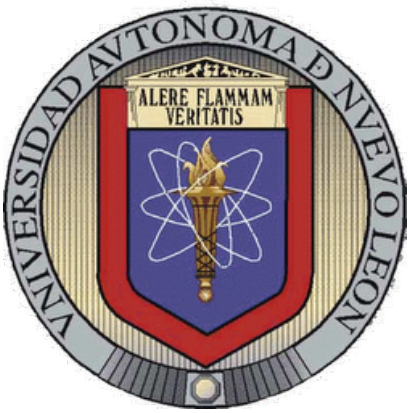
Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado



Optimización de la producción en máquinas en paralelo de inyección de plástico

por

I.Q. Miguel Lorenzo Morales Marroquín

en opción al grado de

Maestría en Ciencias

en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León junio 2012

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

FIME

Optimización de la producción en máquinas en paralelo de inyección de plástico

por

I.Q. Miguel Lorenzo Morales Marroquín

en opción al grado de

Maestría en Ciencias

en Ingeniería de Sistemas

San Nicolás de los Garza, Nuevo León junio 2012

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Optimización de la producción en máquinas en paralelo de inyección de plástico», realizada por el alumno I.Q. Miguel Lorenzo Morales Marroquín, con número de matrícula 1022145, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dra. Yasmín Á. Ríos Solís
Director

Dr. Igor S. Litvinchev
Revisor

Dr. Edgar Possani Espinosa
Revisor

Vo. Bo.

Dr. Moisés Hinojosa Rivera
División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, junio 2012

Este trabajo está dedicado a mis padres Miguel y Blanca por su apoyo incondicional y a Génesis, Regina y Minerva por darme la razón para dar un poco más de mi todos los días.

Índice general

Índice de figuras

Índice de tablas

Agradecimientos

Agradezco a Génesis Suarez Lazalde por su comprensión y apoyo incondicional, a mis hijas Regina y Minerva por darme fuerza todos los días, a mis padres por ayudarme en todo momento.

Les doy las gracias a mis compañeros de clase por tantas horas de convivencia y haber hecho tan agradable mi estancia en la maestría.

Un gran reconocimiento a mis maestros por todos sus consejos y conocimiento transmitidos, muy especialmente a la Dra. Yasmín A. Ríos Solís por su constante apoyo en mi formación cómo investigador y su excelente asesoría a los largo de esta tesis.

A mis revisores el Dr. Igor S. Litvinchev y el Dr. Edgar Possani Espinosa les agradezco por formar parte de este comité de tesis, por sus consejos y total disposición a lo largo de esta tesis.

El apoyo que F.I.M.E. y CONACyT brindaron fue fundametal para la realización de este trabajo.

Resumen

I.Q. Miguel Lorenzo Morales Marroquín.

Candidato para el grado de Maestro en Ciencias
con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.
Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

Optimización de la producción en máquinas en paralelo de inyección de plástico

Número de páginas: 103.

Objetivos y método de estudio: El objetivo es abstraer y modelar matemáticamente el problema de producción de piezas de plástico en máquinas de inyección en paralelo considerando los productos finales.

Se desarrolla un modelo matemático de programación entera. Diseñamos varios algoritmos de resolución de manera a obtener soluciones de buena calidad en un tiempo razonable. Se diseñaron 9 heurísticos constructivos además de probar un algoritmo de ramificación y acotamiento.

Finalmente, construimos un generador de instancias para validar los modelos y los algoritmos propuestos de manera empírica. Hemos abordado el problema de manera innovadora mediante el uso de heurísticos, con los cuales obtuvimos soluciones eficientes en cuanto al tiempo de ejecución y calidad de respuesta.

El objetivo principal de esta investigación, es apoyar a empresas con una metodología que ayude a la rápida toma de decisiones y con esto obtener una producción cercana a la óptima.

La metodología científica que se utiliza para la realización de esta investigación se describe a continuación:

- Revisión bibliográfica: Se hace una extensa revisión bibliográfica de problemas de calendarización de máquinas en paralelo, problemas de asignación y problemas de tamaño de lote. Además, se hace una revisión sobre máquinas que trabajan con moldes de inyección de plástico.
- Formulación matemática: Ya planteado el problema, se hace un modelo matemático de programación entera con la mayoría de sus características como los 3 niveles de asignación producto-piezas, piezas-moldes, moldes-máquinas, máquinas y moldes trabajando en paralelo tiempo finito de trabajo y los moldes trabajan con lotes de piezas. Así mismo, se hace una prueba de complejidad que demuestra que el problema es NP-Duro.
- Evaluación con el método de ramificación y acotamiento: El objetivo de la experimentación es comprobar si se pueden obtener soluciones exactas del problema de una manera rápida y eficaz mediante un algoritmo de ramifiación y acotamiento, aún siendo el problema NP-Duro.
- Diseño de heurísticos: Se desarrollarón 9 algoritmos de tipo heurístico debido a la necesidad de obtener resultados cercanos al óptimo y de forma rápida.
- Evaluación del desempeño de heurísticas: Esta segunda etapa de experimentación es para validar y comparar los 9 heurísticos.

CONTRIBUCIONES Y CONCLUSIONES: Entre las aportaciones mas destacadas podemos mencionar las siguientes:

1. Se plantea el problema el cual tiene caracteristicas como los 3 niveles de asignación productos-piezas, piezas-moldes, moldes-máquinas, máquinas y moldes trabajando en paralelo con tiempo finito, dimensionamiento de la producción y los moldes trabajando por lotes. Lo anterior en conjunto nunca antes había sido abordado en la literatura.

2. Se diseña un nuevo modelo matemático de programación entera lineal, que cubre las caracteristicas mencionadas del problema.

3. Se hace una prueba de complejidad del problema, en la cual se demuestra que pertenece a la clases de complejidad de NP-Duro.

4. Se desarrolla un generador de instancias que se apega las particularidades del problema.

5. Dado que el modelo programación entera no se resuelve en tiempo eficiente mediante un algoritmo de ramificación y acotamiento, se desarrolla una metodología de solución la cual consta de diferentes heurísticos de tipo constructivo.

Concluimos que dado el funcionamiento de los heurísticos, es de suma importancia el escojer el producto correcto a fabricar debido a que pueden presentarse cuellos de botella al momento de decidir el producto a fabricar. Además, también son de suma importancia la asignaciones de las piezas a las combinaciones molde-máquina. También, podemos agregar que tanto la respuesta de los heurístico como la del metodo exacto, dependen del tamaño de instancia y su densidad.

Firma del director: _____

Dra. Yasmín Á. Ríos Solís

Capítulo 1

Introducción

En esta investigación se estudia la planeación de la producción de una empresa dedicada a la manufactura de productos de plástico en máquinas de inyección que trabajan en paralelo. Estas máquinas necesitan de equipo auxiliar para poder trabajar, que en este caso son moldes. Al hacer la planeación hay que hacer una serie de asignaciones para poder dedicar los recursos de manera óptima.

Cabe destacar que este problema nunca ha sido abordado en la literatura. A lo largo de esta tesis se desarrolla una metodología de solución para resolver este problema el cual es de gran importancia en la industria de la manufactura del plástico.

1.1 Planteamiento del problema

Este problema surge en una empresa que se dedica a hacer productos de plástico en máquinas de inyección. La empresa busca aprovechar todos sus recursos al máximo para tratar de cumplir con su demanda y de esta manera aumentar su producción.

Los productos que fabrica esta empresa son hechos a partir de varias piezas. Estos producto tienen la particularidad de que pueden tener piezas que solo sirven para ese producto, exclusivas y piezas que pueden compartir con otros productos, En el ejemplo de la Figura 1.1 se muestran dos productos diferentes que utilzan piezas en común y además tienen piezas exclusivas.

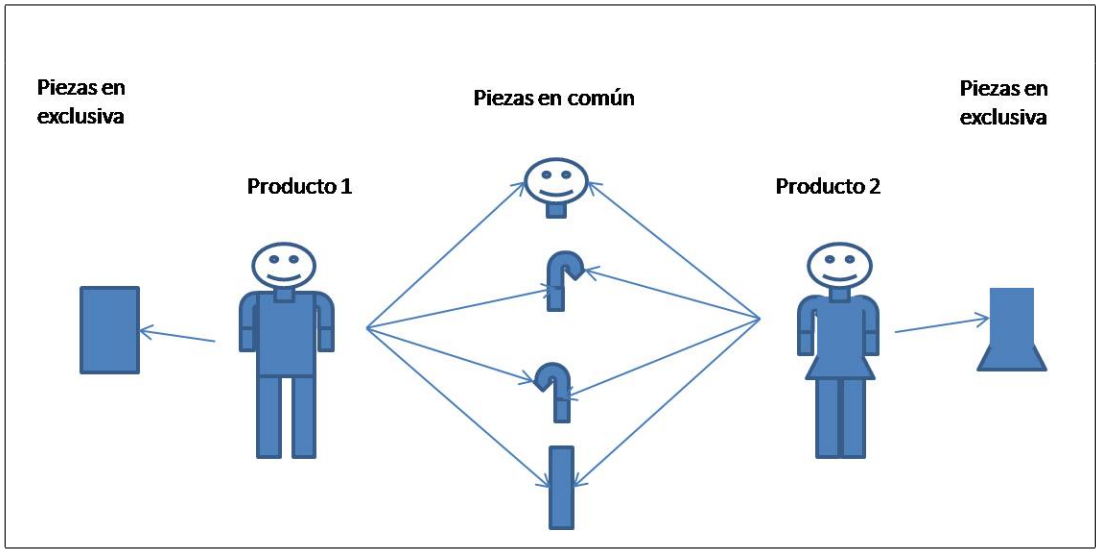


Figura 1.1: Los productos pueden tener piezas exclusivas y piezas en común.

En la Figura 1.2 se muestra un ejemplo donde se describe un producto i, unido a las diferentes piezas p que lo conforman. También, se muestra en la figura que cada producto i tiene un precio v_i y una demanda d_i. El parámetro c_p^i indica el número de piezas de tipo p que el producto i requiere.

Las piezas p están hechas en moldes. Como se comenta en la Sección 2.1, a los moldes se le inyecta el polímero líquido en las cavidades para hacer las diferentes piezas. Ahora bien, un molde puede tener varias cavidades, donde definimos cavidades como diferentes orificios que tiene un molde con la forma de cierta pieza.

Las cavidades de un molde j pueden ser de un mismo tipo de pieza p, o de diferentes tipos. Sin embargo, al momento de estar haciendo la inyección del plástico solo se puede usar las cavidades de un mismo tipo de pieza p.

En el ejemplo de la Figura 1.3, del lado izquierdo notamos cómo un mismo molde j tiene diferentes tipos de cavidades y por lo tanto tiene la capacidad de hacer diferentes tipos de piezas p. En cuanto el lado derecho de la figura se ejemplifica que un molde j solo puede hacer un tipo de pieza p a la vez y estas salen del molde por lotes, donde lotes es la cantidad de cavidades ($b_{p,j,k}$) de un mismo tipo que se producen por cada ciclo de un molde j y el ciclo representa la cantidad de tiempo

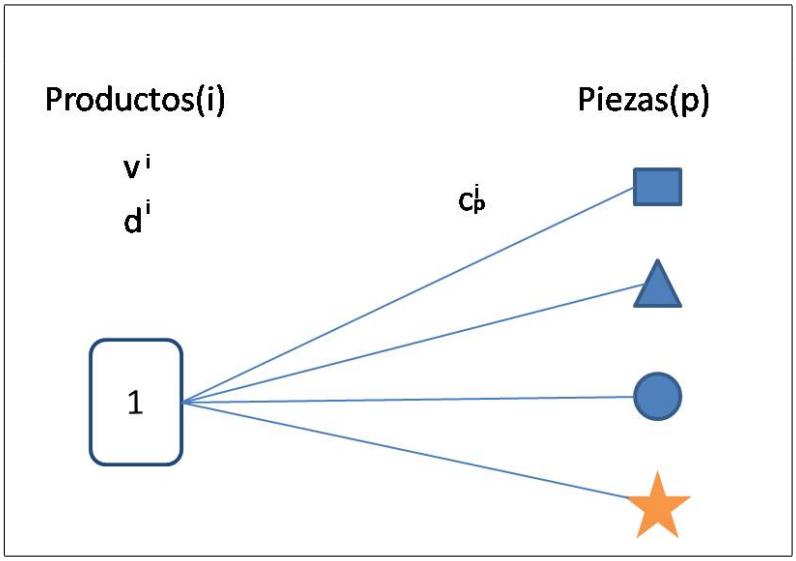


Figura 1.2: El producto 1 está unido por arcos a las pieza que lo conforman. Estos arcos pueden tener diferentes valores los cuales indicarían la cantidad de piezas de un mismo tipo que se necesitan por producto (c_p^i).

necesaria para producir un lote de piezas p en el molde j y la máquina k.

Es importante hacer mención que una misma pieza p puede ser hecha en diferentes moldes j, como se muestra en el ejemplo de la Figura 1.4.

En cuanto a los moldes j, estos deben ser instalados en las máquinas k. Vale la pena mencionar que el tiempo de instalación de los moldes depende de la máquina en que se instalen. Hay que destacar que los tiempos de instalación son mucho mayores a los tiempos de ciclo, ya que los moldes pueden ser muy pesados y se puede necesitar la ayuda de un montacargas o una grúa pequeña para su instalación o desinstalación. El tiempo de instalación y desinstalación son aspectos importantes del problema ya que pueden afectar al tiempo disponible para la producción.

Es importante señalar que los moldes pueden ser compatibles con más de una máquina. Por ejemplo, supongamos que tenemos 2 moldes que son compatibles con la máquina k. Ambos pueden hacer la pieza p que necesitamos producir a una razón de 3 piezas cada 2 minutos y 3 piezas cada 3 minutos, respectivamente. Tienen un tiempo de instalación de 60 minutos y 45 minutos, respectivamente y se tiene un

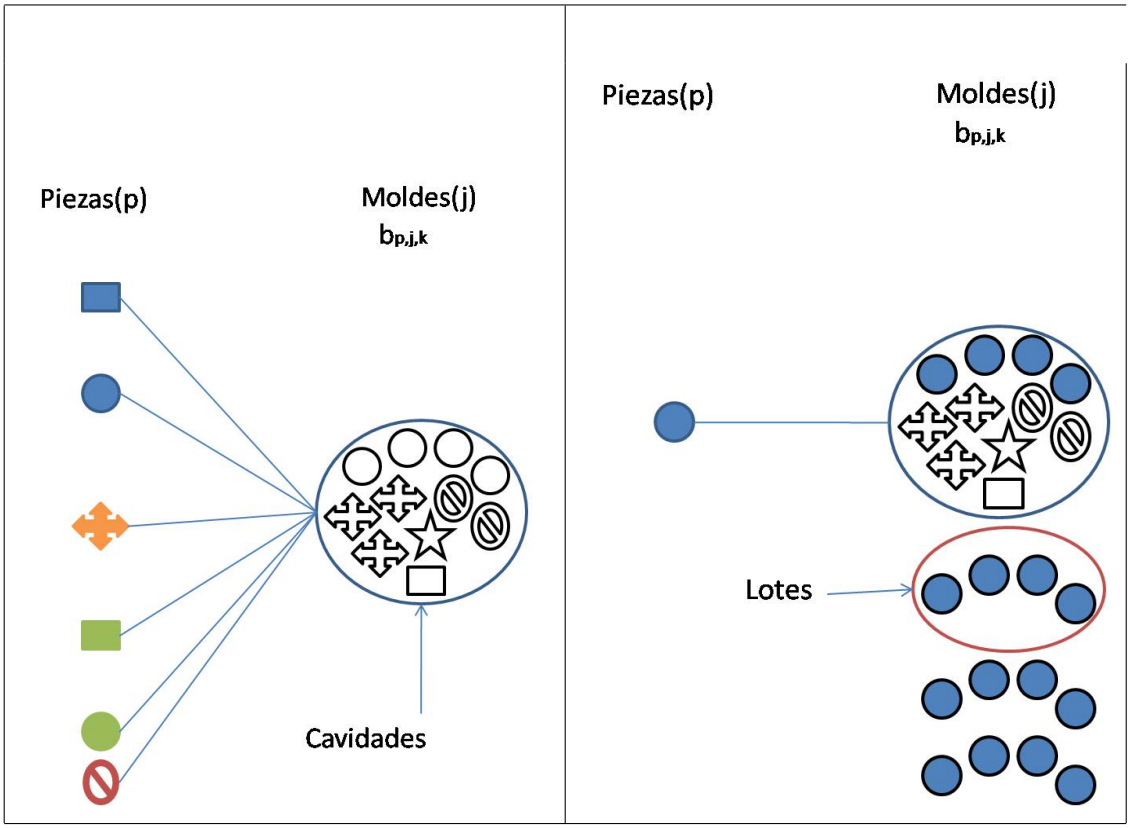


Figura 1.3: En el lado izquierdo de la figura presenta un molde que puede hacer varias piezas y del lado derecho se expone que los moldes producen por lotes.

tiempo disponible de máquina de 80 minutos. En la Figura 1.5 se visualiza esta situación a detalle. En la Figura 1.6 se muestra una grafica de Gantt que representa las dos posibilidades del ejemplo anterior, donde se observa que si se instala el molde 1, solo se obtendrían 30 piezas y en cambio si se instala el molde 2 se obtendrían 33 piezas.

A continuación se presenta un breve resumen de las particularidades y supuestos del problema:

- Cada producto i tiene un valor monetario solo hasta que las piezas que lo componen estén completas.
- Cada producto i está acotado por su demanda que es conocida de antemano y raramente se alcanza.

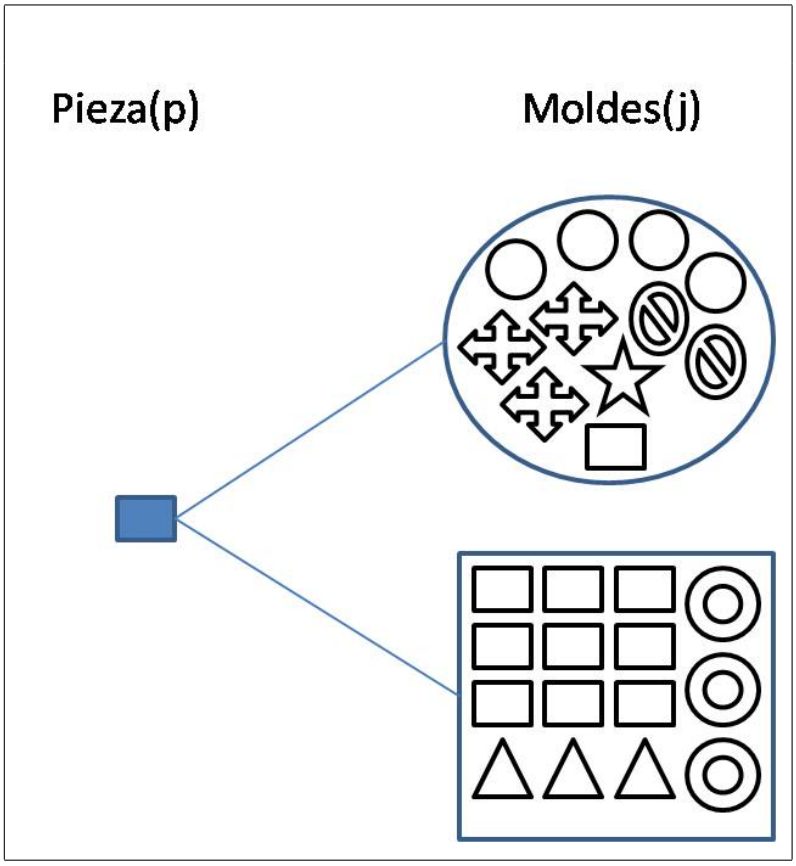


Figura 1.4: En la figura se observa que una misma pieza puede ser hecha en más de un molde.

- Cada producto i puede tener piezas exclusivas y piezas genéricas que comparte con otros productos, lo cual hace que haya competencia entre los productos por las piezas a asignar.

- Las piezas se pueden hacer en varios moldes, así como cada molde j puede hacer varias piezas p pero no a la vez. Por lo tanto, hay una competencia de las piezas por los moldes en los que se van a fabricar.

- Cada molde j puede ser instalado en más de una máquina k y a su vez, a cada máquina k se le puede instalar más de un molde j. Esto genera un espacio de soluciones factibles grande, debido a la diversidad de asignaciones molde-máquina posibles.

- Las máquinas y moldes trabajan con un tiempo finito.

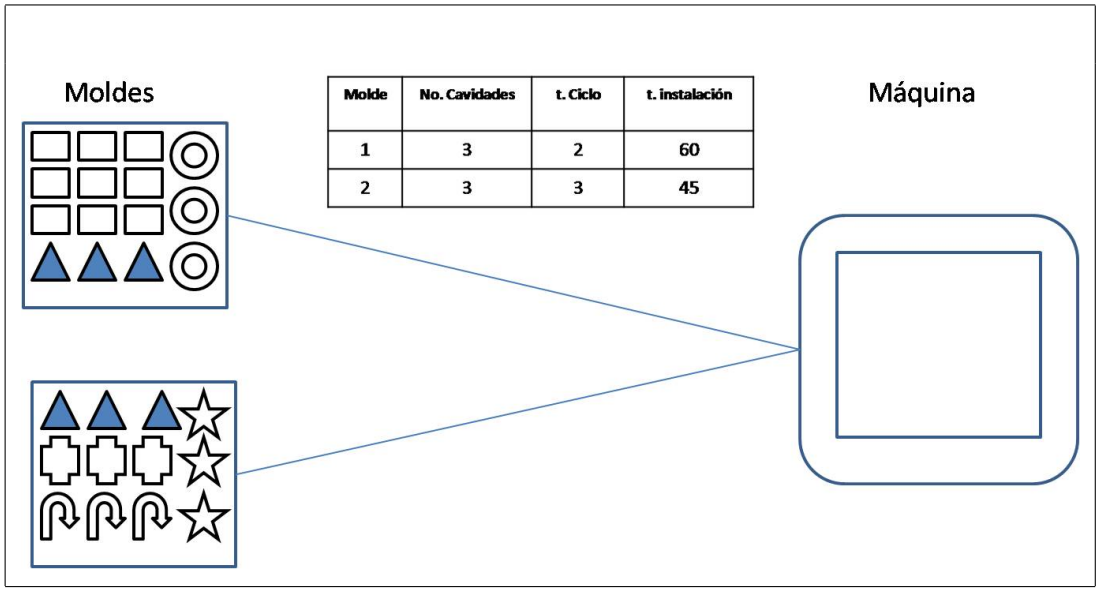


Figura 1.5: En este ejemplo se quiere tomar la decisión de qué molde usar para producir más de un tipo de pieza.

- Los moldes pueden ser instalados en más de una máquina en cada periodo.
- La cantidad de piezas p que produce un molde j por ciclo está dada por el número de cavidades de las piezas entre el tiempo de ciclo.
- Al empezar el periodo no hay moldes instalados y al finalizarlo todos los moldes ya han sido desinstalados.

A lo largo de este trabajo para simplificación llamaremos a el problema como

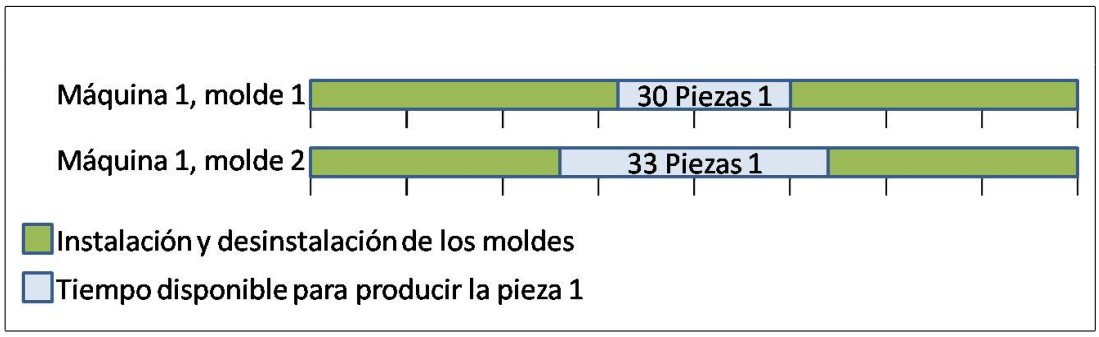


Figura 1.6: En esta figura se muestra un diagrama de Gantt, las opciones de producir la pieza 1 en la máquina 1 con el molde 1 produciendo 30 piezas o el molde 2 donde se producen 33.

PPMM. Debido a que son las siglas de productos, piezas, moldes, máquinas.

Ya por último, se muestra en la Figura 1.7 el diagrama de un ejemplo específico de compatibilidad del PPMM. En este diagrama se muestra qué piezas necesita cada producto y en qué moldes y máquinas pueden ser fabricadas estas piezas.

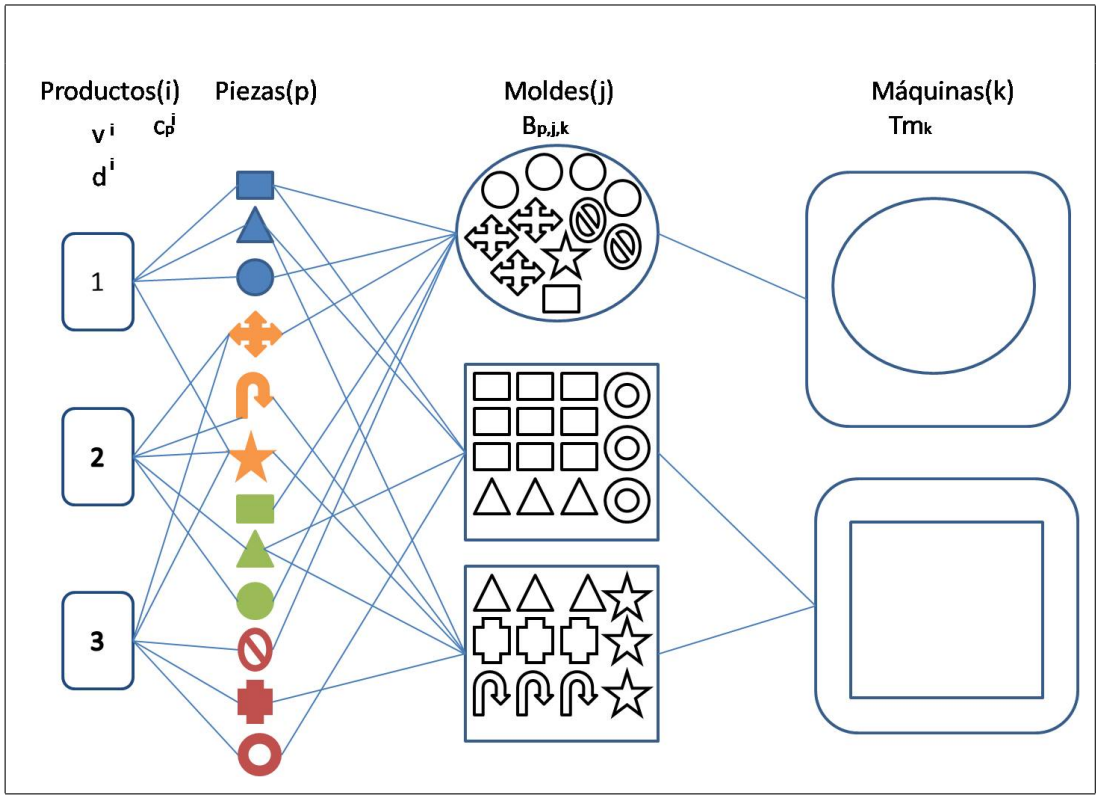


Figura 1.7: Ejemplo de diagrama de compatibilidad del problema PPMM.

1.2 Objetivo

Primeramente plantear las particularidades del PPMM. Asimismo, diseñar varios tipos de instancias que ejemplifiquen al problema real. A continuación hacer un modelo de programación entera que se adapte al PPMM.

Ahora bien, implementar el modelo de programación lineal entera de PPMM en un software que utilice un método exacto, en este caso un algoritmo de ramificación y acotamiento, para resolver los diferentes tipos de instancias del PPMM.

Ademas, implementar un algoritmo tipo heurístico para resolver los diferentes tipos de instancias del PPMM.

El objetivo principal de esta investigación, es apoyar a estas empresas con una metodología que ayude a la rápida toma de decisiones y con esto obtener una producción cercana a la óptima.

1.3 Relevancia del problema

1.3.1 Relevancia en la industria

En una gran cantidad de empresas las decisiones sobre la producción las toma una persona a su criterio, debido a su experiencia y a la falta de herramientas que ayuden a la toma de decisiones. Aunque se tiene gran experiencia en el proceso mismo, debido a la gran cantidad de decisiones a tomar, la posibilidad de haber tomado las decisiones correctas para tener una producción óptima es mínima.

