuera de la red debido a que la capacidad no es suficiente para asignar a todos los *sellers*.