

UNIVERSIDAD DE LOS ANDES

**DESCOMPOSICIÓN ROBUSTA DE HORIZONTES TEMPORALES EN
PROBLEMAS DE REGULACIÓN AGREGADA DE CARGAS DE TRABAJO,
PARA MÚLTIPLES RECURSOS**

Angela María Castro Fontecha

Bogotá, Enero de 2003

TABLA DE CONTENIDO

TABLA DE CONTENIDO		2
1. INTRODUCCIÓN		4
2. MARCO TEÓRICO		6
2.1 ESTRUCTURAS DECISIONALES DE UN SISTEMA DE PRODUCCIÓN		6
2.2 REGULACIÓN AGREGADA DE CARGA		8
2.3 HORIZONTE DE PLANEACIÓN:		9
2.4 REFERENCIALES TEMPORALES		9
2.5 ESTRUCTURA ADYACENTE Y PROPIEDAD DE ADYACENCIA:		10
2.6 CURVAS DE CARGA		12
2.7 MARGEN DE AUTONOMÍA		15
2.8 OPTIMIZACIÓN MÚLTIPLES OBJETIVOS		16
2.8.1 <i>Suma de Pesos:</i>		20
2.8.2 <i>Programación de metas</i>		21
2.8.3 <i>El método de las restricciones</i>		22
2.9 ALGORITMO DE WAGNER - WHITIN		23
2.10	SIMULATED ANNEALING	24
3. PLANTEAMIENTO DEL MODELO		29
4. ALGORITMOS DE LOS MÉTODOS DE SOLUCIÓN PROPUESTOS		31
4.1 ALGORITMO DE PROGRAMACIÓN DINÁMICA (BASADO EN EL ALGORITMO DE WAGNER – WHITIN):		31
4.2 ALGORITMO DE SIMULATED ANNEALING		34

4.3 DESARROLLO DEL PROGRAMA PARA LA SOLUCIÓN DE LOS ALGORITMOS	36
4.3.1 <i>Explicación</i>	36
5. EJEMPLOS APLICADOS AL MODELO	41
6. ANÁLISIS DE RESULTADOS	53
5.1 RESULTADOS DE LOS EJEMPLOS	53
5.2 ANÁLISIS DE RESULTADOS	55
7. CONCLUSIONES	56
8. BIBLIOGRAFÍA	58

1. INTRODUCCIÓN

En los sistemas de producción, las personas encargadas de la toma de decisiones de programación, se ven enfrentadas a la necesidad de manipular eficazmente el horizonte temporal de planificación. La descomposición del horizonte en intervalos adyacentes permite caracterizar la coherencia del plan de operaciones propuesto con las restricciones de capacidad real del taller [7].

El modelo propuesto en esta tesis presenta una solución al problema de la planeación agregada, donde el proceso a solucionar requiere de múltiples operaciones o tareas y cada operación requiere el uso de diferentes tipos recursos para desarrollarse. El modelo consiste en maximizar el margen de carga encontrando una descomposición del horizonte temporal de planeación que cumpla con la *PROPIEDAD DE ADYACENCIA* [TORRES]. Hasta el momento los modelos presentados para maximizar el margen en un plan de operaciones, aceptando la *PROPIEDAD DE ADYACENCIA*, se han desarrollado bajo el supuesto de que cada tarea requiere de un solo tipo de recurso [9].

Para encontrar el modelo que MAXIMICE EL MARGEN DE CARGA en el caso donde es necesario el uso de diferentes tipos de recursos se utilizó el concepto de optimización de MÚLTIPLES OBJETIVOS. El método de solución aplicado fue el de SUMA DE PESOS donde a cada recurso se le dio un valor o costo para así poder unirlos en una única función.

En la última parte de la tesis se presentan dos métodos de optimización el primero un proceso de programación dinámica basado en el algoritmo de Wagner – Whitin

y el segundo método de optimización es utilizando la técnica de Simulated Annealing el cual consiste en optimizar una solución ya dada al problema que se plantea. Los resultados de la implementación de estas dos técnicas de optimización, programados en Java, se presentan para tres casos particulares.

Decisiones estratégicas, relativas a la política general de la empresa (horizonte: algunos años), abarcan por ejemplo:

- La política presupuestal: plan de inversiones en medios materiales, financieros o humanos.
- La selección de nuevas orientaciones: introducción de nuevos productos, técnicas recientes.
- La puesta en marcha de campañas publicitarias, etc.

Decisiones tácticas aseguran la gestión del trabajo y de los recursos de producción en el marco delimitado por las decisiones estratégicas (horizonte: algunos meses). Suponiendo que los pedidos de los clientes, las necesidades de materias primas y las capacidades de los recursos de producción son conocidos, se trata de definir un plan de producción y un plan de utilización de los recursos disponibles sobre el horizonte de planificación.

Decisiones operacionales regulan en detalle la fabricación propiamente dicha (horizonte: algunos días). Y definen por ejemplo, la programación de las operaciones de fabricación sobre los recursos, la afectación de los equipos de hombres, etc.

Toda decisión tomada a un nivel dado constituye un objetivo para el nivel inmediatamente inferior de la estructura. El retorno de informaciones de un nivel hacia su nivel superior es necesario para facilitar la evaluación o modificación de las decisiones tomadas por éste.

2.2 REGULACIÓN AGREGADA DE CARGA

La planeación agregada en una compañía manufacturera es un proceso por medio del cual se determina la cantidad y mezcla óptima de productos que debe seleccionar. Este mecanismo tiene como objetivo traducir los pronósticos de la demanda en un esquema de plan de operaciones, durante un *horizonte predeterminado de planeación* [2].

El plan de operaciones de un recurso agregado es el conjunto de tareas planificadas. Estas tareas se efectuarán sobre el recurso durante un horizonte de planeación fijo. Cada tarea esta definida por la fecha de comienzo a más temprano, la fecha de fin a más tarde y la duración estimada [6,9]. El conjunto del plan de operaciones es:

$$\pi = \{w_i / w_i = (C_i, F_i, D_i, q_i), 1 \leq i \leq N\}$$

i = Operación planificada

C_i = Fecha de Comienzo a más temprano

F_i = Fecha de fin a más tarde

D_i = Duración

q_i = Consumo del recurso de la operación planificada w_i

Representación del plan de operaciones:

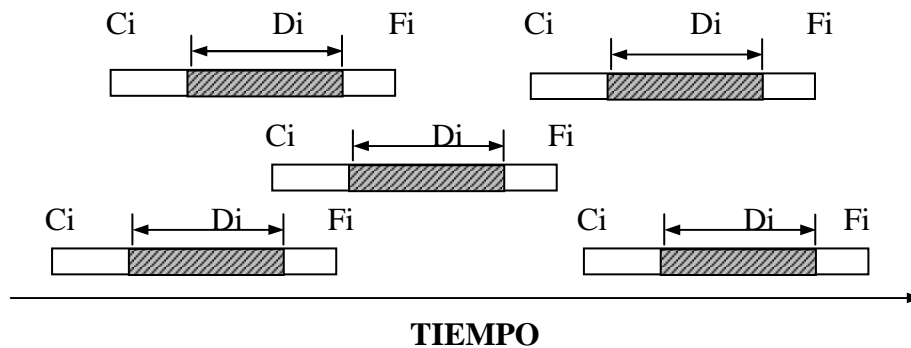


Figura 2.2. Representación gráfica del plan de operaciones

2.3 HORIZONTE DE PLANEACIÓN:

El horizonte de planeación corresponde al intervalo de tiempo en donde se efectúa la planeación de operaciones ver numeral 2.2. El horizonte de planeación puede ser dividido en intervalos de igual longitud, que corresponden a los períodos significativos de una organización colectiva de trabajo (días, turnos,...) [6]. Las personas encargadas de la toma de decisiones de programación, se ven en la necesidad de construir eficazmente el horizonte temporal de planificación [2]. Este horizonte se define como el intervalo de tiempo $[T_1, T_M]$ donde:

$$T_1 = \underset{w_i \in \pi}{\text{Min}}(C_i)$$

$$T_M = \underset{w_i \in \pi}{\text{Max}}(F_i)$$

Con la ayuda de soportes apropiados de decisión, las personas encargadas de tomar decisiones en la planificación, pueden llegar a un horizonte que permita a otros centros de decisión funcionar con márgenes apropiados de maniobra [6]. Un horizonte que puede permitir un margen de maniobra apropiado es el “horizonte de planificación en intervalos adyacentes” [6,9].

