INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY
DIVISIÓN DE INGENIERÍA Y ARQUITECTURA
PROGRAMA DE GRADUADOS EN INGENIERÍA

OPTIMIZACIÓN DE LA VARIEDAD DE COLORES EN UNA EMPRESA DE ACERO RECUBIERTO

TESIS
PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO ACADÉMICO DE:

MAESTRO EN CIENCIAS
ESPECIALIDAD EN SISTEMAS DE CALIDAD Y PRODUCTIVIDAD

POR:
LUIS JACOB ESCOBAR SALDÍVAR

MONTERREY, N. L. **DICIEMBRE DE 2005**

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY
DIVISIÓN DE INGENIERÍA Y ARQUITECTURA
PROGRAMA DE GRADUADOS EN INGENIERÍA

Los miembros del comité de tesis recomendamos que el presente anteproyecto de tesis presentado por el Ing. Luis Jacob Escobar Saldívar sea aceptado como requisito parcial para obtener el grado académico de:

Maestro en Ciencias con Especialidad en:
Sistemas de Calidad y Productividad

Comité de tesis:

_____
Dr. Neale R. Smith Cornejo
ASESOR

_____
Dr. José Luis González Velarde
SINODAL

_____
M.C. Heriberto García Reyes
SINODAL

Aprobado:

_____
Dr. Federico Viramontes Brown
Director del Programa de Graduados en Ingeniería
Diciembre, 2005

AGRADECIMIENTOS

Se agradece el apoyo para la realización de esta tesis a mi asesor, el Dr. Neale Smith y a mis sinodales el Dr. José Luis González Velarde y el MC. Heriberto García por su retroalimentación. Un agradecimiento muy especial al personal de la empresa donde se desarrolla el proyecto, particularmente al Ing. Guillermo Arriaga. También un reconocimiento a los desarrolladores del LP Solve 5.5, que se utilizó como parte de los modelos, y al Ing. Luciano Salvietti quien compartió tips sobre su uso.

INDICE

1. INTRODUCCIÓN

Esta investigación propone un método de solución para un problema real, de una empresa mexicana. El método propuesto se puede aplicar además a otras líneas de productos de la misma empresa o inclusive a otras empresas que requieran racionalizar su oferta de productos y encontrar un punto óptimo de variedad. Se consultó literatura del tema para comparar el método propuesto con otros existentes y situar esta investigación dentro del universo de enfoques y soluciones para este tipo de problemas. Los datos aquí presentados están basados en los del problema real. El problema consiste en ofrecer a los clientes una variedad de productos óptima que maximice la contribución marginal de las líneas de pintado de la empresa.

1.1. LA EMPRESA

La empresa en la cual se desarrolla este proyecto, se especializa en los procesos de recubrimiento del acero: galvanizado y pintado. Está localizada en Nuevo León, atendiendo principalmente el mercado mexicano, aunque también tiene presencia en Estados Unidos y en otros países de Latinoamérica.

La subsidiaria de esta empresa en Estados Unidos opera desde el 2003 y ha tenido un crecimiento constante en sus ventas desde su arranque. Esta subsidiara norteamericana es una red de distribución pero actualmente no cuenta con líneas de pintado, por lo que se abastece de la planta en Nuevo León. El mercado norteamericano tiene mucho potencial de consumo pero también está más acostumbrado a recibir servicios personalizados. Debido a esto, la contraparte norteamericana ha sido el principal promotor de colores nuevos en los catálogos de la empresa mexicana. La empresa ha tenido que desarrollar colores nuevos cada mes y el volumen de venta por color es menor que en el mercado nacional, lo cual activó señales de alerta para analizar el impacto de tal proliferación de colores de su oferta en la eficiencia de sus líneas y sus niveles de inventarios.

1.2. EL MERCADO

Una forma de segmentar el mercado de acero recubierto es en consumidores de productos galvanizados y consumidores de productos pintados, para este estudio nos interesa el mercado de productos pintados. Otra forma de dividir mercado es en nacional (México) y exportación (principalmente Estados Unidos), para este estudio es necesario considerar tanto el mercado nacional como el de exportación ya que ambos mercados se atienden desde las mismas líneas de pintado. Los sectores que atiende la empresa con sus productos pintados son el automotriz, el de envases, el de línea blanca y electrodomésticos, el de construcción y arquitectónico, el de agricultores y ganaderos, entre otros.

1.2.1. Los clientes

La empresa tiene clientes ocasionales o "spot" que son atendidos por su red de distribución. Estos clientes tienen una demanda irregular ya que no siempre consumen los mismos productos, su frecuencia de consumo no es constante y su elección de proveedor puede cambiar en cada compra. Por otra parte, también se tienen clientes con contrato. Estos clientes son generalmente empresas medianas o grandes cuyo patrón de consumo es más estable por lo que se tiene de antemano un mejor estimado de su demanda. Lo anterior permite una mejor planeación de la producción pero a la vez los clientes con contrato exigen un mejor precio con respecto a los clientes ocasionales.

1.3. LOS PRODUCTOS

La empresa vende sus productos en diferentes presentaciones, estas pueden ser rollo, cinta, hoja lisa, hoja acanalada, panel (formado por dos hojas y poliuretano entre ellas), teja (de acero) y tubo. Para todas estas presentaciones existen diversos tipos de galvanizado (galvanizado y galvalum, con diferentes grosores de capa) y diversos tipos

de pintura (acabado flurocarbonado, poliéster modificado, poliéster siliconizado, entre otros). Otras características de los productos son el tipo de acero, el calibre o espesor, el ancho o el diámetro, el largo, el color, el acabado (liso, embozado, entre otros), la capa de polietileno, etc. El universo de productos ofrecido por la empresa es amplio, profundo y complejo, ya que las características de estos se pueden combinar de muchas formas y el cliente tiene la libertad de especificar casi todas ellas.

Para este estudio las características que nos interesan son el tipo de pintura y el color. El tono del color "Gris Perla" es igual en Poliéster Modificado (PM) que en Acabado Flurocarbonado (AF), sin embargo son pinturas diferentes. Por su composición, el AF es de mayor calidad y tiene mayor garantía (25 años AF vs. 15 años PM), por lo que es un tipo de pintura más caro. Debido a esto, el color Gris Perla AF es diferente al color Gris Perla PM ya que si se quiere ofrecer ambos colores se necesitan inventarios diferentes para ambos y setups entre sus corridas de producción en las líneas de pintado. Al ser tipos de pinturas diferentes inclusive sus primers (sustancia para que se adhiera la pintura a la lámina) son diferentes. En este estudio, se considera un "color" como una combinación de "tipo de pintura" y "color"; por ejemplo el Arena PM es un color, el Arena AF es otro color y el Gris Perla AF es otro color. Actualmente, cuando la empresa agrega un color a su catálogo (Color Chart), no agrega un producto sino una infinidad de productos potenciales, ya que el cliente puede especificar ese color en diferentes presentaciones y dimensiones, lo cual multiplica los inventarios y los stockouts (faltantes de producto) en sus sucursales.

3. LITERATURA DE LA VARIEDAD

Esta sección se presenta a manera de síntesis de las conclusiones obtenidas al revisar la literatura de loas diferentes líneas de investigación del tema, sus métodos y resultados. Existen en ella resúmenes integrativos [Da Silveira, 1998; Ramdas, 2003 & Swink 2005], algunos específicamente de modelos aplicados a la industria acerera y metalúrgica [Dutta, 2001], de los cuales se concluye lo siguiente:

Hay muchas dimensiones de variedad de los productos englobadas principalmente en la forma y función. Estas pueden ser: geometría, materiales, color, componentes, marcas, empaque, canales de venta, garantías, servicio post-venta, entre otras. Sin embargo, las dimensiones en las cuales elija la empresa competir deben ser de valor para el cliente [Ramdas 2003]. Además, el conjunto de dimensiones o características elegidas impactan la variedad percibida por el cliente [Tang, 1998]. Por ejemplo, en la industria automotriz tiene un mayor impacto en la variedad percibida el cambiar el chasis, los colores y los interiores que el cambiar el motor o el sistema de frenado. Otra conclusión de los es que generalmente se logra distinguir una tendencia al comparar diferentes empresas y su variedad ofrecida. En muchas ocasiones la variedad percibida por el cliente fue indicador del éxito comercial de la empresa y hubo una correlación positiva entre estos elementos.

El portafolio de una empresa es un indicador comúnmente usado como índice de variedad y se mide generalmente en dos dimensiones diferentes: amplitud y profundidad. Su amplitud es la cantidad de líneas diferentes de productos que tiene mientras que su profundidad es la cantidad de productos diferentes en cada línea. [Stanton, 2005]

Es de común acuerdo entre las diferentes fuentes revisadas que simplemente incrementar la variedad en el portafolio de productos de una empresa no le asegura ganancias en el largo plazo y puede restarle competitividad debido a altos costos. Los procesos de selección del grado de variedad, creación de la variedad e implementación de la variedad son determinantes clave del éxito de la estrategia.

Para selección y creación de la variedad existen varias estrategias en la literatura que englobo como "Estrategias de Adaptabilidad". Estas estrategias consisten en seleccionar los mercados objetivo en base a su rentabilidad, potencial de crecimiento y competencia para decidir hacia qué productos y clientes adaptar su producción, expansión y mezcla mediante ajuste de prioridades, manufactura enfocada, personalización masiva ("mass customization"), entre otras prácticas. Estas estrategias de adaptabilidad obedecen más a factores externos de la empresa, como el mercado.

Para la implementación de la variedad se desarrollan tácticas que denomino como "Flexibilidad Operativa". Estas actividades o tácticas consisten en desarrollar flexibilidad en la entrega, en los procesos, programación, máquinas y prácticas como: manufactura celular, SMED, diseño para la manufactura, entre otras. Esta flexibilidad operativa obedece a factores internos de la empresa, como las tecnologías de información y producción con las que cuenta.

El modelo que se propone en esta investigación, sirve como base para tomar decisiones de adaptabilidad; es decir, selección de un portafolio de productos. El modelo considera la flexibilidad operativa actual y los resultados de ella son una entrada para este modelo. Sin embargo, se debe formar un círculo de retroalimentación entre ambas ramas. En base al mercado y a las capacidades de la planta se selecciona una estrategia a seguir. Al momento de haber un cambio ya sea en el mercado, debido por ejemplo a factores macroeconómicos externos, o un cambio en la capacidad de la planta, por ejemplo la implementación de tecnologías nuevas que permiten ofrecer mayor variedad sin comprometer la productividad; es necesario rehacer el análisis de selección de portafolio.

Se propone un modelo heurístico que, por su complejidad, engloba muchas áreas. Se encontraron en la literatura modelos que con ciertas adaptaciones podrían formar parte de este modelo mayor, como submodelos ya que vienen de diferentes áreas y están planteados desde diferentes enfoques: mercadotecnia, producción, inventarios, economía, logística, entre otros.

Por ejemplo, un modelo encontrado [Morgan, 2001] toma en cuenta en su definición las familias de productos que comparten recursos, capacidades y setups y lo cruza con los diferentes segmentos de clientes. El planteamiento del modelo es muy interesante ya que toma en cuenta el "canibalismo" que puede suceder entre productos al haber un sustituto en el mercado. La premisa de este modelo es que el cliente siempre va a escoger el producto que le maximice su rentabilidad. De acuerdo a lo leído en la literatura, esta premisa se cumple muy a menudo. Este modelo junto con otras teorías investigadas [Schrieber, 2005] se podrían adaptar para formar un submodelo de estimación y proyección de demanda en base a productos sustitutos e integrarlo en el heurístico propuesto.

Se encontraron además otros modelos que relacionan los inventarios made-to-stock y los made-to-order con el precio y la oferta de productos [Dobson, 2002] o que proponen formas alternativas para la generación de órdenes que generen el menor riesgo de inventarios [Keong, 2005]. Debido a las complejidades del problema real que se investiga en este caso, no se podrían aplicar tal cual estos modelos pero el combinando el planteamiento de sus restricciones con otras teorías investigadas [Chopra, 2003] servirían para formar un submodelo de inventarios que complemente el heurístico propuesto y permitan mejorar esas prácticas.

Los modelos anteriores incluyen el precio en sus consideraciones pero hay otros modelos del área de administración de la ganancia en general ("revenue management") [Chen, Kyle D, 2000; Chen, Miao-Sheng, 2001] así como específicamente en el mercado del acero [Hall, 2003]; cuyos planteamientos sirven para en un futuro ampliar el heurístico al grado de poder definir la política de precios óptima para cada portafolio seleccionado.

El caso basado específicamente en la industria del acero [Hall, 2003] hace notar que el problema de la "irracionalidad" en los precios de este caso es un problema generalizado en ese tipo de industria. Por irracionalidad se refiere a que clientes diferentes de una empresa acerera pueden pagar precios distintos, por el mismo volumen del mismo

producto, en el mismo día. Esto se debe a que el precio en esta industria está influenciado fuertemente por las habilidades negociadoras de cada una de las partes, la distancia o familiaridad entre ellos y es generalmente algo privado entre la empresa y cada uno de sus clientes. Esto puede generar "escopetazos" en las gráficas de precio-volumen analizadas, ya que los puntos forman una nube en vez de seguir la tendencia lógica de a mayor volumen menor precio.

Aquí también queda claro que el problema de inventarios y stock-outs es algo difícil de modelar en esta industria ya que en general se encuentra poca relación entre los inventarios, las ventas y las compras. Esto se debe a lo inestable de los precios, ya que así como sube un precio un mes, puede bajar al otro, por lo que las compras de acero pueden ser poco frecuentes y en tamaños grandes. Lo anterior es algo que siempre ha sucedido por la naturaleza de la industria pero actualmente se ha intensificado por el "Efecto China". Debido al tamaño de China y a que se encuentra en desarrollo, alterna fácilmente entre consumidor neto y productor neto de acero y chatarra. Cuando China importa acero y chatarra los precios de este aumentan considerablemente, pero después los exporta y los precios mundiales vuelven a bajar. Estos cambios pueden pasar de un mes a otro y generan una incertidumbre en el análisis de los precios, ya que se pueden obtener gráficas donde se vea que, conforme se aumentó el precio, aumentaron las ventas y viceversa.

Se encontraron además de modelos de programación matemática, metodologías aplicadas a empresas reales [Appelqvist, 2005; Cargille, 2005] que sirvieron como referencia para tratar de involucrar en el heurístico propuesto los elementos que consideran las otras metodologías, aunque no fueran métodos matemáticos. Se trató de asimilar lo relevante para este caso; para lo que no fue posible modelar debido a los datos o al tiempo con el que se contaba, se dejó el heurístico preparado para su ampliación en algún proyecto posterior.

El caso de HP [Cargille, 2005] sirve como referencia del heurístico propuesto ya que pretende ser algo global que involucra inputs de otras áreas también, aunque no es un

modelo matemático es una metodología bien definida por cinco pasos. Los costos que toma en cuenta son los de la rampa de producción, los de inventarios en proceso y en producto terminado, los de mercadotecnia, los de servicios de soporte y los costos organizacionales administrativos ("overhead" / "backoffice"). El problema de esta metodología es que precisamente se basa mucho en la correcta estimación de los costos unitarios, para lo cual se requiere lograr aislar unos productos de otros o lograr dividir de forma equitativa el desperdicio y los costos de oportunidad de manera que no haya subsidios ocultos. Hay ciertas interacciones que sí se pueden eliminar desde antes de plantear el heurístico, como el tratar de identificar productos sustitutos y modelar solo uno; pero otras interacciones como desperdicios, no se pueden aislar certeramente y mejor se dejan en el heurístico propuesto tal cual.