El objetivo principal de esta investigación, es apoyar a estas empresas con una metodología que ayude a la rápida toma de decisiones y con esto obtener una producción cercana a la óptima.

Lo anterior tiene una enorme repercusión ya que esto implica un mayor margen de ganancias en las empresas debido a que se aprovechan de mejor manera los recursos disponibles.

1.3.2 Relevancia académica

En esta investigación el problema que se modela e intentamos resolver, tiene una gran cantidad de peculiaridades interesantes como se menciona previamente. El PPMM dá la posibilidad de abrir nuevas líneas de investigación, ya que no se encontró literatura científica que abarque todas las particularidades de este problema conjuntamente.

1.4 Metodología científica

La metodología científica que se utiliza para la realización de esta investigación se describe a continuación:

- Revisión bibliográfica: Se hace una extensa revisión bibliográfica en problemás de calendarización de máquinas en paralelo, problemás de asignación y problemás de tamaño de lote. Además, se hace una revisión sobre máquinas que trabajan con moldes de inyección de plástico.
- Formulación matemática: habiendo especificado el problema, se plantea un modelo matemático de programación entera lineal que cumpla con todas sus particularidades. Se hace una prueba de complejidad que demuestra que el PPMM es NP-Duro.
- Experimentación: El objetivo de la experimentación es comprobar si se puede resolver el problema de una manera rápida y eficaz, aun siendo el problema NP-Duro.
- Desarrollo de los algoritmos heurísticos: Se busca desarrollar diversos algoritmos de tipo heurístico debido a la necesidad de obtener resultados cercanos al óptimo y de forma rápida.
- Experimentación: Esta segunda etapa de experimentación es para validar los algoritmos heurísticos.

1.5 Estructura de tesis

- En el Capítulo 1, se da una breve introducción de la investigación. Además, se resaltan los objetivos de la investigación y metodología a implementar.
- En el Capítulo 2, se explican brevemente, algunos conceptos necesarios para poder comprender mejor el alcance y totalidad de esta investigación.

- En el Capítulo 3, se aborda la modelación matemática del PPMM. En ésta veremos más a fondo cómo se modelan cada una de las restricciones. En este capítulo presentamos la demostración de que el PPMM es NP-Duro.

- En el Capítulo 4, se justifica la necesidad de diseñar e implementar algoritmos heurísticos para resolver el PPMM. Además, se explica paso a paso el funcionamiento de los mismos.

- En el Capítulo 5, expondremos los resultados experimentales a los cuales se llegaron en el transcurso de la investigación.

- Por último en el Capítulo 6, se exponen las conclusiones obtenidas en esta investigación así como trabajo futuro.

Capítulo 2

Marco teórico

En este capítulo se da una breve introducción al proceso de moldeo por inyección de plástico, así como, una breve introducción a la optimización discreta y los algoritmos y métodos para resolverla. Además, se presentan algunos problemas clásicos de la investigación de operaciones que son necesarios para el desarrollo de esta tesis. Por último se presenta una revisión de literatura relacionada con esta investigación.

2.1 Moldeo por inyección de plástico

En esta sección daremos una introducción al proceso de moldeo por inyección de plástico, la información fue recopilada mayormente de [5].

El moldeo por inyección de plástico es una industria relativamente nueva, desde sus orígenes en 1872 cuando Jonh Wesley Hyatt y su hermano Isaiah patentaron la primera máquina de inyección de plástico revolucionando al mundo de la industria. Aunque no fue hasta principios de siglo XX, cuando la industria de los moldes de inyección fue formada como tal.

La industria de la inyección de plástico tuvo un gran salto en la década de los 40's, como muchas otras industrias, debido a la segunda guerra mundial, ya que ésta provocó un aumento en la demanda de productos que fueran hechos de materiales de bajo costo y que se pudieran producir en masa.

También, en la década de los 40's, se revolucionó la industria de los moldes de inyección, gracias James Hendry y su máquina de tornillo de inyección, la cual sustituiría a la máquina convencional de los Hyatt que funcionaba inyectando el plástico en algo parecido a una aguja a través de un cilindro caliente hacia el molde. Este nuevo sistema usaba una barrena para inyectar el plástico al molde lo cual también ayudaba a mezclar de manera más uniformemente el plástico.

Otra de las ventajas de la barrena es que se necesitaba menos energía que con el método de los Hyatt, ya que la fricción generada por la barrena ayudaba a que se necesitara menos energía externa.

En los siguientes años se produjeron más cambios en la industria de la inyección de plásticos, principalmente en los siguientes puntos:

- Energía, se busca optimizar la energía para gastar la menos posible en el proceso. Se usaron resistencias eléctricas para producir el calor necesario para el proceso.
- Moldes, cada día se trata de hacer moldes más ligeros y rápidos.
- Materiales, siempre hay una constante investigación por diseñar nuevos polímeros, aleaciones o mezclas que sean más fáciles de trabajar, de menor costo y mucha mayor calidad.
- Reciclaje, en nuestros tiempos es muy importante encontrar materiales plásticos que sean reciclables.

Para resumir, el moldeo por inyección de plástico es una industria relativamente nueva y con un gran crecimiento ya que permite sustituir a una gran gama de materiales por polímeros de menor costo, además es un proceso sencillo y se puede hacer prácticamente cualquier tipo de forma en los moldes, esto genera una gran cantidad de áreas de oportunidad para innovar en esta industria desde la investigación de nuevos materiales, ahorro de energía hasta la mejora de la producción usando la investigación de operaciones.

Para más información acerca de moldeo por inyección de plástico se pueden consultar las referencias [5], [20].

2.2 Complejidad computacional

En esta sección explicaremos breve e intuitivamente lo complejidad computacional, en [35], [21] y [23] se puede encontrar una referencia más completa sobre complejidad computacional, así como ejemplos del mismo.

En un enfoque informal de la teoría de la complejidad computacional, se define como la ciencia que estudia la complejidad teórica de los problemas, principalmente de la ciencias de la computación y matemáticas. Por lo general, mide la cantidad de tiempo o pasos necesarios para resolver un problema.

Los tipos de complejidad que usamos en este trabajo son P y NP-Duro. Un problema se dice es de complejidad P cuando puede ser computado en tiempo o cantidad de pasos calculados a partir de una formula polinómica, en otras palabras en tiempo polinómico. Mientras que para definir NP-Duro necesitamos primero definir NP-completo y estos son los problemas que no pueden ser terminados en un tiempo determinista o en otras palabras tiempo polinomial no determinista. Ahora los problemas NP-Duro se refieren a los problemas que al menos son tan difíciles como los problemas NP-Completo.

También, es importante mencionar que la manera de probar la complejidad de un problema es pasando el problema a su versión decisión y comparándolo con otro que de igual manera este en su versión decisión y se puede decir que es al menos tan complejo como el problema con el que se está comparando.

En la Sección 3.2 probamos que el problema PPMM es NP-Duro haciendo una reducción a el problema de multiple mochila.

2.3 Programación lineal

La Programación lineal es una rama de las matemáticas aplicadas y ciencias de la computación, relacionada con la investigación de operaciones y se encarga principalmente de extraer de sucesos físicos, sistemas, procesos, en una serie de relaciones matemáticas que expliquen fenómeno. A las relaciones matemáticas que describen el fenómeno se le denomina modelo matemático, que suelen incluir una función objetivo ya sea minimizar costos, fallos, recursos utilizados, tiempo, personal, distancias, entre otras cosas o bien maximizar ganancias, producción, por mencionar algunas. La obtención de una solución del fenómeno se lleva a cabo especificando una serie de decisiones sobre los valores de las variables en el modelo.

La Programación lineal puede ser usada para cualquier área como medicina, ingeniería, ciencias sociales, economía, química, biología, etcétera. Ahora bien, los modelos de programación lineal están formados por función objetivo, parámetros, restricciones, variables de decisión y espacio de soluciones.

Los parámetros son el conjunto de datos extraídos del procesos al modelo y estos no cambian. Por ejemplo, cantidad de materia prima, demanda, costos, precios, tiempo, numero de máquinas, dinero, entre otros.

Las variables de decisión son representaciones matemáticas de las posibles decisiones que hay en el proceso y con los diferentes valores en estas podemos obtener diferentes soluciones del modelo. Por ejemplo, cantidad de productos a fabricar, camino a escoger, tipo de transporte, qué máquina usar, qué artículos llevar, dónde localizar una fábrica, horarios, qué sembrar, por mencionar algunas.

Las restricciones son una serie de igualdades e inigualdades que acotan el sistema y no permiten que las variables tomen cualquier valor. Por ejemplo mínino a producir, máximo a producir, limitación en la materia prima, tiempo disponible.

El espacio de soluciones, es el conjunto de todas las soluciones que cumplan con las restricciones del sistema, es decir, el conjunto de todas las soluciones factibles.

La función objetivo es la relación matemática que busca tener un valor de máximo o mínimo, dependiendo del caso.

El modelo (2.1) es un ejemplo de un modelo matemático de programación lineal.

$$\text{máx} \sum_{i}^{n} V_i \cdot X_i \tag{2.1}$$

$$sujeto\ a: \tag{2.2}$$

$$\sum_{i=1}^{n} P_i \cdot X_i \ \leq C.$$

En este caso, X_i representa a una variable de decisión, V_i y P_i son parámetros del modelo, la ecuación (2.2) es la restricción y la ecuación (2.1) es la función objetivo, que se trata de maximizar.

Hay que destacar, que este problema por simple que parezca es un problema del tipo NP-Duro lo cual significa que muy probablemente no se puede resolver fácilmente. En los problemas de optimización los problemas tienen una complejidad computacional, pero principalmente los podemos dividir en dos clases P y NP-Duro, en la Sección 2.2 se explica más a detalle.

En [4], [10], [48] y [49] se puede encontrar más información acerca del tema de optimización.

Los problemas de optimización se pueden clasificar en varias ramas, algunas de ellas serán explicadas más a detalle a continuación:

- Programación lineal continua

  Un modelo de programación lineal continua se define como un modelo matemático. Pero este tiene la particularidad de que su función objetivo y restricciones son lineales, además sus variables son de tipo continuo.

  Un ejemplo, de modelo de programación lineal está representado en las ecuaciones (2.3), (2.4) y (2.5).

$$\text{máx} \sum_{i} g(X_i) \tag{2.3}$$

$$f(X_i) \leq c_i \qquad \forall i \tag{2.4}$$

$$X_i \geq 0 \tag{2.5}$$

Donde f y g son funciones lineales, (2.3) es la función objetivo, (2.4) representa a una serie de restricciones y (2.5) se interpreta como que la variable de decisión X_i debe ser mayor a o igual a 0.

La programación lineal es bastante joven, ya que se considera que su nacimiento fue alrededor del año de 1947 cuando Dantzig invento el método simplex, Debido a esto a Dantzig se le considera "el padre de la programación lineal". Los problemas de programación lineal son simples de resolver, debido a que son de complejidad P.

La mayor parte de estos problemas pueden ser resuelto por el método simplex de manera rápida y eficiente, aunque el metodo simplex no es polinomial.

Otro método para la solución de problemas de programación lineal es el método de puntos interiores,por ejemplo, el de Narendra Karmarkar.

En la Figura 2.1, se observa intuitiva y brevemente el funcionamiento y diferencia entre el método simplex y el método de puntos interiores. En esta se muestra una gráfica que esta acotada por las restricciones del modelo representadas en color azul y estas delimitan el espacio de soluciones del problema. Ahora bien, las líneas de color verde representan las iteraciones del método simplex y se ve claramente como este sigue por extremos de las aristas hasta llegar al resultado óptimo, mientras el método de puntos interiores, representado por el color rojo, va por dentro del espacio de soluciones hasta llegar al óptimo.

- Programación lineal entera

  Un modelo de programación entera lineal se define como un modelo matemático, que a diferencia del modelo de programación lineal continua, contiene variables de decisión enteras.

$$\text{máx} \sum_i g(X_i) \tag{2.6}$$

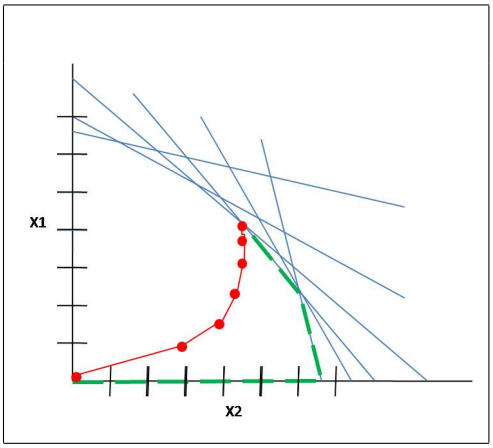


Figura 2.1: En esta figura se muestra una comparación entre el método simplex que está representado en color verde y método de puntos interiores que está representado en color rojo.

$$f(X_i) \leq c_i \qquad \forall i \tag{2.7}$$

$$X_i \in \mathbb{Z}^+ \tag{2.8}$$

En las ecuaciones (2.6), (2.7) y (2.8), se observa un modelo de programación entera, donde g y f son funciones lineales.

Este tipo de modelos son muy usados ya que representan una gran cantidad de procesos, en los cuales tenemos variables que representan cosas enteras como producto, cantidad de personas, fábricas, etcétera.

Este tipo de modelos representa una mayor dificultad de resolución ya que los algoritmos que son eficientes en el tiempo no los resuelven, debido a esto hay una gran cantidad de investigación en desarrollar algoritmos para su resolución y dependiendo del modelo pueden llegar a ser muy complicados.

Un algoritmo que se usa para resolver estos problemas de manera exacta es el de ramificación y acotamiento.

En la Figura 2.2 se muestra un ejemplo del espacio de soluciones para un problema de programación lineal entera, donde las líneas azules representan las restricciones y los vértices donde se intersectan las líneas punteadas de color rojo es el espacio de soluciones del modelo de programación entera.

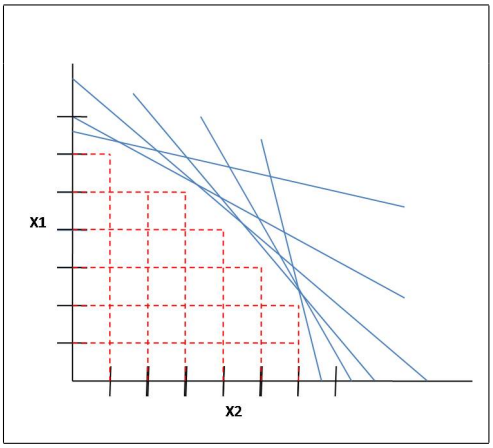


Figura 2.2: En esta gráfica se muestra el espacio de soluciones de un problema de programación lineal entera, representado por los vértices donde se intersectan las líneas punteadas de color rojo.

En [49] se puede encontrar información muy completa acerca de programación entera.

- Programación binaria

  Los modelos de programación binaria son modelos matemáticos en los cuales al menos una de sus variables de decisión es binaria. Es decir, solamente pueden tomar valores 0 y 1.

  En las ecuaciones (2.9), (2.10) y (2.11), se muestra un ejemplo de modelo binario. Donde f yg son funciones lineales.

$$\text{máx} \sum_i g(X_i) \tag{2.9}$$

$$f(X_i) \leq c_i \qquad \forall i \tag{2.10}$$

$$X_i \in \mathbb{B} \tag{2.11}$$

  Estas variables binarias son ampliamente utilizadas para modelar decisiones del tipo asignación, por ejemplo, si el pasajero va en el vuelo o no, si se va a producir un producto, si se va a instalar una fábrica, si se va a mandar un bus, entre otras.

  Por ejemplo, tenemos una empresa de I empleados e I tareas a desarrollar. Ahora bien, cada empleado i tarda una cantidad de tiempo $t_{i,j}$ en desarrollar

la tarea j, la idea es asignar a cada empleado una tarea a forma de minimizar el tiempo en desarrollar todas las tareas, en las ecuaciones (2.12), (2.13), (2.14) y (2.15) se muestra el modelo matemático de este problema donde $X_{i,j}$ es la variable de decisión, 1 si el empleado i hace la tarea j y 0 en otro caso.

$$\text{mín} \sum_{i,j} t_{i,j} \cdot X_{i,j} \tag{2.12}$$

$$\sum_{i} X_{i,j} = 1 \qquad \forall j \tag{2.13}$$

$$\sum_{j} X_{i,j} = 1 \qquad \forall i \tag{2.14}$$

$$X_{i,j} \in \mathbb{B} \tag{2.15}$$

En la Figura 2.3 se muestra un ejemplo que representa este problema, en el cual los círculos de la izquierda representan a los empleados, los de la derecha a las tareas y las aristas representan el tiempo, a este tipo de grafo también se le conoce como grafo bipartito.

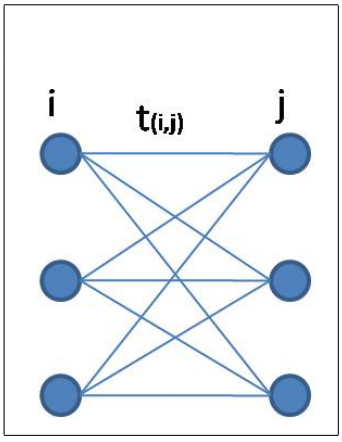


Figura 2.3: Ejemplo de problema con variables de asignación binarias.

Un modelo entero mixto se puede definir como aquel que usa variables continuas y enteras. En esta tesis se propone un modelo entero mixto, para el desarrollo del PPMM en la Sección 3.1.

Método simplex

El método simplex fue inventado por Dantzig en 1947, este método hasta el día de hoy sigue siendo el más usado para resolver problemas de programación lineal, ya que se resuelve de forma rápida y exacta en la mayoría de los casos.

El funcionamiento de este método es el siguiente, se va buscando soluciones factibles en los puntos extremos del espacio de soluciones hasta que la función objetivo no se pueda mejorar más.

En [10] se encuentra información más completa acerca del desarrollo y funcionamiento del método simplex.

2.3.1 Algoritmo de ramificación y acotamiento

La mayor parte de esta sección está basada en la información obtenida en [49].

El algoritmo de ramificación y acotamiento o bien B&B (branch and bound) por sus siglas en inglés es un algoritmo usado para resolver modelos de programación entera lineal. Este método garantiza la optimalidad de la solución.

El primer articulo de programación entera donde se presento el algoritmo de ramificación y acotamiento fue en [26], como se menciona en [49].

El algoritmo de ramificación y acotamiento se puede interpretar como un árbol en donde cada hoja representa un subespacio del problema, es decir, cada subespacio contiene un subproblema del problema original con restricciones adicionales y cuando un subespacio no contiene la solución óptima se poda o descarta.

El algoritmo se desarrolla de la siguiente manera, primero supongamos que tenemos un modelo de programación entera lineal. Ahora, hacemos una relajación continua del problema, es decir, suponemos que todas sus variables son del tipo continuo y como se menciona anteriormente estos problemas son fáciles.

A continuación resolvemos el problema continuo, el resultado será nuestra cota superior, en el caso de que se esté maximizando. Ahora, pasamos a la fase de rami-

ficación, tomamos el valor de una de las variables que relajamos que no tenga valor continuo, por ejemplo, X_1 y agregamos dos ramas al árbol cada rama representa una restricción diferente. La primera rama se le agrega la restricción $X_1 \leq \lfloor X_1 \rfloor$ y la segunda rama se le agrega la restricción $X_1 \geq \lceil X_1 \rceil$, donde $\lfloor\rfloor$ y $\lceil\rceil$ significan redondear al entero inferior próximo y redondear al entero superior próximo respectivamente.

Lo anterior nos da como resultado dos modelos de programación continua, los cuales serán resueltos de manera continua y estos a su vez generan más ramas, es decir, modelos de programación continua.

En la Figura 2.4, se muestra una representación del árbol de soluciones de un algoritmo de ramificación y acotamiento. En esta representación se ven una serie de nodos que representan modelos continuos. Colocamos entre paréntesis (valor de la solución, valor de primera variable..., valor última variable) y en las aristas se muestra la restricción que fue agregada al modelo. En este caso, el modelo deja de buscar soluciones ya que dos de sus ramas se podaron debido a que sus resultados no eran factibles.

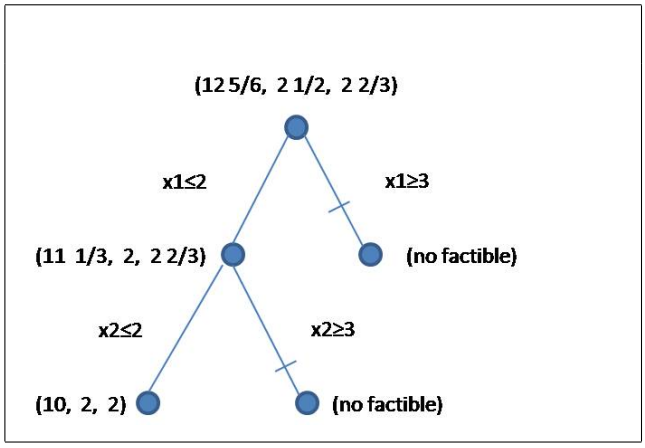


Figura 2.4: Ejemplo de algoritmo de ramificación y acotamiento en su fase de ramificación.

De manera intuitiva en la fase de acotamiento o poda se busca optimizar el árbol a forma de no tener que buscar en todos los nodos. Así podamos los nodos en los que ya no es necesario continuar la búsqueda. Hay que destacar que los no-

dos se cancelan al contar con una solución factible, es decir, que las variables solo tengan valores enteros. Así, obtenemos una cota superior en caso de problemas de maximización y una cota inferior en caso de minimización y al contar con esta cota si cualesquiera otro de los nodos que se están explorando tiene una valor de solución que no sea tan bueno como el de la cota, este nodo se poda ya que en este nodo sería imposible encontrar una mejor solución que la encontrada en el nodo anterior. Otra razón de poda es una nodo que arroja un resultado infactible.

En la Figura 2.5, se ejemplifica el algoritmo de ramificación y acotamiento en su fase de poda. Aquí se muestra como en uno de los nodos tenemos una cota superior factible. Ahora, uno de los nodos es infactible mientras los otros 2 aún pueden ser explorados, sin embargo ya que la cota obtenida tiene un valor a 10 y el valor en los nodos es menor a este, estos nodos se cancelan debido a que ya no se podrá encontrar una mejor solución en estos que la que ya se encontró en el nodo anterior.

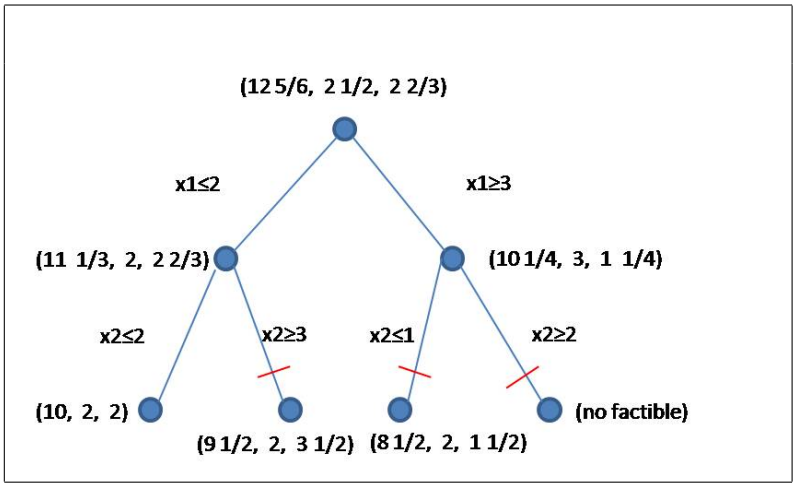


Figura 2.5: Ejemplo de algoritmo de ramificación y acotamiento en su fase de poda.

Para mayor referencia del desarrollo y funcionalidad del algoritmo de ramificación y acotamiento en [49].

En la Sección 5.2 se implementa un algoritmo de ramificación y acotamiento, para resolver instancias del problema PPMM.

2.3.2 Heurísticas

La palabra heurística viene del griego heuriskein, que significa descubrir o encontrar y esto se refiere a el proceso de búsqueda interna el cual trata de descubrir la naturaleza y significado de la experiencia, desarrollar métodos y procedimientos para su investigación y análisis futuro.

El algoritmo heurístico aplicado a la optimización y ciencias de la computación, es una regla extraída de la experiencia, pero no siempre justificada en la teoría. Ahora bien, en el área de la optimización los problemas NP-Duros son muy difíciles de resolver por los métodos exactos, debido a esto se busca la implementación de algoritmos heurísticos para obtener resultados con una mayor eficiencia. Sin embargo el algoritmo heurístico no siempre garantiza la optimalidad.

Los algoritmos heurísticos han sido aplicados desde los años cuarentas como, por ejemplo, en [38]. En sus primeros años a la heurística se le llamaba rule of thumb como se le menciona en [41].

Las heurísticas son bastante utilizadas debido a su corto tiempo de ejecución. Por esto son perfectas para la industria, ya que en una gran cantidad de casos se prefiere una respuesta rápida más que una exacta. Un punto importante a considerar es que no en todas las industrias se puede invertir en un software de optimización exacto.

Podemos clasificar las heurísticas en los siguientes dos tipos:

- Las heurísticas constructivas, comienzan sin una solución a partir de la realización de pasos iterativos de construcción. Usualmente usan reglas específicas basadas en el problema para la construcción de la solución.
- Heurísticas de búsqueda local, se comienza con una solución completa, se genera una estructura de vecindario, donde se hace, una búsqueda dentro del vecindario para encontrar una solución mejor.

En esta tesis se proponen 9 heuristicas del tipo constructivo para resolver el PPMM en la Sección 4.

Se puede encontrar más información acerca de heurísticas en [39], [34], [32] y [27].

El Capitulo 4 se diseñan diversos algoritmos heuristicos tipo constructivos para resolver el PPMM.

2.4 Problemas de optimización clásicos

En está sección, se habla de problemas clásicos en la optimización lo cuales son mencionados a lo largo de esta tesis.

Problema mochila 0/1.

El problema en si trata de una serie de decisiones en las cuales tenemos n artículos i, los cuales tienen un valor V_i y un peso P_i. Ahora bien contamos con una mochila para transportar estos artículos pero la mochila solo puede cargar un peso no mayor a C, la idea es transportar los artículos de mayor valor sin excederse en la capacidad de la mochila.

El problema mochila, ha sido ampliamente estudiado durante el siglo XX, ya que es uno de los problemas clásicos de la investigación de operaciones.

Como se menciona en [24] desde 1897 Mathews muestra como varias restricciones podían ser agregadas a una simple restricción de tipo mochila, [31]pero no fue hasta los años cincuentas que Bellman con su teoría de la programación dinámica produce el primer algoritmo para resolver de forma exacta este problema y hasta el año de 1957 Dantzig encuentra un método para resolver eficientemente la relajación continua de este problema.

En los años sesentas Gilmore and Gomory estudiaron ampliamente la resolución de este problema mediante programación dinámica. A su vez en el año de 1967

Kolesar experimentaba con el algoritmo de ramificación y acotamiento para resolver este problema como se menciona en [31].

A continuación se mostrara el modelo matemático del problema de la mochila.

$$\text{máx} \sum_{i}^{n} V_i \cdot X_i \tag{2.16}$$

$$sujeto\ a: \quad \sum_{i}^{n} P_i \cdot X_i \ \leq C \tag{2.17}$$

$$X \in \mathbb{B}$$

Donde X_i es una variable de decisión binaria, por lo tanto el vector X denota la solución del problema, en otras palabras que artículos van dentro de la mochila.

Es un problema NP-Duro como se puede corroborar en [24], [31], [21] y [35]. Sin embargo, es NP-Duro en el sentido débil ya que puede ser resuelto por un algoritmo pseudo-polinomial. En [24], [31], [37] y [10] se pueden encontrar mas acerca de este tema.

Problema múltiples mochilas con variables 0/1.

Presentamos en esta sección el problema de múltilples mochilas puesto que en la Sección 3.2 es utilizado para hacer una reducción polinomial en la prueba de complejidad del PPMM.

El problema múltiples mochilas con variables 0/1 también conocido como MKP por sus siglas en inglés, es una de las variantes del problema mochila y aunque es muy parecido al problema original este problema es más complicado de solucionar.

En este problema se cuenta con n productos i con diferentes pesos p_i y un valor V_j, al igual que en el problema mochila original. Sin embargo en este problema se cuenta con la particularidad de que se cuenta con m mochilas j y cada mochila tiene una capacidad de c_j. Ahora la idea es colocar los productos en las mochilas de tal manera que no se sobrepase la capacidad de ninguna de las mochilas y maximizar las suma de los valores de productos en ellas.