2.4 REFERENCIALES TEMPORALES

El hecho de introducir, sobre la figura 2.3, una recta vertical de abscisa T, permite percibir los posicionamientos temporales relativos de las operaciones: con relación a la fecha T de una parte, y unas en relación a las otras, de otra parte.

La introducción, en un problema de planificación, de fechas o referenciales temporales permite estructurar el razonamiento en el tiempo. Evidentemente, la

selección de tales referenciales no es independiente de la utilidad que se espera, ni del objetivo que se persigue, o de la estrategia decisional que se espera adoptar en la búsqueda de soluciones que regulen la carga de trabajo [6]

2.5 ESTRUCTURA ADYACENTE Y PROPIEDAD DE ADYACENCIA:

El Horizonte de planeación en intervalos adyacentes, se descompone en $M-1$ intervalos, donde M es el número total de referenciales del horizonte de planeación, y el conjunto intervalos se conoce como *estructura adyacente* [6,7,9]. Los intervalos de la estructura adyacente son divididos cada uno por un referencial ubicado en el horizonte de planeación. En la figura 2.3 se puede ver la representación de un horizonte de planeación que esta formado por 4 intervalos de tiempo y 5 referenciales temporales nombrados como T_1, T_2, \dots, T_5 .

En esta estructura se supone que hay un intercambio de carga entre los intervalos vecinos. Esto facilita la regulación de cargas de trabajo y la programación [6,7]. El horizonte planeación se plantea bajo el supuesto de la *propiedad de adyacencia*, donde se garantiza que cada tarea este presente **máximo en dos intervalos sucesivos de la estructura adyacente**. [6,9].

Cada referencial temporal que pertenece a la estructura adyacente debe satisfacer las siguientes restricciones que conforman la propiedad de adyacencia.

1. $T_j < T_{j+1}$ para $2 \leq j \leq M-1$
2. $C_i + T(1 - Z_{ij}) \geq T_{j-1}$
3. $F_i \leq T_{j+1} + T(1 - Z_{ij})$, para $1 \leq i \leq N$, $2 \leq j \leq M-1$
4. $T_1 = \min C_i \quad w_i \in \pi$
5. $T_M = \max F_i \quad w_i \in \pi$

$$6. \quad Z_{ij} \in \left\{ \begin{array}{l} 1 \quad \text{si } T_{j-1} \leq C_i \leq F_i \leq T_{j+1} \\ 0 \quad \text{de lo contrario} \end{array} \right\}$$

7. $T =$ Un número grande, por ejemplo, $T = T_M + 1$

Donde:

j = Es la posición del referencial en el horizonte de planeación.

La restricción número 1 garantiza que el referencial T_j está ubicado antes del referencial inmediatamente siguiente (T_{j+1}), formando de esta manera un intervalo.

Las restricciones 2 y 3 garantizan que la fecha de comienzo a más temprano de la operación i puede ser menor o igual a la fecha que indica el referencial T_{j-1} y la fecha de fin a más tarde de la operación i no puede ser mayor al referencial ubicado en T_{j+1} . De esta manera se garantiza que la tarea no pertenezca a más de dos intervalos de tiempo.

Las restricciones 4 y 5 indican que la fecha de comienzo más temprana de todas las operaciones, y la fecha de fin a más tarde de todas las operaciones pertenecen al plan de operaciones.

Z_{ij} = Variable binaria que toma el valor de 1 si la operación se encuentra entre dos intervalos adyacentes y toma el valor de cero para el caso contrario.

Un horizonte de planeación que cumpla con estas condiciones se muestra en la siguiente gráfica:

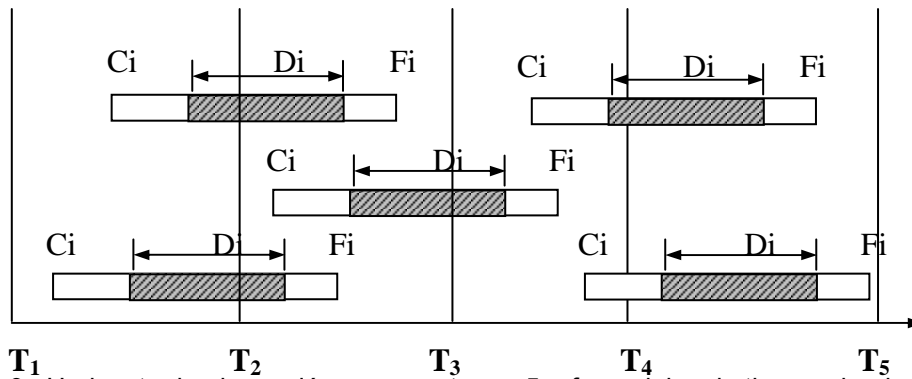


Figura 2.3. Horizonte de planeación compuesto por 5 referenciales de tiempo donde los intervalos cumplen con la propiedad de adyacencia

Se puede notar que ninguna tarea pertenece a más de 2 intervalos y que los referenciales no están necesariamente a una misma distancia.

La elaboración de una Estructura Adyacente permite descomponer el conjunto de operaciones planificadas sobre el recurso agregado considerado, en subconjuntos de tipo Ω_j y Int_j , que pueden definirse de la siguiente manera:

Sea (T_1, T_2, \dots, T_n) la Estructura Adyacente escogida,

1. $\Omega_j = \{w_i / C_i < T_j < F_i\}$ es el conjunto de ventanas que encuentran T_j . Es decir, las operaciones que son atravesadas por el referencial T_j
2. $Int_j = \{w_i / T_j \leq C_i < F_i \leq T_{j+1}\}$ es el conjunto de ventanas contenidas en el intervalo $[T_j, T_{j+1}]$.

2.6 CURVAS DE CARGA

La suma de trabajo requerida para la realización de un conjunto de operaciones se conoce con el concepto de carga. Las curvas de carga fijan los límites de carga

(en unidades de tiempo x recurso) máximo y mínimo, referidas sobre el horizonte de planeación.

Sobre cada intervalo del horizonte, y para un recurso agregado específico, los valores acumulados de las cargas sobre las operaciones que interceptan el intervalo considerado, por medio del concepto de contribución individual, se puede definir las cargas mínima y máxima [7].

Entonces la carga máxima $WMAX_l$ es obtenida posicionando todas las tareas lo más al interior posible del intervalo I comprendido entre $[T_j, T_{j+1}]$:

Por medio del cálculo de los valores acumulados de las contribuciones de todas las operaciones cuyas ventanas son de intersección no vacía con $[T_j, T_{j+1}]$, obtenemos:

$$WMAX_l = \sum_{\omega_i \in \Omega_j} \min(D_i, F_i - T_j) + \sum_{\omega_i \in Int_j} D_i + \sum_{\omega_i \in \Omega_{j+1}} \min(D_i, T_{j+1} - C_i)$$

Donde:

I = Intervalo al cual se le esta calculando la carga.

La $WMAX_l$ presentada corresponde al caso donde el plan de operaciones requiere el uso de un solo tipo de recurso; por esto q_i toma el valor de 1 y no aparece en la multiplicación de cada uno de los términos. De esta manera la ecuación queda con las unidades de tiempo x recurso que corresponden a las cargas agregadas para cada intervalo de tiempo.

El primero y tercer término de la ecuación para la calcular la carga máxima ($WMAX_l$) para el intervalo I , indican las cargas aportadas por las operaciones que son atravesadas por los referenciales T_j y T_{j+1} respectivamente. El segundo

término de la ecuación presenta la carga aportada por cada operación que esta en medio de los referenciales T_j y T_{j+1} .