Uno de los puntos que pocos artículos de la literatura revisada toman en cuenta es el impacto de la variedad en la ecología, el medio ambiente y el desarrollo sostenible. Esto es un punto importante ya que la variedad tiene una correlación negativa con las economías a escala y genera ineficiencias y desperdicios a lo largo de una empresa. La variedad excesiva en una compañía lleva hacia ganancias menores, pero la variedad excesiva en la sociedad lleva al desperdicio debido al reemplazo innecesario de productos usados y a los altos niveles de inventarios obsoletos [Tang, 1996].

2. COMPLEJIDADES DE LA VARIEDAD

Cuando se tiene un solo producto, la asignación de costos y la programación de la producción son tareas relativamente sencillas ya que todos los recursos se asignan a ese producto. En estos casos la complejidad solo resulta en estimar la demanda y definir los niveles de inventarios a lo largo de toda la cadena para satisfacer esa demanda de la forma más rentable. Sin embargo, son pocas las empresas que tienen un solo producto. Conforme se agregan más productos a la mezcla, haciéndola más amplia (más líneas de productos) o profunda (más productos de líneas ya existentes) se torna más difícil calcular la rentabilidad de un producto específico.

2.1. DEMANDA

Al tener un universo de productos muy grande, se generan productos complementos y productos sustitutos de la misma empresa que en ocasiones son difíciles de identificar. Un producto complemento es aquel cuya demanda está ligada a la demanda de otro, por ende sus demandas son dependientes. Por ejemplo, varios clientes del sector de la construcción, compran lámina recubierta, panel y el "despiece" o servicio de ingeniería. Si la empresa dejara de vender uno de esos dos productos o de proporcionar el servicio adicional, es probable que varios clientes se vayan con quien sí les ofrezca todo el paquete o sistema constructivo.

Un producto sustituto es aquel que el cliente aceptaría en lugar de otro. Por ejemplo, un cliente pequeño que compra el color "Blanco Nieve" podría comprar el color "Blanco Imperial" si se le ofrece a mejor precio o si se deja de vender el "Blanco Nieve". Por el contrario, McDonald's no aceptaría otro rojo que no fuera el "Rojo McDonald's" y si se deja de vender ese color se pierde ese cliente. Debido a lo anterior, se puede decir que entre el Blanco Nieve y el Blanco Imperial hay canibalismo, ya que a varios clientes les es indiferente cuál de los dos tonos de blanco elijan, sus demandas se empalman. De igual forma, aunque el "Rojo McDonald's" sea insustituible para McDonald's, ese color sí podría sustituir al rojo consumido por otros clientes más pequeños que estén de

acuerdo en tener el mismo tono de rojo que esa cadena. Otro caso que se puede dar es que productos pintados con un tipo de pintura caro puedan sustituir a los de un tipo de pintura más económico en los mismos tonos o viceversa, dependiendo de las garantías que requiera el cliente y de la diferencia de precio entre los tipos de pintura.

2.2. PROGRAMACIÓN DE LA PRODUCCIÓN

La producción de la empresa es aproximadamente 50% "made to stock" y 50% "made to order". Las sucursales y bodegas tienen niveles de inventario definidos para productos estándar, así que cuando bajan de esos niveles o cuando tienen stockouts se hace un pedido a la planta, por lo que parte de la producción es para reponer esos inventarios. Por otra parte, la oferta de la empresa es tan variada que muchos productos solo tienen ventas puntuales (se vende una vez a un cliente y luego ya no se vuelve a vender) o esporádicas, por lo que otra gran parte de su producción no se hace sino hasta que ya tiene un cliente específico. En el sistema de la empresa se registran todos los pedidos, tanto los de los clientes como los de las sucursales para reponer inventarios.

La programación de la producción se hace cada semana y puede sufrir modificaciones cada día; siempre se da en dos pasos. El primer paso es filtrar los pedidos en base a urgencia, disponibilidad del acero y disponibilidad de las pinturas. Si se cuenta en inventario con los insumos necesarios para fabricar el producto, se procede a la siguiente fase; si no, se hacen las requisiciones pertinentes y se reprograman los pedidos para una fecha posterior, tomando en cuenta los tiempos de entrega de los proveedores. El filtro por urgencia es más subjetivo ya que es una decisión que se toma entre los programadores y los vendedores considerando factores como importancia del cliente para la empresa, la apreciación del vendedor acerca de la necesidad de su cliente y el nivel de servicio que se les quiere dar a los clientes afectados. De igual forma si alguna línea está al tope de su capacidad, se filtran los pedidos en base a la contribución marginal histórica de los productos. Actualmente las líneas de pintado de la empresa son cuello de botella.

Una vez que se tienen los pedidos filtrados, estos son organizados para obtener la secuencia óptima de producción de manera que se haga la menor cantidad posible de setups. Los setups o tiempos de preparación son básicamente paros de línea para cambiar de rodillo aplicador de pintura o para cambiar la pintura o ambas cosas. Estos paros de línea son en los que tiene impacto el programador, ya que los demás paros de línea, como las fallas, no dependen de él.

Los rodillos se cambian ya que conforme se va pintando la lámina se va marcando el ancho de dicha lámina en el rodillo. Por lo tanto, un rodillo que ya pintó láminas de cierto ancho solo puede pintar láminas de anchos iguales o menores. Para que ese rodillo se pueda reutilizar para pintar anchos mayores, es necesario rectificarlo lo cuál se hace en un taller externo. El tiempo de cambio de rodillo es de aproximadamente 15-20 minutos dependiendo de la línea de pintado.

Dado que la lámina de puede pintar por ambos lados, el cambio de pintura se da tanto para el color de la cara como para el color del fondo de la lámina. Debido a que hay mucha más variabilidad en el color de la cara que en el del fondo y a que los clientes son menos estrictos en el color del fondo, el color cara es el de mayor impacto en los setups de pintura. El tiempo de cambio de pintura es de 25-30 minutos dependiendo de la línea de pintado.

Lo que los programadores hacen es ordenar los pedidos por ancho y por color cara de manera que hacen “conos” de producción donde agrupan cada color en anchos de mayor a menor y tratan de hacer el cambio de rodillo y de pintura al mismo tiempo, para pintar el siguiente color otra vez en anchos de mayor a menor, sin embargo no siempre es posible o no siempre es lo más conveniente. Por ejemplo, cuando escasean los rodillos porque están en el taller, se le da prioridad al ancho del producto y se hacen más cambios de pintura.

Después de cada paro de línea, además de desperdicios de solvente, pintura y primer, también puede haber otras ineficiencias y pérdidas. Por ejemplo velocidades de arranque

más lentas que la nominal de la línea, para el tipo de pintura que se esté utilizando, o toneladas de arranque perdidas, ya que en ocasiones las toneladas iniciales de un color no salen con la calidad o el tono solicitado.

2.3. INVENTARIOS

Se hacen estimaciones de demanda para la compra de los insumos, para la definición de los niveles de inventario y para las "apuestas" de las sucursales, especialmente las sucursales nuevas, donde se quiere impulsar algún producto nuevo. Sin embargo la principal forma de definir los inventarios es en base a la historia o estadística de ventas de un producto. Los productos que se venden 8 de los 12 meses del año se clasifican como productos de alto movimiento mientras que los productos con 90 días de inventario se clasifican como obsoletos.

Una consecuencia de ofrecer una gran variedad de colores es que aumenta la probabilidad de tener stockouts y de aceptar pedidos urgentes que modifiquen un plan de producción ya establecido. El efecto anterior se puede disminuir en cierta medida si se multiplican los inventarios de producto terminado y los de pinturas para tener un buen tiempo de entrega en los colores del catálogo. Al tener más productos diferentes en inventario y más pinturas se tiene mayor riesgo de que los productos se vuelvan obsoletos. Ni el producto terminado puede estar mucho tiempo en inventario ni las pinturas. El producto terminado se oxida y se va degradando su calidad hasta llegar a chatarra; la chatarra pintada no es la ideal para reciclar ya que la composición de las pinturas afecta la calidad del acero al fundirse. Las pinturas caducan a los 6 meses y es necesario confinarlas, lo cual tiene un costo, o mandarlas al proveedor para maquilar. Maquilar una pintura quiere decir que se recicla para producir pinturas nuevas en las cuales el proveedor le hace a la empresa algún descuento; sin embargo solo ciertos colores se pueden reciclar.

Algunos clientes piden los productos por tonelada, otros por metro cuadrado y otros por pieza por lo que es necesario hacer las conversiones respectivas. Ya sea al hacer esas conversiones o por mantener cierta productividad de las líneas o debido a que es

necesario protegerse por posibles fallas de calidad, en ocasiones se produce de más generando "overrollings" que también quedan en inventario. Si se ofrece una gran variedad de colores es menos probable que se desfogue ese inventario de sobrantes de producción.

2.4. TIEMPO DE ENTREGA

Al tener pocos colores en el catálogo se puede tener una mayor proporción en inventario, pero al ofrecer una gran variedad es necesario hacer distinciones o categorías de productos. Los colores estándar o del "Color Chart", se busca que sean los de mayor venta y los que se compromete a entregar desde inventario; donde el tiempo de entrega es solo el tiempo de tránsito al destino.

El tiempo de entrega es considerablemente mayor (aproximadamente 6 semanas) para los productos de un color que no está dentro del Color Chart ya que estos no se tienen en inventario de producto terminado y no se pueden programar en cualquier corrida estándar del color sino que se requiere una corrida de producción especial. Además, como se busca tener inventario de pintura sólo para los colores estándar, también hay que agregarle el tiempo de entrega de la materia prima. Por lo tanto, en los colores no estándar, el tiempo de entrega es el tiempo de abastecimiento de la materia prima, más el tiempo de cola o de espera para producción, más el tiempo de producción, más el tiempo de tránsito.

Con esta premisa se aceptan casi todos los pedidos de colores no estándar o especiales, sin embargo como se verá más adelante, es necesario castigar estos pedidos mediante un precio premium para que sean convenientes y no sólo con el tiempo de entrega ya que generan varios costos, desperdicios e ineficiencias ocultas. Uno de los riesgos ocultos generados es que se pierdan ventas de colores estándar al no cumplir con su tiempo de entrega, por no encontrarse en inventario; por fabricar pedidos urgentes de colores especiales en ocasiones no se tiene tiempo de reabastecer productos de stock a las sucursales o se les da menos prioridad. La política de precios y la política de aceptación

de nuevos colores deben ser revisadas y redefinidas, pero para ello es necesario poder cuantificar los costos ocultos generados. De igual forma, al ofrecer una mayor variedad de colores los pedidos tienden a ser de menor volumen y se corre mayor riesgo de no poder consolidar embarques a los destinos, pagando entonces fletes "falsos" (pagar por un medio de transporte y no aprovechar toda su capacidad).

2.5. COSTEO

Es muy común el aumento y la disminución de precios en los productos siderúrgicos debido a que dependen mucho del costo de las materias primas del acero. A su vez, el costo de las materias primas, fierro esponja y chatarra, depende fuertemente de otros factores, como el costo de los energéticos para el fierro esponja (luz y gas principalmente) y de la disponibilidad de la chatarra. Cualquier variación de estos insumos es necesario reflejarla en el precio.

Se asigna a específicamente a cada producto aquellos costos que se le puedan relacionar directamente. Por ejemplo a una tonelada de lámina galvanizada y pintada se le carga específicamente la cantidad de kilogramos del acero que se utilizó, la cantidad de gramos de zinc para el galvanizado y la cantidad de litros de pintura que se utilizaron para pintarla. Ya que hay diferentes proveedores del mismo acero, los cuales tienen precios diferentes, se obtiene un precio promedio y es el que se utiliza para todos los productos que utilicen ese tipo de acero. La misma metodología se utiliza para las demás materias primas, por ejemplo, las pinturas. Así como hay varios proveedores de acero también hay varios proveedores de pintura y lo que se asigna en los costos de un producto es el costo promedio de la pintura que se compró a los diferentes proveedores, del color de dicho producto. La aplicación de ese costo es tanto por tipo de pintura como por color. Ciertos tonos de los tipos de pintura más baratos, pueden ser tan caros como ciertos tonos de los tipos de pintura más caros y viceversa, dependiendo de la brillantez y la rareza del tono además de la composición de la pintura.

Los costos debido a mano de obra, desperdicios y reprocesos, se asignan de forma global a todos los productos que pasan por ese proceso. Por ejemplo, todos los productos pintados llevan en su costo una proporción igual (prorrateada por tonelada) del desperdicio y confinamiento de pintura, solvente y primer del mes, del costo de rectificar los rodillos, del costo de las toneladas desperdiciadas en los arranques de las líneas de pintado, entre otros.

2.6. TRADEOFFS

La variedad en la oferta puede ser algo atractivo a pesar de los costos ocultos que implica y en ocasiones es algo necesario, para ser sobrevivir en los mercados actuales. La variedad de productos aumenta con la competitividad del mercado [Tang 1996]. Esto impacta de forma positiva y negativa en diferentes aspectos del negocio. El primer paso para modelar los diferentes escenarios de portafolios de productos es identificar las ventajas y las desventajas relacionadas con la variedad ofrecida a los clientes.

2.6.1. Ventajas - Beneficios

Mejor imagen: Diversos estudios [Ramdas, 2003] han demostrado que, en general, a mayor variedad percibida por el cliente más confianza en el fabricante. Por ejemplo, un fabricante que puede hacer 20 modelos de refrigeradores diferentes se considera un especialista mientras que aquel que solo puede producir un modelo no.

Acceso a otros mercados: Al aumentar la variedad en la oferta, se puede tener presencia en varios mercados y por ende se aumentan la demanda y los ingresos, siempre y cuando la variedad no sea en productos sustitutos.

Demanda Estable: Con productos complemento se puede impulsar y asegurar la demanda de algún otro producto mediante sinergia. Se puede estabilizar la demanda teniendo productos con diferentes patrones de estacionalidad. Al ofrecer varios productos se aminora el efecto de una caída de ventas en un producto particular.

Demanda Cautiva: Los productos de mayor volumen generalmente son productos "commodities" mientras que productos de menor volumen son generalmente productos diferenciados, únicos o personalizados para clientes específicos de los cuales se puede obtener un mayor margen y a la vez retención de los clientes.

2.6.2. Desventajas - Perjuicios

Mayor complejidad en la planeación: Una gran variedad de productos hace más difícil las proyecciones de demanda debido a las interacciones que hay entre productos, como sustitutos o complementos. Debido a esto, la planeación de los abastecimientos y de la producción se vuelve más compleja.

Mayores inventarios: Al ofrecer una mayor variedad de productos, es necesario tener una mayor cantidad de inventarios tanto de materia prima, como de producto en proceso y producto terminado para poder ofrecer un nivel de servicio adecuado. Además, el riesgo de obsolescencia tanto en producto terminado como en materia prima aumenta. Por ejemplo, no es raro que en la empresa se compre un tambo de pintura especial para un pedido en particular y el sobrante de ese tambo de pintura permanece en inventario hasta caducar; ya no se vuelve a utilizar por ser un color tan poco frecuente.