El modelo se plantea en las ecuaciones (2.18),(2.19) y (2.20) donde X es un vector variables de decisión binarias.

$$\text{máx} \sum_j^m \sum_i^n V_i \cdot X_{i,j} \tag{2.18}$$

$$sujeto\ a:$$

$$\sum_i^n P_i \cdot X_{i,j} \leq c_j \qquad \forall j \tag{2.19}$$

$$\sum_j^m X_{i,j} = 1 \qquad \forall i \tag{2.20}$$

$$X \in \mathbb{B}$$

$$X_{i,j} \begin{cases} = 1 \text{ si el articulo } i \text{ está en la mochila } j. \\ = 0 \text{ en cualquier otro caso.} \end{cases}$$

Hay una gran cantidad de algoritmos para resolver el problema de múltiple mochila, los cuales estan fuera del alcance de este trabajo. Se puede encontrar más información acerca de estos métodos y el problema de múltiple mochila en [37], [31], [31] y [30].

En la Sección 3.2 se reduce el problema de multiples mochilas para probar la complejidad del PPMM.

Problema de asignación.

La información de este capítulo fue extraída de [6].

En 1935 Philip Hall dio las condiciones necesarias y suficientes para una asignación perfecta. Esta es conocida hasta el día de hoy como el teorema de los matrimonios, esto gracias al matemático Hermann Weyl, ya que en 1949 le dio la siguiente interpretación.

Un grupo de señoritas serían los vértices U y un conjunto de caballeros serían los vértices V. Ahora bien, una arista $[i, j]$ indica si la señorita i es amiga del caballero j. Una asignación perfecta consistiría en casar a todos las señoritas con los caballeros donde estos solo pueden hacer pareja si son amigos.

En general, el problema de asignación consiste en asignar n elementos i como pueden ser trabajadores, estudiantes, maestros, pasajeros, máquinas, por mencionar

algunos a n elementos diferentes como tareas, salones, asientos, etcétera.

La complejidad computacional del problema de asignación perfecta en un grafo bipartita es de $O(n^{5/2})$ demostrado en [17]. Donde n es la mitad del numero de vertices totales del problema.

En las ecuaciones (2.21), (2.22) y (2.23) se describe un modelo de asignación en el cual $C_{i,j}$ se puede visualizar como un costo y $X_{i,j}$ es una variable binaria de asignación. En cuanto a las ecuaciones (2.22) y (2.23) sirven para asegurar que solamente un elemento i sea asignado a un elemento j y viceversa.

$$\text{mín} \sum_i^n \sum_j C_{i,j} \cdot X_{i,j} \tag{2.21}$$

$$\sum_i^n X_{i,j} \qquad = 1 \qquad \forall j \tag{2.22}$$

$$\sum_j^n X_{i,j} \qquad = 1 \qquad \forall i \tag{2.23}$$

$$X_{i,j} \in \mathbb{B} \tag{2.24}$$

$$X_{i,j} = \begin{cases} 1 \text{ si el elemento } i \text{ esta en asignado al elemento } j. \\ 0 \text{ en cualquier otro caso.} \end{cases}$$

Algunos problemas destacados de asignación son:

- Problema de asignación de suma lineal: Este se representa por la suma de las asignaciones en su función objetivo, por ejemplo, en la asignación de tareas a una serie de máquinas que están en serie y se busca minimizar el tiempo de trabajo.
- Problema de asignación de cuello de botella lineal: Esta representada por una función min max y esta se utiliza, por ejemplo, en la asignación de tareas a una serie de máquinas en paralelo y se busca minimizar el tiempo de trabajo.

Para más información acerca de esta sección en [6].

En la Sección 3.1 se desarrolla un modelo de programación lineal para resolver el problema PPMM, que contiene varios problemas de asignación.

Problemas de secuenciación.

Es importante mencionar que aunque en este trabajo no se está desarrollando un problema de secuenciación, es importante la comprensión del mismo debido a que está fuertemente relacionado con el mismo. Además, en la mayoría del trabajo relacionado trata de resolver un problema de secuenciación. En esta tesis no se considera el problema de secuenciación debido a que en [19] se plantea una metodología sencilla para implementar la secuenciación que fácilmente se puede aplicar a el problema PPMM.

Mucha de la información acerca de esta sección fue recopilada de [36].

los problemas de secuenciación, consisten en decidir el orden en que una serie de tareas sean procesada o asignandas a los diferente recursos, buscando optimizar un proceso o sistema. Los recursos pueden ser tiempo disponible en máquinas, personal, espacios disponibles para las salidas en un aeropuerto, procesamiento disponible en un procesador de una computadora, almacenamiento, entre otras. Asimismo, las tareas pueden ser piezas por fabricar, tareas por realizar en un restaurante, vuelos por salir, ciclos en un autobús, tareas por realizar en una computadora, entre muchas otras.

Los problemas de secuenciación juegan un importante papel en la industria de la manufactura y también en la industria de la transportación y distribución.

Los problemas de secuenciación puede ser de muchos tipos diferentes pero por lo general se describen con la siguiente notación $(\alpha\,|\beta|\,\gamma)$. Donde α describe el ambiente de las máquinas, β describe características y restricciones del proceso y γ describe el objetivo a ser minimizado.

Algunos de las diferentes descripciones de α pueden ser:

- 1: Una sola máquina.
- Pm: Máquinas idénticas en paralelo (Pm).

- Qm: Máquinas en paralelo con diferentes velocidades (Qm).
- Rm: Máquinas en paralelo no relacionadas (Rm).
- Fm: Flujo de taller (flow shop).
- FFc: Flujo flexible de taller (flexible flow shop).
- Jm: Tarea de taller (job shop).
- Fjc: Tarea flexible de taller (flexible job shop).
- Om: Taller abierto (open shop).
- rj: Fecha de lanzamiento (relase date).

Algunos de los objetivos a ser minimizado son:

- C_{max}:Tiempo en que la última tarea terminar el proceso(makespan).
- L_{max}:Retraso (lateness).
- $\sum w_j C_j$:Tiempo total de realización ponderada (total weighted completion time).
- $\sum w_j(1 - e^{rC_j})$:Tiempo total de realización ponderada descantado (discounted total weighted completion time).
- $\sum w_j T_j$:Tardanza ponderada total (total weighting tardiness).
- $\sum w_j U_j$:Ponderado del número de trabajos tardíos (weighted number of tardy jobs).

El tema de secuenciación es bastante amplio se puede encontrar más información en [11], [3], [22], [14], [9], [42] y [36].

2.5 Trabajos relacionados

En esta sección, hablaremos acerca de trabajos relacionados con las características similares al PPMM, además de las diferentes formas que se usaron para resolver estos problemas.

Es importante mencionar que aunque en este trabajo no se esta atacando la parte de secuenciación implícita en el problema, es usual que cuando se habla de máquinas se mencione el tema de secuenciación. Ya que en muchos de los procesos en la literatura como en [1], [13], [29], [43], [12], [45], [16], [44], entre otros, la secuencia entre tareas es de vital importancia debido a que la precedencia de una tarea puede significar una diferencia en el costo o tiempo de proceso. En el PPMM la precedencia no tiene impacto en la solución del problema por lo tanto se puede implementar fácilmente la metodología expuesta en [19] para dar secuencia a los resultados del PPMM.

Con respecto a la cantidad de máquinas, varios artículos de la literatura tratan con una sola máquina con muchas tareas, por ejemplo, [29], [12], [43], [13]. Esto es debido a que en muchos de ellos la precedencia de una tarea afecta la solución y como se menciona en [12] el problema es complicado de resolver con más de 12 tareas y es necesaria la implementación de heurísticas.

A continuación tenemos problemas más similares al presentado en este trabajo hablando de máquinas en paralelo. Estos problemas se pueden dividir en 2 clases: Máquinas similares como los problemas expuestos en [40], [28] y máquinas no relacionadas como los expuesto en [8], [15], [47], [44], [19], [18], [51], [25]. Particularmente en [2] se estudia casos de máquinas solas, en paralelo sin relación y con relación.

Como se mencionaba al principio, la mayoría de estos trabajos están enfocados a la secuenciación, por ello muchos de ellos se centran minimizar el tiempo o la tardanza de todas las tareas. En nuestro trabajo nos centramos en el dimensionamiento del lote, es decir, cuánto se puede producir. Entre algunos de los trabajos

relacionados que hablan también del dimensionamiento del lote tenemos a [19], [18], [16], [50], [7], [1], [33], [42].

En [46] aunque trata del secuenciamiento de las tareas en un hospital, tiene una estructura de asignaciones de tareas similar a nuestro problema. Sin embargo, en [46] busca minimizar a los pacientes no tratados, mientras que en nuestro problema buscamos solo maximizar las ganancias.

Nuestro problema está basado principalmente en el caso de estudio de [19] y [18]. Sin embargo, al agregar productos dependientes de piezas y la producción por lotes de piezas, el problema cambia totalmente y nos vemos en la necesidad de hacer un nuevo modelo matemático que se presenta en la Sección 3.1 y una nueva metodologia de solución que se expone a detalle en el Capítulo 4.

Capítulo 3

Formulación del modelo de programación entera lineal para el problema PPMM

En este cápitulo en la Sección 3.1, se explica paso a paso cómo se desarrolla el modelo de programación entera para el problema PPMM.

Así mismo, en la Sección 3.2, se describe la prueba de complejidad del PPMM. En donde se demuestra que el PPMM es NP-Duro.

3.1 Modelación matemática

Para formular a PPMM como un problema de programación lineal con números enteros. Hay que considerar que, un producto no tiene un valor hasta que el conjunto de piezas que lo constituyen esté completo y además, que los moldes trabajan por ciclos.

Para poder formular el PPMM se definen I, P, J y K, como el conjunto de productos, piezas, moldes y máquinas, respectivamente.

Además, se considera que para cada producto $i \in I$ existe un conjunto $p(i) \subseteq P$ piezas y a su vez toda pieza $p \in P$ debe formar parte de un conjunto $i(p) \subseteq I$ con el que sea compatible. Para cada pieza $p \in P$ existe un conjunto de moldes $j(p) \subseteq J$

con el que es compatible y cada molde $j \in J$ debe ser compatible con un conjunto de piezas $p(j) \subseteq P$. Para cada molde $j \in J$ existe un conjunto de máquinas $k(j) \subseteq K$ con el que es compatible. A su vez, para cada máquina $k \in K$ existe un conjunto de moldes $j(k) \in J$ compatibles.

Cabe mencionar, que los conjuntos $p(i)$, $i(p)$, $j(p)$, $p(j)$, $k(j)$ y $j(k)$ son diferentes del conjunto vacío.

Los parámetros necesarios para formular el modelo de programación entera, están descritos en la Tabla 3.1.

Parámetros	Descripción
V^i	Precio de cada producto i.
C_p^i	Cantidad de piezas p que necesita el producto i.
D^i	Demanda de cada producto i.
$H_{p,j,k}$	Tiempo que tarda un ciclo del molde j en la máquina k para hacer piezas p.
$B_{p,j,k}$	Cantidad de cavidades de la pieza p que tiene el molde j instalado en la máquina k.
$It_{j,k}$	Tiempo de instalación y desinstalación del molde j en la máquina k.
Tm_k	Tiempo disponible de la máquina k.

Tabla 3.1: Tabla de parámetros para el modelo de programación entera lineal del problema PPMM.

Como se plantea en la Sección 1.1, uno de los supuestos fuertes del PPMM es que al empezar el turno no hay moldes instalados y que todos los moldes se desinstalan al final del mismo. Sin perdida de la generalidad, se considera que $It_{j,k}$ es la suma del tiempo de instalación y desinstalación del molde j en la máquina k.

Otro punto de interés es que el parámetro $B_{p,j,k}$, además de indicarnos la cantidad de cavidades que tiene el molde j instalado en la máquina k para hacer

la pieza p, sirve como parámetro de compatibilidad entre la pieza p, el molde j y la máquina k. Por ejemplo, si el molde j tiene cavidades para hacer la pieza p y es compatible con la máquina k entonces $B_{p,j,k} > 0$, en cualquier otro caso $B_{p,j,k} = 0$.

En la Sección 1.1, también se habla de la necesidad de tratar el PPMM con lotes de fabricación de piezas, ya que se trabaja con moldes, en donde cada lote se define como la cantidad de piezas p que se obtienen de un molde j en una cierta cantidad de tiempo. Este lote se determina por las cavidades que tiene un molde j para fabricar la pieza p y el tiempo que tarda en hacer esa cantidad de piezas que definiremos como tiempo de ciclo y está dado por el parámetro $H_{p,j,k,}$ y es el tiempo que se tarda en producir un lote de piezas p, en el molde j instalado en la máquina k.

Dadas las especificaciones del PPMM, descritas anteriormente en la Sección 1.1, presentamos las variables de decisión y se describen en la Tabla 3.2.

Variables	Descripción
Pf^i	Indica el número de productos i a fabricar.
$X^i_{p,j,k}$	Cantidad de piezas p hechas en el molde j instalado en la máquina k, designadas a fabricar el producto i.
$T_{p,j,k}$	Tiempo que usa el molde j en la máquina k para hacer la pieza p.
$E_{p,j,k}$	Cantidad de ciclos que hace el molde j en la máquina k para hacer la pieza p.
$N_{j,k}$	Variable de asignación binaria que indica si el molde j fue instalado en la máquina k o no.

Tabla 3.2: Tabla de variables para el modelo de programación entera lineal del problema PPMM.

La ecuación (3.1), es la función objetivo del modelo y lo que busca es maximizar

las ganancias de vender el producto final,

$$\text{máx} \sum_{i \in I} V^i \cdot Pf^i. \tag{3.1}$$

El siguiente conjunto de restricciones, cumple con todas las características antes mencionadas, en la Sección 1.1.

$$Pf^i \leq \min_{p \in p(i)} \left[\frac{\left[\sum_{j \in j(p)} \sum_{k \in k(j)} X^i_{p,j,k} \right]}{C^i_p} \right] \qquad \forall i \in I \tag{3.2}$$

$$Pf^i \leq D^i \qquad \forall i \in I \tag{3.3}$$

$$E_{p,j,k} = T_{p,j,k} / H_{p,j,k} \quad \forall p \in P, j \in j(p), k \in k(j) \tag{3.4}$$

$$\sum_{i \in i(p)} \sum_{j \in j(p)} \sum_{k \in k(j)} X^i_{p,j,k} = \sum_{j \in j(p)} \sum_{k \in k(j)} E_{p,j,k} \cdot B_{p,j,k} \qquad \forall p \in P \tag{3.5}$$

$$\sum_{p \in P} T_{p,j,k} \leq Tm_k \cdot N_{j,k} \qquad \forall j \in J, k \in j(k) \tag{3.6}$$

$$\sum_{j \in J} \left[N_{j,k} \cdot It_{j,k} + \sum_{p \in P} T_{p,j,k} \right] \leq Tm_k \qquad \forall k \in K \tag{3.7}$$

$$\sum_{k \in K} \left[N_{j,k} \cdot It_{j,k} + \sum_{p \in P} T_{p,j,k} \right] \leq \max_{k \in K} Tm_k \qquad \forall j \in J \tag{3.8}$$

$$Pf^i \in \mathbb{Z}^+, X^i_p \in \mathbb{Z}^+, E_{p,j,k} \in \mathbb{Z}^+, T_{p,j,k} \in \mathbb{R}^+,$$

$$N_{j,k} \in \mathbb{B}, \quad \forall i \in I, \forall p \in P, \forall j \in J, \forall k \in k.$$

Las ecuaciones (3.2) y (3.3) acotan la cantidad de producto i a ser fabricado. La primera lo acota con el mínimo componente de sus piezas $p(i)$, es decir, que la máxima cantidad que se puede fabricar de un producto es el mínimo existente de las piezas que lo componen. Por ejemplo, si quiero producir un producto i y se compone de 4 piezas diferentes y tenemos 4, 3, 5 y 2 piezas respectivamente y suponiendo que requiera solo una pieza de cada una. El producto se ve acotado a 2 ya que la última pieza acota el numero de productos enteros que se pueden producir, como lo muestra la Figura 3.1 . La restricción (3.2) acota el producto manufacturado a la demanda del mismo.

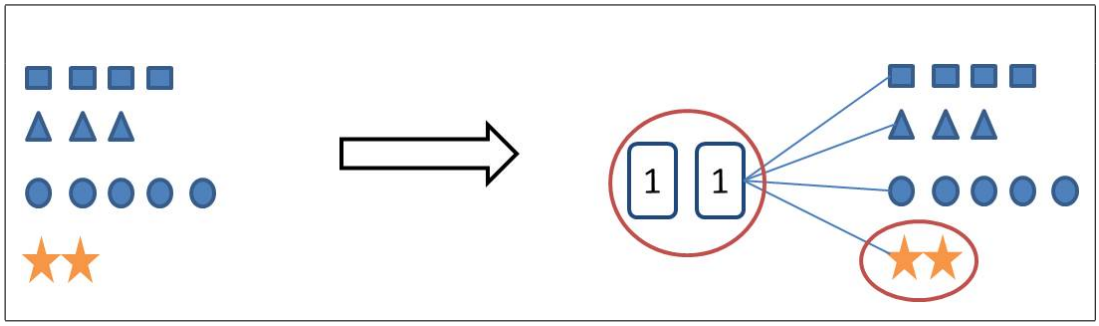


Figura 3.1: Ejemplo restricción (3.2) del PPMM. Si quiero producir un producto i y se compone de 4 piezas cuadro, triangulo, circulo y estrella y tenemos 4, 3, 5 y 2 piezas respectivamente, el producto se ve acotado a 2 ya que la pieza estrella lo acota a 2.

La restricción (3.4) vincula el número de ciclos que usa un molde j en una máquina k para hacer la pieza p, con el tiempo que usa el mismo molde j en la misma máquina k para hacer la pieza p, como se muestra en el ejemplo de la Figura 3.2.

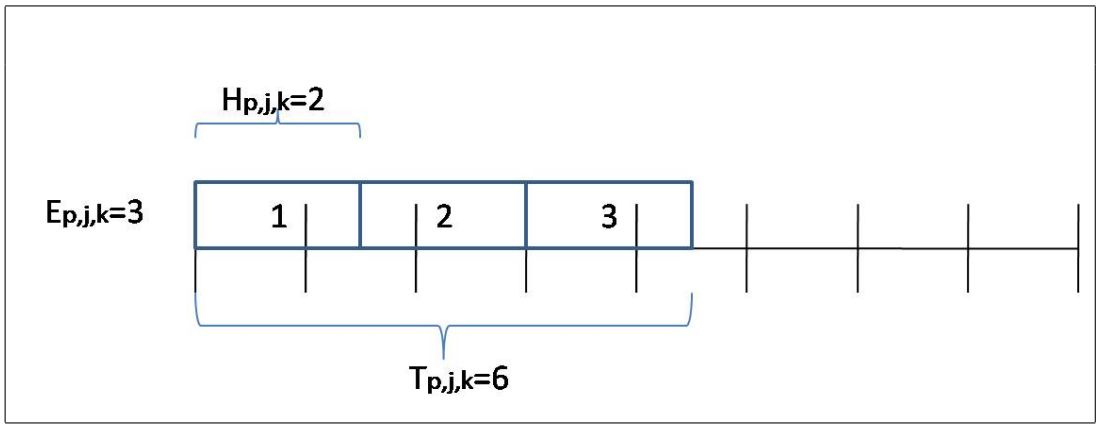


Figura 3.2: Ejemplo restricción (3.4) del PPMM.

La restricción (3.5), asigna valores al vector X de la cantidad de piezas p que se fabricaron para cada producto i, en cada molde j y máquina k. Esto se consigue multiplicando la cantidad de ciclos por las cavidades de esa pieza que tiene ese molde. Se muestra un ejemplo en la Figura 3.3 en donde se va a producir las piezas t en los moldes 1 y 2, que tienen 3 cavidades ambos, se usan durante 3 y 4 ciclos respectivamente y la suma de las piezas t producidas en ambos moldes da un total

de 21 piezas t.

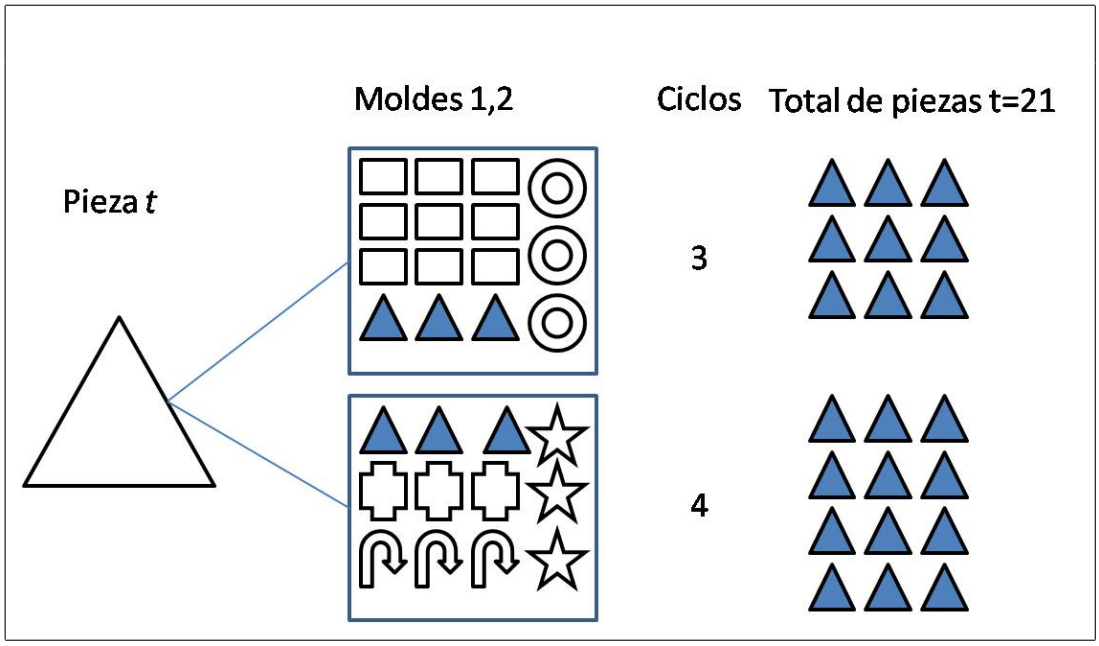


Figura 3.3: Ejemplo restricción (3.5) del PPMM.

La restricción (3.6), se encarga de activar la variable binaria en caso de que un molde j sea instalado en la máquina k dándole valor de 1. En esta restricción usa el valor Tm_k como límite si la suma de tiempo que usa la máquina es mayor a 0 significa que sí se instala.

Por último las restricciones (3.7) y (3.8), limitan que lo que se va a producir no sobrepase el tiempo máquina y molde, respectivamente. Esto se logra en el caso de la primera sumando todos los moldes j que se instalan y el tiempo que usa cada uno de ellos en cada máquina k. De igual manera, la siguiente restricción acota el tiempo molde sumando el tiempo que se usa cada molde j en cada máquina k. En la Figura 3.5, en la parte de superior, la cual hace referencia a la restricción (3.7), se suma el tiempo de instalación del molde 1 y 2 en la máquina 1, más el tiempo de de trabajo de ambos moldes para hacer las piezas 1 y 2. Se muestra que este debe ser menor al tiempo que hay disponible de la máquina 1. En la figura de abajo, se muestra cómo se suma el tiempo que usa el molde 1, más el tiempo de instalación en las máquinas 1 y 2 y esta debe ser menor al tiempo disponible del molde 1 lo cual

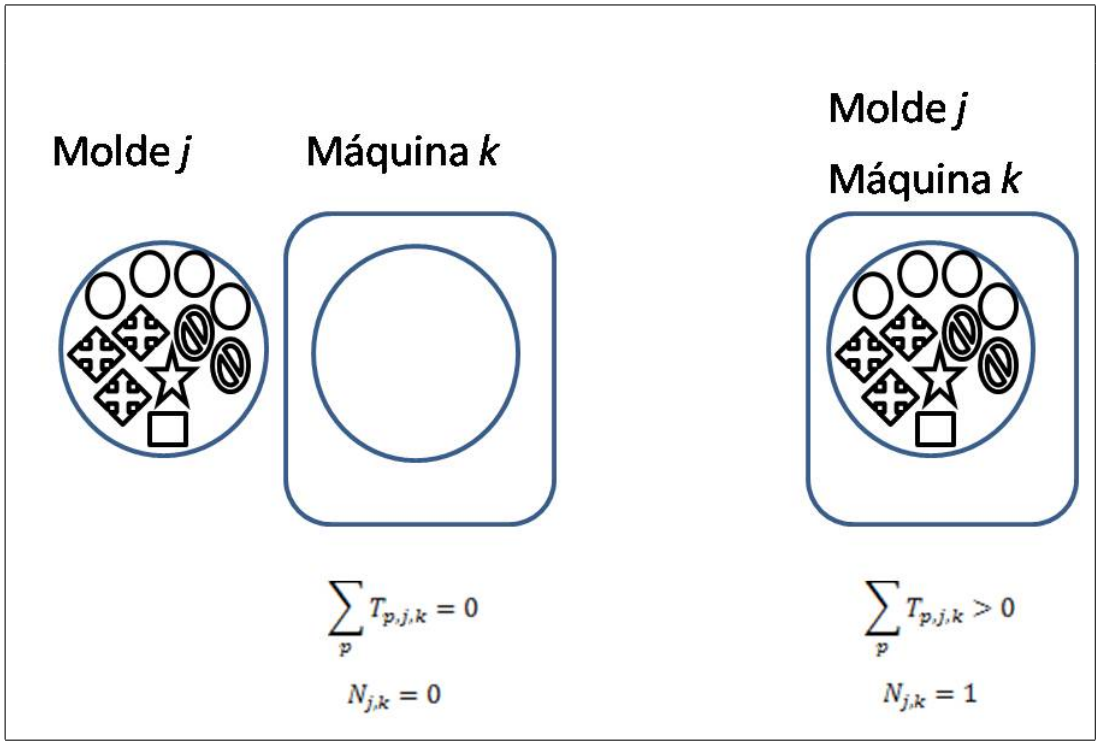


Figura 3.4: Ejemplo restricción (3.6) dle PPMM.

es la representación de la restricción (3.8).

Al formular este modelo, se observa que no es necesario especificar en la variable que representa la cantidad de piezas $X^i_{p,j,k}$ ni de qué molde j o máquina k provienen las piezas p. Por lo tanto la variable queda de la siguiente manera X^i_p. Lo cual simplifica el modelo matemático.

En la restricción (3.2) está presente la función mínimo la cual no es lineal entonces se linealiza el modelo con ayuda de [48] y la restricción queda de la siguiente manera,

$$C^i_p \cdot Pf^i \leq \left[X^i_p + a^i_p\right] \qquad \forall i \in I, p \in p(i). \tag{3.9}$$

Además, se reformula la restricción (3.6), ya que anteriormente usamos una variable continua, tal es el caso de $T_{p,j,k}$ y no es tan buena cota como usar una variable entera tal es el caso de $E_{p,j,k}$, como se muestra en la restricción (3.10),

$$\sum_{p \in P} E_{p,j,k} \leq \frac{Tm_k - It_{j,k}}{\text{mín}_p\, H_{p,j,k}} \cdot N_{j,k} \qquad \forall j \in J, k \in j(k). \tag{3.10}$$

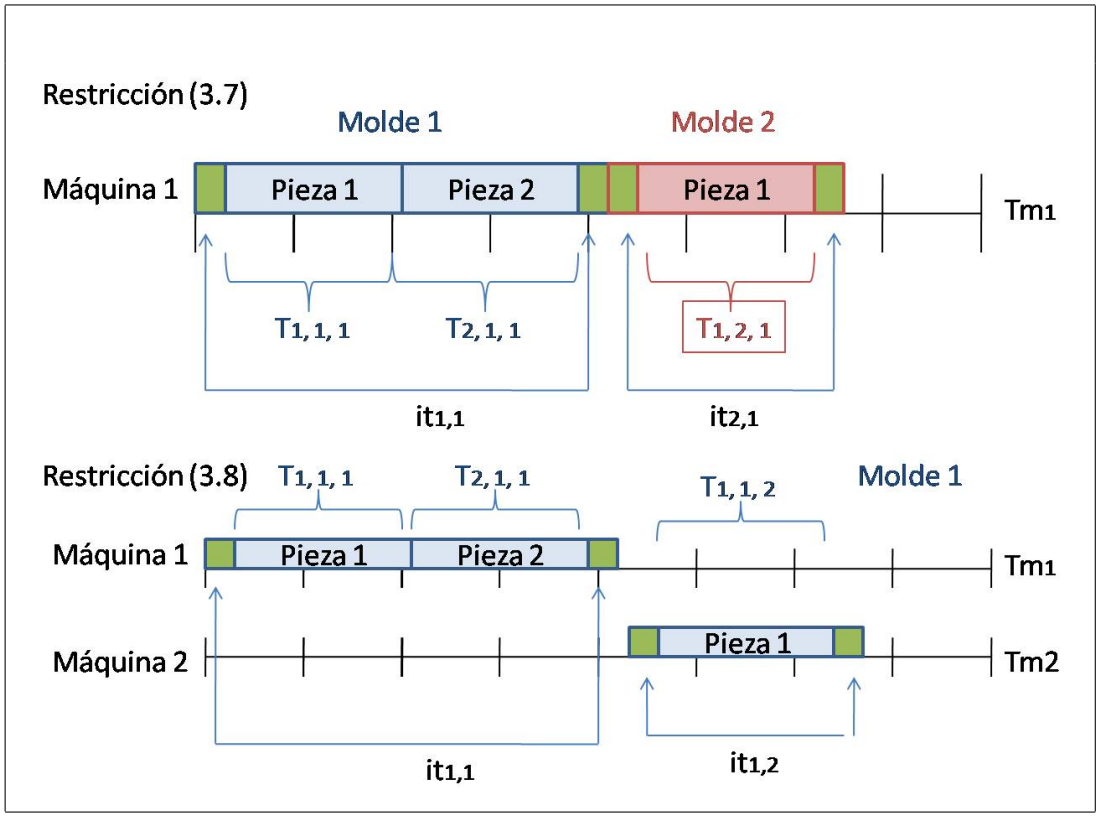


Figura 3.5: Ejemplo restricción (3.7) y (3.8) del PPMM.