La carga máxima para el intervalo $I [T_j, T_{j+1}]$ sería:

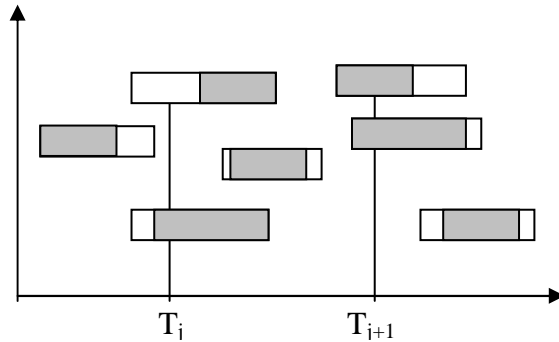


Figura 2.4. Representación de la Carga Máxima para un intervalo comprendido entre T_j y T_{j+1}

La carga mínima **WMIN_i** es obtenida posicionando todas las tareas lo más al exterior posible del intervalo I comprendido entre $[T_j, T_{j+1}]$:

Por el cálculo de los acumulados de las contribuciones de todas las operaciones cuyas ventanas tienen intersección no vacía con el intervalo I comprendido entre $[T_j, T_{j+1}]$, obtenemos:

$$WMIN_i = \sum_{\omega_i \in \Omega_j} \max(0, C_i + D_i - T_j) + \sum_{\omega_i \in Int_j} D_i + \sum_{\omega_i \in \Omega_{j+1}} \max(0, T_{j+1} - (F_i - D_i))$$

La **WMIN_i** presentada corresponde al caso donde el plan de operaciones requiere el uso de un solo tipo de recurso; por esto α_i toma el valor de 1 y no aparece en la multiplicación de cada uno de los términos. De esta manera la ecuación queda con las unidades de tiempo por recurso que corresponden a las cargas agregadas para cada intervalo de tiempo.

De manera similar a la **WMAX_i**, los términos primero y tercero de la ecuación para la calcular la carga mínima **WMIN_i** del intervalo I , indican la carga aportada por las

operaciones que son atravesadas por los referenciales T_j y T_{j+1} respectivamente. El segundo término de la ecuación presenta la carga aportada por cada operación que esta en medio de los referenciales T_j y T_{j+1} .

La carga mínima para el intervalo $[T_j, T_{j+1}]$ sería:

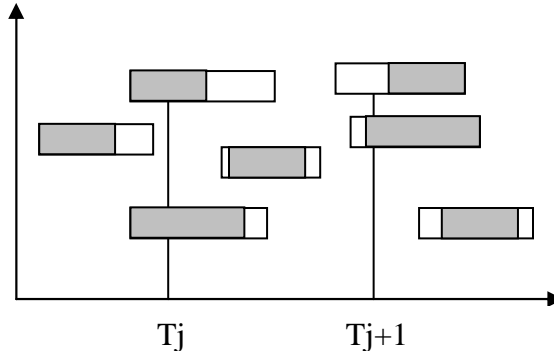


Figura 2.5. Representación de la Carga Mínima para un intervalo comprendido entre T_j y T_{j+1}

2.7 MARGEN DE AUTONOMÍA

El margen de autonomía se define como la diferencia entre las cargas máxima y mínima para cada intervalo de tiempo I comprendido entre $[T_j, T_{j+1}]$ es función del plan de operaciones π y de la descomposición del horizonte de fabricación $E(\pi)$. Para un solo tipo de recurso se tendría la siguiente expresión para el margen de carga por tarea es:

$$MARGEN DE CARGA = W(E(\pi), \pi) = \sum_{l=1}^L [W \max_l - W \min_l]$$

Donde L = al número total de intervalos en el horizonte de planeación.

Gráficamente puede ser representada de la siguiente forma:

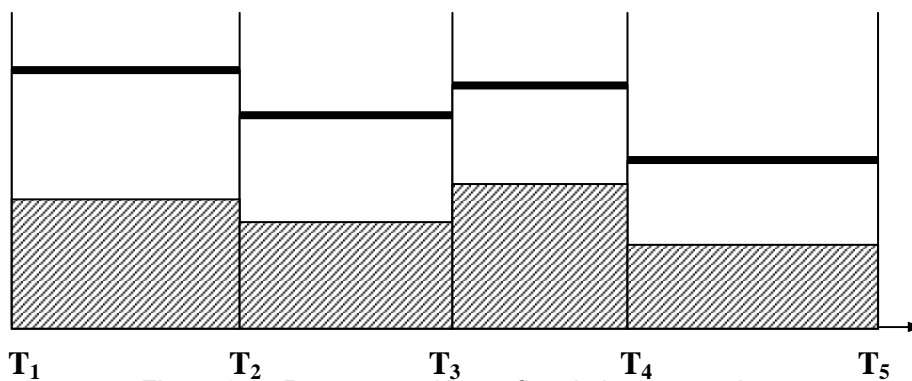


Figura 2.6. Representación gráfica de las curvas de carga

La línea superior señala la carga máxima aportada por cada una de las tareas y la línea inferior marca la carga mínima aportada por cada una de las tareas a los intervalos. Es importante aclarar que esto es para el caso donde las tareas necesitan de un solo tipo de recurso.

2.8 OPTIMIZACIÓN MÚLTIPLES OBJETIVOS

La descomposición robusta de horizontes temporales en problemas de regulación agregada de cargas de trabajo, para múltiples recursos, puede ser un caso típico para ser resuelto por métodos de optimización de múltiples objetivos.

Casi todos los problemas reales involucran la optimización de varios objetivos que a menudo son conflictivos y competitivos entre sí. Mientras que en la optimización con un simple objetivo usualmente la solución óptima está bien definida, no sucede lo mismo en problemas con objetivos múltiples. En lugar de un óptimo simple, existen un conjunto de alternativas conocidas como soluciones óptimo Pareto (conjunto de soluciones no inferiores o no dominadas). Éstas soluciones son óptimas en el amplio sentido de que ningunas otras soluciones son superiores a éstas cuando todos los objetivos son considerados. [4].

Entonces, la optimización con objetivos múltiples (llamada también optimización vectorial y con criterios múltiples) puede definirse como [1]:

El problema de encontrar un vector de variables de decisión que satisfaga las restricciones y optimice una función vectorial cuyos elementos representen las funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que usualmente están en conflicto entre sí. Por lo tanto, el término "optimizar" significa encontrar una solución tal que proporcione valores para todos los objetivos que resulten aceptables para el diseñador.

El problema puede ser definido de una manera simple de la siguiente forma:

$$\text{Maximizar } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))$$

Sujeto a

$$R(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_m(\mathbf{x}))$$

Donde

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

$$F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))$$

Entonces: \mathbf{x} es el vector de decisión, $F(\mathbf{x})$ es el vector objetivo. Las restricciones $R(\mathbf{x})$ determinan el conjunto soluciones factibles.

Estos problemas de optimización multiobjetivo son muy comunes en el área de la ingeniería y las técnicas para resolverlos son muy diferentes a las de la optimización simple [4].

El modelo de regulación de cargas para un solo tipo de recurso puede ser planteado como un modelo de optimización de múltiples funciones objetivo donde se tienen las cargas máximas y mínimas que aporta cada intervalo por tipo de recurso que se utilice. Para el intervalo l que este comprendido entre los referenciales $[T_j, T_{j+1}]$. Las ecuaciones quedarían expresadas de la siguiente forma:

WMAX_{l,k} = Carga Máxima por intervalo, para cada tipo de recurso

$$WMAX_{l,k} = \sum_{\omega_i \in \Omega_j} [q_{ik} * \min(D_{ik}, F_i - T_j)] + \sum_{\omega_i \in Int_j} (q_{ik} * D_{ik}) + \sum_{\omega_i \in \Omega_{j+1}} [q_{ik} * \min(D_{ik}, T_{j+1} - C_i)] \quad \forall k$$

WMIN_{l,k} = Carga Mínima por intervalo, para cada tipo de recurso

$$WMIN_{l,k} = \sum_{\omega_i \in \Omega_j} [q_{ik} * \max(0, C_i + D_{ik} - T_j)] + \sum_{\omega_i \in Int_j} (q_{ik} * D_{ik}) + \sum_{\omega_i \in \Omega_{j+1}} [q_{ik} * \max(0, T_{j+1} - (F_i - D_{ik}))] \quad \forall k$$

q_{ik} = Cantidad del recurso tipo k que es consumido por la operación i

k = Tipo de recurso que esta consumiendo la operación i

$D_{ik} = D_i$ = Duración del consumo del recurso k por la tarea i es igual a la duración de la tarea i . Esto para todo k

Y la maximización del margen de autonomía para cada tipo de recurso sería:

$$\text{MAX MARGEN}(\varepsilon(\pi), \pi) = \sum_{l=1}^L (WMAX_l - WMIN_l) \quad \forall k$$

Donde L = al número total de intervalos en el horizonte de planeación.