Mayores costos administrativos: La cantidad de pedidos para llenar la capacidad de producción con una mayor variedad de productos aumenta, ya que se atienden pedidos de menor volumen. Esto conlleva a más tiempo invertido por parte de administradores y vendedores al procesamiento de estos pedidos.

Costo de oportunidad: Este no es fácil de cuantificar en dinero pero sí en tiempo. El tiempo perdido en setups disminuye la capacidad de la línea de producción. Además, se pueden generar ineficiencias por desperdicios y velocidades de arranque.

4. JUSTIFICACIÓN

Una de las premisas de la empresa que le da una ventaja competitiva, es el enfoque de servicio al cliente. Este enfoque le ha permitido ingresar al mercado del acero recubierto en los Estados Unidos, ofreciendo una gama de productos muy amplia y flexible. Esto ha llevado a la empresa a una disyuntiva, ya que otro de sus objetivos, como el de todas las empresas privadas, es satisfacer las expectativas de rentabilidad de sus accionistas. El ofrecerle al cliente total flexibilidad en las características de sus pedidos (tonelaje, color, calibre, entre otras) crea mucha variabilidad en las líneas de pintado por lo que su productividad se ve disminuida. El presente proyecto busca enfrentar esta problemática con un modelo que le sirva a la empresa como herramienta en su toma de decisiones en cuanto al manejo de la variedad de productos. Se busca la cantidad óptima en por lo menos una de las características de la mezcla de productos: el tipo de pintura y color.

Este modelo podría servir más adelante para poder definir una política de precios y una política de aceptación de nuevos colores; ya que al poder cuantificar correctamente los costos de oportunidad se puede definir cuánto hay que cobrar por un color nuevo para que este sea rentable, considerando todos los costos e ineficiencias que genera el aumentar la gama de productos ofrecida. Lo que se busca es que no subsidien unos productos a otros o que si lo hacen sea por alguna estrategia de comercialización y no porque simplemente la empresa no se da cuenta de ello.

5. MÉTODO Y MODELOS

El método propuesto para la solución del problema planteado, consiste en un modelo heurístico de optimización que se presenta más adelante, a detalle. Debido a la complejidad del problema, plantearlo de forma heurística, nos da más flexibilidad y precisión en su construcción. Por ejemplo, en esta forma nos permite tener una relación entre cantidad de colores y número de setups más realista que no esté limitada a ser lineal, inclusive se puede tener todo un simulador que obtenga un valor de cantidad de setups estimada en base a una demanda simulada estadísticamente o en base a la demanda real, tomando en cuenta también otras características de los productos como lo son el calibre y el ancho o su relación con la decisión de atender la demanda de otros productos complementos o sustitutos. Esto se debe a que el método seleccionado funciona a través de varias iteraciones.

En cada iteración se puede llamar a una función o un simulador para que nos otorgue valores de cantidad de setups, inventarios, contribuciones marginales y demandas en base al escenario que se está analizando. Por ejemplo, si el problema decide dejar de hacer algún color blanco se puede pensar que aumentaría la demanda de otros tonos blancos o que si se decide dejar de hacer algún color en tipo de pintura flurocarbonada podría aumentar la demanda de otros colores flurocarbonados para sustituir esa demanda. Como se indicó antes, podría ser que al dejar de vender cierto color disminuya la demanda de otro que se vende en conjunto, como complemento o que aumente la de un sustituto. Para modelar de forma entera la complejidad del problema incrementa excesivamente y aún así no se podrían representar todas las interacciones entre las variables explicadas anteriormente.

Se encontraron en la literatura del tema, métodos que ilustran costos ocultos o indirectos de la complejidad. Uno de estos métodos, el de Hewlett-Packard [Cargille, 2005], estima estos costos de complejidad por unidad. Un problema que se le ve a esa metodología es que los costos de complejidad generalmente son costos en función de alguna otra variable del escenario analizado. Además, al tener muchos productos ¿a cuál se le hecha la culpa

por la existencia de los demás? Por ejemplo, al pintar un rollo de color rojo y hacer un cambio de pintura para pintar ahora un rollo de color verde, se tira pintura roja de las charolas y se pierde tiempo en lo que se preparan las charolas para pintar el color verde. ¿A qué color se le carga la pintura roja tirada y qué color el tiempo de setup? La pintura que se tiró fue roja, pero si el programador los hubiera puesto en diferente orden la pintura tirada sería verde y el tiempo perdido sería en preparar la pintura roja. Por lo tanto, conviene estudiar el impacto de la variedad como un total en cada escenario de portafolio de productos y no como una medida por unidad de producto.

5.1. CASO GENERAL

Max $\boldsymbol{G(X, P)}$ (Modelo General)

s.t.

$\boldsymbol{(X, P) \in F}$

G: Ganancia (sin considerar costos administrativos)
X: Demanda que se elige atender o ventas discrecionales
P: Precios de venta
F: Conjunto de soluciones factibles

Las ventas de los productos dependen de muchos factores, entre ellos está la relación precio/valor del producto de la empresa. Esto implica que la política de precios que define la compañía tiene influencia sobre las ventas, lo cual es cierto. El portafolio de productos ofrecido también afecta las ventas, debido a los productos complemento y sustitutos por lo que hay interacciones entre las ventas de los productos. Para el modelo general, las ventas son función de **X** y de **P**. Sin embargo, para el modelo particular **P** se mantendrá constante por lo que las ventas y la ganancia serán solo función de **X**.

$$\boldsymbol{G(X)} = \boldsymbol{X}^T \boldsymbol{P} - Cvariables(\boldsymbol{X}) - Cdesperdicios(\boldsymbol{X}) - Cinventarios(\boldsymbol{X}) \quad (1)$$

En esta ecuación, la ganancia se define como los ingresos por ventas ($\mathbf{X}^T \mathbf{P}$) menos los costos variables, de desperdicios y de inventarios. Se selecciona esta definición para cubrir todos los costos en que influye el seleccionar una demanda **X**. El único costo que no está incluido directamente aquí es el de los stockouts o ventas perdidas, sin embargo ese se toma en cuenta indirectamente en los costos de inventarios, los cuales se explican más delante.

En los costos variables, **X** puede tener influencia en cuanto a los precios de compra de la materia prima. Si se decidiera pintar un solo color, por ejemplo el Blanco Imperial, probablemente se obtendrían descuentos por volumen por parte de los proveedores de esa pintura.

En los costos de desperdicios, **X** tiene influencia ya que los desperdicios generados dependen de los productos que se decida producir. Algunos de estos costos pueden ser considerados dentro del costo variable de cada producto ya que su causa es directamente la producción del mismo. Pero otros costos por desperdicios no se pueden incluir en ese elemento ya que en realidad dependen de las interacciones entre uno o más productos, por ejemplo el caso del setup entre el rollo rojo y el rollo verde descrito en la sección anterior. Ese tipo de costos va en este rubro.

En el modelo propuesto, el tiempo de entrega y la disponibilidad de producto son variables definidas por la empresa de acuerdo al nivel de servicio que se desea ofrecer al cliente en cada producto. En base a lo anterior, se ajustan los niveles de inventario requeridos por producto y es aquí donde entra el cuarto elemento de la función objetivo (1), Cinventarios(**X**). Para el modelo propuesto, los inventarios requeridos por producto deben calcularse con algún estudio aparte que considere tiempos de abastecimiento, de tránsito, de cola, de producción, de ciclo, etc. El modelo general toma como entrada los niveles requeridos de inventario por producto y además calcula otros inventarios, como los de materias primas. Ya que hay productos que utilizan las mismas materias primas, este costo también depende de **X** debido a las interacciones entre productos. Por ejemplo, el inventario de pintura roja pudiera ser el mismo para pintar un producto rojo que para

pintar tres productos rojos, pero el inventario de producto terminado total sí sería menor con un producto rojo que con tres.

La capacidad disponible de producción, en unidades, depende de la mezcla de productos que se esté analizando, o sea **X**. La mezcla impacta en el tiempo utilizado debido a las ineficiencias que se generan por los setups y a las diferentes velocidades de producción de los productos. La velocidad de las líneas de pintado en toneladas por hora varía dependiendo del tipo de pintura-color, el calibre y el ancho. El modelo general se plantea para considerar todas las características de los productos de **X**. En el modelo particular, por cuestiones de simplificación y aplicabilidad se considera solo el tipo de pintura-color de **X**. Limitar la capacidad de producción y por lo tanto **X**, dependiendo de la mezcla, es la forma de estos modelos de considerar el costo de oportunidad.

Ya que la ganancia **G** se definió en función de las ventas y de los costos, y a su vez estos están en función de **X** y de **P**, **G** es función de **X** y de **P** como lo plantea el modelo general. Una vez que se tienen bien definidas las funciones de las ventas y de los costos en base a **X** y **P** se pudiera probar, en un futuro, con varios escenarios de precios para llegar a definir una política de precios.

5.2. CASO PARTICULAR

El modelo particular tiene ciertas simplificaciones con respecto al modelo general por dos motivos. Por una parte, no se contaba con los recursos necesarios para llevar a cabo estudios extensivos en las áreas que implica el tratar de cubrir todos los costos de la empresa, como se explica más delante. Por otra parte, lo que se desea probar es que dicho modelo se puede programar de forma heurística y se propone un heurístico para ello.

Para poder evaluar el desempeño del heurístico se compara con un método de programación entero mixto con variables binarias el cuál sí asegura optimalidad dentro de ciertos supuestos. Dado que la programación entera solo acepta relaciones lineales entre

las variables, se utiliza la misma relación lineal para calcular la cantidad de setups por pintura tanto en el programa entero como en el heurístico, de manera que sean comparables. Sin embargo, con el heurístico tenemos más opciones ya que las relaciones lineales pueden ser más complejas que una simple ecuación; pudieran incluso ser funciones, sub-modelos o simuladores y eso no afecta el planteamiento del heurístico propuesto debido a su construcción modular. Ver Figuras 1 y 2 más delante.

La cantidad de setups por pintura, se eligió como una de las variables más importantes a modelar ya que tienen impacto doble. Impactan como costos de desperdicio en Cdesperdicios(**X**) (1) e impactan en la misma **X** al ponerle un tope por el tiempo perdido que se le resta a la capacidad o tiempo disponible. Debido a que la alerta sobre la necesidad de racionalización de productos fue la cantidad excesiva de colores, el color es la característica particular que este modelo particular busca racionalizar.

La relación lineal para los setups por pintura se obtuvo con datos reales de las líneas de pintado, relacionando la cantidad de colores que se pintan al mes con la cantidad de paros para cambio de pintura que se realizan al mes y quedó de la siguiente forma:

$$CSp(\mathbf{X}) = k * \Sigma_i Z_i + b \qquad (2)$$

CSp: Cantidad de setups por pintura

Z(**X**): Cantidad de colores seleccionados,

$$Z_i = \begin{cases} 0, & X_i = 0 \\ 1, & X_i > 1 \end{cases}$$

k: Pendiente de la relación lineal

b: Constante de la relación lineal

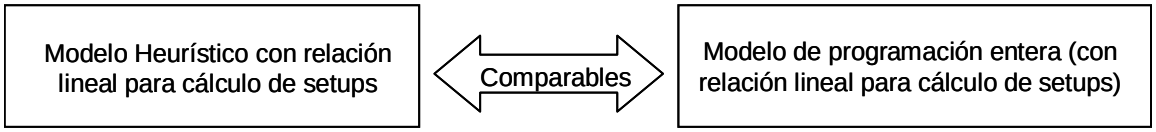

Figura 1. Heurístico comparable con modelo de programación entera.

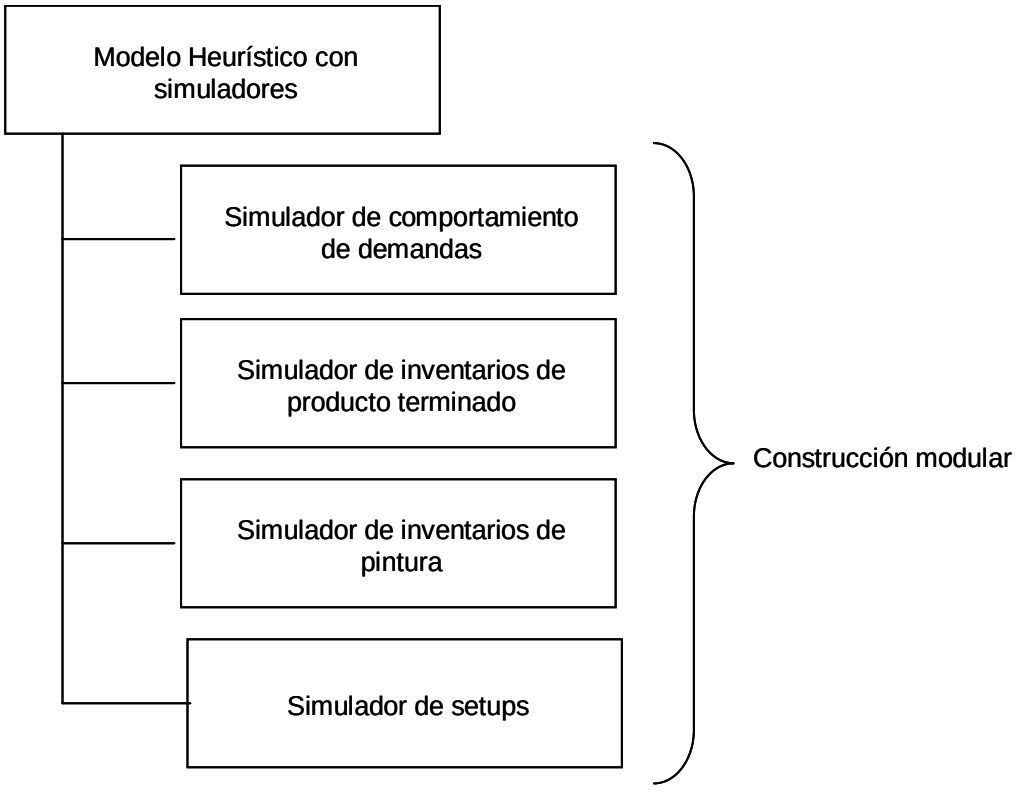

Figura 2. Potencial del heurístico propuesto

Se consideró la modelación lineal de los cambios de pintura como un enfoque aceptable para obtener conclusiones válidas; mayor detalle en simular la programación actual no hubiera generado mucho más valor agregado. Una vez que se tenga un método estandarizado para la programación de la producción sería adecuado incluirlo en este modelo debido a que realmente aumentaría su precisión; actualmente dos programadores de la empresa podrían programar de forma diferente la misma producción.

El tiempo perdido por setups de pintura y el desperdicio por setups de pintura se obtienen en base a la cantidad de setups, por lo que dependen de la misma relación lineal.

$$TSp(\mathbf{X}) = sp * CSp(\mathbf{X}) \qquad (3)$$

TSp: Tiempo total perdido en setups
sp: Tiempo perdido por cada setup de pintura

$$Cdesperdicios(\mathbf{X}) = Cpint * CSp(\mathbf{X}) \quad (4)$$

Cdesperdicios: Costo total por desperdicios de pintura, en función de CSp y por ende **X**
Cpint: Costo de pintura, solvente y primer desperdiciado en cada setup

Se consideró que en cada setup se pierden 30 minutos y se desperdician 50 Lts con costo promedio de 7.0 dlls / Lt. La pintura desperdiciada se debe al remanente que queda tanto en los rodillos aplicadores de pintura como en las charolas de donde absorben la pintura. Esa pintura y primer se remueven con solvente y son litros que hay que confinar debido a que ya están contaminados.