Esta nueva restricción tiene dos funciones, la primera activa la variable binaria N y la segunda acota E.

Después de los cambios en el conjunto de restricciones, el modelo queda de la siguiente manera.

$$\begin{aligned}
\text{máx} \sum_{i \in I} V^i \cdot Pf^i & \\
C_p^i \cdot Pf^i \leq \left[X_p^i\right] & \qquad \forall i \in I, p \in p(i) \\
Pf^i \leq D^i & \qquad \forall i \in I \\
E_{p,j,k} = T_{p,j,k} / H_{p,j,k} & \quad \forall p \in P, j \in j(p), k \in k(j) \\
\sum_{i \in I} X_p^i = \sum_{j \in j(p)} \sum_{k \in k(j)} E_{p,j,k} \cdot B_{p,j,k} & \qquad \forall p \in P \\
\sum_{p \in P} E_{p,j,k} \leq \frac{Tm_k - It_{j,k}}{\text{mín}_p \, H_{p,j,k}} \cdot N_{j,k} & \qquad \forall j \in J, k \in j(k)
\end{aligned}$$

$$\sum_{j\in J}\left[N_{j,k}\cdot It_{j,k}+\sum_{p\in P}T_{p,j,k}\right]\leq Tm_k \qquad \forall k\in K$$

$$\sum_{k\in K}\left[N_{j,k}\cdot It_{j,k}+\sum_{p\in P}T_{p,j,k}\right]\leq \max_k Tm_k \qquad \forall j\in J$$

$$Pf^i\in\mathbb{Z}^+, X_p^i\in\mathbb{Z}^+, E_{p,j,k}\in\mathbb{Z}^+, T_{p,j,k}\in\mathbb{R}^+,$$

$$N_{j,k}\in\mathbb{B}, \quad \forall i\in I, \forall p\in P, \forall j\in J, \forall k\in k.$$

Algunas de las mejoras del primero al segundo modelo son las siguientes:

- Se elimina el mínimo de la restricción (3.2).
- Se encuentra una mejor cota en la restricción (3.6).
- Al quitar los subindices j y k del vector X, se reduce el número de variables.

En la Sección 5 se muestra este modelo resuelto mediante un algoritmo de ramificación y acotamiento. No obstante, en la Sección 3.2 se hace una prueba de complejidad del problema PPMM.

3.2 Prueba de complejidad.

Para demostrar que el PPMM es NP-Duro se reduce el problema de múltiple mochila, que definiremos como MKP, a nuestro problema. Como primer paso presentamos el PPMM a su versión decisión, la cual llamaremos dPPMM.

[ENTRADA]: Los conjuntos I, P, J, K y los parámetros V^i, C_p^i, a_p^i, D^i, $H_{p,j,k}$, $B_{p,j,k}$, $It_{j,k}$ y Tm_k para toda $i\in I$, $p\in P$, $j\in J$, $k\in K$, definidos en la formulación del PPMM en la Sección 3.1, así como una constante $G>0$.

[PREGUNTA]: ¿Existe alguna asignación de variables $Pf^i\in\mathbb{Z}^+, X_p^i\in\mathbb{Z}^+$, $E_{p,j,k}\in\mathbb{Z}^+, T_{p,j,k}\in\mathbb{R}^+$, $N_{j,k}\in\{0,1\}$ tal que:

$$F(Pf)=\sum_{i\in I}V^i\cdot Pf^i\geq G \tag{3.11}$$

$$C_p^i \cdot Pf^i \leq \left[X_p^i\right] \qquad \forall i \in I, p \in p(i) \tag{3.12}$$

$$Pf^i \leq D^i \qquad \forall i \in I \tag{3.13}$$

$$E_{p,j,k} = T_{p,j,k}/H_{p,j,k} \quad \forall p \in P, j \in j(p), k \in k(j) \tag{3.14}$$

$$\sum_{i \in I} X_p^i = \sum_{j \in j(p)} \sum_{k \in k(j)} E_{p,j,k} \cdot B_{p,j,k} \qquad \forall p \in P \tag{3.15}$$

$$\sum_{p \in P} E_{p,j,k} \leq \frac{Tm_k - It_{j,k}}{\min_p H_{p,j,k}} \cdot N_{j,k} \qquad \forall j \in J, k \in j(k) \tag{3.16}$$

$$\sum_{j \in J} \left[N_{j,k} \cdot It_{j,k} + \sum_{p \in P} T_{p,j,k} \right] \leq Tm_k \qquad \forall k \in K \tag{3.17}$$

$$\sum_{k \in K} \left[N_{j,k} \cdot It_{j,k} + \sum_{p \in P} T_{p,j,k} \right] \leq \max_k Tm_k \qquad \forall j \in J \tag{3.18}$$

$$Pf^i \in \mathbb{Z}^+, X_p^i \in \mathbb{Z}^+, E_{p,j,k} \in \mathbb{Z}^+, T_{p,j,k} \in \mathbb{R}^+, \tag{3.19}$$

$$N_{j,k} \in \mathbb{B}, \quad \forall i \in I, \forall p \in P, \forall j \in J, \forall k \in k?$$

Para ejemplificar mejor, a continuación se muestra en la Figura 3.6 un diagrama con un ejemplo del PPMM, de lado izquierdo tenemos los productos y estan unidos por aristas a las piezas que los componen. A su vez, las piezas estan unidas por aristas a los moldes con los que son compatibles y estos a las máquinas con los que son compatibles.

Por simplicidad llamaremos el problema de multiple mochila como MKP. A continuación presentamos el MKP en su version desición al cual llamaremos dMKP y es NP-completo.

[ENTRADA]: Sean $(w_1, w_2, ...w_{n'}) \in \mathbb{Z}^{n'}$ los pesos y $(u_1, u_2, ...u_{n'}) \in \mathbb{Z}^+$ el beneficio de n' artículos, $(f_1, f_2, ...fm)$ la capacidad de las m mochilas y sea G' una constante positiva.

[PREGUNTA]: ¿Existe alguna forma de asignar los artículoss a las diferentes mochilas de tal manera a no sobrepasar las capacidades de estas y al mismo tiempo que el beneficio total de los artículoss escogidos sea mayor que G'?

Teorema 3.2.1 *El problema dPPMM es NP-completo.*

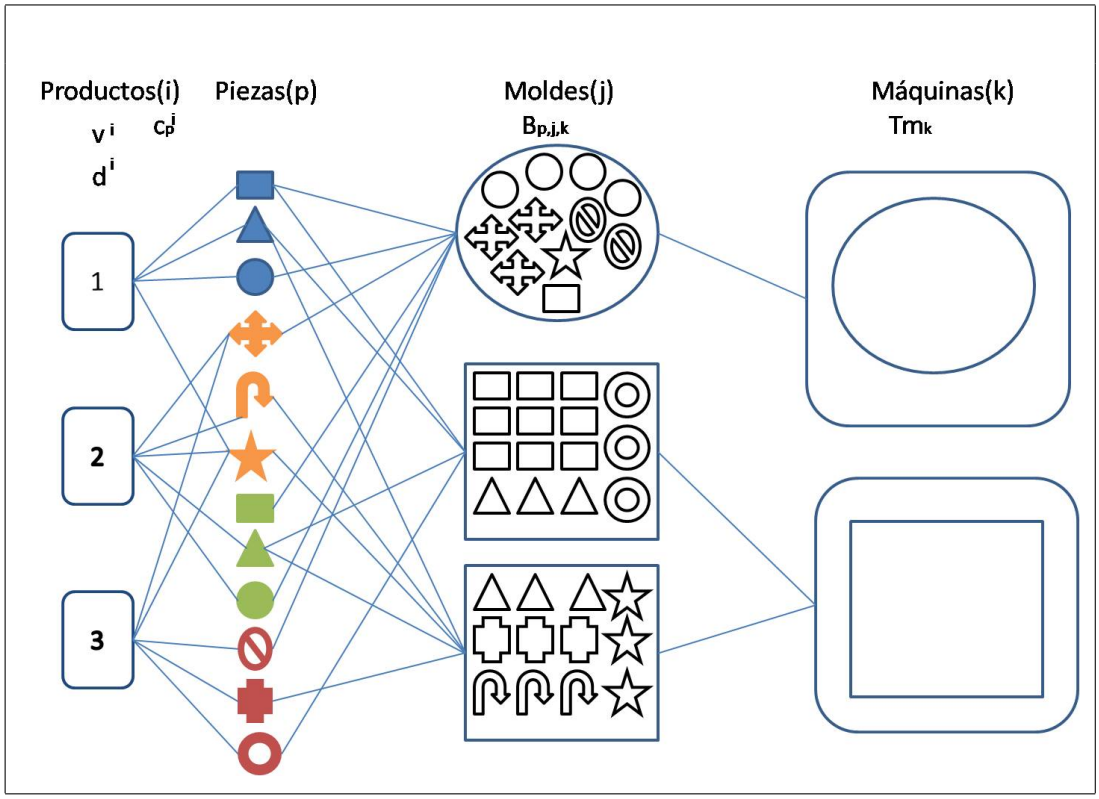


Figura 3.6: Diagrama de ejemplo de PPMM.

Demostración: Primero notamos que dPPMM está en NP, ya que requiere $pjk + jk + ip + i + p + j + k$ número de pasos para verificar que una solución dada es factible, dado que, sería el número de restricciones que hay que verifcar. Ahora, reducimos el problema de múltiples mochilas a nuestro problema, es decir, construimos una reducción polinomial para llegar de una instancia cualquiera del problema decisión de las múltiple mochilas a una instancia particular del problema dPPMM.

Considerando una instancia cualquiera del MKP en su versión decisión construimos la instancia particular del problema dPPMM como sigue.

Definimos tantos productos i y piezas p como artículos n' ($i = p = 1, 2, 3...n'$), así cada producto está conformado por una sola pieza y no se comparten piezas entre productos, es decir, si $i = p$ entonces $C_p^i = 1$, si no $C_p^i = 0$ $\forall i \in I, p \in P$. Esto se puede visualizar en la Figura 3.7.

Ahora, consideramos que hay tantos moldes como piezas, es decir, $|J| = |P|$ y

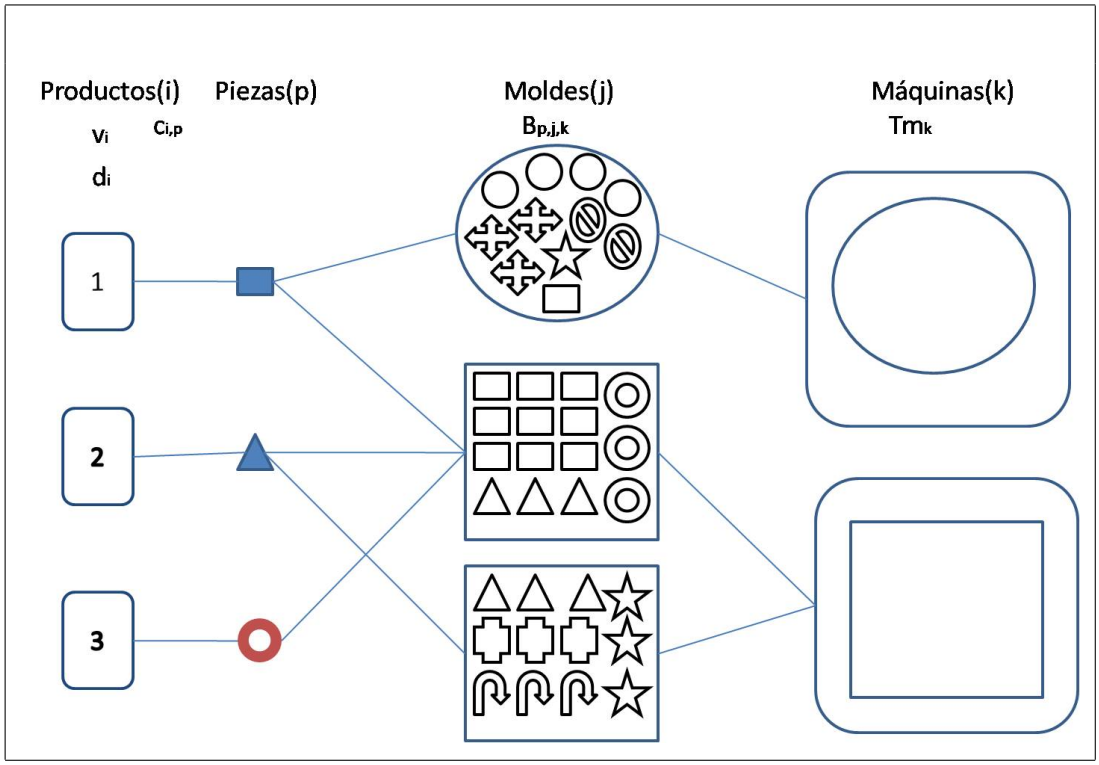


Figura 3.7: Diagrama de ejemplo de PPMM, si $i = p$ entonces $C_p^i = 1$, si no $C_p^i = 0$.

que cada pieza p es compatible con un solo molde j y cada molde j es compatible únicamente con una pieza p. Además, el número de máquinas es igual al número de mochilas de dMKP y que todas las máquinas son compatibles con todos los moldes. Esto se ejemplifica en la Figura 3.10.

Cabe denotar, que $\forall j = p$, $p \in P$, $j \in J$, $k \in K$, $B_{p,j,k} = 1$ y en cualquier otro caso $B_{p,j,k} = 0$. Además, la demanda de cada producto es infinita ($D^i = \infty$ $\forall i \in I$), el tiempo instalación es cero en todos sus casos ($It_{j,k} = 0$ $\forall j \in J$, $k \in K$) y $maxTm_k = \infty$, en el caso de D^i y $maxTm_k$ se les da el valor de infinito para simplificar 2 restricciones del modelo.

Por último, definimos que el valor de cada producto i de dPPMM es igual a el valor de cada artículo i de dMKP ($v_i = u_i$), que el tiempo disponible de cada máquina k de dPPMM es igual a la capacidad de cada mochila en dMKP respectivamente, es decir, $Tm_k = f_k$ $\forall k \in K$.

Debido a que cada producto i tiene una sola pieza p específica y cada pieza p

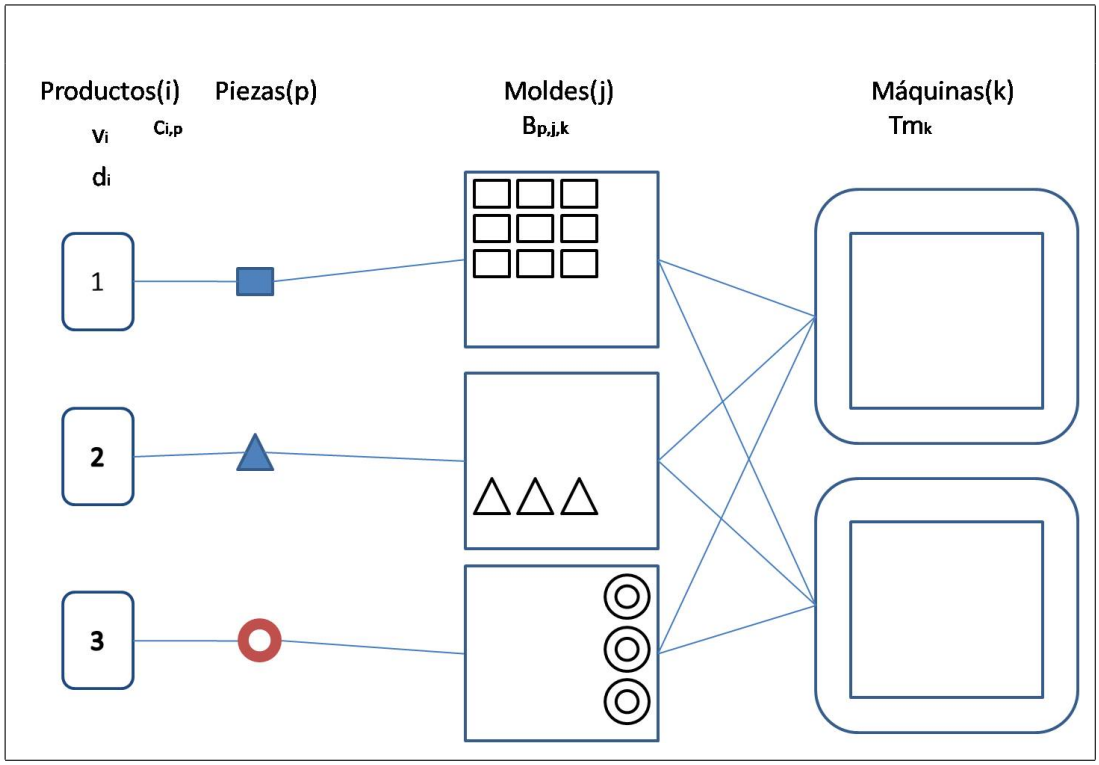


Figura 3.8: Diagrama de ejemplo de una instancia particular de dPPMM, en donde, cada producto esta compuesto por una sola pieza exclusiva para cada producto y además todos los moldes son compatibles con todas las máquinas.

tiene un molde j exclusivo entonces $Pf^i = X_p^i \ \forall i, p$, $X_p^i = \sum_k E_{p,j,k} \ \forall i$ y $E_{p,j,k} = \frac{T_{p,j,k}}{H_{p,j,k}}$ y que $H_{p,j,k} = w_i \ \forall i = p \ \vee \ p = j \ , i \in I, \ p \in P, \ j \in J, \ k \in K$ y en cualquier otro caso $H_{p,j,k} = 0$, dado lo anterior podemos simplificar el vector H de la siguiente manera $H^i = w_i \ \forall i \in I$.

A continuación se muestra un diagrama donde se resume la reducción men-

cionada anteriormente.

$$\text{Parámetros}\begin{cases} V^i = u^i \\ C_p^i \begin{cases} 1 \text{ si } \ i = p \ \forall i \in I,\ p \in P. \\ 0 \text{ en cualquier otro caso.} \end{cases} \\ D^i = \infty \\ H_{p,j,k} \begin{cases} w_i \text{ si } \ i = p = j \ \forall i \in I,\ p \in P,\ j \in J,\ k \in K. \\ 0 \text{ en cualquier otro caso.} \end{cases} \\ B_{p,j,k} \begin{cases} 1 \ \text{ si } \ p = j \ \forall p \in P,\ j \in J,\ k \in K. \\ 0 \text{ en cualquier otro caso.} \end{cases} \\ It_{j,k} = 0 \ \forall j \in J,\ k \in K. \\ Tm_k = f_k \ \forall k \in K. \end{cases}$$

Dado lo anterior también podemos simplificar las variables del dPPMM y quedan de la siguiente manera:

$$\text{Variables}\begin{cases} Pf^i \in \mathbb{Z}^+ \\ X^i \in \mathbb{Z}^+ \\ T_k^i \in \mathbb{R}^+ \\ E_k^i \in \mathbb{Z}^+ \\ N_k^i \in \{0, 1\} \end{cases}$$

Ahora, sustituimos los valores de los parámetros particulares en el modelo dPPMM y simplificando, queda de la siguiente forma:

$$F(Pf) = \sum_i V^i \cdot Pf^i \geq G' \tag{3.20}$$

$$Pf^i = X^i \qquad \forall i \tag{3.21}$$

$$E_k^i \cdot h^i = T_k^i \qquad \forall i, k \tag{3.22}$$

$$X^i \leq \sum_k E_k^i \qquad \forall i \tag{3.23}$$

$$\sum_i T_k^i \leq Tm_k \qquad \forall k \tag{3.24}$$

$$E_k^i \leq Tm_k \cdot N_{i,k} \qquad \forall i, k \tag{3.25}$$

$$Pf^i \in \mathbb{Z}^+, X_p^i \in \mathbb{Z}^+,\ E_p \in \mathbb{Z}^+,\ \ T_p \in \mathbb{R}^+, N_{i,k} \in \{0,1\}\, \forall i,k.$$

En las Figuras 3.9 y 3.10 se ilustra un ejemplo de una instancia MKP y la correspondiente instancia particular de dPPMM', respectivamente.

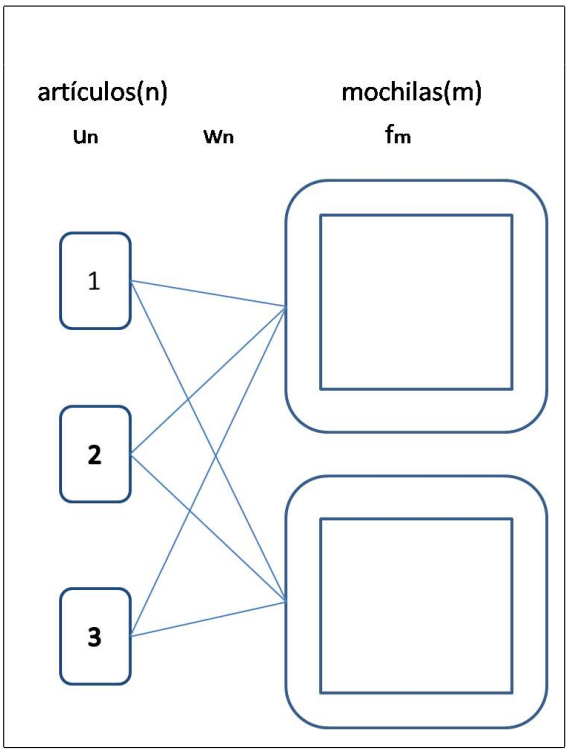


Figura 3.9: Diagrama de MKP, para 2 mochilas con 3 artículos con un beneficio u_n, un peso para cada artículo de w_n y una capacidad de f_p para cada mochila.

El valor de $F(Pf)$ en dPPMM' , está representado por la siguiente expresión:

$$F(Pf) = \sum_i V^i \cdot Pf^i.$$

Si consideramos que los valores de cada producto en dPPMM' son los mismos de cada artículo en MKP, entonces podemos decir que dPPMM' se puede calcular en función de las variables de dMKP y viceversa. Por lo tanto, $F(Pf)$ para la instancia dPPMM' puede verse como sigue.

$$F(Pf) = \sum_i u^i \cdot Pf'^i = F'(Pf').$$

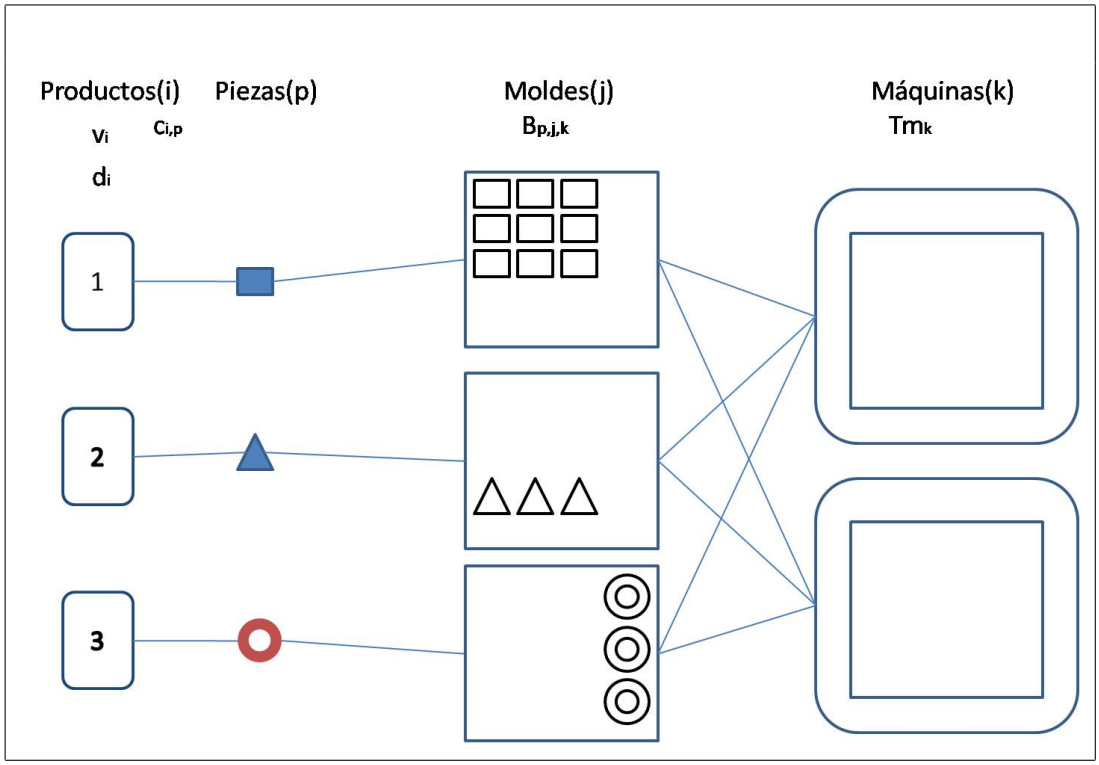


Figura 3.10: Diagrama de una instancia particular de PPMM, para 3 productos, 3 piezas, 3 moldes y 2 máquinas.

Lema 3.2.2 *Los valores de $F(Pf)$ y $F'(Pf')$ son iguales para cualquier solución de $Pf = Pf'$ factibles en dMKP y dPPMM'.*

Ahora demostraremos que dada una solución de $Pf = Pf'$ para ambos problemas, la respuesta de la pregunta de dPPMM' es Sí, si y solo si, la respuesta de dMKP es Sí.

Paso 1: Sea X una solución factible de dPPMM tal que la respuesta es Sí. Entonces,

$$\sum_i T_k^i \leq Tm_k \ \forall k.$$

Haciendo que $Pf^i = Pf'^i$ para toda $i \in I$, se cumple que

$$\sum_i Pf'^i \cdot w_i \leq f_k \ \forall k,$$

lo cual indica que no se excede el peso de la mochila y por el Lema 3.2.2 se tiene que $F(Pf) = F(Pf') \geq G = G'$. Entonces, la respuesta la pregunta de dMKP es Sí.

Paso 2: Sea Pf' una solución de dMKP, tal que la respuesta es Sí. Entonces Pf' cumple con lo siguiente,

$$\sum_i Pf'^i \cdot w_i \leq f_k \ \forall k,$$

haciendo que $Pf^i = Pf'^i$ para toda $i \in I$, se cumple que

$$\sum_{p,i} X_p^i / B_p \leq tm_k \ \forall k,$$

ya que la instancia dPPMM' existen solo asignaciones producto-pieza (i, p) para cada $i = p$ y tomando en cuenta que la demanda es infinita la restricción (3.4) es redundante y la restricción (3.3), solo nos dice que cada producto está compuesto con una pieza, por lo tanto, el número de productos es igual al número de piezas para cada i.

Tomando en cuenta que solo hay un molde y una máquina y $H_{p,j,k} = 1$ para toda p, la ecuación (3.5) iguala la cantidad de ciclos con el tiempo quedando redundante al sustituir la igualdad. De igual manera, al considerar $B_{p,j,k} = 1$ para toda p, la ecuación (3.5) iguala la cantidad de piezas con la cantidad de ciclos y es redundante después de sustituirla.