Uno de los problemas para el caso de múltiples recursos es que las cargas de trabajo para cada uno de los intervalos no son independientes, luego el máximo margen de carga no podría ser expresado como la suma de las cargas de trabajo para cada tipo de recurso.

$$\text{MAX MARGEN}(\varepsilon(\pi), \pi) \neq \sum_{k=1}^K \text{MAX MARGEN}(\varepsilon(\pi), \pi)_k$$

El problema de maximización de cargas para múltiples objetivos puede ser planteado por alguno método de optimización para múltiples objetivos. Esta sección incluye las técnicas más conocidas.

Probablemente la idea más simple que se podría proponer para trabajar con varios objetivos sería combinarlos en uno solo usando una suma, una multiplicación o cualquier otra combinación de operaciones aritmética. Hay, sin embargo, problemas para este tipo de técnicas.

Esto implica que se debe conocer, en la medida de lo posible, el comportamiento de cada una de las funciones objetivo, lo cual normalmente (al menos en la mayoría de las aplicaciones del mundo real) es un proceso muy costoso (en términos de tiempo de CPU) [1]. Obviamente, si esta combinación de objetivos es posible (y lo es, en algunos casos), esta técnica no sólo es la más simple de implementar, sino que además es la más eficiente, porque no se requiere posterior interacción con el usuario; y si el algoritmo tiene éxito en el proceso de optimización, entonces los resultados serán al menos sub-óptimos para la mayoría de los casos.

Al proceso de combinar objetivos en una sola función se le denomina normalmente función agregada, y se le ha utilizado en diversas ocasiones con

éxito relativo en problemas en los cuales el comportamiento de las funciones objetivo se conoce más o menos bien [1, 5].

2.8.1 Suma de Pesos:

Este método consiste en sumar todas las funciones objetivo usando diferentes pesos para cada una de ellas. Esto significa que el problema con objetivos múltiples se transforma en un problema de optimización de la forma:

$$\max \sum_{k=1}^K \gamma_k * F_k(x)$$

Donde γ_k son los pesos que representan la importancia relativa de los objetivos y usualmente se supone que:

$$\sum_{k=1}^K \gamma_k = 1$$

Los resultados obtenidos en un problema de optimización usando el método de optimización por suma de pesos pueden variar significativamente conforme se modifiquen los pesos debido a que no se conoce muy poco acerca de la forma más adecuada de seleccionar estos coeficientes. En problema con múltiples objetivos se requiere resolver el mismo problema usando diferentes valores de γ_k .

Los pesos no reflejan proporcionalmente la importancia relativa de los objetivos, sino que son sólo factores que, al variarse, localizan puntos diferentes en el conjunto de posibles soluciones o conjunto de Pareto.

Análisis de la técnica

Esta técnica es muy eficiente desde el punto de vista de recursos de cómputo, y puede usarse para generar una solución fuertemente dominada que pueda usarse como punto inicial para otras técnicas [1]

El principal problema de esta técnica es cómo determinar los pesos apropiados cuando no se tiene suficiente información acerca del problema. En este caso, cualquier punto óptimo obtenido será función de los coeficientes usados para combinar los objetivos.

2.8.2 Programación de metas

En este método, el tomador de decisiones tiene que asignar objetivos o metas que desee alcanzar para cada objetivo [4]. Estos valores se incorporan en el problema como restricciones adicionales. La técnica tratará entonces de minimizar las desviaciones absolutas de cada objetivo con respecto a lo deseado. La forma más simple de este método puede formularse de la manera siguiente [1]:

$$\text{Minizar } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) - T_i$$

Sujeto a

$$R(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_n(\mathbf{x}))$$

Donde

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

$$F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))$$

$R(\mathbf{x})$ = Región factible

T_i = denota la meta u objetivo establecido para la i -ésima función objetivo.

El criterio es, entonces, minimizar la suma de los valores absolutos de las diferencias entre los valores deseados y los obtenidos.

Hay dos clases de problemas de programación por metas. No restrictiva en donde todas las metas son de importancia comparable. Restrictiva en donde existe una jerarquía de niveles de prioridades para las metas, de tal manera que las metas de

importancia primaria reciben atención de primera prioridad, las de segunda importancia reciben atención de segunda prioridad y así sucesivamente. Por defecto el término programación por metas se refiere a éste último concepto [4].

Análisis de la técnica

Esta técnica producirá una solución dominada si la meta seleccionada yace en la región factible [1]. La técnica puede ser muy eficiente (en términos de tiempo de CPU) si conocemos las metas que deseamos obtener y si éstas se encuentran en la zona factible. Sin embargo, el tomador de decisiones tiene la tarea de encontrar pesos o prioridades adecuadas para los objetivos que eliminen las características no conmensurables del problema, lo cual es difícil en la mayor parte de los casos, a menos que exista conocimiento previo acerca de la topología del espacio de búsqueda. Asimismo, si la región factible es difícil de localizar, este método puede volverse muy ineficiente.

Sin embargo, la técnica puede ser muy útil en casos en los que se pueden efectuar aproximaciones lineales parciales de las funciones objetivo, debido a la disponibilidad de excelentes programas para esa tarea y a la posibilidad de eliminar las metas dominadas fácilmente. Por otra parte, en los casos no lineales, otras técnicas resultarán generalmente más eficientes.

2.8.3 El método de las restricciones

Este método se basa en la minimización de una función objetivo y considera a los demás objetivos como restricciones que están acotadas por ciertos niveles permisibles [4]. Por lo tanto, se efectúa una minimización con un solo objetivo para la función objetivo más relevante f_1 sujeta a restricciones adicionales en las otras funciones objetivo. Se continúa optimizando tomando la segunda función objetivo en importancia para el tomador de decisiones [1].

Puede ser necesario repetir el proceso de elección por orden de importancia de la función objetivo para que se pueda llegar a la solución deseada.

Para obtener valores adecuados de las funciones objetivo, se efectúan normalmente optimizaciones individuales para cada una de las funciones objetivo en turno, usando técnicas de programación matemática.

Análisis de la técnica

Este método tiende a encontrar soluciones débilmente no dominadas, lo cual puede no ser muy adecuado en ciertas aplicaciones. Sin embargo, la simplicidad relativa de la técnica la ha hecho muy popular entre algunos investigadores [1].

2.9 ALGORITMO DE WAGNER - WHITIN

Este método generalmente es usado para el manejo de inventarios, su objetivo consiste en mantener un inventario mínimo. Es un modelo de programación dinámica, que busca determinar la forma más económica para satisfacer las demandas para cierto horizonte de planeación, encontrando las fechas en las cuales deben hacerse los pedidos.

Este algoritmo es modificado para ser utilizado como un método de solución del modelo de *MAXIMIZACIÓN DEL MARGEN DE CARGA* para múltiples recursos generando un **nuevo algoritmo de programación dinámica** que se explica en la sección 4.1.

Aplicándolo al modelo se tiene que el algoritmo de Wagner – Whitin se toman algunas variables de este y se modifican para el horizonte de planeación donde se requiere encontrar la estructura adyacente que maximice el margen de carga.

Toma como matriz base la obtenida con los márgenes de carga para cada período.

La relación del algoritmo de Wagner – Whitin con el método de optimización propuesto para la solución del modelo se presenta a continuación:

- La variable costo de procesamiento del período es cambiado, para el algoritmo de solución propuesto, por el Margen de Carga del período evaluado.
- La Fase = F_i , la cual representa el costo acumulado del procesamiento del período mas el costo unitario por la demanda del producto y el costo de almacenamiento. Esta variable es cambiada, en el algoritmo propuesto, por la variable P_i , la cual representa el margen acumulado de carga.
- La solución dominada, es decir la que genera menos costos de inventario, es llamado en el método de solución el vector $R(j)$, el cual da como resultado los márgenes de carga mas altos.
- A partir de la solución dominada se encuentra las fechas en el horizonte de planeación donde se realizan los pedidos que generen menor costo de inventario. En el algoritmo propuesto se encuentra la estructura adyacente donde se maximiza el margen de carga.
- Adicionalmente al modelo propuesto se penaliza con $-\infty$ cuando el período evaluado no cumple con la propiedad de adyacencia, esto ocurre para los casos donde las tareas pertenecen a más de 2 intervalos (ver 2.4).