Además de los setups por pintura hay otros paros de línea debidos a otras causas, como fallas o cambios de rodillo. Los cambios de rodillo son afectados más que nada por el ancho de los productos y no se simulan en este caso particular. De igual forma, hay ineficiencias en la línea al cambiar de calibre entre un producto y otro ya que es necesario pasar "hebras" o toneladas en falso que sirven para pegar láminas de diferentes espesores. Esas hebras se reutilizan y por lo tanto no representan un gran costo en términos de costo por desperdicio pero sí impactan en términos de costo de oportunidad; no obstante dado que el calibre tampoco es característica de interés en este análisis, tampoco se toma en cuenta. De igual manera los paros debido a fallas en las líneas tampoco se simulan. Lo único que se considera con respecto a estos otros setups es su tiempo total mensual, en base a la estadística de la empresa para el 2004, se le resta a la cantidad de horas disponibles de la línea y el modelo parte desde esa capacidad de línea modificada.

$$Cap = Tdisponible - Totros \quad (5)$$

$$Cap\_restante(\mathbf{X}) = Cap - TSp(\mathbf{X}) \quad (6)$$

$$Cap\_necesaria(\mathbf{X}) = \mathbf{A}^{T}\mathbf{X} \quad (7)$$

Cap: Capacidad modificada, es la capacidad total para propósitos del modelo particular
Cap_restante(**X**): Capacidad restante debido al tiempo perdido por setups de pintura
Cap_necesaria(**X**): Capacidad necesaria para atender **X**

A: Tiempos de producción por tonelada

Tdisponible: Tiempo disponible total de la línea de pintado

Totros: Tiempo perdido total por paros debidos a fallas, mantenimientos o setups de rodillo entre otros.

Algunos otros datos que utiliza el modelo particular están basados en el promedio de la estadística de la empresa para el 2004 y son: la demanda, el precio, el costo variable y los niveles de inventario. Las razones se describen a continuación.

La empresa no cuenta con simuladores de demanda con las interacciones que se quieren probar y no se cuenta con el conocimiento suficiente del tema para simularlas. La empresa tiene desarrollados más de 1,500 colores diferentes y vende aproximadamente 240 colores diferentes al mes y 600 colores diferentes al año. Estos colores engloban en sí varias naturalezas de pintura y varios productos (diferentes aceros, formas, calibres, anchos, acabados, etc), por lo que es difícil para este caso modelar las ventas en función de **X** y **P**. Hay colores que pueden ser sustitutos o complementos de otros pero solo en ciertos mercados (por ejemplo mercado nacional vs exportación) o solo en ciertas condiciones (algunos mercados piden más garantías) y se hacen en la misma línea de pintado. Para este caso en particular, el modelo de ventas en función de **X** y **P** sería tema de otra tesis que requeriría otro estudio y mayor acercamiento con los vendedores y clientes.

En el modelo particular, se utiliza $\mathbf{CM}^T\ \mathbf{X}$ en vez de los términos $\mathbf{X}^T\mathbf{P}$ – Cvariable(**X**) del modelo general. Algebraicamente es lo mismo ya que para cada producto se está tomando el precio y los costos variables del 2004, como constantes y no como variables en función de **X**.

$$\boldsymbol{CT(X) = CM}^T\boldsymbol{X} = \boldsymbol{X}^T\boldsymbol{P} - \mathit{Cvariables(X)} = \boldsymbol{X}^T\boldsymbol{P} - \boldsymbol{X}^T\boldsymbol{CV} \qquad (8)$$

CT(**X**) = Contribución total

CM: Contribución marginal por producto

X: Demanda que se elije atender o ventas discrecionales

P: Precios por producto (como constantes)

CV: Costos variables por producto (como constantes)

La empresa no tiene un modelo para simular los niveles de inventarios de manera sistemática. Hay varias propuestas de políticas nuevas para manejar los inventarios, que se pudieran probar con este modelo, pero siguen en definición. La lógica que se siguió fue que, para modelar un escenario con un nivel de servicio igual al actual, sin importar si es bueno o malo, se utilizaron los inventarios que se requirieron para obtener las ventas registradas en el mismo período, el 2004. Para los inventarios lo que se le asigna a cada color son los inventarios promedio de pintura en los almacenes y los inventarios promedio de todos los productos terminados de ese color en las sucursales. El costo total de inventarios por cada color se obtiene como:

$$Cinventarios(i) = TIR * Cpint(i) * Ltsinv(i) + TIR * Cprod(i) * Tonsinv(i) \quad (9)$$

Cinventarios(i): Costo de inventarios debido al color i

Cpint(i): Costo/Lt de la pintura de color i

Ltsinv(i): Litros promedio en inventario de la pintura de color i

Cprod(i): Costo/ton de los productos de color i

Tonsinv(i): Toneladas promedio en inventario de los productos de color i

TIR: Tasa de para evaluar proyectos en la empresa

Como se puede constatar, todo el modelo está en función de **X**. Con las variables y ecuaciones previamente definidas se plantea el modelo particular así:

$$Max\ \boldsymbol{G} = \boldsymbol{CM}^T \boldsymbol{X} - Cinventarios(\boldsymbol{X}) - Cdesperdicios(\boldsymbol{X}) \qquad \text{(Modelo Particular)}$$

s.t.

$$\boldsymbol{D}_{min} \leq \boldsymbol{X} \leq \boldsymbol{D}_{max}$$

$$\boldsymbol{A}^T \boldsymbol{X} \leq Cap\_restante(\boldsymbol{X})$$

$$\boldsymbol{X} \geq 0$$

CM: Contribución marginal del 2004, de cada producto (precio – costo variable)

A: Velocidades de producción de **X**

$\mathbf{D_{max}}$: Demandas máximas, en base a un promedio histórico

$\mathbf{D_{min}}$: Demandas mínimas, en base a un promedio histórico

Como se puede ver en este modelo particular, los precios **P** se han tomado como constantes. La evaluación del modelo es para el escenario de precios actual, por lo que la ecuación (1) se transformó con las simplificaciones en:

$$\boldsymbol{G(X)} = CT(\boldsymbol{X}) - Cinventarios(\boldsymbol{X}) - Cdesperdicios(\boldsymbol{X}) \qquad (10)$$

En resumen, los supuestos del modelo matemático particular son:

- Precio, costos variables y contribución marginal constantes o fijos por color
- La demanda se puede manejar a discreción dentro de un rango pequeño (+10%, -15% del promedio del 2004)
- El tiempo total perdido en setups es proporcional al número de colores seleccionados. Se estimó utilizando una regresión lineal que modela la cantidad de setups en función de la cantidad de colores seleccionados (datos de la empresa para el 2004).
- Se utiliza una tasa financiera del 20% para los costos de inventario.

Toda la programación se hizo en Microsoft Visual C++ 6.0 y para la parte de programación lineal se utilizó la librería LP_SOLVE 5.5.

5.2.1. Programación entera mixta

El planteamiento del problema mediante programación entera mixta es el siguiente:

$$Max\ G = \Sigma_i\, CM_i\, X_i - \Sigma_i\, (k*Cpint + Cinventarios_i)\, Z_i - b*Cpint$$

s.t.

$X_i \leq D_{max,i}$

$X_i \geq D_{min,i}$

$\Sigma_i A_i X_i + sp*k*\Sigma_i Z_i \leq Capm$

$X_i \geq 0$

$Z_i : \{0,1\}$

G: Ganancia total del plan

X_i: Producción de i, en toneladas

CM_i: Contribución marginal del producto i

$Cinventarios_i$: Costo de inventarios del producto i

$D_{max,i}$: Demanda máxima de i

$D_{min,i}$: Demanda mínima de i

A_i: Tiempo de producción de i, hrs/ton

k: Pendiente de la ecuación lineal de la cantidad de setups (2)

b: Constante de la ecuación lineal de la cantidad de setups (2)

sp: Tiempo perdido por cada setup de pintura (3)

Cpint: Costo de pintura, solvente y primer desperdiciado en cada setup (4)

Z(**X**): Cantidad de colores seleccionados,

$$Z_i = \begin{cases} 0, & X_i = 0 \\ 1, & X_i > 1 \end{cases}$$

Capm: Capacidad de producción disponible, restándole el tiempo de la cantidad constante de setups de pintura

$$Capm = Cap - sp*b \qquad (11)$$

5.2.2. Heurístico llenador por ganancia total

Este método consiste en varias iteraciones. Su algoritmo es el siguiente:

1. Se considera que no se ha seleccionado ningún color, G(**X**=Ø) = 0, (1).
2. Se repite el siguiente proceso mientras siga aumentando G(**X**) y mientras Cap_restante(**X**) (6) > Cap_necesaria(**X**) (7), cuando ya no se cumplan estas condiciones se pasa al punto 3.

a. Para todo color i que no forme parte de **X**:
    i. Se agrega temporalmente el color i al conjunto de ventas discrecionales, **X**
    ii. Se calculan la cantidad de colores Z(**X**+i) y la cantidad de setups de pintura en base a la cantidad de colores, CSp(**X**+i), (2).
    iii. Se calculan el tiempo perdido en setups, TSp(**X**+i) (3), la capacidad restante Cap_restante(**X**+i) (6) y el costo de la pintura tirada en los setups, Cdesperdicio(**X**+i), (4)
    iv. Se calculan los costos por inventarios pintura y producto terminado, Cinventarios(**X**+i) (9).
    v. Si Dmax(i) ≤ Cap_restante(**X**+i), se atiende toda su demanda; si Dmax(i) > Cap_restante(**X**+i), se produce sólo hasta llenar Cap_restante(**X**+i).
    vi. Se calcula y se guarda la ganancia **G**(**X**+i) (10) generada al agregar temporalmente el color i al conjunto **X** y se remueve i del conjunto **X.**
b. Se comparan todas las **G**(**X**+i), si se cumple que una o varias **G**(**X**+i) > **G**(**X**), se escoge la mayor de ellas y se agrega definitivamente el color i a **X**.
   X = **X** + i
c. Se regresa al inciso a.

3. Se despliegan los colores seleccionados como elementos de **X**, el orden en que fueron seleccionados, su producción y la ganancia total generada por ese escenario

Este modelo heurístico nos permite tomar decisiones basadas en el impacto global de agregar un color a **X**. Es decir, cada color i puede tener una contribución marginal por sí solo, pero al agregarlo a la producción genera costos, ineficiencias y desperdicios que no son fáciles de ver debido a sus interacciones complejas y a que varían de acuerdo a **X**. Aún y cuando la contribución marginal del color i sea positiva, puede resultar que G(**X**+i) < G(**X**) debido a los otros costos que se generan y que no contempla la contribución marginal.

A manera de ejemplo y demostración, se resolvió este mismo "Heurístico llenador por ganancia" de dos formas alternas a la que se describió; donde lo único que cambia es el parámetro para selección del color i a agregar a **X**. La segunda forma, "Heurístico llenador por ganancia/ton" lugar de comparar la ganancia total generada, **G**(**X**+i), compara la ganancia total generada / tonelada, **G**(**X**+i) / $\mathbf{A}^{T}$ (**X+i**). La tercera forma, "Heurístico llenador por ganancia/hr" lugar de comparar la ganancia total generada, **G**(**X**+i), compara la ganancia total generada / hora, **G**(**X**+i) / Cap_necesaria(**X**+i). Los resultados de los tres heurísticos se comparan en la sección de resultados.

No es raro ver que en la industria se aplique el criterio de seleccionar el producir un producto en base solo a su contribución marginal por tonelada o a su contribución marginal por hora. Aunque de las dos anteriores, la contribución marginal por hora es generalmente un mejor parámetro de comparación ya que ahí se pueden incluir más fácilmente las ineficiencias; lo mejor es analizar el escenario de producción y ventas como un todo y basarse en la ganancia generada por todo el portafolio de productos y sus interacciones.

5.2.3. Heurístico llenador mediante programación lineal

Este método también consiste en varias iteraciones y su algoritmo es el siguiente:

1. Se considera que no se ha seleccionado ningún color, G(**X**=Ø) = 0, (1).
2. Se repite el siguiente proceso mientras siga aumentando G(**X**) y mientras Cap_restante(**X**) (6) > Cap_necesaria(**X**) (7), cuando ya no se cumplan estas condiciones se pasa al punto 3.
    a. Para todo color i que no forme parte de **X**:
        i. Se agrega temporalmente el color i al conjunto de ventas discrecionales, **X**
        ii. Se calculan la cantidad de colores Z(**X**+i) y la cantidad de setups de pintura en base a la cantidad de colores, CSp(**X**+i), (2).

iii. Se calculan el tiempo perdido en setups, TSp(**X**+i) (3), la capacidad restante Cap_restante(**X**+i) (6) y el costo de la pintura tirada en los setups, Cdesperdicio(**X**+i), (4)

iv. Se calculan los costos por inventarios pintura y producto terminado, Cinventarios(**X**+i) (9).

v. Se resuelve el siguiente modelo de programación lineal:

$$Max\ CT = \Sigma_i\, CM_i\, X_i$$

$$s.t.$$

$$X_i \leq D_{max,i}$$

$$X_i \geq D_{min,i}$$

$$\Sigma_i\, A_i X_i \leq Cap\_restante$$

$$X_i \geq 0$$

vi. Se calcula y se guarda la ganancia **G**(**X**+i) (10) generada al agregar temporalmente el color i al conjunto **X** y se remueve i del conjunto **X.**

b. Se comparan todas las **G**(**X**+i), si se cumple que una o varias **G**(**X**+i) > **G**(**X**), se escoge la mayor de ellas y se agrega definitivamente el color i a **X**.
X = **X** + i

c. Se regresa al inciso a.

3. Se despliegan los colores seleccionados como elementos de **X**, el orden en que fueron seleccionados, su producción y la ganancia total generada por ese escenario

En este caso el heurístico es similar al anterior solo que la decisión de cuánto producir de cada color ϵ {**X**+i} se toma mediante programación lineal en cada iteración. En otras palabras se compara el verdadero óptimo de cada escenario mientras que el heurístico anterior nos comparaba soluciones buenas de cada escenario. Este heurístico tiene la función de buscar en el universo de escenarios posibles y comparar sus soluciones óptimas.

5.2.4. Heurístico llenador mediante programación lineal con solución inicial

Con el objetivo de darle más velocidad al heurístico anterior, se plantea este heurístico donde la única diferencia es que se parte desde un conjunto **X** inicial bueno en vez de un conjunto **X** inicial vacío, para llegar más rápido a una solución final buena.