Al considerar que $It_{j,k} = 0$ y al sustituir, se terminan de cumplir todas las restricciones de dPPMM' y por el Lema 3.2.2 se tiene que $F(Pf) = F(pf') \geq G = G'$. Entonces, la respuesta a la pregunta de dPPMM' es Sí.

De lo anterior, tenemos que dMKP se reduce a dPPMM. Entonces, dPPMM es a su vez NP-Completo. Por lo tanto, surge el siguiente corolario.

Corolario 3.2.3 *El problema de optimización PPMM es NP-Duro.*

Capítulo 4

Diseño del algoritmos heurísticos

En este capítulo se presentan nuevos algoritmos del tipo heurístico para resolver el PPMM. En el Capítulo 5 se demuestra que el PPMM no se puede resolver de manera exacta en un tiempo razonable por lo cual es necesario el uso de heurísticas.

En esta sección, se explica paso a paso el funcionamiento y desarrollo de los algoritmos heurísticos tipo constructivo. Primero recordemos en la Tabla 4.1 los parámetros que son necesarios para desarrollar nuestros heurísticos.

Parámetros	Descripción
V^i	Precio de cada producto i.
C_p^i	Cantidad de piezas p que necesita el producto i.
D^i	Demanda de cada producto i.
$H_{p,j,k}$	Tiempo que tarda un ciclo del molde j en la máquina k para hacer piezas p.
$B_{p,j,k}$	Cantidad de cavidades de la pieza p que tiene el molde j instalado en la máquina k.
$It_{j,k}$	Tiempo de instalación y desinstalación del molde j en la máquina k.
Tm_k	Tiempo disponible de la máquina k.

Tabla 4.1: Tabla de parámetros del problema PPMM.

En experimentos preliminares sobre instancias del PPMM, notamos que por sus características, es difícil llegar a una buena solución factible de manera aleatoria. Por esta razón, escojimos los heurísticos constructivos como método apropiado para la resolución del PPMM.

Es de notarse que para que haya un cambio significativo en la función objetivo, es necesario que se combinen muchos factores: que se hagan toda las piezas del producto y que se tengan las asignaciones pieza-molde y molde-máquina.

Diseñamos 9 algoritmos constructivos con la denominacion HPPMM y se le agrega al final un número que diferencia al algoritmo. Los algoritmos se describen la siguientes secciones.

4.1 Diseño del HPPMM1

Esta heurística busca primero decidir cuántos productos hacer y después en qué moldes y máquinas se harán.

- Paso 1:

  Decidir el producto i con el que se va a trabajar. Esto se hace buscando el producto i que tenga el mayor valor V_i entre su número de piezas como se muestra en la ecuación (4.1). En otras palabras, el producto i que mayor valor aporte dividido por el número de piezas necesarias para que esté completo. En la Figura 4.1 se ejemplifica este Paso.

$$\max_i \frac{V^i}{\sum_p C_p^i} \tag{4.1}$$

  En el ejemplo de la Figura 4.1 se muestran tres artículos respectivamente el primero con un valor de 300, el segundo con un valor de 250 y el tercero con un valor de 240, están conformados por 4, 5 y 5 piezas respectivamente, en este

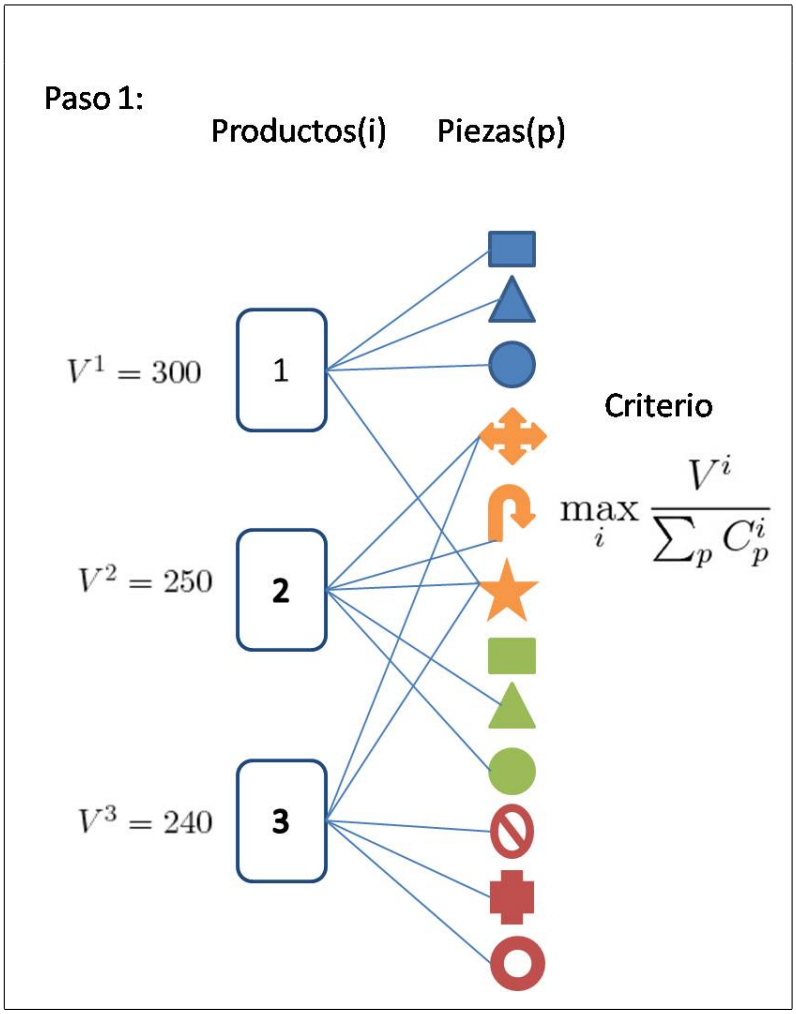


Figura 4.1: Ejemplo del Paso 1 del heurístico constructivo, en el cual se muestra cómo se escoge el producto que se va a fabricar.

ejemplo el heurístico escogería el producto 1, ya que, es el que mejor beneficio por pieza tiene.

- Paso 2:

  Hacer una lista de prioridades para cada pieza contenida en el producto seleccionado. Se busca el mejor molde j y la mejor máquina k dónde se pude producir de cada pieza.

  Para esto se toma en cuenta la cantidad piezas p que necesita el producto i (C_p^i), la cantidad de cavidades ($B_{p,j,k}$), tiempo de ciclo ($H_{p,j,k}$), tiempo molde ($Tmol_j$) y el tiempo máquina (Tm_k). Dicho de otra forma, buscamos dónde se

puede producir la cantidad máxima de cada pieza.

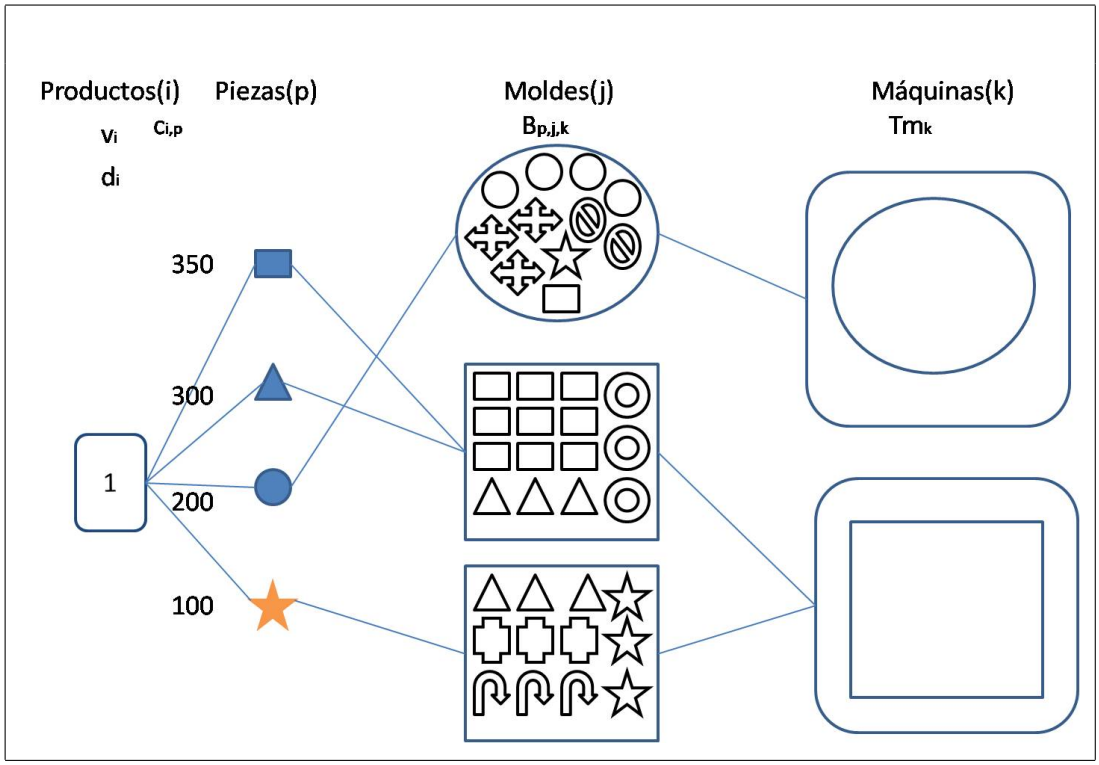


Figura 4.2: Ejemplo del Paso 2 del heurístico constructivo HPPMM1, en donde se ejemplifica cómo se decide la cantidad máxima de piezas que se pueden hacer para cada producto.

En el ejemplo de la Figura 4.2, se muestra que las piezas que componen el producto uno son cuadrado, triángulo, círculo y estrella y el máximo que se pueden hacer son 350, 300, 200 y 100, respectivamente.

- Paso 3:

  Teniendo la cantidad de piezas que obtuvimos del Paso anterior, hacemos una lista en orden de menor a mayor, como se muestra en el Ejemplo de la Figura 4.3.

  Esta lista tiene la finalidad de que a la pieza p que sea más limitante para el producto i, se le asignen recursos primero. De otra forma cabe la posibilidad de que los recursos que pudieran ser asignados a esta pieza sean asignados a otra. Lo anterior afectaría al resultado ya que si fueran asignados a otra

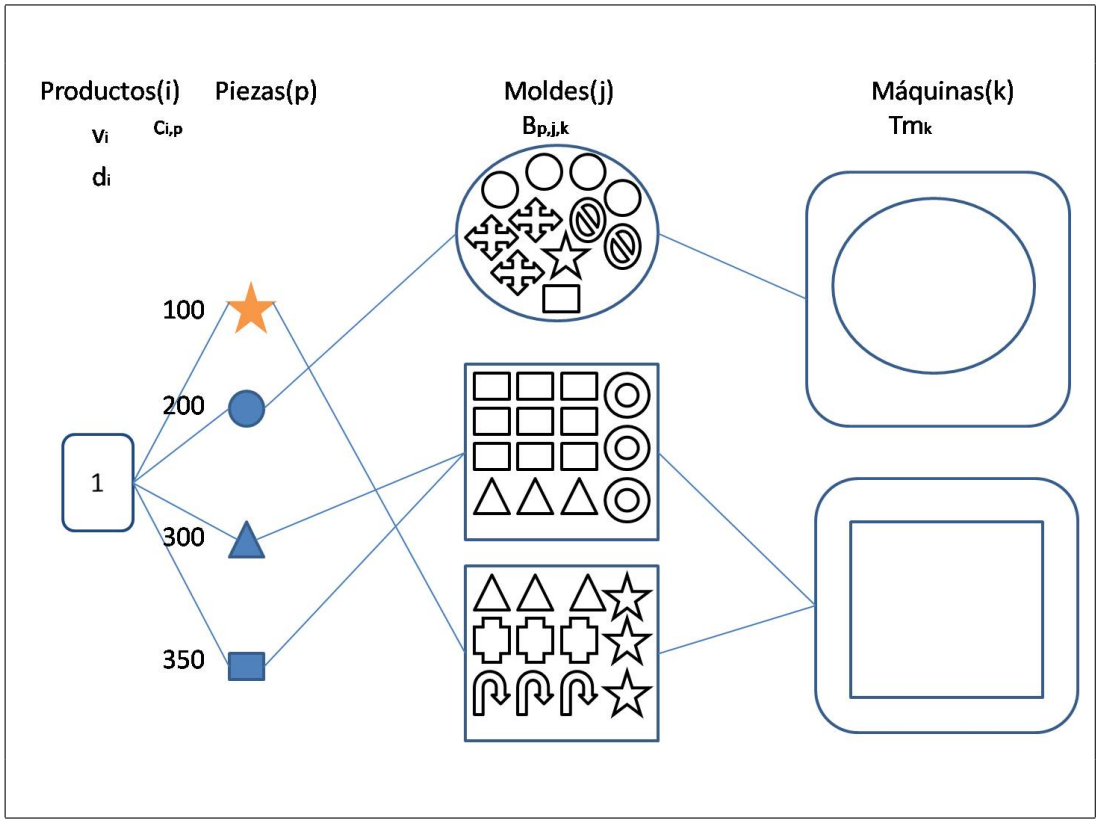


Figura 4.3: Ejemplo del Paso 3 del heurístico constructivo HPPMM1. Se hace una lista de prioridades dando la mayor prioridad a la pieza más limitante.

pieza primero, esta ya no se podría producir al máximo lo cual causaría una reducción de la cantidad de artículos a fabricar del producto i.

- Paso 4:

  Se fija la cantidad de producto a fabricar la cual definimos como cota. Esto se decide en base a la cantidad máxima de producto que se puede fabricar con la cantidad de la pieza limitante que se puede producir.

  Esto debido a que suponemos que una buena cota es la cantidad propuesta en el Paso anterior y también una de las particularidades del PPMM es que no queremos hacer más que la demanda del producto i.

- Paso 5:

  Ahora, se hace un ciclo que se repite tanto como el número de piezas p que tenga el producto i por fabricar, con el orden de la lista que se hizo en el

Paso 3.

Primero se busca el mejor molde y la mejor máquina en que se pueda fabricar la pieza como se muestra en la Figura 4.4. En esta figura se observa cómo se escoge la mejor opción y se tacha la arista no seleccionada.

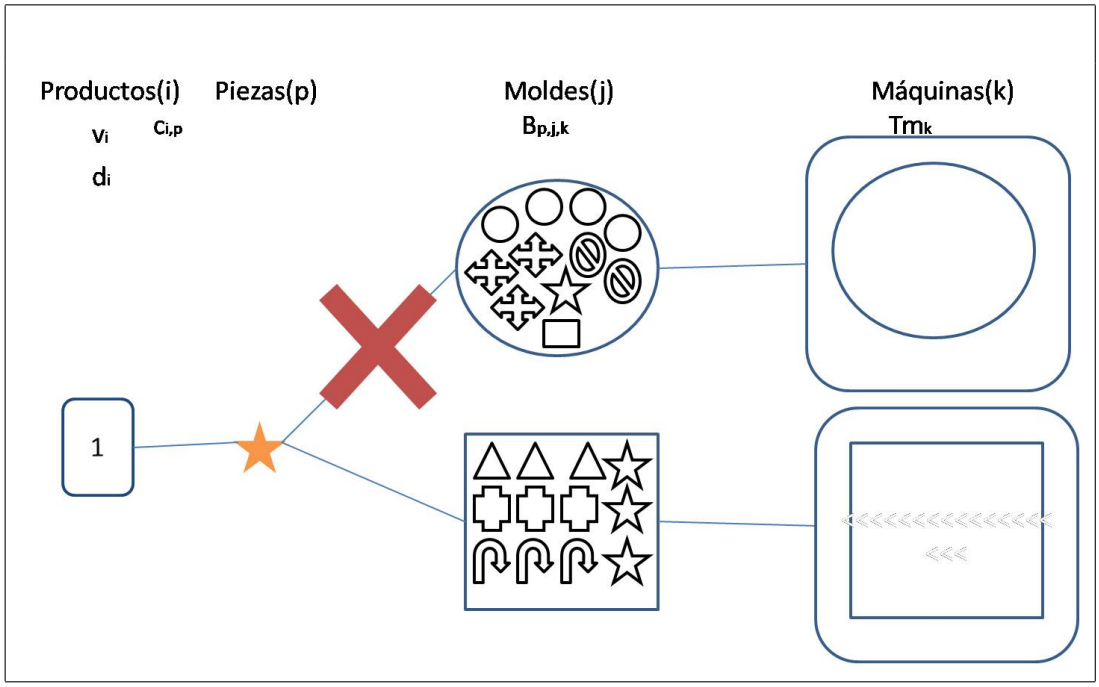


Figura 4.4: Ejemplo del Paso 5 del heurístico constructivo HPPMM1. Se escoge el mejor molde j y máquina k de las combinaciones molde-máquina disponibles, para la pieza p en cuestión.

- Paso 5.1:

  Si las piezas a hacer son suficientes para completar la cantidad de producto acotado anteriormente, se pasa al Paso 5.1a, si no se pasa al Paso 5.1b.

- Paso 5.1a:

  En este Paso se actualiza el tiempo del molde j y la máquina k que se usaron. Además, si el molde no se ha instalado aún, se instala y se regresa al principio del ciclo continuando con la siguiente pieza p en la lista.

- Paso 5.1b:

En el caso que no se complete el número de piezas y la cantidad de piezas a hacerse fuera mayor a 0, entonces se hacen la cantidad de piezas que se pueden hacer en ese molde y esa máquina.

Simultáneamente, esa cantidad de piezas se le resta a la cantidad de piezas que se iban a hacer inicialmente y se actualizan los tiempos de molde y máquina.

Además, se marca si se tuvo que instalar un molde y por último se regresa al principio del ciclo continuando otra vez con la misma pieza.

Si las piezas que se pueden hacer son 0 se pasa al Paso 5.1c.

- Paso 5.1c

  Cuando llegamos a este Paso significa que no se pudo completar la cantidad de producto i que planeamos, así reducimos la cota que obtuvimos en el Paso 4.

  Se reinicializa el ciclo junto con todos los parámetros y se vuelve al inicio solo que esta vez la cota esta reducida.

- Paso 6:

  Después de haber acabado de planear la producción del producto i, se regresa al Paso 1 borrando de la lista al producto i.

  Esto se repite hasta que ya no haya productos en la lista. En la Figura 4.5, se muestra un ejemplo del Paso 6, en este ejemplifica como se selecciona el producto 1 se hacen 2000 productos y se borra de la lista antes de entrar en la siguiente iteración y así sucesivamente.

A continuación, veremos un ejemplo del Paso 5:

Caso 5.1a:

Supongamos que necesitamos hacer 100 piezas estrella y en el mejor molde y máquina que se escogieron, hay capacidad para hacer 120 piezas. Por lo tanto se actualizan los parámetros y se pasa a la siguiente pieza.

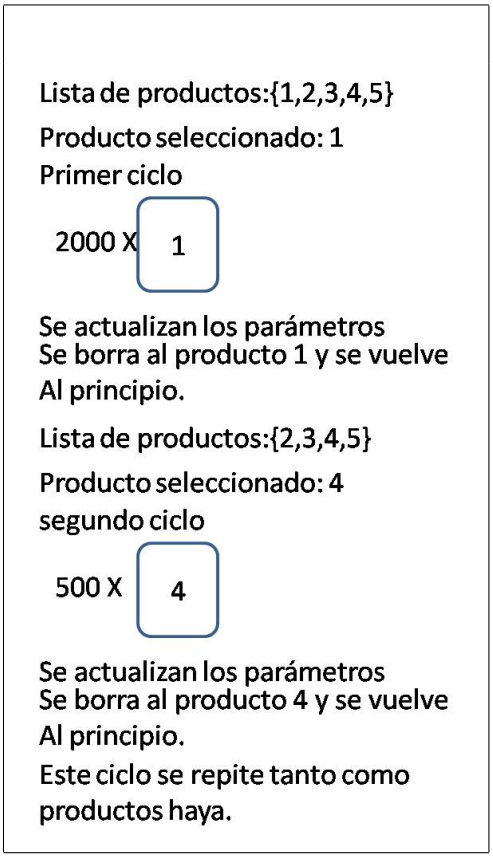


Figura 4.5: Ejemplo del Paso 6 del heurístico constructivo.

Caso 5.1b:

Ahora, se necesitan hacer 100 piezas triángulo y en el mejor molde y máquina hay capacidad para hacer 90 piezas. Se actualizan los parámetros y se regresa al inicio del ciclo pero ahora solo se necesitan hacer 10 piezas triángulo.

Caso 5.1c:

Ya por último, se necesitan hacer 100 piezas rectángulo y en el mejor molde y máquina se pueden hacer solo 90. Por lo tanto, se actualizan la cota y los parámetros.

Ahora, se regresa otra vez al inicio pero solo se necesitan hacer 10 piezas. Pero al querer hacer la pieza en el mejor molde y máquina, lo más que se puede hacer es cero.

Por esto se reinicia el ciclo regresando todos los parámetros y variables a sus valores antes de empezar el ciclo. Pero ahora la cota se ve reducida y en lugar de hacer 100 productos i se hacen solo 90.

En el ejemplo de la Figura 4.6, se muestra en la parte superior como en la primera iteración no se pudo completar la cantidad de productos propuestos en el Paso anterior y en la parte inferior como después de una iteración se busca hacer una cantidad menor de productos y se logra la nueva propuesta. Esto muestra el funcionamiento de los Pasos 5.1 a, 5.1b y 5.1c.

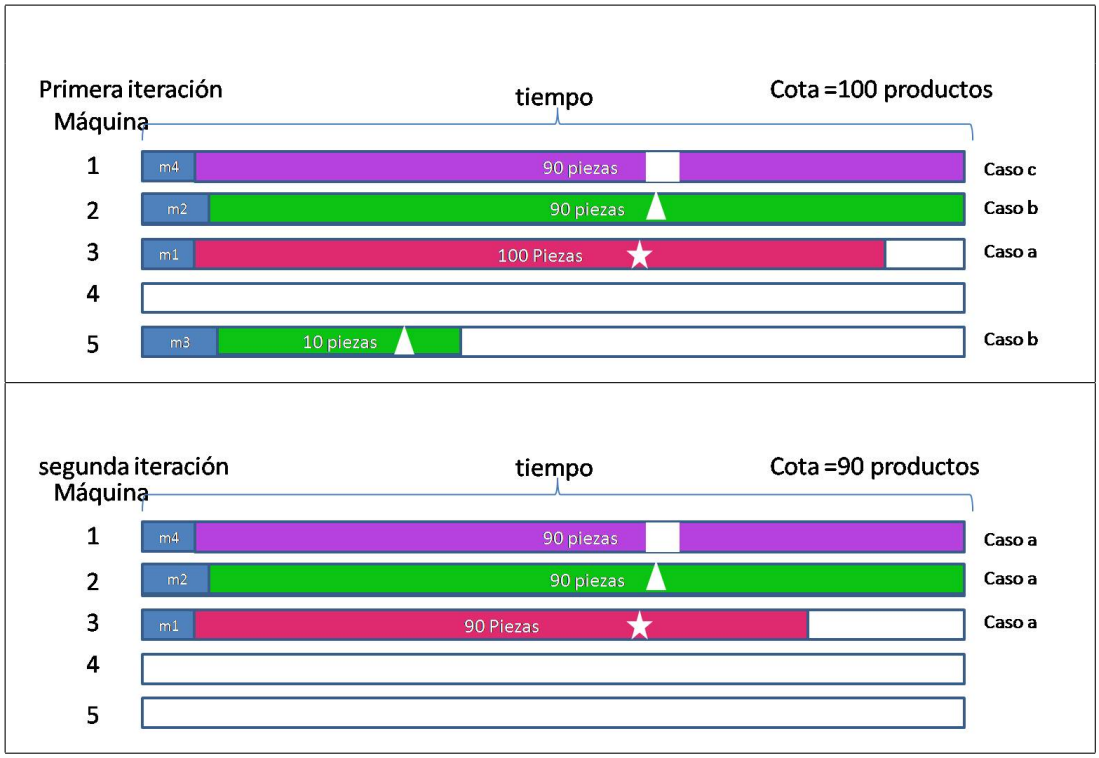


Figura 4.6: Ejemplo del Paso 5 del heurístico constructivo HPPMM1.

4.2 Diseño del HPPMM2

En esta heurística, el cual llamamos HPPMM2, se deriva del algoritmo HPPMM1 y aborda la idea de que la cota obtenida en el Paso 4, es mucho menor a la cantidad de producto i que se pude fabricar. Por lo tanto, se busca la manera de

fabricar más producto i. Se modifica el Paso 6 de la siguiente forma.

Paso 6 modificado:

Después de producir el producto i en el Paso 5, se regresa al Paso 1 y se borra de la lista al producto i solo en el caso de que ya se haya llegado la cantidad de producto i fabricado sea menor a un límite de 100 o la cantidad de producto de la demanda menos lo que ya se produjo sea menor a un límite de 100. Si no, se vuelve al Paso 1 y se vuelve a escoger el producto i tantas veces sea necesario. Esto se repite hasta que ya no haya productos.

En la Figura 4.7 se puede observa un ejemplo del Paso 6 modificado en el cual se muestra cómo el producto i es introducido al ciclo de planeación en más de una ocasión hasta que cumple con la restricción, después de esto pasa al siguiente producto.

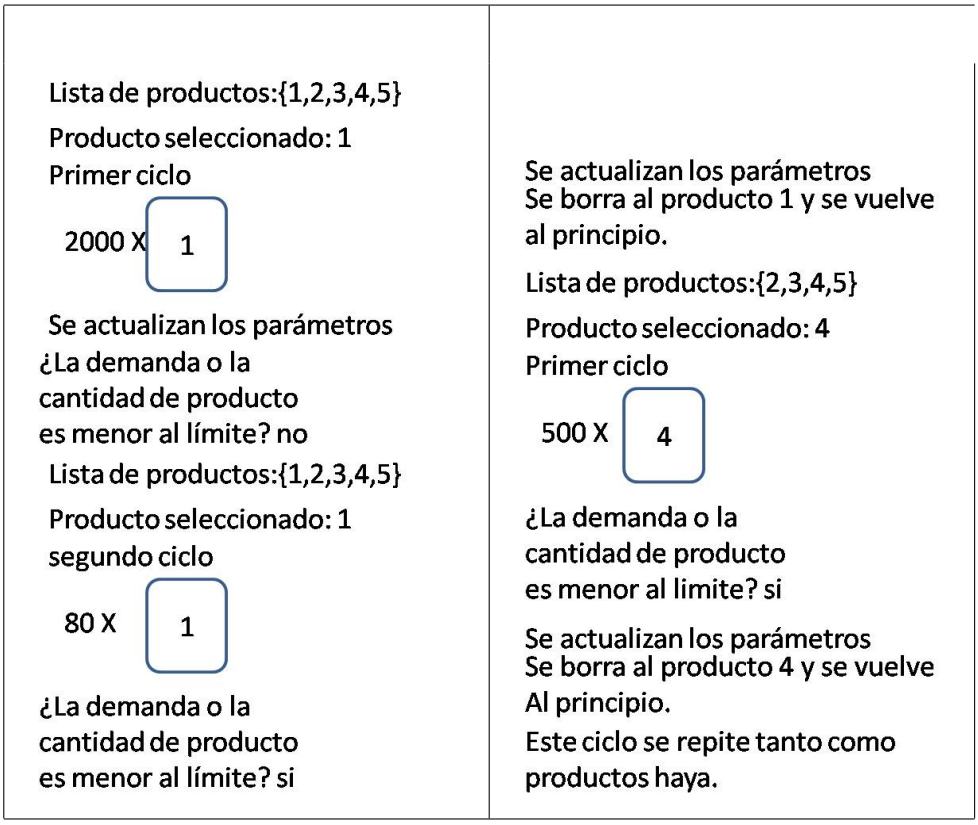


Figura 4.7: Ejemplo del Paso 6 modificado del heurístico constructivo HPPMM2.

En este algoritmo se espera obtener un mejor resultado, sin embargo también se espera que requiera un mayor tiempo de procesamiento por el aumento de ciclos.

4.3 Diseño del HPPMM3

Esta heurística la llamaremos HPPMM3 y esta basado en HPPMM1.

Con una idea similar a HPPMM2, se piensa que la cota puede estar muy por abajo de lo que realmente se puede producir. Sin embargo, a diferencia de la variación anterior, en esta se remarca la idea de que las piezas tienen un orden de fabricación debido a la prioridad de su hechura.

Dicho esto, se trata de fabricar el máximo de cada pieza dependiendo de su prioridad y la manera en que esto se implementa es cambiando el Paso 4. Se iguala la cota a la demanda.