2.10 SIMULATED ANNEALING

El algoritmo de simulated annealing tiene su origen en un proceso de Física y Metalurgia, conocido con el nombre de "recocido". Consiste en un proceso térmico para obtener estados de baja energía de un sólido en un baño caliente.

El proceso físico de recocido puede ser modelado satisfactoriamente usando métodos de simulación informática. En 1953, Metropolis, Rosenbluth y Teller introdujeron un algoritmo simple para simular la evolución de un sólido en un baño caliente hasta su equilibrio térmico este algoritmo es conocido como ALGORITMO METROPOLIS. El algoritmo ideado por estos autores se genera una secuencia de estados del sólido.

Si el decremento de la temperatura se realiza lo suficientemente lento, el sólido puede alcanzar el equilibrio térmico a cada temperatura. En el algoritmo Metropolis esto se consigue mediante la generación de un gran número de transiciones a cada valor de la temperatura.

El algoritmo de simulated annealing busca generar una secuencia de soluciones de un problema de optimización combinatoria. Para este propósito, asumimos una analogía entre un sistema físico de muchas partículas y un problema de optimización combinatoria, basándonos en las siguientes equivalencias:

- Las soluciones en un problema de optimización combinatoria son equivalentes a los estados de un sistema físico.
- El coste de una solución es equivalente a la energía de un estado.

Seguidamente, se introduce un parámetro que juega el papel de la temperatura. Este parámetro se denomina *parámetro de control*.

El algoritmo de simulated annealing puede ser visto como una repetición de algoritmos Metropolis, evaluados a valores decrecientes del parámetro de control. Como en los algoritmos de búsqueda local, se asume la existencia de una estructura de entornos y un mecanismo de generación. Se Introduce las siguientes definiciones:

Definición 1. Sea (S, f) un caso de un problema de optimización combinatoria, y sean i y j dos soluciones con costes $f(i)$ y $f(j)$ respectivamente. Entonces, el *criterio de aceptación* determina si j es aceptado a partir de i aplicando la siguiente *probabilidad de aceptación*:

$$P\{\text{aceptar } j\} = \begin{cases} 1 & \text{si } f(j) \leq f(i) \\ \exp\left(\frac{f(i) - f(j)}{T}\right) & \text{si } f(j) > f(i) \end{cases}$$

donde T es el parámetro de control.

Definición 2. Una *transición* es una acción combinada que resulta en la transformación de la solución actual en otra subsecuente. Dicha acción se compone de los siguientes dos pasos

1. Aplicación del mecanismo de generación.
2. Aplicación del criterio de aplicación.

Una característica típica del algoritmo de simulated annealing es que, aceptando siempre mejoras en el coste, también acepta eventualmente empeoramientos de la solución hacia costes peores. Inicialmente, a grandes valores de T , se aceptan grandes deterioros; a medida que T disminuye, solamente deterioros más pequeños van siendo aceptados, y finalmente, cuando T se aproxima a 0, ningún empeoramiento es ya aceptado. Esta singular característica hace que el algoritmo de simulated annealing, en contraste con los algoritmos de búsqueda local, pueda escapar de mínimos locales, exhibiendo sin embargo aún los rasgos favorables de dichos algoritmos de búsqueda local: simplicidad y aplicabilidad general.

La probabilidad de aceptar deterioros de la solución se implementa mediante la comparación del valor de $\exp((f(i)-f(j))/T)$ con un número aleatorio generado de una distribución uniforme en el intervalo $[0,1)$.

Parámetros de Simulated Annealing

- Valor Inicial de la Temperatura. Se fija un radio de aceptación tal que cuando el SA está en sus primeras iteraciones, la probabilidad de aceptación sea lo más cercana a 1.
- Se fija la Temperatura inicial.
- Inicia el Simulated Annealing, con el parámetro Temperatura inicializado a esa temperatura.
- Razón de Cambio de la Temperatura ¿Cuándo debería cambiar la Temperatura del Sistema? El sistema debe alcanzar un cuasi equilibrio, por lo tanto se necesita fijar un nuevo parámetro denominado Época o constante de concavidad de la curva. Se mantiene un contador del número de veces en que las transiciones son aceptadas, y es comparado con este parámetro época que mantiene un número fijo, llamado x, que dice que la Temperatura cambia (decrece) sólo si se aceptaron x transiciones. Cuando se alcanza este límite, el costo de la época será el valor de la energía en que estaba evaluada la última configuración aceptada y se procede a enfriar la Temperatura con alguna de las siguientes razones:

$$T_{k+1} = \alpha T_k$$

$$T_{k+1} = \alpha^k T_k$$

$$T_{k+1} = \frac{T_k}{\ln(k)}$$

- Vecindario Extendido En la práctica, la probabilidad de que la solución factible inicial generada por el algoritmo esté cercana al mínimo global es pequeña debido a la gran cantidad de posibilidades que implica un espacio de búsqueda altamente dimensional. Por lo tanto, se cree que en los primeros pasos del algoritmo, es decir a altas temperaturas, debe tenerse

un vecindario grande, es decir permitirle al algoritmo moverse libremente a través del espacio total de búsqueda. Con el progreso del algoritmo, la temperatura va decreciendo y la localización espacial de la configuración actual en ese momento debería ser cercana al mínimo global, por lo tanto el vecindario del SA debiera restringirse aún más.

- Condición de Parada La condición de parada se cumple cuando se alcanza la Temperatura Final o cuando luego de un número "n" de épocas la configuración que tiene mejor evaluación con respecto a la función objetivo no cambia por una nueva.

Para el problema se calcula el valor total de iteraciones por la siguiente ecuación:

$$N = \frac{\left[\frac{\ln T_f}{\ln T_o} \right]}{\ln \alpha}$$

3. PLANTEAMIENTO DEL MODELO

Para obtener un modelo donde el margen de carga se maximice, teniendo en cuenta el tipo de recurso utilizado y la cantidad consumida de cada recurso por cada tarea, se aplicará el método de optimización por suma de pesos expuesto en el numeral 2.8.1

Se parte de las ecuaciones para la solución de problemas con múltiples funciones objetivo. A cada una de las funciones objetivo se les asignará un peso según sea la importancia del tipo de recurso utilizado [RARDIN]. Los pesos son representados en el modelo como el costo por uso del recurso tipo k en la tarea i .

Inicialmente se presentan las ecuaciones que representan la carga máxima y la carga mínima para cada uno de los recursos tipo k que son utilizados en el plan de operaciones. Es importante aclarar que el margen de autonomía y por lo tanto las cargas máxima y mínima son diferentes según el tipo de recurso utilizado. Entonces, Las cargas máxima y mínima para el intervalo I comprendido entre $[T_j, T_{j+1}]$ son:

$$WMAX_I = \sum_{\omega_i \in \Omega_j} \sum_{k=1}^K [q_{ik} * \min(D_{ik}, F_i - T_j)] + \sum_{\omega_i \in Int_j} \sum_{k=1}^K (q_{ik} * D_{ik}) + \sum_{\omega_i \in \Omega_{j+1}} \sum_{k=1}^K [q_{ik} * \min(D_{ik}, T_{j+1} - C_i)]$$

$$WMIN_I = \sum_{\omega_i \in \Omega_j} \sum_{k=1}^K [q_{ik} * \max(0, C_i + D_{ik} - T_j)] + \sum_{\omega_i \in Int_j} \sum_{k=1}^K (q_{ik} * D_{ik}) + \sum_{\omega_i \in \Omega_{j+1}} \sum_{k=1}^K [q_{ik} * \max(0, T_{j+1} - (F_i - D_{ik}))]$$

q_{ik} = Cantidad del recurso tipo k que es consumido por la operación i

k = Tipo de recurso que esta consumiendo la operación i

$D_{ik} = D_i$ = Duración del consumo del recurso k por la tarea i es igual a la duración de la tarea i . Esto para todo k

K = Número total de recursos utilizados.