El conjunto inicial **X** se construye resolviendo primero el problema mediante el "Heurístico llenador por ganancia" y el "Heurístico llenador por ganancia/hr", los cuales son heurísticos muy rápidos porque involucran pocas operaciones. Por otro lado, el "Heurístico llenador mediante programación lineal" realiza más operaciones al momento de incluir un modelo de programación lineal en su procedimiento. La solución inicial es la intersección de ambas respuestas.

$$\boldsymbol{X}_{solución\ inicial} = \boldsymbol{X}_{ganancia\ total} \cap \boldsymbol{X}_{ganancia\ /\ hr} \qquad (12)$$

Otra opción para obtener una solución inicial, es plantear el problema de forma tipo "knapsack" o "problema de la mochila" y resolverlo para después hacer una búsqueda local. Los problemas tipo "knapsack" son de programación entera binaria y tienen una sola restricción. La función objetivo involucra los beneficios que se obtienen al seleccionar un elemento y la restricción involucra los costos o la capacidad que ocupa cada elemento.

Este planteamiento requiere incluir en la restricción de capacidad el tiempo perdido en setups de pintura. Debido a la relación lineal que se está utilizando, se puede incluir esa pérdida de capacidad en el tiempo de producción de cada color. Sin embargo, si se encuentra una relación no lineal más adecuada o se simulan de forma más precisa los paros por cambio de pintura, ya no se podrían incluir en la restricción de capacidad.

De igual forma, este planteamiento requiere incluir en la función objetivo los costos de pintura desperdiciada, los cuales están en función de la misma relación lineal mencionada en el punto anterior. Otro costo a incluir en la función objetivo es el costo de inventarios

de pintura y de producto terminado. En este caso particular ya que no se plantearon interacciones que afecten los inventarios se puede poner un costo de inventarios independiente por color; sin embargo si se desea en un futuro darle más precisión al modelo al utilizar algún simulador de inventarios, ya no se podría incluir en la función objetivo esos costos.

$Max\ G = \Sigma_i\, CT_i\, Z_i - Cpint * b$

s.t.

$\Sigma_i\, AT_i Z_i \leq Capm$

CT_i: Contribución total del color i

$$CT_i = Dmax_i * CM_i - Cinventarios_i - sp * k \qquad (13)$$

AT_i: Tiempo requerido para producir $Dmax_i$

$$AT_i = Dmaxi * Ai + sp * k \qquad (14)$$

Capm: Capacidad modificada (11)

k: Pendiente de la ecuación lineal de la cantidad de setups (2)

b: Constante de la ecuación lineal de la cantidad de setups (2)

Z_i: Indica los estados "seleccionado" o "no seleccionado"

$$Z_i = \begin{cases} 0, & color\ no\ seleccionado \\ 1, & color\ seleccionado \end{cases}$$

Para el método knapsack, se pueden ordenar los productos o en este caso los colores de acuerdo a su razón beneficio / capacidad ocupada. Ese ordenamiento es similar a lo que hace el "heurístico llenador por hora" que se describió anteriormente. Como la misma literatura demuestra, no siempre los "mejores" elementos son seleccionados en la solución óptima. Esto se debe a que en ocasiones, el incluir al "mejor" elemento se puede impedir que se incluyan otros elementos buenos que en conjunto proveen un beneficio total mayor al aprovechar toda la capacidad disponible. En el caso knapsack el elemento con la mejor relación beneficio / capacidad ocupada puede dejar capacidad disponible que nadie más puede utilizar, en el caso nuestro el incluir varios de los mejores elementos provoca que la capacidad disponible disminuya. [Winston, 2003]

Como se puede ver, este método solo es útil si se utilizan las relaciones lineales planteadas para calcular los setups de pintura en función de la cantidad de colores y los inventarios por color. Una ventaja de los heurísticos es que no dependen de esa relación lineal mientras que la programación entera mixta y la solución inicial tipo knapsack sí. De igual forma, el método knapsack tampoco permite valores intermedios de producción ni considera las interacciones entre las ofertas, inventarios y demandas de los colores.

5.2.5. Heurístico vaciador mediante programación lineal

Este método consiste en varias iteraciones y su algoritmo es el siguiente:

1. Se considera que se han seleccionado todos los colores, con su respectiva G(**X**) (1).
2. Se repite el siguiente proceso mientras siga aumentando G(**X**), cuando ya no se cumpla esta condición se pasa al punto 3.
    d. Para todo color i que forme parte de **X**:
        i. Se resta temporalmente el color i al conjunto **X**
        ii. Se calculan la cantidad de colores Z(**X**–i) y la cantidad de setups de pintura en base a la cantidad de colores, CSp(**X**–i), (2).
        iii. Se calculan el tiempo perdido en setups, TSp(**X**–i) (3), la capacidad restante Cap_restante(**X**–i) (6) y el costo de la pintura tirada en los setups, Cdesperdicio(**X**–i), (4)
        iv. Se calculan los costos por inventarios pintura y producto terminado, Cinventarios(**X**–i) (9).
        v. Se resuelve el siguiente modelo de programación lineal:

$$Max\ CT = \Sigma_i\, CM_i\, X_i$$

$$s.t.$$

$$X_i \leq D_{max,i}$$

$$X_i \geq D_{min,i}$$

$$\Sigma_i\, A_i X_i \leq Cap\_restante$$

$$X_i \geq 0$$

vi. Se calcula y se guarda la ganancia **G**(**X**–i) (10) generada al restar temporalmente el color i al conjunto **X** y se regresa i al conjunto **X.**

e. Se comparan todas las **G**(**X**–i), si se cumple que una o varias **G**(**X**–i) > **G**(**X**), se escoge la mayor de ellas y se elimina definitivamente el color i a **X**.
   X = **X** – i

f. Se regresa al inciso a.

3. Se despliegan los colores seleccionados como elementos de **X**, su producción, el orden en que fueron removidos los elementos de **X**' (colores no seleccionados) y la ganancia total generada por ese escenario

Este heurístico avanza hacia la solución óptima partiendo desde el otro extremo del universo de soluciones posibles, donde todos los colores son elementos de **X**. Sus resultados se comparan en las tablas de la siguiente sección.

Si se quiere mejorar la velocidad de este heurístico mediante una solución inicial, se puede partir de la misma solución que parte el heurístico anterior. El heurístico vaciador tendría como restricción el no poder eliminar colores del conjunto inicial seleccionado. Esto aumentaría enormemente la eficiencia de este heurístico ya que sin la solución inicial el problema que tiene este método vaciador es que al principio busca eliminar los colores de mayor volumen para cumplir con la capacidad disponible. El hecho de que elimine color por color provoca lo anterior, por lo que en los primeros pasos puede eliminar colores que probablemente sí estarían en la solución óptima. Otra opción para resolver la problemática descrita, es combinar estos heurísticos de tipo greedy con tabu search; de esta forma se podrían volver a seleccionar colores previamente eliminados por su alta demanda mínima, al momento de tener mayor capacidad disponible.

En la práctica, los heurísticos propuestos con solución inicial; ya sea el llenador o el vaciador, serían los más propensos a utilizarse debido a que algunos de los colores es "obvio" que quedarían en la solución óptima; probablemente los que conforman el 80% de los ingresos (Pareto). En realidad las ventas marginales son las que habría que confirmar si su precio de venta paga realmente por las ineficiencias que provocan.

6. **RESULTADOS**

Los métodos anteriores se probaron con dos problemas, uno de prueba de solo 10 colores para analizar más fácil su funcionamiento y uno basado en la empresa real, de 346 colores. Se pueden obtener conclusiones muy importantes al comparar los diferentes resultados. En primer lugar, vemos que el heurístico llenador x utilidad es muy bueno, aún cuando no utiliza programación lineal y confirma que es mejor analizar la utilidad total generada por el escenario que una medida de utilidad/hr o utilidad/ton, como se ve en todos los casos.

Como no se tiene capacidad de sobra, resulta lógico que los escenarios que seleccionan una mayor cantidad de colores tienen una producción en toneladas menor a los que seleccionan una menor cantidad de colores; ya que la cantidad de setups generados es proporcional a la cantidad de colores del escenario y le restan capacidad de producción a la línea. Esto se cumple siempre que el resto de las condiciones se mantengan igual y no se apliquen tácticas operativas de flexibilidad en la producción o de reducción de setups como SMED, entre otras. En dado caso, habría que analizar el escenario con las nuevas condiciones. Por ejemplo, si la empresa adquiriera una nueva tecnología que le permitiera cambiar de pintura en la línea sin necesidad de hacer un paro, la productividad de las líneas no se vería afectada, pero los problemas con los inventarios seguirían.

Además, salen a la luz ciertas características útiles para la empresa que rompen algunos paradigmas. En primer lugar, no sólo basta con seleccionar los colores con mayor contribución marginal x hora, ya que ese no fue el mejor método y sin embargo es algo común en muchas empresas. Tampoco basta con seleccionar los colores de mayor volumen, ya que la solución óptima escogió más colores, y por ende menos volumen total. Aunque los métodos anteriores otorgaron buenas soluciones, el que genera la utilidad óptima está en una cantidad de colores y de producción intermedia. No es el escenario que más colores tiene ni el que más tonelaje produce pero sí el que produce una mayor utilidad total, esto se ve más claramente en el caso de la empresa real.

Las diferencias entre las tablas 1 y 2 es simplemente la capacidad disponible de producción, en uno de los casos solo se consideran dos turnos de trabajo y por lo tanto en ese escenario es más notoria la disminución de colores. En ese escenario se ve más claramente el 80/20 de los ingresos.

346 colores 3 turnos	Situación actual	Prog entero (óptimo)	Llenador x G(X)	Llenador x G(X)/hr	Llenador x G(X)/ton	Llenador con prog lineal	Vaciador con prog lineal	Llenador con soln inicial
Colores elegidos	346	268	267	328	335	267	327	267
Ganancia total (mdlls)	$40,419.4	$44,538.0	$44,533.7	$42,621.6	$42,120.4	$44,533.7	$42,305.7	$44,533.7
Producción (tons)	11,279.00	12,230.70	12,233.10	11,567.10	11,489.60	12,233.10	11,449.90	12,233.10

Tabla 1. Prueba de los modelos con el caso de 346 colores, trabajando 3 turnos diarios

346 colores 2 turnos	Prog entero (óptimo)	Llenador x G(X)	Llenador x G(X)/hr	Llenador x G(X)/ton	Llenador con prog lineal	Vaciador con prog lineal	Llenador con soln inicial
Colores elegidos	92	83	326	332	85	319	85
Ganancia total (mdlls)	$39,505.9	$39,362.0	$33,391.4	$28,430.1	$39,402.2	$32,883.8	$39,402.2
Producción (tons)	10,190.00	10,254.20	8,329.73	8,282.03	10,245.70	8,103.70	10,245.70

Tabla 2. Prueba de los modelos con el caso de 346 colores, trabajando 2 turnos diarios

10 colores	Prog entero (óptimo)	Llenador x G(X)	Llenador x G(X)/hr	Llenador x G(X)/ton	Llenador con prog lineal	Vaciador con prog lineal	Llenador con soln inicial
Colores elegidos	6	6	10	10	6	6	6
Ganancia total (mdlls)	$16,889.1	$16,889.1	$16,634.5	$16,634.5	$16,889.1	$16,889.1	$16,889.1
Producción (tons)	4,548.49	4,548.49	4,535.19	4,535.19	4,548.49	4,548.49	4,548.49

Tabla 3. Prueba de los modelos con el caso de 10 colores

Soluciones iniciales	XG(X) ∩ XG(X)/hr			Knapsack
	10 colores	346 colores, 3 turnos	346 colores, 2 turnos	346 colores, 3 turnos
Colores elegidos	6	260	75	268
Ganancia total (mdlls)	$16,889.1	$42,479.7	$28,625.9	$44,537.4
Producción (tons)	4,548.49	11,427.70	6,417.00	12,229.80

Tabla 4. Soluciones iniciales al combinar los heurísticos "Llenador x utilidad" y "Llenador x utilidad/hr"

7. CONCLUSIONES

El método propuesto sirve para analizar la variedad óptima de la gama de productos pintados de la empresa. Esta metodología se puede aplicar a otras áreas e inclusive a otras empresas que requieran llevar a cabo una racionalización de sus productos; logrando un balance entre flexibilidad estratégica y flexibilidad operativa. Lo que se busca es la selección de un portafolio de productos adecuado para las tecnologías, capacidades y tácticas operativas de flexibilidad con que se cuenta.

El método presentado otorga claridad para determinar si el precio que se cobra por los colores marginales es el adecuado. La ventaja de la variedad está en que se puede seleccionar entre los mercados y clientes más rentables.

Además, esta metodología arroja luz sobre ciertos costos difíciles de detectar y los cuantifica, de manera que se pueda obtener un costo de oportunidad y poder llegar más adelante a políticas de precios y de introducción de nuevos productos adecuadas. El tener identificados esos costos le permite a la empresa una mayor claridad a la hora de elegir ciertas estrategias de comercialización, ya que puede impulsar ventas de productos realmente rentables y no ventas de productos que estén subsidiados por otros. La transparencia en los costos de producción también permite más adelante definir un lote de producción mínimo en base a la rentabilidad de las líneas (utilidad generada / mes) y no sólo en base a la productividad de las líneas (toneladas producidas / mes).

Así como se pueden reasignar los costos de producción, los gastos administrativos también se pueden reasignar dependiendo de los pedidos que utilizan ciertos procesos, por ejemplo los que requieren análisis de crédito. En todos los casos lo importante es identificar correctamente los conductores (“drivers”) de los costos. Por ejemplo, en los gastos administrativos el conductor puede ser la cantidad de facturas o de pedidos procesados, mientras que en el empaque, carga y manejo (“handling”) puede ser la cantidad de partidas de los pedidos y en producción puede ser el tonelaje producido.

Cabe señalar que la metodología presentada puede ser mejorada con simuladores que sustituyan las relaciones lineales utilizadas; sin embargo debido a las complejidades reales y a las interacciones entre muchas de las variables, cada simulador puede ser objeto de tesis futuras. Ver Figuras 1 y 2 en la sección 5.

Por último, es necesario recalcar que conforme se realizó este estudio se detectaron varias áreas de oportunidad en la empresa. Se considera que la implementación de los siguientes elementos debe ser un paso previo ya que eficientarían sus procesos y automáticamente simplificarían el modelo propuesto, debido a que eliminarían algunas de las interacciones entre las variables del problema.

- Homologación de productos claramente sustitutos.
- Desarrollo de metodologías claras para la definición de inventarios y la programación de la producción.
- Prácticas de "flexibilidad operativa" como Lean Manufacturing, Poka-Yokes, SMED, tecnologías nuevas de pintado, entre otras.
- Estandarización de los sistemas administrativos de compras, producción y ventas para tener claridad en la información y contar con los indicadores correctos, tales como precios reales, descuentos aplicados, tiempos de producción, entre otros.

BIBLIOGRAFÍA

[1] Anderson, Shannon. *"Direct and Indirect Effects of Product Mix Characteristics on Capacity Management Decisions and Operating Performance"*. International Journal of Flexible Manufacturing Systems; Jun 2001; 13, 3; pg. 241

[2] Appelqvist, Patrik & Ebbe Gubi. *"Postponed Variety Creation: Case Study In Consumer Electronics Retail"*. International Journal of Retail & Distribution; 2005; 33, 10; pg. 734

[3] Arnett, Rob. *"The Big Picture"*. Marketing Research; Winter 2002; 14, 4; pg 32

[4] Askin, Ronald & Standridge, Charles. *Modeling and Analysis of Manufacturing Systems*.USA: John Wiley & Sons, Inc, 1993

[5] Balakrishnan, Anantaram & Geunes, Joseph. *"Production Planning with Flexible Product Specifications: An Application to Specialty Steel Manufacturing"*. Operations Research. Linthicum: Jan/Feb 2003; 51, 1; pg 94

[6] Ballou, Ronald H. *Business Logistics Management*. Upper Saddle River, NJ; Prentice Hall; 1999; Fourth Ed.