Al igual que en HPPMM2, se busca obtener un mejor resultado. Sin embargo, en esta variación se aumenta el tiempo de procesamiento debido al número de iteraciones en el Paso 5, ya que es mas difícil converger al número de productos i que se pueden fabricar.

En el siguiente ejemplo, se muestra una comparación entre HPPMM1, HPPMM2 y HPPMM3.

Supongamos que el mejor producto a fabricar según el criterio del Paso 1 es el producto 3. En HPPMM1 se acota a 100 productos 3 y después se continúa con el siguiente producto. Asimismo, en HPPMM2 se hacen 100 productos 3, posteriormente se hacen 15 productos 3 y ya en el tercer ciclo se cambia de producto. Mientras que en HPPMM3 en el primer ciclo se acota a 500 productos 3, que es el tamaño de su demanda, pero resulta que no se logra fabricar esa cantidad de producto, por lo tanto se hacen varias iteraciones. Tras esto se consiguieron fabricar 130 productos 3 y en el segundo ciclo se pasa al siguiente producto.

4.4 Diseño del HPPMM4

A esta heurística la definiremos como HPPMM4 y se basa en la estructura del HPPMM3.

De forma similar a HPPMM3, en esta variación la cota superior está dada por la demanda. Sin embargo, al hacer la lista de prioridades en lugar de obtener la cantidad de piezas que se pueden hacer en la mejor combinación de un molde j y una máquina k, se hace un promedio de cuántas piezas se pueden producir en el mejor molde para cada máquina.

En la Figura 4.8, se muestra un ejemplo, de la diferencia entre el criterio en HPPMM1 y HPPMM4 en el Paso 3, como se muestra en la parte superior de la figura, al momento de contabilizar la cantidad de piezas que se pueden fabricar, solamente se toman una máquina y un molde, en este caso la máquina 3. En la parte de abajo de la figura, se muestra como se suman todas las piezas que se puedan hacer en donde haya compatibilidad de piezas con moldes y máquinas.

El objetivo principal de HPPMM4 es tomar en cuenta que una pieza puede ser más compatible que otra. Por lo tanto, aunque haya piezas que tal vez se puedan hacer muy bien en un molde, si son poco compatibles, la cantidad de piezas que se puedan producir será menor que una pieza que sea muy compatible.

Debido a lo anterior, HPPMM4 busca mejorar la lista de prioridades y por lo tanto el resultado final.

4.5 Diseño del HPPMM5

Esta heurística la definiremos como HPPMM5 y su origen es el algoritmo HPPMM3.

HPPMM5, es igual a HPPMM3 con la diferencia de que se hace una actua-

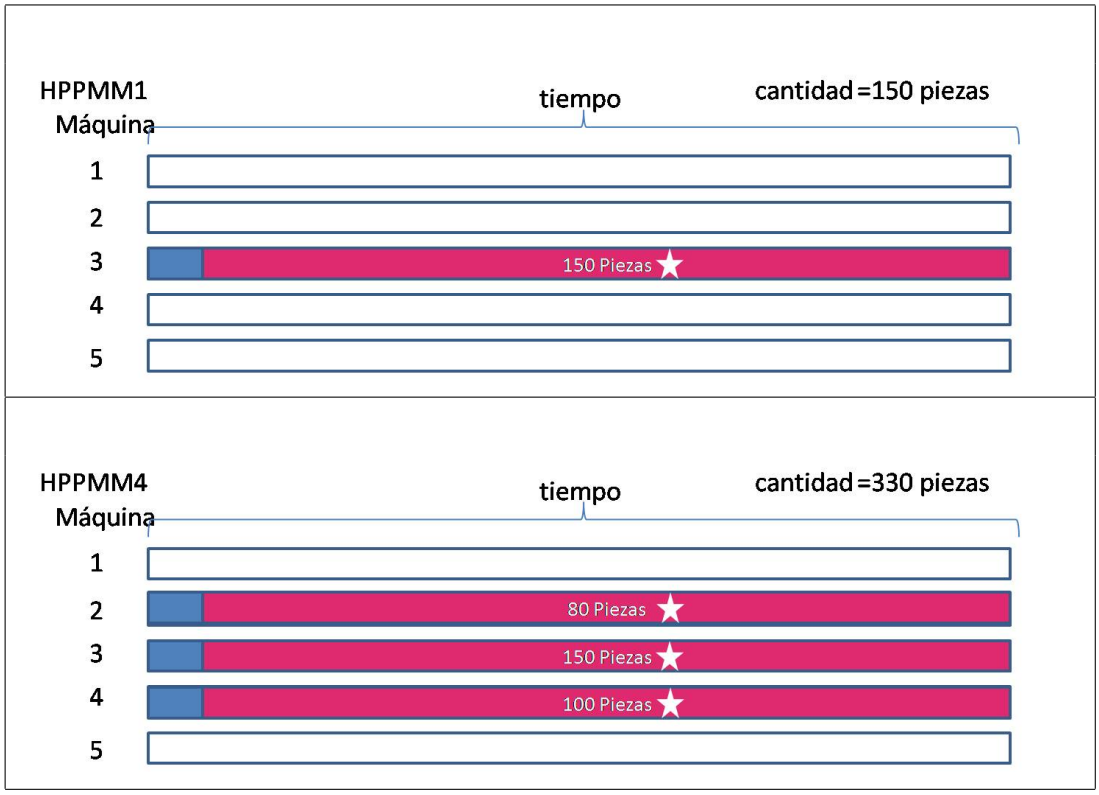


Figura 4.8: Ejemplo de HPPMM4. En esta figura se muestra la diferencia entre cómo se considera la cantidad de producto para la lista de prioridades. En una se considera la mejor combinación molde máquina y en la segunda la suma del mejor molde instalado en cada máquina.

lización a la lista de prioridades cada vez que se produce una pieza y se pasa a la siguiente, en el Paso 5.

Hay que destacar que en HPPMM5 se busca destinar de una mejor manera los recursos a las piezas limitantes. Esto basado en el hecho que al hacer la lista original de prioridades no se había destinado ningún recurso y esta podría cambiar al momento de asignar recursos. En el siguiente ejemplo se expone lo mecionado.

Supongamos que tenemos 3 piezas: fabricar estrellas, triángulos y cuadrados. En la lista de prioridades las cantidades que se pueden fabricar son las siguientes, 100, 200 y 150, respectivamente. En la lista original el orden de asignación de los recursos es el siguiente: estrellas, cuadrados y triángulos.

Supongamos que la mejor máquina para fabricar estrellas es la máquina 1, por lo tanto las 100 estrellas se hacen en la máquina 1. Según la lista, las siguientes piezas a fabricar son los cuadrados. Pero supongamos que la máquina que es la mejor para fabricar triángulos era también la máquina 1 pero esta ya está ocupada por las estrellas, por lo tanto la evaluación anterior donde se menciona que se pueden hacer 200 triángulos es errónea.

Si se asignan los cuadrados primero en la máquina 2, que es la mejor para hacer cuadrados, se hacen los 100 cuadrados sin problemas. Sin embargo, al querer hacer los triángulos, la mejor máquina para hacerlos es la 1 que ya está ocupada y la siguiente es la 2 que también está ocupada. Por lo tanto, se va hasta la máquina 3 pero en ésta solo puede hacer 80 triángulos, por lo tanto no se completa la producción.

Si hubiéramos actualizado la lista después de asignarle los recursos a la estrella, en la máquina 2 se hubieran hecho los triángulos y en la máquina 3 los cuadrados. Por lo tanto, se hubiera completado la producción.

En la Figura 4.9 se muestra el ejemplo anterior.

4.6 Diseño del HPPMM6

A esta heurística la llamaremos HPPMM6 y se deriva del algoritmo HPPMM3.

En HPPMM6, es similar HPPMM3 pero en esta la lista de prioridades del Paso 3 se hace aleatoriamente. Esto dado que se quiere explorar la posibilidad de encontrar mejores soluciones pensando que en HPPMM1 y las variaciones hasta ahora propuestas se puede llegar a un máximo local.

Aunque por lógica, al hacer la lista de prioridades es más probable encontrar un mejor resultado ya que se toma en cuenta la idea de que hay que asignarle primero los recursos a la pieza más limitada del producto.

Sin embargo, hay casos en los cuales la lista no es muy efectiva porque como se

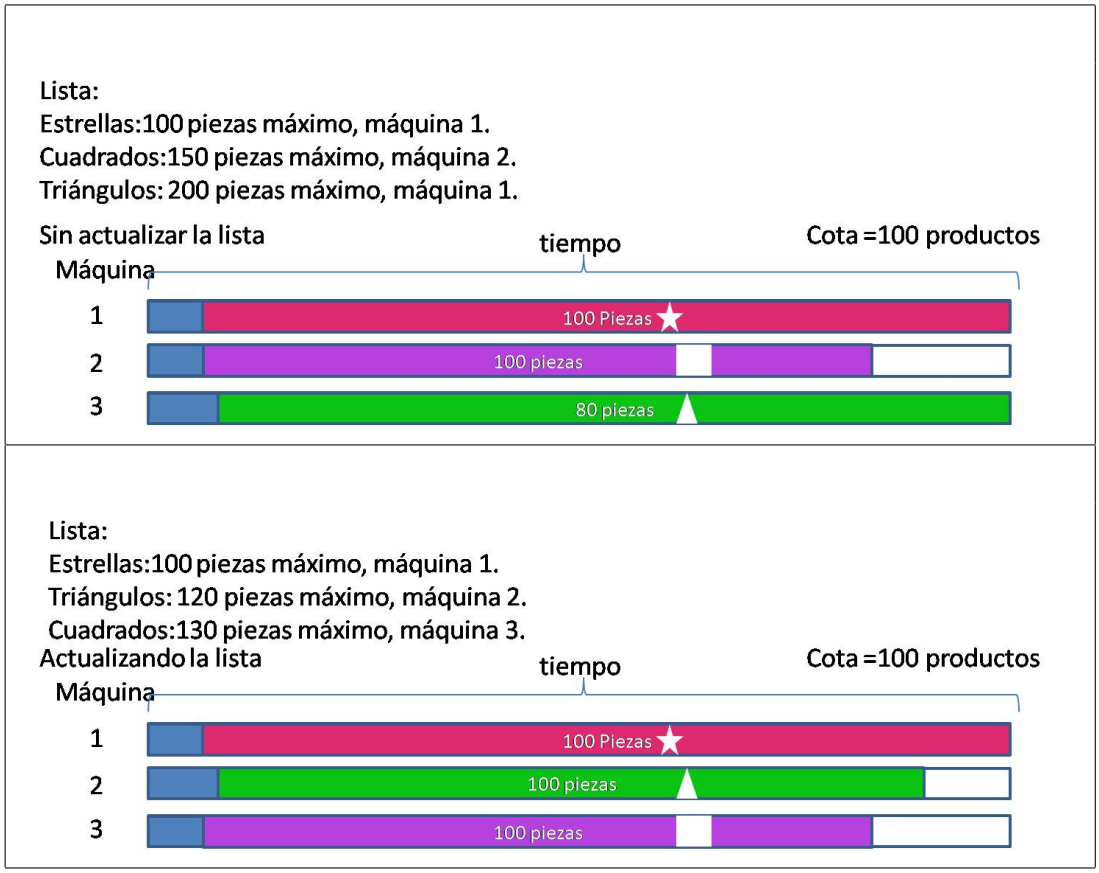


Figura 4.9: Ejemplo de HPPMM5. Aquí se muestra la diferencia entre una lista sin actualizar y cuando se actualiza la lista cada vez que se usan los recursos disponibles.

mencionaba en HPPMM5 al momento de asignarles recursos a las piezas a fabricar, se pierde un poco el sentido ya que alguna pieza se pudiera limitar en su producción ya que como se menciona anteriormente, pudiera ser que se asigne el recurso a otra pieza primero.

Esto se muestra de forma más clara en el siguiente ejemplo:

Consideramos que cada máquina tiene un molde especifico y no se pueden compartir, además los moldes pueden hacer todas las piezas en diferente cantidad, este ejemplo es solo con fin explicativo.

Supongamos que el producto que vamos a fabricar está formado por dos piezas, triángulos y estrellas. Ahora tenemos dos máquinas, en la máquina 1 se pueden hacer 100 triángulos o 150 estrellas, mientras que en la máquina 2 se pueden hacer 60

estrellas o 90 triángulos. Ahora según la lógica de la lista, la pieza limitante es el triángulo.

Debido a esto, se hacen 100 triángulos en la máquina 1 y al momento de querer hacer 100 estrellas en la máquina 2 solo se pueden hacer 60. Por lo tanto, hay que entrar en el proceso iterativo en el cual va a resultar que solo se pueden hacer 60 productos.

Si se hubiera hecho la lista aleatoriamente habría un 50 % de posibilidades de obtener un mejor resultado.

Este caso se visualiza mejor en la Figura 4.10. Sin embargo, en algunos casos

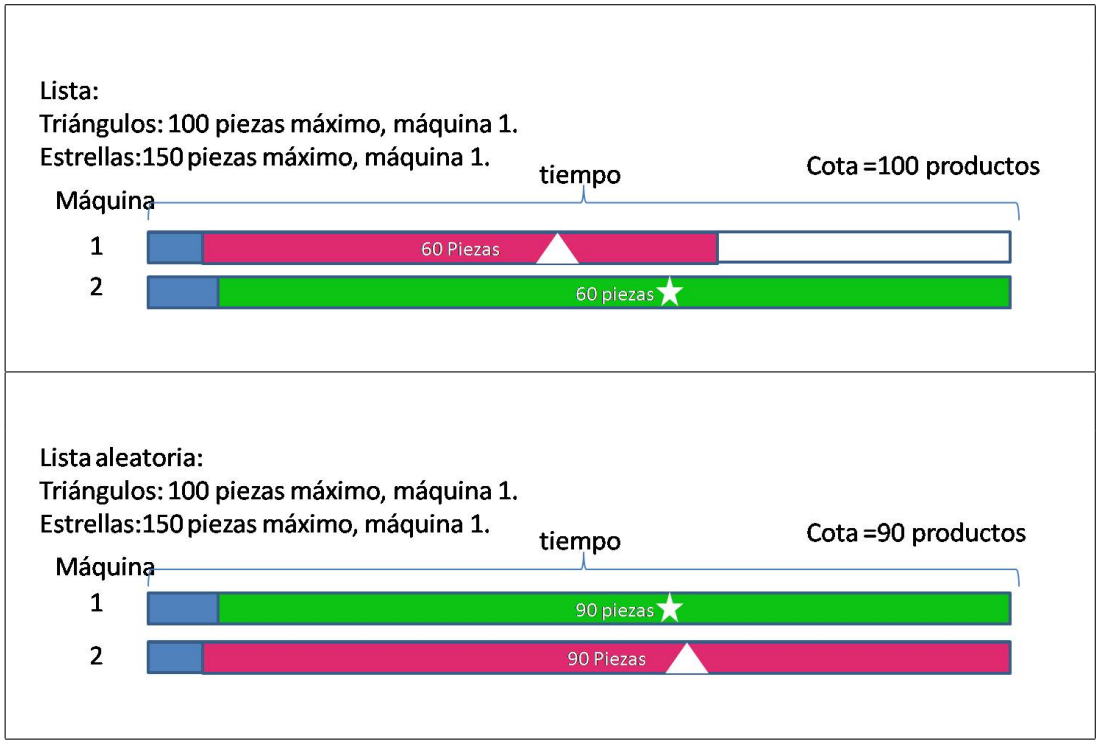


Figura 4.10: Ejemplo de HPPMM6. En esta figura se muestra la diferencia entre usar la lista de prioridades con las piezas y hacer la lista aleatoriamente sin prioridades.

es difícil llegar a una buena solución debido a que no siempre la lista aleatoria da un buen resultado. Además, en algunos casos se complica y tarda más en llegar al resultado debido a que el número de iteraciones para converger a un resultado es mucho mayor.

4.7 Diseño del HPPMM7

A esta heurística la definimos como HPPMM7 y se desarrolla similar a HPPMM3. En HPPMM7, se sigue buscando diversidad y se logra variando aleatoriamente el Paso 5.1c, en la cantidad a reducir. Además, también se varía aleatoriamente las listas de prioridades del Paso 3 como en HPPMM6.

Se busca hacer esto una serie veces (30) a ver si este resultado mejora. Sin embargo, aunque en algunos casos sí mejora en la mayoría tiende a ya no mejorar después de un tiempo y alcanza resultados muy similares a otras variaciones.

Como se menciona en HPPMM6 también en algunos casos puede complicarse y caer en ciclos iterativos bastante largos.

4.8 Diseño del HPPMM8

Esta heurística la llamaremos HPPMM8 y esta basado en la estructura de HPPMM2.

En HPPMM8, después de hacer un revisión de los resultados anteriores, se intenta correr el HPPMM2 para cada producto i.

De los resultados de las corridas anteriores hacemos una lista por orden de importancia en cada producto donde la importancia está dada dependiendo de la ganancia que se obtine con cada producto i.

Ya con esta lista, se busca hacer una última corrida con HPPMM2 dándo prioridad a los productos dependiendo del orden de importancia de la lista. Con esto se busca escoger de una manera más inteligente los productos a fabricar.

En las anteriores variaciones, se pueden dar casos en los cuales las piezas que componen un producto tengan muy poca compatibilidad. Por lo tanto, aunque fuera un buen producto este se vería afectado en su producción, por su compatibilidad.

Por ejemplo, supongamos que tenemos dos productos a fabricar 1 y 2. Según el criterio en HPPMM2, el producto 2 es mejor para fabricar.

Supongamos que 2 tiene baja compatibilidad. En otras palabras, las piezas del producto 2 son más difíciles de hacer que las piezas del producto 1. Por lo tanto las ganancias obtenidas por hacer el producto 2 son bajas.

De esta manera al tratar de hacer el producto 1, resulta que varios recursos para hacerlo ya fueron usados en la fabricación del producto 2. Por lo tanto, el producto 2 se hace en menor cantidad. Debido a lo anterior, la ganancia total es baja.

Supongamos que tomamos en cuenta la cantidad de producto que se puede hacer. Hubiera sido más productivo primero hacer el producto 1.

En la Figura 4.11 se visualiza mejor el ejemplo anterior.

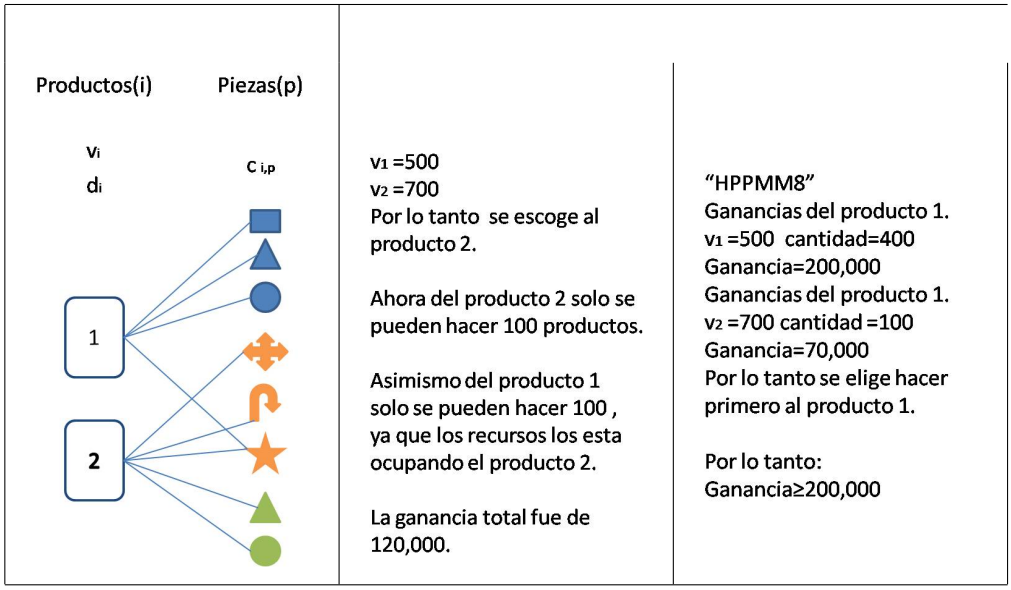


Figura 4.11: Ejemplo de HPPMM8. En este ejemplo se muestra la diferencia entre ordenar con anterioridad los productos y cuando no se hace.

4.9 Diseño del HPPMM9

Esta heurística la definimos como HPPMM9, y esta basado en la estructura de HPPMM2.

Los puntos que se observan que más afectan la calidad de la respuesta y con ello me refiero al resultado de la función objetivo, es al momento de escoger la prioridad del producto. Es decir, la parte del algoritmo que escoge qué producto se va hacer primero.

Aunque por lo general escoge el mejor producto a producir, en ciertos casos como se menciona en la HPPMM8, hay producto que aunque tiene muy buen valor tiene algún cuello de botella en su alguna de sus piezas, es decir, una de las piezas que lo conforma está bastante limitada y esto afecta la cantidad a producir.

Esto podría arreglarse cómo se menciona en la variación anterior. Sin embargo, esto podría influir en el desempeño del algoritmo. Además, aún haciendo esto se puede caer en el caso de que haya una mejor combinación de productos, que mejoren la calidad de la respuesta.

Por lo anterior, se opta por lo siguiente: dejar la parte de escoger el producto igual a HPPMM2 con la ligera variación que un 30 % de las veces que se escoge un producto al azar, dando un poco de variabilidad a la solución.

En la Figura 4.12 se muestra como se modifico el Paso 1. Por ejemplo, supongamos que tenemos el producto 1 con valor de 400, el producto 2 con valor de 350 y cada uno tiene 4 y 5 piezas respectivamente, sin embargo, supongamos que el producto 1 solo podemos fabricar 50 productos y del 2 100, sería mejor opción escoger el producto 2.

Dicho lo anterior es mejor opción escoger el producto 2 ya que este produce una ganancia de $ 3500 mientras que el producto 1 solo $ 2000.

Al haber modificado el Paso 1 se logra que se pueda considerar el producto 2

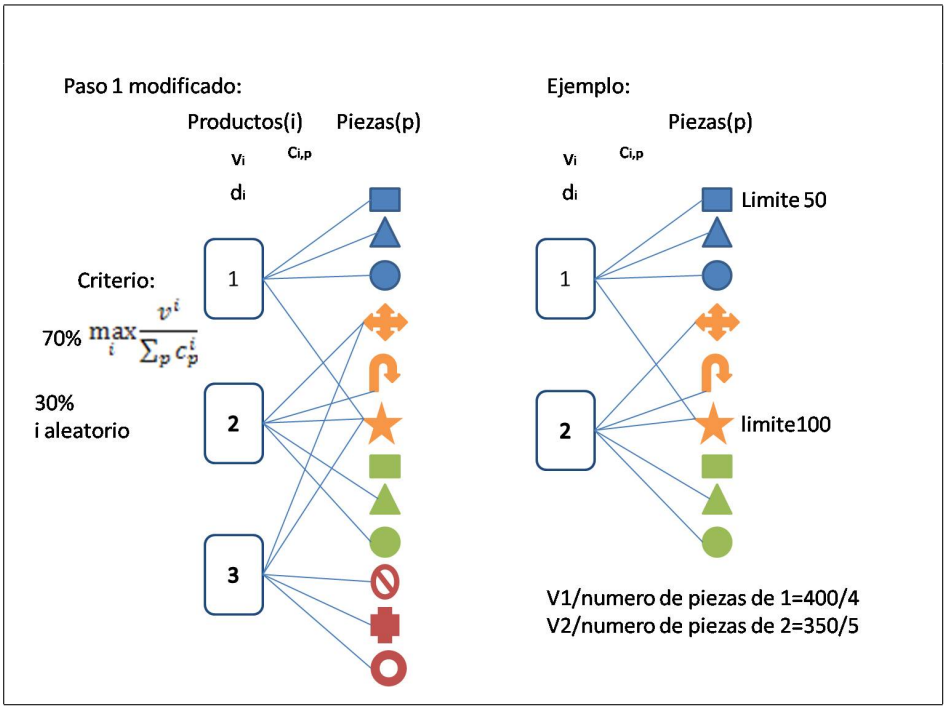


Figura 4.12: Del lado izquierdo se muestra el Paso 1 modificado para HPPMM9, del lado derecho se muestra un ejemplo del Paso 1 modificado para HPPMM9.

y cuando esto sucede ofrece una mejor ganancia.

En la Figura 4.12, se muestra este ejemplo de manera más clara.

También, al comparar las respuestas del heurístico contra las del algoritmo de ramificación y acotamiento de CPLEX, se observa que las respuestas son muy similares entre que moldes y máquinas que son utilizados para producir las piezas.

Esto logra identificar que una ligera variación, al momento de decir qué molde se instala en qué máquina, se puede obtener un mejor resultado. Lo anterior debido a que al agregarle un poco de variabilidad a la instalación de los moldes en las máquinas, se logra que haya un mejor acomodo de los mismos, dando la posibilidad de aumentar el número de piezas a producir.

En esta variación es necesario que no cambie demasiado el algoritmo. Por la tanto se decidió que un 50 % del tiempo se utilice la segunda mejor combinación molde-máquina para hacer la pieza. En la Figura 4.13, se muestra cómo funciona

esta modificación.

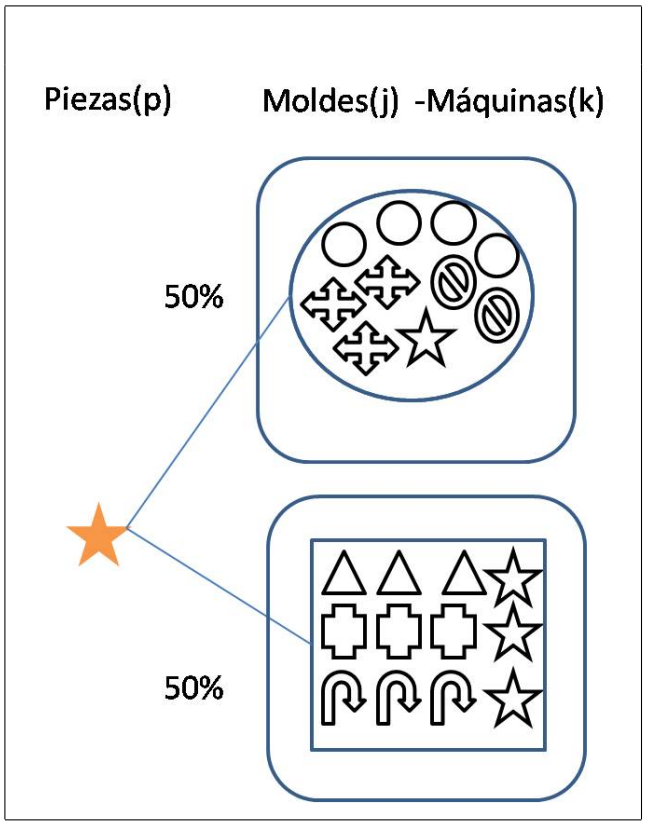


Figura 4.13: En esta figura se muestra como el 50 % del tiempo se usa la mejor combinación molde-máquina y el resto del tiempo la segunda mejor combinación molde-máquina.

Por último, también un problema en general de todas las heurísticas anteriores, es que al momento de hacer la lista de prioridades del Paso 2, no tenemos las seguridad de que esta siga estando en orden de prioridad al momento de actualizar los recursos.

Es decir, qué pasa si una pieza necesita los mismos recursos que otra, al momento de hacer la lista esto no se toma en cuenta y las dos utilizan estos para decidir cuántas piezas se pueden hacer de cada una. Al momento de asignar recursos esta lista puede perder este orden de prioridad.

Sin embargo, esto ya había sido tratado en HPPMM6. Pero, al momento de actualizar la lista tiene dos grandes defectos. Primero, el actualizar contantemente

esta lista reduce mucho el desempeño del algoritmo. Segundo, al actualizar la lista, aunque contrario a lo que se piensa disminuye la calidad de la respuesta debido a que reduce la aleatoriedad del mismo.

Una manera de minimizar este problema es haciendo pequeños cambios en la lista de prioridades. Esto se logra dando un poco más de prioridad a una pieza si esta resulta ser la pieza limitante en el Paso 5.1c.

Por ejemplo, estamos produciendo la pieza 4 que es la cuarta en la lista de prioridad, tenemos que producir 100 piezas y solamente podemos producir 50 piezas, entonces se cambia la prioridad de la pieza 4 y ahora es la tercera en prioridad. En la Figura 4.14 se muestra el ejemplo de una manera más clara.