El modelo se presenta a continuación:

$$\text{MAX. MARGEN}(\varepsilon(\pi), \pi) = \sum_l^L \sum_{k=1}^K (C_k (WMAX_l - WMIN_l))$$

Sujeto a:

1. $T_j < T_{j+1}$ para $2 \leq j \leq M-1$
2. $C_i + T(1 - Z_{ij}) \geq T_{j-1}$
3. $F_i \leq T_{j+1} + T(1 - Z_{ij})$, para $1 \leq i \leq N$, $2 \leq j \leq M-1$
4. $T_1 = \min C_i \quad \omega_i \in \pi$
5. $T_M = \max F_i \quad \omega_i \in \pi$
6. $Z_{ij} \in \begin{cases} 1 & \text{si } T_{j-1} \leq C_i \leq F_i \leq T_{j+1} \\ 0 & \text{de lo contrario} \end{cases}$
7. $T =$ Un número grande, por ejemplo, $T = T_M + 1$

Donde:

L = Número total de intervalos

K = Número total de recursos utilizados.

Partiendo de este modelo se encuentra una solución para el plan de operaciones en los casos donde estos necesitan tanto el uso de 1 o más tipos de recursos.

4. ALGORITMOS DE LOS MÉTODOS DE SOLUCIÓN PROPUESTOS

Los algoritmos utilizados para la solución de modelo propuesto, y aceptando como válida la Propiedad de adyacencia, son:

4.1 ALGORITMO DE PROGRAMACIÓN DINÁMICA (BASADO EN EL ALGORITMO DE WAGNER – WHITIN):

Este permite encontrar un plan de operaciones que maximiza el Margen de Carga (ver secciones 2.9 y 3) para referenciales posicionados en el conjunto de los enteros positivos, siguiendo el siguiente algoritmo de programación dinámica.

1. Se define la variable $M_{j,n}$ que es el margen de autonomía del plan de operaciones, tomando a partir de aquí el intervalo I como el intervalo comprendido entre $[T_j, T_n]$, si éste es factible y $M_{j,n}$ toma el valor de $-\infty$ cuando el intervalo es no factible (esto ocurre para los casos donde las tareas pertenecen a más de 2 intervalos).

$$M_{j,n} = \begin{cases} \sum_{k=1}^K (C_k (WMAX_{l,k} - WMIN_{l,k})) & \text{si el intervalo } (T_j, T_n) \text{ es factible} \\ -\infty & \text{de lo contrario} \end{cases}$$

$$T_1 \leq T_j < T_n \leq T_M$$

2. Sea P_I el máximo margen de autonomía posible entre los referenciales T_1 y T_j .

P_1 se define como:

$$P_1 = 0$$

$$P_2 = P_1 + M_{12}$$

$$P_3 = \text{Max} \begin{cases} P_1 + M_{13} \\ P_2 + M_{23} \end{cases}$$

$$P_4 = \text{Max} \begin{cases} P_1 + M_{14} \\ P_2 + M_{24} \\ P_3 + M_{34} \end{cases}$$

En general:

$$P_l = \{ \text{Max}(P_j + M_{j,n}) \}$$

3. Sea j^* la posición j para la cual $P_l + M_{j,n}$ es máximo.
4. A partir de j^* se obtiene el vector solución $E(\pi)$. Éste vector es subconjunto de todos los referenciales posibles del horizonte de planeación y muestra la posición de los referenciales donde se maximiza el margen de carga del plan de operaciones.

A continuación se ilustra el algoritmo de programación dinámica en el siguiente ejemplo:

Se tiene el siguiente plan de operaciones para una fábrica de manufactura donde cada operación requiere el uso de un solo recurso y cuyo costo es igual a 1

i	Di	Ci	Fi
1	1	1	3
2	1	2	4
3	1	4	6
4	1	5	7
5	2	3	6

Tabla 4.1

Gráficamente se representa de la siguiente manera:

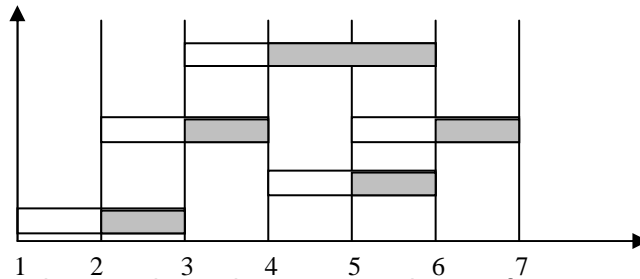


Figura 4.1. Plan de operaciones de un proceso de manufactura con todos los referenciales posibles

Se calcula la matriz $M_{j,n}$ de la siguiente forma:

$M_{j,n}$	2	3	4	5	6	7
1	1	1+1=2	1+3=4	2+5=7	1+8=9	0+11=11
2		2+1=3	2+3=5	3+5=8	2+8=10	1+11=12
3			2+3=5	3+5=8	2+8=10	1+11=12
4				$-\infty$	2+8=10	1+11=12
5					3+8=11	1+11=12
6						1+11=12
P	1	3	5	8	11	12
J*	1	2	3	3	5	6

Tabla 4.2

Para encontrar el vector solución se parte del último referencial ubicado en la primera fila y la última columna, para este ejemplo el referencial T7 y se lee la posición j^* que lo acompaña en la misma columna en este caso el referencial T6. Luego el referencial encontrado se ubica en la primera fila y se lee el referencial j^* que lo acompaña en este caso T5, se continúa haciendo de esta manera hasta llegar a la posición T1. El vector solución $E(\pi)$ se halla organizando estos referenciales en forma ascendente.

Para este ejemplo se tiene que el vector solución donde se maximiza el margen es:

$$E(\pi)=[1,2,3, 5,6,7] \text{ y}$$

Margen de carga =12

Gráficamente se representa de la siguiente manera:

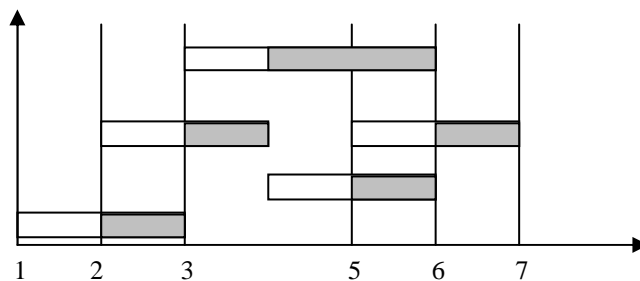


Figura 4.2. Estructura adyacente para el plan de operaciones propuesto

4.2 ALGORITMO DE SIMULATED ANNEALING

Con el algoritmo de Simulted Annealing se busca encontrar un mejor horizonte de planeación que el encontrado con el algoritmo de Wagner – Whitin. Con este algoritmo se encuentra un referencial modificado que mejora el Margen de Carga y no necesariamente la nueva posición del referencial este en una posición entera.

Definiciones:

- $E(\pi)$ = Vector solución actual, es vector resultado por el algoritmo de Wagner – Whitin.

- L = tamaño del vector $E(\pi)$.
- $E(\pi)^*$ = Vector modificado.
- W = Margen de autonomía de la solución $E(\pi)$.
- W^* = Margen de autonomía de la solución $E(\pi)^*$.
- T_0 y T_f = Temperatura inicial y final en el proceso de enfriamiento.
- N = Número máximo de iteraciones por nivel de temperatura.
- n = Número de iteraciones por nivel de temperatura.
- T = Temperatura.
- Rnd = Variable aleatoria uniforme continua de dominio $[0,1)$.

Algoritmo:

1. $E(\pi)^* = E(\pi)$, $W^* = W$, $T = T_0$
2. $N = 1$
3. Generar Posición
4. Generar referencial
5. $E(\pi)$ posición = Referencial
6. Evaluar factibilidad de la solución generada $E(\pi)$.
7. Si $E(\pi)$ es factible calcular W
8. Calcular P (ver 2.9)
Generar **rnd**
Si $rnd \leq P$ entonces $E(\pi)^* = E(\pi)$
 $W^* = W$
Fin si
Fin si.
9. $N = n+1$
10. Si $n \leq N$ entonces regresar al paso 3
Si no disminuir T .
Fin si
11. Si $T > T_f$.