[7] Bramham, Jo & Brat MacCarthy & Jane Guinery. *"Managing Product Variety in Quotation Processes"*. Journal of Manufacturing Technology Management; 2005; 16, 4; p. 411

[8] Cargille, Brian & Chris Fry & Aaron Raphel. *"Managing Product Line Complexity"*. OR/MS Today; 2005; 32, 3; pg. 34

[9] Chen, Kyle D. & Warren H. Hausman. *"Technical Note: Mathematical Properties of the Optimal Product Line Selection Problem Using Choice-Based Conjoint Analysis"*. Management Science; February 2000; 46, 2; pg. 327

[10] Chen, Miao-Sheng & Mei-Chen Chu. *"The Analysis of Optimal Price Control Model in Matching Problem Between Production and Sales"*. Asia-Pacific Journal of Operational Research; 2001; 18, 2; pg. 131

[11] Chopra, Sunil & Martin A. Lariviere. *"Managing Service Inventory to Improve Performance"*. MIT Sloan Management Review; Fall 2003; 47, 1; pg. 54

[12] Da Silviera, Giovani. *"A Framework for the Management of Product Variety"*. International Journal of Operations & Production Management; 1998; 18, 3; pg. 271

[13] Dobson, Gregory & Candace Arai Yano. *"Product Offering, Princing, and Make-To-Stock/Make-To-Order Decisions with Shared Capacity"*. Production and Operations management; Fall 2002; 11, 3; pg. 293

[14] Dutta, Goutam & Fourer, Robert. *"A Survey of Mathematical Programming Applications in Integrated Steel Plants*. Manufacturing & Service Operations Management. Linthicum: Fall 2001; 3, 4; pg. 387

[15] Holström, Jan. *"Handling Product Range Complexity: A Case Study on Re-engineering Demand Forecasting"*. Business Process Management Journal; 1998; 4, 3; p. 241

[16] Hunley, Terry. *"A Better Return Requires New Ideas"*. Pulp & Paper; San Francisco; Apr 2002; 76, 4; pg. 74.

[17] Keong, Ooi Chee & M. Ahmad & S. Sulaiman & Y. Ismail. *"Proposing a Non-traditional Ordering Methodology in Achieving Optimal Flexibility with Minimal Inventory Risk"*. Asia Pacific Journal of Marketing and Logistics; 2005; 17, 2; pg. 31

[18] LP Solve 5.5: http://lpsolve.sourceforge.net/5.5/, Noviembre 2005

[19] Hall, George & Hiu Man Chan & John Rust. *"Price Discrimination in the Steel Market"*. Yale University; December; 2003

[20] Morgan, Leslie Olin & Richard L. Daniels & Panos Kouvelis. *"Marketing / Manufacturing Trade-Offs in Product Line Management"*. IIE Transactions; Nov 2001; 33, 11; pg. 949

[21] Panico, C. Richard. *"From Bankruptcy to New Business"*. Global Cosmetic Industry; New York; Oct 2004; 172, 10; pg. 42

[22] Prassad, Biren. *"Designing Products for Variety and How to Manage Complexity"*. Journal of Product & Brand Management; 1998; 7, 3; pg. 208

[23] Ramdas, Kamalini. *"Managing Product Variety: An Integrative Review and Research Directions"*. Production and Operations Management Society; USA; Spring 2003; 12, 1; pg. 79

[24] Sackett, J & Douglas J. Maxwell & Paul A. Lowenthal. *"Customizing Manufacturing Strategy"*. Integrated Manufacturing Systems; 1997; 8, 6; pg. 359

[25] Schrieber, Pared. *"Demand Visibility Improves Demand Forecasts"*. The Journal of Business Forecasting; Fall 2005; 24, 3; pg 32

[26] Songini, Marc L. *"Setting the Price Right"*. Computerworld. Farmingham: Jun 2002; 36, 23; pg. 48

[27] Stanton, William J. & Michael J. Etzel & Bruce J. Walker. *Fundamentos de Marketing*. Mc Graw Hill; México. 2004; 13a Ed.

[28] Swink, Morgan & Ram Narasimhan & Soo Wook Kim. *"Manufacturing Practices and Strategy Integration: Effects on Cost Efficiency, Flexibility, and Market-Based Performance"*. Decision Sciences; Aug 2005; 36, 3; pg. 427

[29] Tang, Esther P.Y. & Richard C.M. Yam. *"Product Variety Strategy – An Environmental Perspective"*. Integrated Manufacturing Systems; 1996; 7, 6; pg. 24

[30] Winston, Wayne & Venkataramanan, Munirpallam. *Introduction to Mathematical Programming*. Thompson Brooks Cole; 2003; Fourth Ed.

8. ANEXO 1: Algunos resultados de los diferentes métodos

```
Model name:  'LP_ENTERO' - run #1
Objective:   Maximize(R0)

SUBMITTED
Model size:       21 constraints,      20 variables,           60 non-zeros.
Sets:                                   0 GUB,                  0 SOS.

Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.


Relaxed solution       16904.8938407 after         18 iter is B&B base.

Feasible solution      16903.8110988 after         19 iter,         1 nodes (gap 0.0%)
Improved solution      16903.8230988 after         25 iter,         3 nodes (gap 0.0%)

+Optimal solution      16903.8230988 after         26 iter,         4 nodes (gap 0.0%).

Excellent numeric accuracy ||*|| = 1.11022e-016

 MEMO: lp_solve version 5.5.0.4 for 32 bit OS, with 64 bit REAL variables.
      In the total iteration count 26, 7 (26.9%) were bound flips.
      There were 2 refactorizations, 0 triggered by time and 0 by density.
       ... on average 9.5 major pivots per refactorization.
      The largest [LUSOL v2.2.1.0] fact(B) had 55 NZ entries, 1.0x largest basis.
      The maximum B&B level was 3, 0.1x MIP order, 3 at the optimal solution.
      The constraint matrix inf-norm is 2200, with a dynamic range of 14379.1.
      Time to load data was 0.000 seconds, presolve used 0.000 seconds,
       ... 0.000 seconds in simplex solver, in total 0.000 seconds.


 La solucion con el PROGRAMA ENTERO BINARIO es:
 El color 1 tiene produccion de: 2200
 El color 2 tiene produccion de: 935
 El color 3 tiene produccion de: 495
 El color 4 tiene produccion de: 396
 El color 5 tiene produccion de: 0
 El color 6 tiene produccion de: 280.488
 El color 7 tiene produccion de: 0
 El color 8 tiene produccion de: 242
 El color 9 tiene produccion de: 0
 El color 10 tiene produccion de: 0
 Se seleccionaron 6 colores.
 La CM total total fue: 16889.1
 La produccion total fue: 4548.49
```

```
Buscando por CM_total / HR generada el resultado fue:
El color 1 fue seleccionado 10 con produccion de: 1191.19
El color 2 fue seleccionado 1 con produccion de: 935
El color 3 fue seleccionado 9 con produccion de: 495
El color 4 fue seleccionado 8 con produccion de: 396
El color 5 fue seleccionado 7 con produccion de: 297
El color 6 fue seleccionado 6 con produccion de: 286
El color 7 fue seleccionado 5 con produccion de: 253
El color 8 fue seleccionado 2 con produccion de: 242
El color 9 fue seleccionado 3 con produccion de: 220
El color 10 fue seleccionado 4 con produccion de: 220
Se seleccionaron 10 colores.
La CM_total total fue: 16634.5
La produccion total fue: 4535.19
```

```
Buscando por CM_total TOTAL generada el resultado fue:
El color 1 fue seleccionado 2 con produccion de: 2200
El color 2 fue seleccionado 1 con produccion de: 935
El color 3 fue seleccionado 3 con produccion de: 495
El color 4 fue seleccionado 4 con produccion de: 396
El color 5 no fue seleccionado
El color 6 fue seleccionado 6 con produccion de: 280.488
El color 7 no fue seleccionado
El color 8 fue seleccionado 5 con produccion de: 242
El color 9 no fue seleccionado
El color 10 no fue seleccionado
Se seleccionaron 6 colores.
La CM_total total fue: 16889.1
La produccion total fue: 4548.49
```

```
A partir de la solucion inicial, buscando con el HEURISTICO LLENADOR el resultado fue:
El color 1 fue seleccionado por la solucion inicial con produccion de: 2200
El color 2 fue seleccionado por la solucion inicial con produccion de: 935
El color 3 fue seleccionado por la solucion inicial con produccion de: 495
El color 4 fue seleccionado por la solucion inicial con produccion de: 396
El color 5 no fue seleccionado
El color 6 fue seleccionado por la solucion inicial con produccion de: 280.488
El color 7 no fue seleccionado
El color 8 fue seleccionado por la solucion inicial con produccion de: 242
El color 9 no fue seleccionado
El color 10 no fue seleccionado
Se seleccionaron 6 colores.
La CM total total fue: 16889.1
La produccion total fue: 4548.49
```

```
La solucion con el HEURISTICO LLENADOR CON PROG LINEAL es:
El color 1 fue seleccionado 2 con produccion de: 2200
El color 2 fue seleccionado 1 con produccion de: 935
El color 3 fue seleccionado 3 con produccion de: 495
El color 4 fue seleccionado 4 con produccion de: 396
El color 5 no fue seleccionado
El color 6 fue seleccionado 6 con produccion de: 280.488
El color 7 no fue seleccionado
El color 8 fue seleccionado 5 con produccion de: 242
El color 9 no fue seleccionado
El color 10 no fue seleccionado
Se seleccionaron 6 colores.
La CM total fue: 16889.1
La produccion total fue: 4548.49
```

```
La solucion con el HEURISTICO VACIADOR CON PROG LINEAL es:
El color 1 fue seleccionado con produccion de: 2200
El color 2 fue seleccionado con produccion de: 935
El color 3 fue seleccionado con produccion de: 495
El color 4 fue seleccionado con produccion de: 396
El color 5 fue eliminado 3
El color 6 fue seleccionado con produccion de: 280.488
El color 7 fue eliminado 1
El color 8 fue seleccionado con produccion de: 242
El color 9 fue eliminado 4
El color 10 fue eliminado 2
Se seleccionaron 6 colores.
La CM total fue: 16889.1
La produccion total fue: 4548.49
```

9. ANEXO 2: Código del modelo de programación entera

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>
#include <iostream.h>
#include <fstream.h>
#include <afx.h> // Para poder usar types "CString"

#include "lp_lib.h"


int main()
{

int conteo1, conteo2;
int colores_total, colores_seleccionados, capacidad;
double cantidad_leida, objetivo, CM_total_max, produccion;
double *Color_cara, *Demanda, *CM, *Precio, *Inv_pint, *Inv_PT, *Tiempo_de_prod;
double *row0, *row1, *row2, *row3, *duals, *Prod;
//double cantidad_setups, CT_Inv_pint, CT_Inv_PT;

// Leer el archivo datos.txt para obtener los inputs requeridos
ifstream ff("datos.txt");

ff >> cantidad_leida; colores_total = int(cantidad_leida);
//cout << "El proyecto es de " << colores_total << " colores_total " << endl;

Color_cara = (double *) calloc (colores_total, sizeof (double) );
Demanda = (double *) calloc (colores_total, sizeof (double) );
CM = (double *) calloc (colores_total, sizeof (double) );
Precio = (double *) calloc (colores_total, sizeof (double) );
Inv_pint = (double *) calloc (colores_total, sizeof (double) );
Inv_PT = (double *) calloc (colores_total, sizeof(double) );
Tiempo_de_prod = (double *) calloc (colores_total, sizeof (double) );
Prod = (double *) calloc (colores_total*2, sizeof (double) );
duals = (double *) calloc (colores_total*3+1, sizeof (double) );

row0 = (double *) calloc (colores_total*2+1, sizeof (double) );
row1 = (double *) calloc (colores_total*2+1, sizeof (double) );
row2 = (double *) calloc (colores_total*2+1, sizeof (double) );
row3 = (double *) calloc (colores_total*2+1, sizeof (double) );

ff >> cantidad_leida; capacidad = int(cantidad_leida);
//cout << "La restriccion de capacidad es: " << capacidad << " toneladas por periodo " << endl << endl;

for (conteo1=0; conteo1<colores_total; conteo1++)
{
        // Leer y guardar los valores del archivo

        ff >> cantidad_leida;  Color_cara[conteo1] = cantidad_leida;
        //cout << " El Color es: " << Color_cara[conteo1] << endl;
```

```
        ff >> cantidad_leida;  Demanda[conteo1] = cantidad_leida;
        //cout << " La Demanda es: " << Demanda[conteo1] << endl;

        ff >> cantidad_leida;  CM[conteo1] = cantidad_leida;
        //cout << " La Contribucion Marginal es: " << CM[conteo1] << endl;

        ff >> cantidad_leida;  Precio[conteo1] = cantidad_leida;
        //cout << " Precio es: " << Precio[conteo1] << endl;

        ff >> cantidad_leida;  Inv_pint[conteo1] = cantidad_leida;
        //cout << " El $ de inventario de pintura es: " << Inv_pint[conteo1] << endl;

        ff >> cantidad_leida;  Inv_PT[conteo1] = cantidad_leida;
        //cout << " El $ de inventario Producto Terminado es: " << Inv_PT[conteo1] << endl;

        ff >> cantidad_leida;  Tiempo_de_prod[conteo1] = cantidad_leida;
        //cout << " El Tiempo de Produccion (hrs/ton) para este color es: " << Tiempo_de_prod[conteo1]
<< "\n" << endl;

} // end FOR


// ******************************************************************************
// lp solve comienza, descripción
// ******************************************************************************
/*
cout  <<  "  ********************************************************************************  "  <<
endl;
cout << " *                          LP SOLVE                              * " << endl;
cout  <<  "  ********************************************************************************  "  <<
endl;
cout << " * Description: Open Source (Mixed-Integer) Linear Programming System  * " << endl;
cout << " * Language: Multi-Platform, pure ANSI C / POSIX Source Code           * " << endl;
cout << " * Official Name: lp_solve                                             * " << endl;
cout << " * Release Data: Version 5.5.0.4. , dated: 2005                        * " << endl;
cout << " * Co-developers: Michael Berkelaar, Kjell Eikland, Peter Notebaert    * " << endl;
cout << " * Licence Terms: GNU LGPL (Lesser General Public Licence              * " << endl;
cout  <<  "  ********************************************************************************  "  <<
endl;
cout << " " << endl;
cout << " " << endl;
*/

lprec *lp;
HINSTANCE lpsolve;
lpsolve = LoadLibrary("lpsolve55.dll");
if (lpsolve==NULL)
{
        printf("Unable to load lpsolve shared library\n");
        return(FALSE);
}

lp = make_lp(0, colores_total*2); // (# de restricciones, # de variables)
set_lp_name(lp, "LP_ENTERO"); // Pone nombre al programa
set_maxim(lp);
set_add_rowmode(lp, TRUE);
```