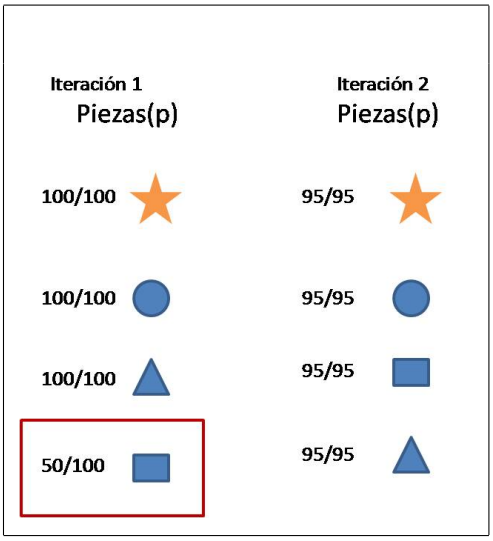


Figura 4.14: En esta figura se muestra como en el Paso 5.1c modificado se reduce la cota y se cambia la prioridad de las piezas.

En el ejemplo anterior, no se nota claramente la utilidad de esta variación, pero cuando se habla de 5000 piezas a producir y solo se pueden producir 100 sí afecta esta modificación. Aunque aumentando un nivel la prioridad en la lista, es difícil que la pieza anterior use los mismo recursos, después de 2 o 3 iteraciones puede darse el caso que llegue a una pieza que al cambiar su lugar sí afecte esto.

Además, debido a la aleatoriedad que se le agrega a HPPMM9, se repite 50

veces o las iteraciones logradas en 1 minuto, con la intención de obtener resultados de mayor calidad.

En el Algoritmo 1 se muestra el pseudocódigo de HPPMM9. La variable Q^i es una variable auxiliar que nos permite contar cuánta demanda se ha cubierto, QP_p^i es otra variable que nos sirve para contabilizar las piezas de p que se han hecho para la iteración *, $\widehat{UB}^i$ nos sirve para guardar la cota superior de productos que se van a fabricar en al iteración *. UB_p se calcula a partir del Paso 2 de HPPMM9 y es la cantidad de piezas máximas que podemos hacer en la mejor combinación molde máquina. L_i es la lista de piezas requeridas por el producto i, Tm_k es el tiempo disponible de la máquina k, $Tmol_j$ es el tiempo disponible del molde j, $N_{j,k}$ es si el molde j está instalado en la máquina k, δ es un parámetro de reducción de la cota superior que ajustamos a 0.99, dado que, con esta reducción tan pequeña se busca en muchas de las soluciones sin dejar grandes espacios sin explorar.

La línea 4 hace referencia al Paso 1 donde se escoje producto solo con la diferencia de que se agrego el factor de aleatoriedad y se escoge un 30 % de las veces un producto al azar. Asimismo, en la línea 9 se escoje la siguiente pieza en la lista y en la línea 12 se escoje j y k dependiendo del cuales le puedan dar mejor cota superior a la pieza. En la línea 13, se actualizan los valores de los parámetros, es decir, se quita el tiempo molde y máquina que se use y si se instala el molde, sino estaba instalado todavía. En la línea 14 se escoje la siguiente pieza en la lista. Igual que en la línea 13, en la línea 17 se actualizan las piezas que se pudieron hacer para completar la cota superior y se actualizan los tiempos e instalaciones. Si no se pudo completar las piezas necesarias para hacer la cantidad propuesta de productos, en la línea 18 se disminuye la cota, se incrementa de rango la pieza p^* en la lista de prioridades y se regresan los parámetros al inicio y por último en la línea 23 se actualiza la demanda con los productos fabricados.

Este algoritmo heurístico de tipo constructivo funciona de manera satisfactoria dando soluciones de excelente calidad en un tiempo razonable. Más adelante en el Capítulo 5 se muestran los resultados de las diferentes pruebas que se hicieron. En

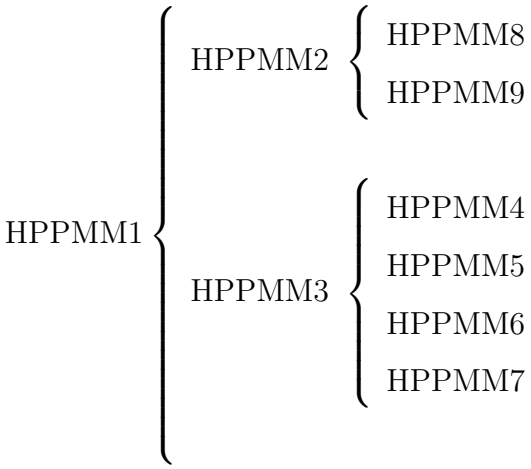
Algorithm 1 heurística constructiva HPPMM9.

$Q^i = D^i \ \forall i$
$QP^i_p = 0 \ \forall i, p \in l_i$
while $I \neq \emptyset$ **do**
  Escoje producto $i^* \in I$
  compute $\widehat{UB}^{i^*} = \text{mín}_{p \in L_{i^*}} UB_p$
  if $\widehat{UB}^{i^*} = 0$ or $Q^i = 0$ **then**
    $I = I - i^*$
  else
    Escoje pieza p^* en L_i
    repeat
      Escoje molde j^* y máquina k^* de tal manera que máx UB_{p^*}
      if $\widehat{UB}^{i^*} - QP^{i^*}_{p^*} \leq UB_{p^*}$ **then**
        Actualiza N_{j^*,k^*}, Tm_{k^*}, $Tmol_{j^*}$, $QP^{i^*}_{p^*}$
        $L_{i^*} = L_{i^*} - p^*$ escoje nueva pieza $p* \in L_{i^*}$
      else
        if $UB_{p^*} \geq 0$ **then**
          Actualiza N_{j^*,k^*}, Tm_{k^*}, $Tmol_{j^*}$, $QP^{i^*}_{p^*}$
        else
          Reduce δ, $\widehat{UB}^{i^*} = \delta\widehat{UB}^{i^*}$, ClearAll($i^*$)
        end if
      end if
    until $L(i^*) = \emptyset$
    $Q^{i^*} = Q^{i^*} - \widehat{UB}^{i^*}$ y $QP^{i^*}_p = 0 \forall p$
  end if
end while

estos resultados, se demuestra que los algoritmos HPPMM8 y HPPMM9 son los que funcionan de mejor manera. Sin embargo, HPPMM9 tiene un mucho menor tiempo de cómputo que HPPMM8.

Cabe mencionar que en algunos casos, HPPMM9 supera la respuesta de obtenida por el algoritmo de ramificación y acotamiento de CPLEX, con el que se prueba el modelo de programación entera mixta de la Sección 3.1.

A continuación, se muestra un diagrama, donde se ven las relaciones entre los diferentes algoritmos propuestos en esta sección.



Capítulo 5

Experimentación

5.1 Diseño de instancias del problema PPMM

En cuanto al diseño de las instancias, se tomaron los datos usados en [19] y en [18] como referencia para crear instancias propias para este problema.

Recordemos los parámetros del PPMM, estos se muestran en la tabla 5.1.

Parámetros	Descripción
V^i	Precio de cada producto i.
C_p^i	Cantidad de piezas p que necesita el producto i.
D^i	Demanda de cada producto i.
$H_{p,j,k}$	Tiempo que tarda un ciclo del molde j en la máquina k para hacer piezas p.
$B_{p,j,k}$	Cantidad de cavidades de la pieza p que tiene el molde j instalado en la máquina k.
$It_{j,k}$	Tiempo de instalación y desinstalación del molde j en la máquina k.
Tm_k	Tiempo disponible de la máquina k.

Tabla 5.1: Tabla de parámetros del problema PPMM.

El primer parámetro en la Tabla 5.1 es V^i, el cual representa el precio de venta del producto i. En las primeras pruebas del generador de instancias este parámetro

afectó fuertemente la dificultad del problema, esto se debe a que se calculaba aleatoriamente el valor de cada producto sin tomar en cuenta de cuantas piezas consta este. En la Figura 5.1 se observa, un ejemplo en el cual se muestra la diferencia entre cómo se calculaba y como se calcula. En la parte izquierda se muestra el producto 1 y 2 que constan de 4 y 2 piezas respectivamente y tienen valores de 200 y 150 respectivamente. Esto nos indica que aunque el producto 2 tiene un valor de venta menor, el tener menos piezas hace que se mas fácil de fabricar y tenga ventaja al momento de seleccionar el producto a fabricar.

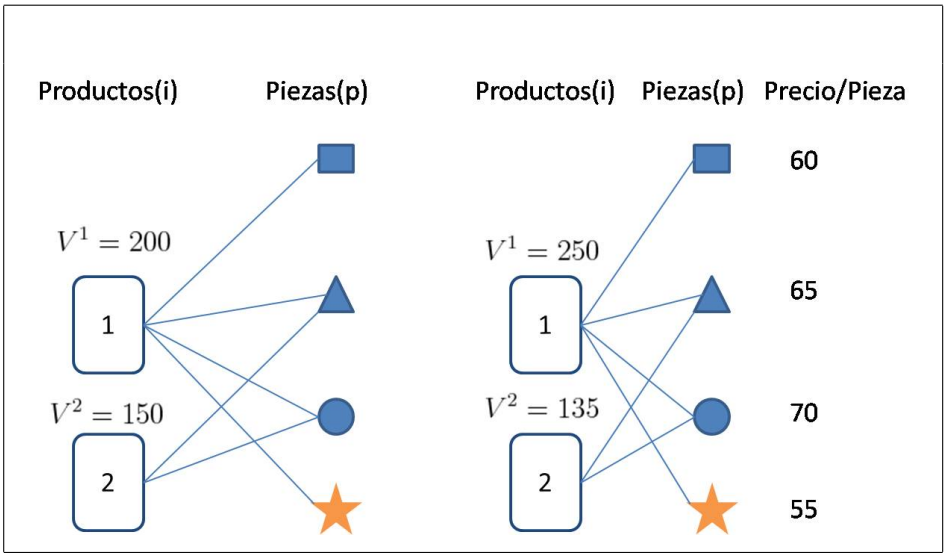


Figura 5.1: Ejemplo del cálculo del precio de un producto en el generador de instancias.

Ahora bien, lo anterior se arregla dando valores directamente a las piezas y el producto adquiere su valor dependiendo de las piezas que lo compongan, esto hace que las instancias sean un poco más complicadas. Pero permite que la instancia se apegue mejor a la realidad, dado que supongamos que tenemos 2 productos con piezas similares y uno tiene un mejor precio se vendería más y si depende de las piezas los precios tienen que ser similares. En el ejemplo de la Figura 5.1 del lado derecho se muestra como se calcula ahora el precio V^i.

Para poder explicar la generación de algunos de los siguientes parámetros hay que definir el termino densidad, en esta investigación se usa para definir la com-

patibilidad. Al momento de generar las instancias hay que decidir, por ejemplo, que piezas lleva cada producto y esto lo hacemos con la densidad, por decir una densidad de 0.25 significa que el producto es compatible con el 25 % de las piezas. La densidad la usamos en tres términos para la compatibilidad producto-piezas, pieza-moldes y molde-máquinas. En la Figura 5.2 se muestra un claro ejemplo donde en la figura de la izquierda se muestra la compatibilidad producto-piezas con densidad de 0.25, de igual manera en el centro pero con densidad de 0.5 y a la derecha con densidad de 1.

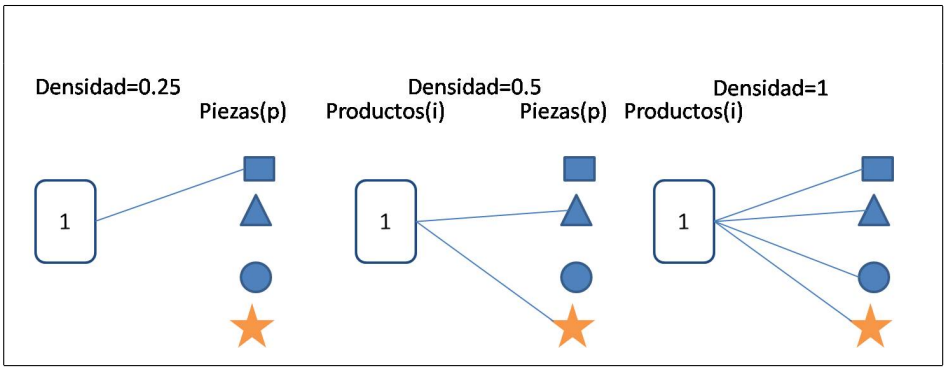


Figura 5.2: Ejemplo de comparacion de densidades producto-piezas.

Ahora bien, después de explicar el termino densidad, para calcular el parámetro C_p^i se usa la densidad producto-piezas, es decir, si el producto i es compatible con la pieza p .C_p^i se distribuye uniformemente entre 1 y 3 piezas que son necesarias para formar el producto, Sino son compatibles es $C_p^i = 0$.

La demanda D^i es otra de las particularidades del problema, ya que casi nunca se logra cumplir con la ella, se calcula uniformente entre 1000 y 18000 productos.

$H_{p,j,k}$ es el tiempo que tarda un ciclo del molde j en la máquina k para hacer la pieza p. Este parámetro se calcula aleatoriamente entre 1 y 2 minutos de manera uniforme.

$B_{p,j,k}$ es el segundo parámetro en el que se necesita utilizar el término de densidad. En este se usa las densidades pieza-moldes y molde-máquinas, es decir, si la pieza p es compatible con el molde j y el molde j es compatible con la máquina

k entonces y solo entonces $B_{p,j,k}$ es mayor a 0. Ya habiendo explicado esto, este parámetro se distribuye uniformente entre 4 y 17 cavidades. Siendo de gran importancia que cuando ya se tiene un valor para la cantidad cavidades de una pieza esta no varía entre las diferentes máquinas solo entre los moldes y que todas las piezas compatibles con el molde j se pueden producir en este molde j y todas las máquinas con el que este sea compatible.

En [19] y en [18], existe un parámetro para el tiempo que se tarda instalar un molde j en una máquina k y otro diferentes para el tiempo de desinstalación. Dado que una de las suposiciones del problema es que todo molde que se instala se debe desinstalar, juntamos todo el tiempo de instalación y desinstalación en un solo parámetro $It_{j,k}$.

El parámetro Tm_k, nos indica el tiempo que hay disponible en cada máquina k y se usa un periodo de 24 horas en todas la máquinas.

En la Tabla 5.2 se resume la distribución de los parámetros para las instancias del PPMM que se usaron para este trabajo.

parámetros	**Distribución**	**Unidades**
V^i	$U[5, 10]$	x pieza
D^i	$U[1000, 18000]$	productos
$It_{j,k}$	$U[90, 138]$	minutos
Tm_k	$U[1440, 1440]$	minutos
C_p^i	$U[1, 3]$	piezas
$B_{p,j,k}$	$U[4, 17]$	cavidades
$H_{p,j,k}$	$U[1, 2]$	minutos

Tabla 5.2: Tabla de distribuciones de parámetros para el diseño de instancias del problema PPMM.

En cuanto el tamaño de las instancias, experimentamos con 3 tamaños diferentes, uno chico que consta de 5 productos, 50 piezas, 30 moldes y 5 máquinas, uno mediano que consta de 10 productos, 120 piezas, 80 moldes y 20 máquinas y uno

grande que tiene 20 productos, 200 piezas, 120 moldes y 25 máquinas.

El efecto de la densidad ya fue abordado en [19] y en [18] para las densidades pieza-moldes y molde-máquinas. Sin embargo, en nuestra investigación se incluye un nuevo nivel de asignación producto-piezas, por lo tanto nos vemos en la necesidad de estudiar la influencia de la densidad producto-piezas. Una parte complicada fue decidir la densidad producto-pieza, pero al momento de usar el generador de instancias se nota que una densidad menor a 0.25 es muy difícil de conseguir por eso se opto por 0.25, mientras que la densidad de 1 es para probar el caso más complicado y la de 0.5 simplemente fue por usar un caso intermedio.

En la Tabla 5.3 se muestran los diferentes tipos de instancias.

	Densidad		
Instancia	$i-p$	$p-j$	$j-k$
5-50-30-5	0.25	0.15	0.60
5-50-30-5	0.5	0.15	0.60
5-50-30-5	1	0.15	0.60
10-120-80-20	0.25	0.15	0.60
10-120-80-20	0.5	0.15	0.60
10-120-80-20	1	0.15	0.60
20-200-120-25	0.25	0.15	0.60
20-200-120-25	0.5	0.15	0.60
20-200-120-25	1	0.15	0.60

Tabla 5.3: Tipos de instancias experimentales para el problema PPMM. Donde $i-p$ representa compatibilidad producto-piezas, $p-j$ pieza-moldes y $j-k$ molde-máquinas.

En la Figura 5.3, del lado izquierdo se muestra un diagrama de compatibilidad de una de las instancias chicas del PPMM con densidades producto-piezas de 0.25, pieza-moldes de 0.15 y molde-máquinas de 0.6 y del lado derecho otro diagrama de compatibilidad con la diferencia de que este tiene densidad producto-piezas de 1. En

la figura se puede observar que al aumentar las densidades aumentan la cantidad de aristas y por lo tanto la complejidad del problema.

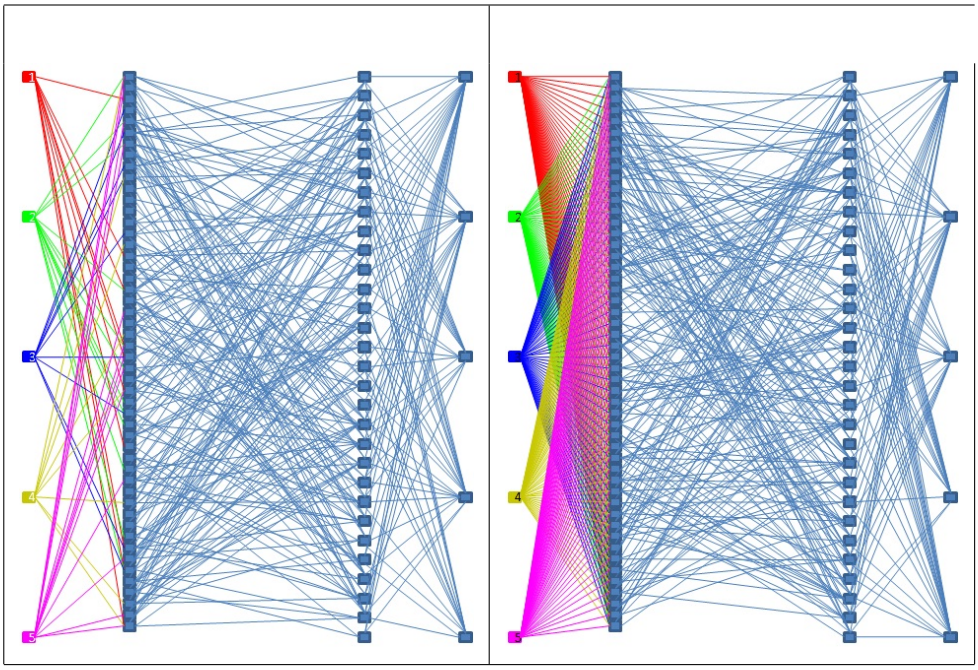


Figura 5.3: Diagramas de compatibilidad del PPMM. del lado izquierdo instancia chica con densidad producto-piezas de 0.25 y de lado derecho instancia chica con desidad producto-piezas de 1

En cuanto a la programación del generador de instancias, se usó EXCEL Visual Basic para aplicaciones y este genera un archivo en código Gams. Una de las necesidades de hacer este generador es que cualquier tamaño de instancia para el PPMM es demasiado grande como para poder escribir el código a mano.

Programando el generador surgieron detalles de suma importancia para hacer que las instancias emularan la realidad. Dado que para decidir las compatibilidades se usa aleatoridad tiene que verificarse los siguientes puntos:

- Cada producto debe estar compuesto por al menos una pieza.
- Cada pieza debe de poder ser usada en al menos un producto.
- Cada pieza debe de poder se fabricada en al menos un molde.

- Cada molde debe de tener cavidades para al menos una pieza.
- Cada molde debe ser compatible con al menos una máquina.
- Cada máquina debe ser compatible con al menos un molde.
- Las cavidades de un molde no dependen de la máquina en la que se instalen.
- La compatibilidad de un molde con una máquina no depende de las piezas.

Las complicaciones de lo anterior, vienen cuando se tienen bajas densidades y esto es debido a que al tener bajas densidades hay menos conexiones y lo anterior no se cumple. Debido a esto al momento de diseñar las instancias había que estar iterando varias veces para que se pudiera cumplir las condiciones.

Para llevar a cabo el experimento se generaron 10 instancias de cada tipo.

5.2 Resultados experimentales del algoritmo de ramificación y acotamiento para las instancias del PPMM

Para resolver las instancias generadas en el procedimiento de la Sección 5.1, usamos GAMS/CPLEX versión 11.2, en una computadora Sun Fire V440 con 4 procesadores ultra sparc III a 1.064 GHZ y 8 Gb de RAM.

Los criterios de paro para el algoritmo de ramificación y acotamiento fueron los siguientes:

- $100,000,000$ de iteraciones.
- optimalidad de $1X10^{-8}$.
- 3600 segundos.

Los resultados de este experimento están presentados en las Tablas de 5.4 a 5.12. En la primera columna tenemos el nombre de la instancia el cual está compuesto por densidad # numero de instancia-tamaño de la instancia, en la segunda columna tenemos el valor de Gap en porcentaje el cual es calculado por Cplex y mide la diferencia de la mejor solución factible contra la mejor posible solución entre la mejor solución factible. En cuanto a la tercera columna, contiene el tiempo que demora cada instancia en ser resuelta por Cplex. En la cuarta columna tenemos la mejor solución factible encontrada por Cplex y en la última columna se encuentra la mejor solución relajada encontrada por Cplex.

En la parte inferior de la tabla tenemos el promedio de Gap de las 10 instancias, el promedio de tiempo, el promedio de nodos explorados por el algoritmo de ramificación y acotamiento de Cplex en las 10 instancias y por último el promedio de iteraciones realizadas para resolver cada instancia de las 10 instancias.

Nombre	**Gap %**	**Tiempo**	**Solución Factible**	**Cota Superior**
0.25#1-5-50-30-5	0.00	425.34	707408.00	707408.00
0.25#2-5-50-30-5	0.00	143.49	618293.00	618293.00
0.25#3-5-50-30-5	0.00	63.45	633424.00	633424.00
0.25#4-5-50-30-5	0.00	114.93	643815.00	643815.00
0.25#5-5-50-30-5	0.00	2449.07	498848.00	498848.00
0.25#6-5-50-30-5	0.00	21.74	615524.00	615524.00
0.25#7-5-50-30-5	0.00	3496.49	613046.00	613046.00
0.25#8-5-50-30-5	0.00	29.01	692228.00	692228.00
0.25#9-5-50-30-5	0.00	712.13	640640.00	640640.00
0.25#10-5-50-30-5	0.00	411.17	601328.00	601328.00
			Nodos	**Iteraciones**
Promedio	0.00	786.68	125672.40	1963406.20

Tabla 5.4: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.25 y tamaño 5-50-30-5.

En las Tablas 5.4, 5.5 y 5.6, se muestran los resultados del algoritmo de ram-

Nombre	Gap %	tiempo(sec)	Solución Factible	Cota Superior
0.25#1-10-120-80-20	11.47	3637.27	3098561.00	3453836.70
0.25#2-10-120-80-20	6.76	3635.54	3269236.00	3490307.99
0.25#3-10-120-80-20	8.13	3635.97	3063600.00	3312764.07
0.25#4-10-120-80-20	11.50	3633.82	3121837.00	3480694.30
0.25#5-10-120-80-20	8.79	3639.16	3100800.00	3373475.98
0.25#6-10-120-80-20	8.82	3637.00	3219320.00	3503337.22
0.25#7-10-120-80-20	7.78	3636.23	3233005.00	3484409.73
0.25#8-10-120-80-20	11.40	3636.54	3186881.00	3550260.03
0.25#9-10-120-80-20	9.41	3635.85	2991456.00	3272820.95
0.25#10-10-120-80-20	7.91	3638.29	3186928.00	3438985.76
			Nodos	**Iteraciones**
Promedio	9.20	3636.57	7134.70	157052.80

Tabla 5.5: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.25 y tamaño 10-120-80-20.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
0.25#1-20-200-120-25	27.51	3703.57	3387732.00	4319756.34
0.25#2-20-200-120-25	13.45	3701.13	3935931.00	4465362.69
0.25#3-20-200-120-25	17.93	3701.82	3686944.00	4348140.66
0.25#4-20-200-120-25	14.43	3703.05	3858880.00	4415765.95
0.25#5-20-200-120-25	37.78	3702.13	3141089.00	4327812.68
0.25#6-20-200-120-25	22.68	3702.30	3557148.00	4363785.37
0.25#7-20-200-120-25	24.23	3703.64	3463852.00	4303161.83
0.25#8-20-200-120-25	17.61	3706.80	3786750.00	4453509.78
0.25#9-20-200-120-25	13.88	3706.31	3855416.00	4390508.77
0.25#10-20-200-120-25	12.35	3702.63	4028694.00	4526372.27
			Nodos	**Iteraciones**
Promedio	20.19	3703.34	660.30	30090.90

Tabla 5.6: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.25 y tamaño 20-200-120-25.

ificación y acotamiento para las instancias de densidad 0.25 y de tamaño chico, mediano y grandes, respectivamente. Ahora bien, en la Tabla 5.4 se muestra que todas las instancias fueron resueltas hasta la optimalidad, en un tiempo promedio de 786.68 segundos. Sin embargo, en las Tablas 5.5 y 5.6, se obtuvieron resultados de 9.2 % de GAP y 20.19 % de GAP en promedio. Se puede observar que al aumentar el tamaño de la instancia es más complicado resolver cada relajación.

Nombre	**Gap %**	**Tiempo**	**Solución Factible**	**Cota Superior**
0.5#1-5-50-30-5	0.00	602.33	730320.00	730320.00
0.5#2-5-50-30-5	0.00	682.02	542724.00	542724.00
0.5#3-5-50-30-5	1.16	3616.91	569856.00	576457.30
0.5#4-5-50-30-5	0.06	3615.37	635250.00	635600.00
0.5#5-5-50-30-5	0.08	3621.00	589050.00	589512.00
0.5#6-5-50-30-5	0.00	177.82	578565.00	578565.00
0.5#7-5-50-30-5	0.00	105.97	633080.00	633080.00
0.5#8-5-50-30-5	0.00	3426.17	538704.00	538704.00
0.5#9-5-50-30-5	1.53	3621.37	560896.00	569467.28
0.5#10-5-50-30-5	1.45	3619.61	562968.00	571153.07
			Nodos	**Iteraciones**
Promedio	0.43	2308.86	364112.10	7992460.90

Tabla 5.7: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.5 y tamaño 5-50-30-5.