Fin si

12. Fin

4.3 DESARROLLO DEL PROGRAMA PARA LA SOLUCIÓN DE LOS ALGORITMOS

Para la solución al algoritmo de Wagner – Whitin se plantea la solución suponiendo que la descomposición del horizonte de fabricación $E(\pi)$ pertenece al conjunto de Z^+ . Aplicando el algoritmo (ver sección 4.1) se encuentra una solución probable del plan de operaciones. El algoritmo de SA parte del resultado dado por el método de optimización modificado de Wagner – Whitin.

4.3.1 Explicación

El programa fue realizado en **JAVA** y consta de 5 clases o partes:

1. **Principal:** en esta clase se manejan los procesos de mayor relevancia al interior del programa.
2. **Intervalos:** en esta clase se maneja una matriz, la cual posee el margen de carga de cada intervalo para un recurso dado.
3. **Recurso:** en esta clase se maneja el recurso con su respectivo costo.
4. **Referente:** en esta clase se maneja cada intervalo con cada una de sus características: inicio, fin, duración y carga.
5. **Tarea:** en esta clase se manejan los aspectos de cada una de las tareas como son inicio, final, duración y cantidad de recursos necesarios por cada tipo.

También se crearon objetos del tipo Vector los cuales contienen las diferentes instancias de cada clase, por ejemplo, se tiene un Vector tareas, el cual contiene las diferentes instancias de la clase Tarea.

El programa consta en una primera parte de los siguientes pasos:

- 1) Se ingresa el tiempo máximo con el cual se va a trabajar en el ejemplo.
- 2) Se ingresa el número de tareas y el número de recursos necesario.
- 3) Se ingresa el costo de cada uno de los recursos.
- 4) Se ingresan cada una de las características de las tareas como son inicio, fin, duración y cantidad de recursos necesarios por cada tipo.
- 5) En otra función se calcula la carga máxima y mínima para cada intervalo. Con esta información se calcula el margen, teniendo en cuenta el costo del recurso al cual pertenece la matriz de intervalos. Así se hace con cada matriz para cada uno de los recursos.
- 6) Otra función llamada “sumarMatricesRecursos” se encarga de sumar todas las matrices de intervalos, generando así una matriz global llamada “matrizTotal”, la cual contiene el margen de carga total para cada intervalo.
- 7) En esta parte se desarrolla el algoritmo de Wagner-Whitin.
- 8) Ahora en una nueva matriz llamada: “matrizResultados” se almacena la máxima carga de cada columna de la “matrizTotal” y la fila en la cual se encuentra.
- 9) El resultado de la función objetivo se obtiene mirando la última posición de la “matrizResultados”.

En la segunda parte del programa se calcula una nueva función objetivo a partir de ciertos cambios que se efectúan a los resultados obtenidos en la primera parte.

Se trabajará con alfa igual a 0.97.

Esta segunda parte se puede resumir en los siguientes pasos:

- 1) Se ingresan las temperaturas inicial y final.
- 2) Se halla el número de iteraciones con el cual se trabajará a partir de la siguiente fórmula:

$$\text{numerolt} = \ln(\text{temperaturafinal}/\text{temperaturalnicial})/\ln(\text{alfa})$$

- 3) Se halla el horizonte de planeación y se almacena.
- 4) Se selecciona un referente con el cual no se desea trabajar y se elimina del arreglo.
- 5) Ahora se selecciona un nuevo referente, el cual se desplazará cada vez más hacia la derecha, dependiendo del número de iteraciones.
- 6) Dependiendo de la posición del referente que se desee modificar y del arreglo se crean objetos de la clase Referente, los cuales, en otras palabras se comportan como intervalos, con inicio, fin, duración y carga.
- 7) Con los nuevos se repite básicamente el mismo proceso de la primera parte del programa para encontrar la función objetivo, con la diferencia de que ahora se trabaja con intervalos definidos, y que no es necesario usar el algoritmo de Wagner-Whitin, ya que no se trabaja con matrices, sino con estructuras dinámicas.
- 8) Los pasos 6 y 7 se repiten tantas veces como número de iteraciones haya, ya que el referencial que se escogió cambia de posición.
- 9) Los diferentes resultados obtenidos son comparados y en cada iteración se les aplica el algoritmo de Simulated Annealing.
- 10) Se muestran los resultados finales por pantalla.

Para el cálculo de la carga máxima y la carga mínima el programa tiene en cuenta la relación entre las fechas de comienzo a más temprano y fin a más tarde con respecto a los referenciales, tomando como casos posibles las siguientes situaciones de las tareas con respecto a los intervalos evaluados.

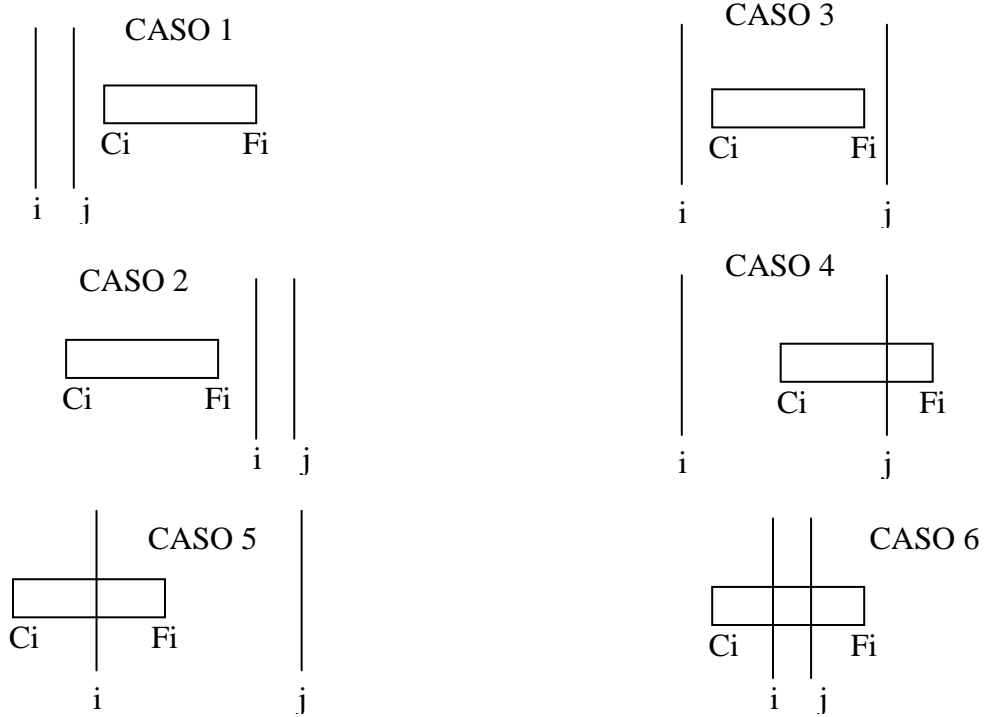


Figura 4.3. Relación de las tareas respecto a la posición de los referenciales, para el calculo de las cargas máxima y mínima.

La evaluación para cada caso es la siguiente:

Caso 1 = $C_i > i, C_i \geq j, F_i > i, F_i > j$.

Caso 2 = $C_i < i, C_i < j, F_i \leq i, F_i < j$.

Caso 3 = $C_i \geq i, C_i < j, F_i > i, F_i \leq j$.

Caso 4 = $C_i \geq i, C_i < j, F_i > i, F_i > j$.

Caso 5 = $C_i < i, C_i < j, F_i > i, F_i \leq j$.

Caso 6 = $C_i < i, F_i > j$.

De esta manera se calcularon la carga para cada uno de los intervalos posibles y se penalizo con un valor de $-\infty$ los intervalos donde ocurra el caso número 6. En el programa desarrollado en Java el valor menos infinito es evaluado dándole el valor de -10.000.