```
for (conteo1=0; conteo1<colores_total; conteo1++) // Forma la fn objetivo
{
        row0[conteo1+1] = CM[conteo1];
        row0[conteo1+1+colores_total] = -Inv_pint[conteo1]-Inv_PT[conteo1]-0.007*50*1.0176; // Los costos de inventarios de pintura, PT y charloa (7dlls*50lts de charola en c/setup)
}

set_obj_fn(lp, row0); // Define la función objetivo

for (conteo1=0; conteo1<colores_total; conteo1++) // Para formar la restricción de capacidad y las de demanda
{
        row1[conteo1+1] = Tiempo_de_prod[conteo1];
        row1[conteo1+1+colores_total] = 0.5*1.0176; // Media hora por cada setup
        for (conteo2=0; conteo2<colores_total*2; conteo2++)  // Para formar las restricciones de demanda
        {
                row2[conteo2+1] = 0;
                row3[conteo2+1] = 0;
        }
        row2[conteo1+1] = 1.0;
        row2[conteo1+1+colores_total] = -Demanda[conteo1]*1.10;
        row3[conteo1+1] = 1.0;
        row3[conteo1+1+colores_total] = -Demanda[conteo1]*0.85;
        add_constraint(lp, row2, LE, 0);      // Agrega las restricciones de demanda maxima
        add_constraint(lp, row3, GE, 0);      // Agrega las restricciones de demanda minima
}
add_constraint(lp, row1, LE, (capacidad-0.5*42.057)); // Agrega la restricción de capacidad, quitando la constante de los setups
set_add_rowmode(lp, FALSE);

for (conteo1=colores_total+1; conteo1<= colores_total*2; conteo1++)
{
        set_binary(lp, conteo1, TRUE);
}

// Imprime la estructura del problema
/*
print_lp(lp);
cout << " " << endl;
cout << " " << endl;
cout << " " << endl;
cout << " *** Solucion *** " << endl;
cout << " " << endl;
*/

// Resuelve el problema y lo presenta en pantalla
solve(lp);
cout << " " << endl;
cout << " " << endl;
objetivo=get_objective(lp); // Obtiene las soluciones del problema
get_variables(lp, Prod);

// Imprime la solucion del problema, 1 = numero de columnas para poner valores
/*
print_solution(lp,1);
```

```
cout << " " << endl;
cout << " " << endl;
*/

// Imprime en pantalla los valores duales del problema //
/*
print_duals(lp);  // Marca error al imprimir los valores duales!!! (al definir variables BINARIAS)
cout << " " << endl;
cout << " " << endl;
get_dual_solution(lp, duals); // Marca error al obtener los valores duales!!! (al definir variables BINARIAS)
*/

CM_total_max = objetivo - 0.007*50*42.057; // Al resultado hay que quitarle los costos constantes, en este caso los setups por default y su pintura tirada
colores_seleccionados = 0;
produccion = 0;

for (conteo1=0; conteo1<colores_total; conteo1++) // FOR que imprime todas las producciones
{
        if (Prod[conteo1]>0)
        {
                colores_seleccionados++;
        }
        produccion = produccion + Prod[conteo1];
        cout << " El color " << Color_cara[conteo1] << " tiene produccion de: " << Prod[conteo1] << endl;
}

cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;


FreeLibrary(lpsolve);