En las Tablas 5.7, 5.8 y 5.9, se muestran los resultados del algoritmo de ramificación y acotamiento para las instancias de densidad 0.5 y de tamaño chico, mediano y grandes, respectivamente. En La Tabla 5.7 se puede observar que la mitad de los resultados se resuelven hasta la optimalidad y la otra mitad llega a resultados menores al 2 % de GAP, obteniendo un promedio de 0.43 %. Ahora bien, las Tablas 5.8 y 5.9, obtuvieron resultados de 16.37 % y 43.87 % de GAP en promedio. En estas tablas se logra apreciar que de nuevo un factor determinante en la dificultad del problema es su tamaño.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
0.5#1-10-120-80-20	15.18	3634.13	2795905.00	3220417.32
0.5#2-10-120-80-20	15.57	3659.35	2902410.00	3354212.70
0.5#3-10-120-80-20	18.16	3656.19	2986560.00	3528961.69
0.5#4-10-120-80-20	16.83	3656.51	2883600.00	3368851.88
0.5#5-10-120-80-20	15.55	3651.56	2847585.00	3290494.84
0.5#6-10-120-80-20	12.39	3634.36	2982552.00	3352074.84
0.5#7-10-120-80-20	19.91	3633.13	2859948.00	3429308.05
0.5#8-10-120-80-20	16.87	3635.14	2864070.00	3347141.67
0.5#9-10-120-80-20	15.57	3632.90	2849180.00	3292806.34
0.5#10-10-120-80-20	17.67	3656.30	2981902.00	3508703.37
			Nodos	**Iteraciones**
Promedio	16.37	3644.96	4391.90	101306.30

Tabla 5.8: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.5 y tamaño 10-120-80-20.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
0.5#1-20-200-120-25	54.98	3702.15	2801904.00	4342483.43
0.5#2-20-200-120-25	65.77	3704.52	2682974.00	4447583.64
0.5#3-20-200-120-25	55.17	3703.15	2791020.00	4330864.07
0.5#4-20-200-120-25	32.82	3702.64	3147600.00	4180663.21
0.5#5-20-200-120-25	30.09	3704.27	3323177.00	4323254.33
0.5#6-20-200-120-25	50.99	3702.45	2805330.00	4235799.55
0.5#7-20-200-120-25	33.40	3702.97	3217840.00	4292567.11
0.5#8-20-200-120-25	35.63	3702.02	3217080.00	4363438.57
0.5#9-20-200-120-25	34.95	3703.62	3381506.00	4563362.03
0.5#10-20-200-120-25	44.92	3702.57	2923524.00	4236745.50
			Nodos	**Iteraciones**
Promedio	43.87	3703.04	536.80	27997.60

Tabla 5.9: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 0.5 y tamaño 20-200-120-25.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
1#1-5-50-30-5	7.68	3621.28	450350.00	484945.72
1#2-5-50-30-5	7.36	3635.99	494816.00	531250.21
1#3-5-50-30-5	13.28	3621.20	451288.00	511236.73
1#4-5-50-30-5	13.81	3616.76	519769.00	591533.52
1#5-5-50-30-5	17.66	3617.03	434468.00	511183.42
1#6-5-50-30-5	10.52	3618.83	462292.00	510913.45
1#7-5-50-30-5	15.25	3620.01	442800.00	510323.30
1#8-5-50-30-5	11.91	3619.95	428499.00	479543.15
1#9-5-50-30-5	19.06	3618.14	537307.00	639739.68
1#10-5-50-30-5	8.18	3615.47	473396.00	512131.90
			Nodos	**Iteraciones**
Promedio	12.47	3620.47	266183.20	9403588.10

Tabla 5.10: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 1 y tamaño 5-50-30-5.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
1#1-10-120-80-20	29.61	3632.80	2488873.00	3225870.21
1#2-10-120-80-20	63.15	3634.38	1984089.00	3237119.61
1#3-10-120-80-20	33.71	3635.85	2428920.00	3247750.52
1#4-10-120-80-20	33.53	3633.11	2376891.00	3173791.98
1#5-10-120-80-20	54.00	3634.01	2189204.00	3371408.68
1#6-10-120-80-20	34.69	3633.95	2438586.00	3284449.90
1#7-10-120-80-20	42.11	3632.66	2411920.00	3427599.94
1#8-10-120-80-20	39.16	3633.91	2415904.00	3361956.24
1#9-10-120-80-20	37.90	3636.47	2435913.00	3359078.08
1#10-10-120-80-20	25.84	3631.36	2602157.00	3274552.08
			Nodos	**Iteraciones**
Promedio	39.37	3633.85	1994.50	82538.20

Tabla 5.11: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 1 y tamaño 10-120-80-20.

Nombre	Gap %	Tiempo	Solución Factible	Cota Superior
1#1-20-200-120-25	419.05	3700.80	832557.00	4321386.14
1#2-20-200-120-25	100.64	3700.38	2127307.00	4268278.50
1#3-20-200-120-25	370.66	3700.50	934524.00	4398469.51
1#4-20-200-120-25	73.84	3700.72	2449890.00	4258838.34
1#5-20-200-120-25	75.85	3700.48	2454120.00	4315562.82
1#6-20-200-120-25	147.64	3706.32	1722165.00	4264828.88
1#7-20-200-120-25	114.70	3707.74	1965315.00	4219541.60
1#8-20-200-120-25	171.39	3720.36	1549848.00	4206084.65
1#9-20-200-120-25	85.27	3711.19	2303540.00	4267667.47
1#10-20-200-120-25	88.43	3858.16	2226390.00	4195224.66
			Nodos	**Iteraciones**
Promedio	164.75	3720.66	297.50	23418.00

Tabla 5.12: Resultados experimentales del algoritmo de ramificación y acotamiento para el problema de PPMM con densidad 1 y tamaño 20-200-120-25.

En las Tablas 5.10, 5.11 y 5.12, se muestran los resultados del algoritmo de ramificación y acotamiento para las instancias de densidad 1 y de tamaño chico, mediano y grandes, respectivamente. Los resultados de las 3 tablas son lejanos al óptimo, la densidad en estas instancias, a diferencia de las anteriores, sí afecta de gran manera el resultado. Obteniendo en la Tabla 5.12 un valor de 164.75 % de GAP promedio.

En la Tabla 5.13, se muestra un resumen de los resultados de las Tablas 5.4 a la Tabla 5.12.

En la Tabla 5.13, se puede observar que el factor que más influye en la solución del PPMM es el tamaño de la instancia. Se puede observar en la tabla, que al aumentar el tamaño de la instancia reduce dramáticamente la cantidad de nodos explorados, esto nos indica que aun la relajación contínua del algoritmo de relajación y acotamiento es complicada. En cuanto a la densidad también se puede observar que al aumentar el problema se vuelve más complicado y de igual forma la cantidad

D	Tamaño	Gap %	Tiempo(seg)	Nodos	Iteraciones
0.25	5-50-30-5	0.00	786.68	125672.40	1963406.20
0.50	5-50-30-5	0.43	2308.86	364112.10	7992460.90
1.00	5-50-30-5	12.47	3620.47	266183.20	9403588.10
0.25	10-120-80-20	9.20	3636.57	7134.70	157052.80
0.50	10-120-80-20	16.37	3644.96	4391.90	101306.30
1.00	10-120-80-20	39.37	3633.85	1994.50	82538.20
0.25	20-200-120-25	20.19	3703.34	660.30	30090.90
0.50	20-200-120-25	43.87	3703.04	536.80	27997.60
1.00	20-200-120-25	164.75	3720.66	297.50	23418.00

Tabla 5.13: Resumen de resultados experimentales del algoritmo de ramificación y acotamiento para el problema PPMM.

de nodos explorados es menor siendo que en las instancias de la Tabla 5.12 solo logra abrir 297 nodos en promedio.

Con los resultados del algoritmo de ramificación y acotamiento, se justifica la implementación de otro tipo de metodología, como los algoritmos heurísticos propuestos en el Capítulo 4 para resolver las instancias del PPMM, en la Seccion 5.3 se muestran los resultados y comparación de los algoritmos heurísticos propuestos en el Capítulo 4

5.3 Resultados experimentales de los algoritmos heurísticos para las instancias del PPMM

En esta sección, mostraremos los resultados obtenidos por los diferentes métodos heurísticos y hablaremos de los resultados de los mismo y su comparación contra los obtenidos por el método exacto presentados en la Sección 5.2.

En la Tabla 5.14, se muestra el promedio del Gap de cada tipo de instan-

cia. Para los diferentes algoritmos heurísticos expuestos en la Sección 4, este %Gap fue calculado a partir del resultado de la solución factible de Cplex de la siguiente manera: $\frac{(SF-H)}{SF}$, donde H representa el resultado de nuestras heurísticas y SF el resultado de Cplex. Los espacios vacios en la tabla indican que el tiempo de ejecución fue mayor a 900 segundos y tuvo que ser terminado el experimento sin resultado. Sin embargo, en la Sección 6.3 se menciona como uno de los trabajos a futuro encontrar los motivos porque estos heurísticos tienen un tiempo de ejecución tan largo.

Un punto importante de mencionar, es que por la aleatoridad de HPPMM9 se corre cada instancia 10 veces y el resultado que se presenta en esta sección es el promedio.

En la Tabla 5.14, se puede observar en los resultados que los mejores algoritmos son HPPMM8 y HPPMM9, debido a que sus promedios de GAP son mas cercanos al óptimo.

Tamaño	densidad	HPPMM1 %Gap	HPPMM2 %Gap	HPPMM3 %Gap	HPPMM4 %Gap	HPPMM5 %Gap	HPPMM6 %Gap	HPPMM7 %Gap	HPPMM8 %Gap	HPPMM9 %Gap
5-50-30-5	0.25	20.37	20.18	19.47	16.05	23.97	21.15	17.02	12.80	10.70
5-50-30-5	0.5	18.55	18.55	18.04	15.93	21.68	22.64	17.98	15.57	13.46
5-50-30-5	1	22.38	22.37	21.30	17.34	26.00	29.60	21.25	18.36	18.14
10-120-80-20	0.25	6.30	5.72	5.82	5.18	8.51	8.29	6.58	4.87	5.46
10-120-80-20	0.5	5.84	5.83	4.83	3.79	8.11	8.69	6.19	4.38	4.59
10-120-80-20	1	-1.06	-1.06	-2.52	-3.48	5.34	3.41	-0.56	-3.65	-3.56
20-200-120-25	0.25	-3.66	-3.69	-4.27	-4.56	—	—	—	-4.46	-4.88
20-200-120-25	0.5	-13.34	-13.29	—	-14.26	—	—	—	-14.38	-12.49
20-200-120-25	1	-92.18	-92.18	—	-97.21	—	—	—	-97.28	-93.88

Tabla 5.14: Tabla de comparación de Gap % de los métodos heurísticos.

Ahora bien en la Tabla 5.3, se muestra el promedio del tiempo de cada tipo de instancia, para los diferentes tipos de heurísticos. En la columna de izquierda de la tabla podemos observar los diferentes tamaños de las instancias y en la segunda columna las diferentes densidades.

En cuanto a él Gap se obtiene usando el mejor valor factible del algoritmo de ramificación y acotamiento para cada instancia. Cabe mencionar, que los valores de Gap negativos refieren a resultados de mayor calidad que los obtenidos en el algoritmo de ramificación y acotamiento.

Las heurísticas que tiene el mejor desempeño en cuanto al tiempo es HPPMM1, HPPMM2 y HPPMM4. En cuanto a las heurísticas HPPMM3, HPPMM5 y HPPMM7 no tuvieron buen desempeño en cuanto el tiempo, HPPMM6 tuvo un desempeño con las instancias pequeñas y medianas. Sin embargo, con las instancias grandes rebaso el límite fijado de los 900 segundos.

Por último HPPMM8 y HPPMM9 tuvieron un desempeño regular en cuanto al tiempo, sin embargo, HPPMM9 tuvo muy buenos resultados en cuanto al tiempo en las instancias más complicadas.

		HPPMM1	HPPMM2	HPPMM3	HPPMM4	HPPMM5	HPPMM6	HPPMM7	HPPMM8	HPPMM9
Tamaño	**densidad**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**
5-50-30-5	0.25	0.2113	0.2032	2.59	0.2632	3.63	2.51	4.693	0.847	8.86
5-50-30-5	0.5	0.27	0.294	2.59	0.347	4.48	2.54	5.13	1.237	11.79
5-50-30-5	1	0.48	0.508	2.89	0.627	15.57	2.95	8.91	2.38	25.70
10-120-80-20	0.25	5.936	5.467	127	7.84	166	129	178.9	24.4	63.92
10-120-80-20	0.5	6.48	6.99	133	8.28	201	138	212.33	36.7	65.64
10-120-80-20	1	16.6	17.4	138	21.9	707	141	454.2	86.2	69.31
20-200-120-25	0.25	38.53	39.54	869	23.19	—	—	—	154	93.93
20-200-120-25	0.5	26.4	27.1	—	40.4	—	—	—	277	84.86
20-200-120-25	1	174	178	—	232	—	—	—	715	149.90

Tabla 5.15: Tabla de comparación de tiempo de los métodos heurísticos.

Para concluir este capítulo, podemos mencionar que el método exacto funciona muy bien con las instancias de tamaño pequeño y de baja densidad, y que entre mayor tamaño y mayor densidad las instancias es mejor el uso de algoritmos heurísticos. Hablando de los diferentes heurísticos podemos mencionar que en promedio el que mejor funciono fue HPPMM8. Sin embargo, al complicarse las instancias su tiempo de ejecución crece demasiado mientras que el tiempo de ejecución de HPPMM9 permanece razonable menor a 200 segundos. Por lo tanto podemos mencionar que HPPMM9 es en general el de mejor desempeño. Ahora bien por los resultados podemos mencionar que Cplex tiene un mejor funcionamiento para los instancias sencillas, y que como se muestra en la Tabla 5.14 los algoritmos heurístico HPPMM4, HPPMM8 y HPPMM9 superan en las instancias grandes al método exacto en calidad de respuesta y tiempo de ejecución.

El mejor funcionamiento de HPPMM8 y HPPMM9 se debe en parte que al tener una estructura similar a HPPMM2 tienen un poco de aleatoriedad al momento de seleccionar moldes y máquinas para un producto con la libertad de poder volver hacerlo en una segunda ronda. Sin embargo, en HPPMM8 se corre varias veces el algoritmo para cada producto con el fin de evitar cuellos de botella y escoger el mejor producto lo cual funciona bien para la calidad de la respuesta, pero no para el tiempo y en HPPMM9 se opta por dar un poco de aleatoriedad al escoger los productos y la selección de moldes y máquinas lo cual permite que el problema converja a un buen resultado en menor tiempo.

Capítulo 6

Conclusiones

En este capítulo hablaremos de las conclusiones a las que se llega en este trabajo de tesis. Además, hablaremos de las aportaciones que en este trabajo fueron obtenidas y trabajo a futuro que podría complementar la investigación aquí presentada.

6.1 Conclusiones

El PPMM surge de un problema en la industria de la manufacturación de plásticos. En el cual hay que tomar una serie de decisiones de cuánto y qué producto fabricar a partir de los recursos disponibles, como las piezas que se pueden fabricar de las que está compuesto el producto. A su vez, estas piezas dependen del tiempo disponible de las máquinas y moldes. Otro punto que se toma en consideración fue el tiempo de instalación de los moldes en máquinas y además que las piezas salen en lotes de los moldes.

Se hizo el análisis del PPMM, problema con caracteristicas nunca antes abordado en la literatura. También, se hizo una formulación con programación lineal entera del PPMM y una serie de heurísticos para resolver este problema.

Para llevar a cabo la experimentación, se hizo un generador de instancias y en cuanto al tamaño de las instancias se pueden leer de la siguiente manera cantidad de productos-cantidad de piezas- cantidad de moldes -cantidad de máquinas y los

tamaños fueron los siguientes: $5-50-30-5$, $10-120-80-20$ y $20-200-120-25$.

Ahora bien, un punto importante a considerar son las compatibilidades entre los productos-piezas, piezas-moldes, moldes-máquinas, para las cuales la llamamos densidad y variamos la densidad productos-piezas (0.25, 0.5 y 1) mientras que las otras densidades se decidió dejarlas fijas en 0.15 y 0.6 respectivamente.

Esto nos da un total 9 tipos de instancias diferentes, las cuales fueron resultas por un algoritmo de ramificación y acotamiento y por los diferentes algoritmos heurísticos que desarrollamos especialmente para resolver el PPMM.

En la Sección 5.2, se muestra cómo la solución obtenida medíante el algoritmo de ramificación y acotamiento fue satisfactorio solamente para las instancias con densidad de 0.25 y 0.5 en tamaño 5-50-30-5. En cuanto a las demás instancias nos vimos en la necesidad de diseñar una métodología de solución de donde obtuvimos varios algoritmos heurísticos tipo constructivo.

El heurístico tipo constructivo HPPMM9 fue más eficiente que Cplex en las instancias del PPMM de densidad 1 tamaño 10-120-80-20 y las de densidades 0.25, 0.5, y 1 en tamaño 20-200-120-25. Aunque entre la ventaja principal del HPPMM9 es el tiempo de ejecución ofrece una buena alternativa al método exacto (algoritmo de ramificación y acotamiento).

Hay que hacer notar que la densidad con la que cuenta la instancia del PPMM es parte esencial de la dificultad del mismo como se menciona en [19] y [18]. Debido a que mayor densidad significa mayor cantidad de asignaciones posibles. Cabe destacar, que es más importante el tamaño de la instancia del PPMM. Debido que a mayor tamaño es mucho más complicado encontrar una solución factible tanto para el método exacto como para los algoritmos heurísticos.

Ahora bien los resultados del algoritmo heurístico tipo constructivo tienen un menor tiempo de ejecución que el método exacto. Sin embargo, aunque estos superan al método exacto no se cuentan con buenas cotas superiores para poder probar su eficacia.

Concluimos que dado el funcionamiento del algoritmo heurístico y sus variaciones, es de suma importancia el escoger el producto correcto a fabricar debido a que pueden presentarse cuellos de botella al momento de decidir que producto escoger. Además, también son de suma importancia la asignaciones de las piezas a las combinaciones molde-máquina. También, podemos agregar que tanto la respuesta del algoritmo heurístico como la del método exacto, dependen del tamaño de instancia y su densidad.

6.2 Aportaciones

En esta sección se resumen las diferentes aportaciones obtenidas a partir de la investigación del problema en este trabajo planteado. Entre las aportaciones más destacadas podemos mencionar las siguientes:

- Se plantea el problema PPMM el cual tiene características que en conjunto nunca antes habían sido abordadas, en la literatura.
- Se diseña un nuevo modelo matemático de programación entera, que cubre las características del problema PPMM.
- Se hace una prueba de complejidad del problema PPMM, en la cual se demostró que es NP-Duro.
- Se desarrolla un generador de instancias que emula las particularidades del problema PPMM.
- Dado que el modelo programación entera no cumplió con las exceptivas de eficiencia y eficacia, se desarrollo una métodología de solución en la cual consta de diversos algoritmos heurísticos tipo constructivo. Los cuales fueron evaluados hasta encontrar las características que mejor se ajustan al modelo de PPMM.

6.3 Trabajo a futuro

En esta sección mencionaremos algunas aspectos que son necesarios de implementar en este problema y algunas ideas para nuevas líneas de investigación.

Para empezar resulta imperativo implementar un algoritmo secuenciación similar al implementado en [19]. Para que este problema pueda ser implementado en la industria.

También es necesario eliminar suposiciones implícitas en la modelación de problema como suponer que al principio del día laborar los moldes tienen que ser instalados y todos los moldes se desinstalan al final del día.

El desarrollar y mejoramiento de cotas superiores, para poder evaluar los diferentes heurísticos. Así como, investigar porque el largo tiempo de ejecución en algunos de los heurísticos propuestos.

Otro punto es darle una ponderación a las piezas. Aunque, en este problema se busca darle solo valor al producto terminado es de suma importancia analizar los casos en que ya no se completen productos pero se puedan seguir fabricando piezas para otro día laboral. Dicho lo anterior, sería importante plantear el problema en multiperiodos.

Con respecto a la solución, sería interesante implementar un algoritmo de búsqueda local y si es necesario una metaheuristica. Esto para mejorar la calidad de la respuesta.

También se puede buscar atacar el problema desde los refuerzos a la modelación o relajaciones lagrangenas.

Otros de los aspectos que son muy importantes es tratar de implementar este problema en diferentes fábricas de productos de plásticos y ver si se puede adaptar a sus particularidades.

Otros de los posibles intereses en este problema es tratar de adaptar el modelo a otras aéreas que se trabaje con moldes y máquinas como la industria del fundición.

Implementar el desgaste de moldes y máquinas como una restricción.

Ahora también es importante considerar que las líneas de ensambles y darle prioridad a piezas necesarias para empezar a armar un producto.

Y por último considerar particularidades de las empresas para una metodología de solución. Por ejemplo, que pasa cuando en medio de producción manda un pedido un cliente muy importante y lo requiere de inmediato ¿La empresa para su producción o no? y sí para, como se ajusta las asignaciones ya antes decididas.

Como se menciona en este capítulo todavía hay una gran cantidad de trabajo por realizar en este problema.

Bibliografía

[1] Almada-Lobo, B., D. Klabjan, M. Carravilla y J. Oliveira, «Multiple machine continuous setup lotsizing with sequence-dependent setups», *Computational Optimization and Applications*, **47**(3), págs. 529–552, 2010.

[2] Angel, E., E. Bampis y A. Kononov, «On the approximate tradeoff for bicriteria batching and parallel machine scheduling problems», *Theoretical computer science*, **306**(1), págs. 319–338, 2003.

[3] Baker, K., *Introduction to sequencing and scheduling*, Wiley, 1974.

[4] Bazaraa, M. S., J. J. Jarvis y H. D. Sherali, *Programación lineal y flujo en redes*, segunda edición, Limusa, México, DF, 2004.

[5] Bryce, D., *Plastic injection molding: manufacturing process fundamentals*, tomo 1, Sme, 1996.

[6] Burkard, R., M. Dell'Amico y S. Martello, *Assignment problems*, Society for Industrial Mathematics, 2009.

[7] Chen, B., Y. Ye y J. Zhang, «Lot-sizing scheduling with batch setup times», *Journal of Scheduling*, **9**(3), págs. 299–310, 2006.

[8] Chen, J. y T. Wu, «Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints», *Omega*, **34**(1), págs. 81–89, 2006.

[9] CHRÉTIENNE, P., *Scheduling theory and its applications*, John Wiley & Sons, 1995.

[10] CHVÁTAL, V., *Linear programming*, WH Freeman, 1983.

[11] CONWAY, R., W. MAXWELL y L. MILLER, *Theory of Scheduling*, Dover Books on Mathematics, Dover Publications, 2003.

[12] EREN, T., «A multicriteria scheduling with sequence-dependent setup times», *Applied Mathematical Sciences*, **1**(58), págs. 2883–2894, 2007.

[13] EREN, T. y E. GÜNER, «A bicriteria scheduling with sequence-dependent setup times», *Applied mathematics and computation*, **179**(1), págs. 378–385, 2006.

[14] FRENCH, S., *Sequencing and scheduling: an introduction to the mathematics of the job-shop*, Ellis Horwood series in mathematics and its applications, E. Horwood, 1982.

[15] GRIGORIEV, A., M. SVIRIDENKO y M. UETZ, «Unrelated parallel machine scheduling with resource dependent processing times», *Integer Programming and Combinatorial Optimization*, págs. 53–86, 2005.

[16] HAASE, K. y A. KIMMS, «Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities», *International Journal of Production Economics*, **66**(2), págs. 159–169, 2000.

[17] HOPCROFT, J. y R. KARP, «An $n\hat{5}/2$ Algorithm for Maximum Matchings in Bipartite Graphs», *SIAM Journal on Computing*, **2**(4), págs. 225–231, 1973.

[18] IBARRA-ROJAS, O., *Programación pieza-molde-máquina en planeación de producción mediante una Búsqueda Local Iterativa*, Tesis de Maestría, Universidad Autónoma de Nuevo León, México, 2009.

[19] IBARRA-ROJAS, O., R. RÍOS-MERCADO, Y. RIOS-SOLIS y M. SAUCEDO-ESPINOSA, «A decomposition approach for the piece-mold-machine manufacturing problem», *International Journal of Production Economics*, 2011.

[20] JOHANNABER, F., *Injection Molding Machines: a user's guide*, Hanser Verlag, 2008.

[21] JOHNSON, D. y M. GAREY, «Computers and Intractability: A Guide to the Theory of NP-completeness», *Freeman&Co, San Francisco*, 1979.

[22] KAN, A., *Machine scheduling problems: classification, complexity and computations*, Stenfert Kroese, 1976.

[23] KARP, R., «Reducibility among combinatorial problems», *Complexity of Computer Computations*, **43**(4), págs. 85–103, 1972.

[24] KELLERER, H., U. PFERSCHY y D. PISINGER, *Knapsack problems*, Springer Verlag, 2004.

[25] KIM, S., H. CHOI y D. LEE, «Tabu search heuristics for parallel machine scheduling with sequence-dependent setup and ready times», *Computational Science and Its Applications-ICCSA 2006*, págs. 728–737, 2006.

[26] LAND, A. y A. DOIG, «An automatic method of solving discrete programming problems», *Econometrica: Journal of the Econometric Society*, págs. 497–520, 1960.

[27] LAU, H., *Combinatorial Heuristic Algorithms With Fortran*, Lecture Notes in Economics and Mathematical Systems, Springer, 1986.

[28] LI, S., G. LI y S. ZHANG, «Minimizing makespan with release times on identical parallel batching machines», *Discrete Applied Mathematics*, **148**(1), págs. 127–134, 2005.

[29] LUO, X. y F. CHU, «A branch and bound algorithm of the single machine schedule with sequence dependent setup times for minimizing total tardiness», *Applied mathematics and computation*, **183**(1), págs. 575–588, 2006.

[30] MARTELLO, S. y P. TOTH, «A bound and bound algorithm for the zero-one multiple knapsack problem», *Discrete Applied Mathematics*, **3**(4), págs. 275–288, 1981.

[31] MARTELLO, S. y P. TOTH, *Knapsack problems: algorithms and computer implementations*, John Wiley & Sons, Inc., 1990.

[32] MICHALEWICZ, Z. y D. FOGEL, *How To Solve It: Modern Heuristics*, Springer, 2004.

[33] MOHAMMADI, M., S. FATEMI GHOMI, B. KARIMI y S. TORABI, «Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups», *Journal of Intelligent Manufacturing*, **21**(4), págs. 501–510, 2010.

[34] MOUSTAKAS, C., *Heuristic Research: Design, Methodology, and Applications*, Sage Publications, 1990.

[35] PAPADIMITRIOU, C., *Computational Complexity*, Addison Wesley Pub. Co., 1994.

[36] PINEDO, M., *Scheduling: Theory, Algorithms, and Systems*, Springer, 2012.

[37] PISINGER, D., *Algorithms for knapsack problems*, Tesis Doctoral, University of Copenhagen, 1995.

[38] POLYA, G., «How to solve it: A new aspect of mathematical model», , 1945.

[39] ROTHLAUF, F., *Application and Adaptation of Heuristic Optimization Methods*, Natural Computing Series, Springer, 2011.

[40] SHIM, S. y Y. KIM, «A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property», *Computers & Operations Research*, **35**(3), págs. 863–875, 2008.

[41] SIMON, H. y A. NEWELL, «Heuristic problem solving: The next advance in operations research», *Operations research*, **6**(1), págs. 1–10, 1958.

[42] STAGGEMEIER, A. y A. CLARK, «A survey of lot-sizing and scheduling models», *33º Simpósio Brasileiro de Pesquisa Operacional*, 2001.

[43] STECCO, G., J. CORDEAU y E. MORETTI, «A branch-and-cut algorithm for a production scheduling problem with sequence-dependent and time-dependent setup times», *Computers & Operations Research*, **35**(8), págs. 2635–2655, 2008.

[44] TSAI, C. y C. TSENG, «Unrelated parallel-machines scheduling with constrained resources and sequence-dependent setup time», en *Proceedings of the 37th International Conference on Computers and Industrial Engineering, Alexandria, Egypt*, págs. 2087–2095, 2007.

[45] VILÍM, P., «Batch processing with sequence dependent setup times», en *Principles and Practice of Constraint Programming-CP 2002*, Springer, págs. 153–167, 2006.

[46] VLAH, S., Z. LUKAČ y J. PACHECO, «Use of VNS heuristics for scheduling of patients in hospital», *Journal of the Operational Research Society*, **62**(7), págs. 1227–1238, 2010.

[47] WENG, M., J. LU y H. REN, «Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective», *International Journal of Production Economics*, **70**(3), págs. 215–226, 2001.

[48] WILLIAMS, H., *Model building in mathematical programming*, tomo 4, Wiley, 1999.

[49] WOLSEY, L., *Integer Programming*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, 1998.

[50] YAVUZ, S., «Scheduling with Batching and Lot-sizing», *Department of Industrial Engineering, Bilkent University*, 1999.

[51] YU, L., H. SHIH, M. PFUND, W. CARLYLE y J. FOWLER, «Scheduling of unrelated parallel machines: an application to PWB manufacturing», *IIE Transactions*, **34**(11), págs. 921–931, 2002.

Ficha autobiográfica

I.Q. Miguel Lorenzo Morales Marroquín

Candidato para el grado de Maestría en Ciencias
con especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Optimización de la producción en máquinas en paralelo de inyección de plástico

Nací un día jueves 11 de marzo de 1982 a las 17:30 horas, en la ciudad de Monterrey, Nuevo León, México. Soy el primogénito del Sr. Miguel Morales Garza y la Sra. Blanca Mireya Marroquín de Morales. En junio de 2009 me gradué como Ingeniero Químico de la facultad de Ciencias Químicas de la Universidad Autónoma de Nuevo León. El miércoles 23 de septiembre de 2009 fui bendecido con el nacimiento de mi primera hija Regina Morales Suarez a la edad de 27 años. En enero de 2010 ingresé al posgrado en ingeniería de sistemas de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León donde actualmente estoy finalizando la maestría en ciencias con especialidad en sistemas bajo la asesoría de la Dra. Yasmín A. Ríos Solís y el apoyo de CONACyT. El lunes 25 de abril de 2011 fui

bendecido por segunda ocasión con el nacimiento de mi segunda hija Minerva Morales Suarez. Actualmente vivo en compañía de mi adorada Génesis Suarez Lazalde y mis dos hijas en Guadalupe, Nuevo León, México.