} // end MAIN
```

10. ANEXO 3: Código del modelo heurístico llenador con solución inicial

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>
#include <iostream.h>
#include <fstream.h>

#include "lp_lib.h"

int main()
{

int conteo1, conteo2, seleccion, salir;
int colores_total, colores_seleccionados, rankeo;
double cantidad_leida, capacidad, capacidad_ocupada, cantidad_setups, CT_Inv_pint, CT_Inv_PT;
double opportunity_lost, setup_cost, objetivo, CM_total_max, Ratio_max, produccion;
double *Color_cara, *Demanda, *CM, *Precio, *Inv_pint, *Inv_PT, *Demanda_backup;
double *CM_total, *Tiempo_de_prod, *Prod, *Prod_final, *Ratio1, *Ratio2;
double *row0, *row1, *variable, *duals;
bool *Seleccionados, *Seleccionados1, *Seleccionados2, *Seleccionados3;
int *posicion, *Ranking, *Ranking1, *Ranking2, *Ranking3;

// Leer el archivo datos.txt para obtener los inputs requeridos
ifstream ff("datos.txt");

ff >> cantidad_leida; colores_total = int(cantidad_leida); // Cantidad de colores del problema

Color_cara = (double *) calloc (colores_total, sizeof (double) );
Demanda = (double *) calloc (colores_total, sizeof (double) );
Demanda_backup = (double *) calloc (colores_total, sizeof (double) );
Prod = (double *) calloc (colores_total, sizeof (double) );
Prod_final = (double *) calloc (colores_total, sizeof (double) );
CM = (double *) calloc (colores_total, sizeof (double) );
Precio = (double *) calloc (colores_total, sizeof (double) );
Inv_pint = (double *) calloc (colores_total, sizeof (double) );
Inv_PT = (double *) calloc (colores_total, sizeof(double) );
Tiempo_de_prod = (double *) calloc (colores_total, sizeof (double) );
CM_total = (double *) calloc (colores_total, sizeof (double) );
Ratio1 = (double *) calloc (colores_total, sizeof (double) );
Ratio2 = (double *) calloc (colores_total, sizeof (double) );
Seleccionados = (bool *) calloc (colores_total+1, sizeof (double) );
Seleccionados1 = (bool *) calloc (colores_total+1, sizeof (double) );
Seleccionados2 = (bool *) calloc (colores_total+1, sizeof (double) );
Seleccionados3 = (bool *) calloc (colores_total+1, sizeof (double) );
Ranking = (int *) calloc (colores_total, sizeof (double) );
Ranking1 = (int *) calloc (colores_total, sizeof (double) );
Ranking2 = (int *) calloc (colores_total, sizeof (double) );
Ranking3 = (int *) calloc (colores_total, sizeof (double) );
row0 = (double *) calloc (colores_total+1, sizeof (double) );
row1 = (double *) calloc (colores_total+1, sizeof (double) );
variable = (double *) calloc (1, sizeof (double) );
posicion = (int *) calloc (1, sizeof (double) );
```

```
duals = (double *) calloc (colores_total*3+1, sizeof (double) );

ff >> cantidad_leida; capacidad = int(cantidad_leida); // Capacidad disponible

for (conteo1=0; conteo1<colores_total; conteo1++)
{
        // Leer y guardar los valores del archivo
        ff >> cantidad_leida;  Color_cara[conteo1] = cantidad_leida;
        ff >> cantidad_leida;  Demanda[conteo1] = cantidad_leida;  Demanda_backup[conteo1] =
cantidad_leida;
        ff >> cantidad_leida;  CM[conteo1] = cantidad_leida;
        ff >> cantidad_leida;  Precio[conteo1] = cantidad_leida;
        ff >> cantidad_leida;  Inv_pint[conteo1] = cantidad_leida;
        ff >> cantidad_leida;  Inv_PT[conteo1] = cantidad_leida;
        ff >> cantidad_leida;  Tiempo_de_prod[conteo1] = cantidad_leida;

} // end FOR para leer datos


// Inicialización de algunas variables para agregar por CM_total
for (conteo1=0; conteo1<colores_total; conteo1++) // Inicializa arreglo de Seleccionados1
{
        Seleccionados1[conteo1] = false;
        CM_total[conteo1] = 0;
        Ranking1[conteo1] = 0;
        Prod_final[conteo1] = 0;
        Prod[conteo1] = 0;
}

CM_total_max = 0;
capacidad_ocupada = 0;
rankeo = 0;
salir = 1;

while (salir<2) // Comienza WHILE para agregar colores 1 x 1 por medio de CM_totales
{

salir = 1;

seleccion = colores_total; // Inicializa seleccion
for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para guardar todas las CM_totales y
ratios
{
        if (Seleccionados1[conteo1]==false) // Si ese color no ha sido seleccionado, lo evalúo como
candidato para llenar
        {
        // Inicialización de algunas variables
        Prod[conteo1] = 0;
        CM_total[conteo1] = 0;
        colores_seleccionados = 1;// El color i será agregado para ver si hay mejoría
        CT_Inv_pint = Inv_pint[conteo1];
        CT_Inv_PT = Inv_PT[conteo1];

        for (conteo2=0; conteo2<colores_total; conteo2++) // Comienza FOR para considerar los colores
activos
        {
```

```
                if ((Seleccionados1[conteo2])&&(conteo2!=conteo1))          // Activa los colores ya Seleccionados1
                {
                        CM_total[conteo1] = CM_total[conteo1] + Prod_final[conteo2] * CM[conteo2];
                        colores_seleccionados++;  // Calcula la cantidad de colores_total usados para determinar el # de setups
                        CT_Inv_pint= CT_Inv_pint + Inv_pint[conteo2]; // Calcula costos de inventario pintura y producto terminado con la cantidad de colores_total real
                        CT_Inv_PT = CT_Inv_PT + Inv_PT[conteo2];
                }
        }

        cantidad_setups = colores_seleccionados * 1.0176 + 42.057;   // Relación lineal entre la cantidad de colores_total y la cantidad de setups
        setup_cost = 0.007 * 50 * cantidad_setups;                   // Se tiran 50 lts en c/cambio de charola a un precio promedio de 7 dlls/lt
        opportunity_lost = 0.5 * cantidad_setups;                    // Cada setup toma aproximadamente 30  minutos


        if ((capacidad-opportunity_lost-capacidad_ocupada)>=(1.10*Demanda[conteo1]*Tiempo_de_prod[conteo1]))
        {
                Prod[conteo1] = 1.10 * Demanda[conteo1];
        } else if ((capacidad-opportunity_lost-capacidad_ocupada)>0)
        {
                Prod[conteo1] = (capacidad - opportunity_lost - capacidad_ocupada)/Tiempo_de_prod[conteo1];
        }

        CM_total[conteo1] = CM_total[conteo1] + Prod[conteo1] * CM[conteo1] - CT_Inv_pint - CT_Inv_PT - setup_cost;

        } // Termina IF que analiza colores que no han sido agregados

} // Termina FOR que guarda todas las opciones de CM_total

for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para comparar todas las CM_totales y ratios
{
        if (CM_total[conteo1]>CM_total_max)
        {
                        seleccion = conteo1;
                        CM_total_max = CM_total[conteo1];
        }
}

if (seleccion<colores_total)
{
        Seleccionados1[seleccion] = true;
        Prod_final[seleccion] = Prod[seleccion];
        capacidad_ocupada = capacidad_ocupada + Prod_final[seleccion]*Tiempo_de_prod[seleccion];
        rankeo++;
        Ranking1[seleccion] = rankeo;
        salir--;
}
```

```
salir++;
if (capacidad == capacidad_ocupada) salir = 2;
} // Termina WHILE que agrega colores 1 x 1 por CM_totales

colores_seleccionados = 0;
produccion = 0;

cout << " " << endl;
cout << " Buscando por CM_total TOTAL generada el resultado fue: " << endl;

for (conteo1=0; conteo1<colores_total; conteo1++) // FOR que imprime todas las producciones
{
        if (Seleccionados1[conteo1])
        {
                colores_seleccionados++;
                produccion = produccion + Prod_final[conteo1];
                cout<< " El color " << Color_cara[conteo1] << " fue seleccionado " <<
Ranking1[conteo1] << " con produccion de: " << Prod_final[conteo1] << endl;
        }
        else
        {
                cout << " El color " << Color_cara[conteo1] << " no fue seleccionado " << endl;
        }

}

cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM_total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;

// Inicialización de algunas variables, ahora para agregar por CM_total x hr
for (conteo1=0; conteo1<colores_total; conteo1++) // Inicializa arreglo de Seleccionados2
{
        Seleccionados2[conteo1] = false;
        CM_total[conteo1] = 0;
        Ranking2[conteo1] = 0;
        Prod_final[conteo1] = 0;
        Prod[conteo1] = 0;
        Ratio1[conteo1] = 0;
}

CM_total_max = 0;
capacidad_ocupada = 0;
rankeo = 0;
salir = 1;

while (salir<2) // Comienza WHILE para agregar colores 1 x 1 por medio de CM_total x hr
{

salir = 1;
seleccion = colores_total; // Inicializa seleccion
Ratio_max = 0;
for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para guardar todas las CM_totales y
ratios
```

```
{
	if (Seleccionados2[conteo1]==false) // Si ese color no ha sido seleccionado, lo evalúo como candidato para llenar
	{
	// Inicialización de algunas variables
	Prod[conteo1] = 0;
	CM_total[conteo1] = 0;
	colores_seleccionados = 1;// El color i será agregado para ver si hay mejoría
	CT_Inv_pint = Inv_pint[conteo1];
	CT_Inv_PT = Inv_PT[conteo1];

	for (conteo2=0; conteo2<colores_total; conteo2++) // Comienza FOR para considerar los colores activos
	{
		if ((Seleccionados2[conteo2])&&(conteo2!=conteo1))		// Activa los colores ya Seleccionados2
		{
			CM_total[conteo1] = CM_total[conteo1] + Prod_final[conteo2] * CM[conteo2];
			colores_seleccionados++;  // Calcula la cantidad de colores_total usados para determinar el # de setups
			CT_Inv_pint= CT_Inv_pint + Inv_pint[conteo2]; // Calcula costos de inventario pintura y producto terminado con la cantidad de colores_total real
			CT_Inv_PT = CT_Inv_PT + Inv_PT[conteo2];
		}
	}

	cantidad_setups = colores_seleccionados * 1.0176 + 42.057;   // Relación lineal entre la cantidad de colores_total y la cantidad de setups
	setup_cost = 0.007 * 50 * cantidad_setups;			// Se tiran 50 lts en c/cambio de charola a un precio promedio de 7 dlls/lt
	opportunity_lost = 0.5 * cantidad_setups;			//  Cada setup toma aproximadamente 30  minutos


	if ((capacidad-opportunity_lost-capacidad_ocupada)>=(1.10*Demanda[conteo1]*Tiempo_de_prod[conteo1]))
	{
		Prod[conteo1] = 1.10 * Demanda[conteo1];
	} else if ((capacidad-opportunity_lost-capacidad_ocupada)>0)
	{
		Prod[conteo1] = (capacidad - opportunity_lost - capacidad_ocupada)/Tiempo_de_prod[conteo1];
	}

	CM_total[conteo1] = CM_total[conteo1] + Prod[conteo1] * CM[conteo1] - CT_Inv_pint - CT_Inv_PT - setup_cost;
	Ratio1[conteo1] = CM_total[conteo1]/(capacidad_ocupada+opportunity_lost+Prod[conteo1]*Tiempo_de_prod[conteo1]);   // Para tomar en cuenta el tiempo total consumido

	} // Termina IF que analiza colores que no han sido agregados

} // Termina FOR que guarda todas las opciones de CM_total

for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para comparar todas las CM_totales y ratios
```

```
{
        if ((Ratio1[conteo1]>Ratio_max)&&(CM_total[conteo1]>CM_total_max))
        {
                seleccion = conteo1;
                Ratio_max = Ratio1[conteo1];
        }
}

if (seleccion<colores_total)
{
        Seleccionados2[seleccion] = true;
        Prod_final[seleccion] = Prod[seleccion];
        capacidad_ocupada = capacidad_ocupada + Prod_final[seleccion]*Tiempo_de_prod[seleccion];
        rankeo++;
        Ranking2[seleccion] = rankeo;
        CM_total_max = CM_total[seleccion];
        salir--;
}

salir++;
if (capacidad == capacidad_ocupada) salir = 2;
} // Termina WHILE que agrega colores 1 x 1 por CM_total x hr

colores_seleccionados = 0;
produccion = 0;

cout << " " << endl;
cout << " Buscando por CM_total / HR generada el resultado fue: " << endl;

for (conteo1=0; conteo1<colores_total; conteo1++) // FOR que imprime todas las producciones
{
        if (Seleccionados2[conteo1])
        {
                colores_seleccionados++;
                produccion = produccion + Prod_final[conteo1];
                cout<< " El color " << Color_cara[conteo1] << " fue seleccionado " <<
Ranking2[conteo1] << " con produccion de: " << Prod_final[conteo1] << endl;
        }
        else
        {
                cout << " El color " << Color_cara[conteo1] << " no fue seleccionado " << endl;
        }

}

cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM_total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;

// Inicialización de algunas variables, ahora para agregar por ratio
for (conteo1=0; conteo1<colores_total; conteo1++) // Inicializa arreglo de Seleccionados3
{
        Seleccionados3[conteo1] = false;
        CM_total[conteo1] = 0;
        Ranking3[conteo1] = 0;
```

```
        Prod_final[conteo1] = 0;
        Prod[conteo1] = 0;
        Ratio2[conteo1] = 0;
}

CM_total_max = 0;
capacidad_ocupada = 0;
rankeo = 0;
salir = 1;

while (salir<2) // Comienza WHILE para agregar colores 1 x 1 por medio de CM_total x ton
{

salir = 1;
seleccion = colores_total; // Inicializa seleccion
Ratio_max = 0;
for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para guardar todas las CM_totales y
ratios
{
        if (Seleccionados3[conteo1]==false) // Si ese color no ha sido seleccionado, lo evalúo como
candidato para llenar
        {
        // Inicialización de algunas variables
        Prod[conteo1] = 0;
        CM_total[conteo1] = 0;
        colores_seleccionados = 1;// El color i será agregado para ver si hay mejoría
        CT_Inv_pint = Inv_pint[conteo1];
        CT_Inv_PT = Inv_PT[conteo1];

        for (conteo2=0; conteo2<colores_total; conteo2++) // Comienza FOR para considerar los colores
activos
        {
                if ((Seleccionados3[conteo2])&&(conteo2!=conteo1))          // Activa los colores ya
Seleccionados2
                {
                        CM_total[conteo1] = CM_total[conteo1] + Prod_final[conteo2] * CM[conteo2];
                        colores_seleccionados++;  // Calcula la cantidad de colores_total usados para
determinar el # de setups
                        CT_Inv_pint= CT_Inv_pint + Inv_pint[conteo2]; // Calcula costos de inventario
pintura y producto terminado con la cantidad de colores_total real
                        CT_Inv_PT = CT_Inv_PT + Inv_PT[conteo2];
                }
        }

        cantidad_setups = colores_seleccionados * 1.0176 + 42.057;   // Relación lineal entre la cantidad
de colores_total y la cantidad de setups
        setup_cost = 0.007 * 50 * cantidad_setups;                                  // Se tiran 50 lts
en c/cambio de charola a un precio promedio de 7 dlls/lt
        opportunity_lost = 0.5 * cantidad_setups;                                   // Cada setup
toma aproximadamente 30  minutos


        if                                                                  ((capacidad-opportunity_lost-
capacidad_ocupada)>=(1.10*Demanda[conteo1]*Tiempo_de_prod[conteo1]))
        {
                Prod[conteo1] = 1.10 * Demanda[conteo1];
```

```
	} else if ((capacidad-opportunity_lost-capacidad_ocupada)>0)
	{
		Prod[conteo1] = (capacidad - opportunity_lost - capacidad_ocupada)/Tiempo_de_prod[conteo1];
	}

	CM_total[conteo1] = CM_total[conteo1] + Prod[conteo1] * CM[conteo1] - CT_Inv_pint - CT_Inv_PT - setup_cost;
	Ratio2[conteo1] = CM_total[conteo1]/Prod[conteo1]; // OJO: CM_total entre Produccion sería la forma tradicional, pero hay que tomar en cuenta el tiempo total consumido

	} // Termina IF que analiza colores que no han sido agregados

} // Termina FOR que guarda todas las opciones de CM_total

for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para comparar todas las CM_totales y ratios
{
	if ((Ratio2[conteo1]>Ratio_max)&&(CM_total[conteo1]>CM_total_max))
	{
		seleccion = conteo1;
		Ratio_max = Ratio2[conteo1];
	}
}

if (seleccion<colores_total)
{
	Seleccionados3[seleccion] = true;
	Prod_final[seleccion] = Prod[seleccion];
	capacidad_ocupada = capacidad_ocupada + Prod_final[seleccion]*Tiempo_de_prod[seleccion];
	rankeo++;
	Ranking3[seleccion] = rankeo;
	CM_total_max = CM_total[seleccion];
	salir = 0;
}

salir++;
if (capacidad == capacidad_ocupada) salir = 2;
} // Termina WHILE que agrega colores 1 x 1 por CM_total x ton

colores_seleccionados = 0;
produccion = 0;

cout << " " << endl;
cout << " Buscando por CM_total / TON generada el resultado fue: " << endl;

for (conteo1=0; conteo1<colores_total; conteo1++) // FOR que imprime todas las producciones
{
	if (Seleccionados3[conteo1])
	{
		colores_seleccionados++;
		produccion = produccion + Prod_final[conteo1];
		cout<< " El color " << Color_cara[conteo1] << " fue seleccionado " << Ranking3[conteo1] << " con produccion de: " << Prod_final[conteo1] << endl;
	}
	else
```

```
		{
			cout << " El color " << Color_cara[conteo1] << " no fue seleccionado " << endl;
		}

}

cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM_total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;

lprec *lp;
HINSTANCE lpsolve;
lpsolve = LoadLibrary("lpsolve55.dll");
if (lpsolve==NULL)
{
		printf("Unable to load lpsolve shared library\n");
		return(FALSE);
}

colores_seleccionados = 0;
CT_Inv_pint = 0;
CT_Inv_PT = 0;
for (conteo1=0; conteo1<colores_total; conteo1++) // Inicializa arreglo de seleccionados y otras variables
{
		if ((Seleccionados1[conteo1])&&(Seleccionados2[conteo1]))
//&&(Ranking1[conteo1]<=300)&&(Ranking2[conteo2]<=300))
		{
			Seleccionados[conteo1] = true;
			Ranking[conteo1] = colores_total+1;
			Demanda[conteo1]= Demanda_backup[conteo1];
			colores_seleccionados = colores_seleccionados++;  // Calcula la cantidad de colores_total
usados para determinar el # de setups
			CT_Inv_pint= CT_Inv_pint + Inv_pint[conteo1]; // Calcula costos de inventario pintura y
producto terminado con la cantidad de colores_total real
			CT_Inv_PT = CT_Inv_PT + Inv_PT[conteo1];

		}
		else
		{
			Seleccionados[conteo1] = false;
			Ranking[conteo1] = 0;
			Demanda[conteo1] = 0;
		}
}

cantidad_setups = colores_seleccionados * 1.0176 + 42.057; // Relación lineal entre la cantidad de
colores_total y la cantidad de setups
setup_cost = 0.007 * 50 * cantidad_setups; // Se tiran 50 lts en c/cambio de charola a un precio promedio
de 7 dlls/lt
opportunity_lost = 0.5 * cantidad_setups; // Cada setup toma aproximadamente 30  minutos


lp = make_lp(0, colores_total); // (# de restricciones, # de variables)
set_lp_name(lp, "LP_LLENADOR_CON_SOLN_INICIAL"); // Pone nombre al programa
set_maxim(lp);
```

```
set_add_rowmode(lp, TRUE);

for (conteo1=0; conteo1<colores_total; conteo1++) // Forma la fn objetivo
{
        row0[conteo1+1] = CM[conteo1];  // Como ya no cambia la fn objetivo, ya no la tengo que volver a formar más adelante
}
set_obj_fn(lp, row0); // Define la función objetivo

for (conteo1=0; conteo1<colores_total; conteo1++) // Para formar la restricción de capacidad y las de demanda
{
        row1[conteo1+1] = Tiempo_de_prod[conteo1]; // Como ya no cambia el lado izquierdo de la restricción de capacidad, no tengo que volver a formarlo más adelante
        variable[0] = 1.0;
        posicion[0] = conteo1+1;
        add_constraintex(lp, 1, variable, posicion, LE, Demanda[conteo1]*1.10);          // Agrega las restricciones de demanda maxima
        add_constraintex(lp, 1, variable, posicion, GE, Demanda[conteo1]*0.85); // Agrega las restricciones de demanda minima
}
add_constraint(lp, row1, LE, (capacidad - opportunity_lost)); // Agrega la restricción de capacidad
set_add_rowmode(lp, FALSE);

// Resuelve el problema y lo presenta en pantalla (con set_verbose ya no lo presenta)
set_verbose(lp, IMPORTANT);
solve(lp);
objetivo = get_objective(lp); // Obtiene las soluciones del problema
get_variables(lp, Prod);
//get_dual_solution(lp, duals);
free_lp(&lp);
CM_total_max = objetivo - CT_Inv_pint - CT_Inv_PT - setup_cost;

produccion = 0;

cout << " " << endl;
cout << " Combinando CM_total TOTAL y CM_total/HR, la SOLUCION INICIAL es: " << endl;

for (conteo1=0; conteo1<colores_total; conteo1++)
{
        if (Seleccionados[conteo1])
        {
                produccion = produccion + Prod[conteo1];
                cout<< " El color " << Color_cara[conteo1] << " fue seleccionado como solucion inicial con produccion de: " << Prod[conteo1] << endl;
        } else
        {
                cout<< " El color " << Color_cara[conteo1] << " no fue seleccionado como solucion inicial." << endl;
        }
}

cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM_total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;
```

```
cout << " " << endl;
cout << " Buscando por medio del Heuristico Llenador... " << endl;

salir = 1;
rankeo = colores_seleccionados;

while (salir<2) // Comienza WHILE para agregar colores 1 x 1
{

salir = 1;
seleccion = colores_total; // Inicializa seleccion

for (conteo1=0; conteo1<colores_total; conteo1++) // Comienza FOR para guardar todas las CM_totales
{
        if (Seleccionados[conteo1]==false) // Si ese color no ha sido seleccionado, lo evalúo como candidato para llenar
        {
        // Inicialización de algunas variables para obtener CM_total total "si agregara este color"
        colores_seleccionados++;  // El color i será agregado para ver si hay mejoría
        CT_Inv_pint = CT_Inv_pint + Inv_pint[conteo1];
        CT_Inv_PT = CT_Inv_PT + Inv_PT[conteo1];
        Demanda[conteo1] = Demanda_backup[conteo1]; // Activa el color i

        cantidad_setups = colores_seleccionados * 1.0176 + 42.057; // Relación lineal entre la cantidad de colores_total y la cantidad de setups
        setup_cost = 0.007 * 50 * cantidad_setups; // Se tiran 50 lts en c/cambio de charola a un precio promedio de 7 dlls/lt
        opportunity_lost = 0.5 * cantidad_setups; // Cada setup toma aproximadamente 30  minutos


        lp = make_lp(0, colores_total); // (# de restricciones, # de variables)
        set_lp_name(lp, "LP_LLENADOR_CON_SOLN_INICIAL"); // Pone nombre al programa
        set_maxim(lp);
        set_add_rowmode(lp, TRUE);

        set_obj_fn(lp, row0); // Define la función objetivo

        for (conteo2=0; conteo2<colores_total; conteo2++) // Para formar la restricción de capacidad y las de demanda
        {
                variable[0] = 1.0;
                posicion[0] = conteo2+1;
                add_constraintex(lp, 1, variable, posicion, LE, Demanda[conteo2]*1.10);          // Agrega las restricciones de demanda maxima
                add_constraintex(lp, 1, variable, posicion, GE, Demanda[conteo2]*0.85); // Agrega las restricciones de demanda minima
        }
        add_constraint(lp, row1, LE, (capacidad - opportunity_lost)); // Agrega la restricción de capacidad
        set_add_rowmode(lp, FALSE);

        set_verbose(lp, IMPORTANT);
        //set_presolve(lp, PRESOLVE_ROWS, 1); //prueba para eficientizar con PRESOLVE_REDUCEMIP o PRESOLVE_ROWS o PRESOLVE_COLS, colores total - colores seleccionados
        solve(lp);
        objetivo = get_objective(lp); // Obtiene las soluciones del problema
```

```
		CM_total[conteo1] = objetivo - CT_Inv_pint - CT_Inv_PT - setup_cost;

		if (CM_total[conteo1]>CM_total_max) // Escoge la CM_total mayor
		{
			seleccion = conteo1;
			CM_total_max = CM_total[conteo1];
			get_variables(lp, Prod);
			//get_dual_solution(lp, duals);
		}

		free_lp(&lp);

		// Regreso a las condiciones antes de agregar este color
		colores_seleccionados--;
		CT_Inv_pint = CT_Inv_pint - Inv_pint[conteo1];
		CT_Inv_PT = CT_Inv_PT - Inv_PT[conteo1];
		Demanda[conteo1] = 0; // Activa además el color i

		} // Termina IF que analiza colores que no han sido agregados


} // Termina FOR que guarda todas las opciones de CM_total

if (seleccion<colores_total)
{
		Seleccionados[seleccion] = true;
		rankeo++;
		Ranking[seleccion] = rankeo;
		salir--;
		colores_seleccionados++;  // El color seleccionado es agregado
		CT_Inv_pint = CT_Inv_pint + Inv_pint[seleccion];
		CT_Inv_PT = CT_Inv_PT + Inv_PT[seleccion];
		Demanda[seleccion] = Demanda_backup[seleccion]; // Activa el color seleccionado
}

cout << " La cantidad de colores elegida es: " << colores_seleccionados << endl;
salir++;
} // Termina WHILE que agrega colores 1 x 1

colores_seleccionados = 0;
produccion = 0;

cout << " " << endl;
cout << " " << endl;
cout << " A partir de la solucion inicial, buscando con el HEURISTICO LLENADOR el resultado fue: "
<< endl;

for (conteo1=0; conteo1<colores_total; conteo1++) // FOR que imprime todas las producciones
{
		if (Seleccionados[conteo1])
		{
			colores_seleccionados++;
			produccion = produccion + Prod[conteo1];
			if (Ranking[conteo1] == colores_total +1)
			{
```

```
                        cout << " El color " << Color_cara[conteo1] << " fue seleccionado por la solucion inicial con produccion de: " << Prod[conteo1] << endl;
                } else
                {
                        cout << " El color " << Color_cara[conteo1] << " fue seleccionado " << Ranking[conteo1] << " con produccion de: " << Prod[conteo1] << endl;
                }
        }
        else
        {
                cout << " El color " << Color_cara[conteo1] << " no fue seleccionado " << endl;
        }

}
cout << " Se seleccionaron " << colores_seleccionados << " colores." << endl;
cout << " La CM total total fue: " << CM_total_max << endl;
cout << " La produccion total fue: " << produccion << endl;
cout << " " << endl;


FreeLibrary(lpsolve);

} // end MAIN
```