Para hacer una aplicación de los métodos de optimización propuestos para resolver la función objetivo se presentan a continuación 3 ejemplos de diferentes niveles de complejidad. Cada uno de ellos presenta el caso donde es necesario el uso de múltiples recursos para el desarrollo de cada una de las tareas, es bueno anotar que este programa también puede se utilizado cuando el plan de operaciones solo requiere el uso de un solo tipo de recurso.

5. EJEMPLOS APLICADOS AL MODELO

- a. Se tiene el siguiente plan de operaciones, determinar la descomposición del horizonte temporal donde se tenga el máximo margen de carga:

i	Ci	Fi	Di	qi1	qi2
1	2	7	3	2	1
2	4	14	5	0	1
3	11	18	6	2	1
4	8	20	7	0	1

Tabla 4.1

q_{i1} = El costo por el uso del recurso 1 para el desarrollo de cada tarea es = 200.

q_{i2} = El costo por el uso del recurso 2 para el desarrollo de cada tarea es = 137.

Utilizando como método de optimización de Wagner-Whitin se obtiene la siguiente matriz M solución.

Mi,j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	0	0	537	1074	1211	811	411	548	822	822	822	1359	1359	1222	1222	1085	948	274	137	3381	
2			537	1074	1211	811	411	548	822	822	822	1359	1359	1222	1222	1085	948	274	137	0	
3				E	E	E	1485	1622	1896	1896	1896	2433	2433	2296	2296	2159	2022	1348	1211	1074	
4					E	E	2559	2696	2970	2970	2970	3507	3507	3370	3370	3233	3096	2422	2285	2148	
5						E	E	E	E	E	E	E	E	3644	3644	3507	3370	2696	2559	2422	
6							E	E	E	E	E	E	E	2844	2844	2707	2570	1896	1759	1622	
7								E	E	E	E	E	E	4192	4192	4055	3918	3244	3107	2970	
s									E	E	E	E	E	4466	4466	4329	4192	3518	3381	3244	
9										E	E	E	E	E	E	E	E	E	E	E	3792
10											E	E	E	E	E	E	E	E	E	E	3792
11												E	E	E	E	E	E	E	E	E	3792
12													E	E	E	E	E	E	E	E	4866
13														E	E	E	E	E	E	E	4866
14															E	E	E	E	E	E	5688
15																E	E	E	E	E	5688
16																	E	E	E	E	5414
17																		E	E	E	5140
18																			E	E	3792
19																					3518
P	0	0	537	1074	1211	811	2559	2696	2970	2970	2970	3507	3507	4466	4466	4329	4192	3518	3381	3244	5688
J*	1	1	1	1	1	1	4	4	4	4	4	4	4	8	8	8	8	8	8	8	14

Tabla 4.2

A partir de la matriz **M** se calculan los vectores P y J* para este ejemplo se obtiene como descomposición del horizonte de planeación para el valor máximo del margen de autonomía es:

$$E(\pi) = [1, 4, 8, 14, 20].$$

Con un margen de carga máximo de **5688**, y con un número total de referenciales de **5**.

La solución propuesta anteriormente para descomponer el horizonte de fabricación genera el máximo número de intervalos posibles para el plan de operaciones dado ubicados en posiciones enteras de tiempo en el horizonte de planeación.

El Diagrama de la solución se ilustra a continuación:

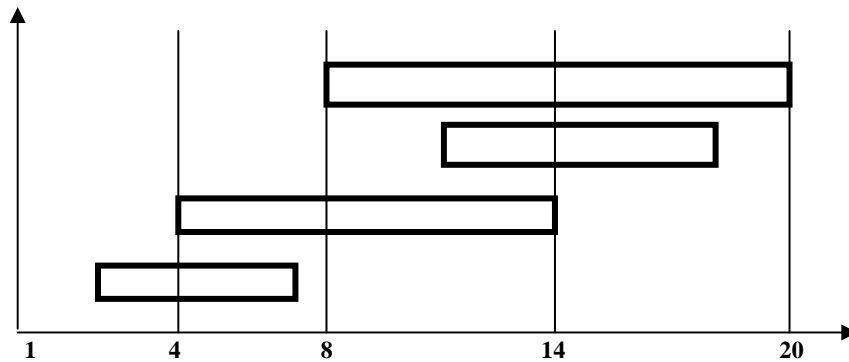


Figura 4.2. Diagrama de Gantt para el ejemplo 1.

Solución por SA:

Para los tres ejemplos propuestos se tomo como parámetros de evaluación los siguientes:

Temperatura inicial = 100000

Temperatura final = 1

$\alpha = 0,97$

Para solucionar problema aplicado el algoritmo de SA el programa pregunta cual referente desea modificar. Para el ejemplo 1 se modifica el referente 14 con el cual se obtuvo el siguiente resultado:

```
C:\angela>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
4
8
14
20
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5688.0
REFERENTE MODIFICADO: 5646.9
REFERENTE MODIFICADO: 5605.7999999999999
REFERENTE MODIFICADO: 5564.7000000000001
REFERENTE MODIFICADO: 5523.6
REFERENTE MODIFICADO: 5482.5
REFERENTE MODIFICADO: 5441.40000000000015
REFERENTE MODIFICADO: 5400.2999999999999
REFERENTE MODIFICADO: 5359.2000000000001
REFERENTE MODIFICADO: 5318.1
REFERENTE MODIFICADO: 5277.0
REFERENTE MODIFICADO: 5235.9000000000001
REFERENTE MODIFICADO: 5194.7999999999999
REFERENTE MODIFICADO: 5153.7000000000001
REFERENTE MODIFICADO: 5005.2000000000003
REFERENTE MODIFICADO: 4803.0
REFERENTE MODIFICADO: 4600.8000000000002
REFERENTE MODIFICADO: 4398.6000000000004
REFERENTE MODIFICADO: 4196.40000000000015
REFERENTE MODIFICADO: 3994.20000000000025
REFERENTE MODIFICADO: 3792.0
REFERENTE MODIFICADO: 3750.9000000000005
REFERENTE MODIFICADO: 3709.8000000000001
REFERENTE MODIFICADO: 3668.7000000000003
REFERENTE MODIFICADO: 3627.5999999999995
REFERENTE MODIFICADO: 3586.5
REFERENTE MODIFICADO: 3545.4000000000005
REFERENTE MODIFICADO: 3504.2999999999997
REFERENTE MODIFICADO: 3463.1999999999999
REFERENTE MODIFICADO: 3422.0999999999995
REFERENTE MODIFICADO: 3381.0
REFERENTE MODIFICADO: 3339.8999999999996
REFERENTE MODIFICADO: 3298.7999999999993
REFERENTE MODIFICADO: 3257.6999999999999
iteraciones: 377.9784970029197
Funcion objetivo FINAL: 5688.0
```

Se observa claramente que el margen de carga máximo es igual al obtenido por el método de programación dinámica. Esto indica que el referencial 14 sufre de ninguna modificación.

Buscando implementar una buena solución, utilizando el algoritmo del SA, se restringió la cantidad de referenciales para el plan de operaciones en un referencial menos de los obtenidos por el método de programación dinámica. Esto se hizo para los tres ejemplos propuestos. El resultado fue el siguiente:

Basándose en la figura 4.2 para el ejemplo se cancela el referencial número 14 y modificando el referencial 8 se obtuvo el siguiente resultado:

```

C:\> Símbolo del sistema - java Principal
C:\programaref>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
4
8
14
20
¡NUEVO CAMINO!
1
4
8
20
REFERENTE MODIFICADO: 3244.0
Posicion final del referencial modificado: 12.049999999999999
Funcion objetivo FINAL: 4866.0
    
```

Después de retirar el referencial 14 se obtiene como margen de carga para el ejemplo:

$$E(\pi) = [1, 4, 8, 20].$$

Con un margen de carga de **3244**.

Aunque el margen de carga aumenta de **3244**, sin tener el referencial 14, a **4866** solucionándolo con el algoritmo de SA, comparado con el horizonte de planeación que contiene todos los referenciales disminuye el margen de carga. El programa

encuentra la posición nueva del referencial modificado. Esta posición busca suplir la ausencia del referencial eliminado ubicándolo en la posición **12,049**.

- b. El segundo ejemplo que ilustra la aplicación donde el proceso tiene múltiples tareas, y requiere el uso de múltiples recursos es:

i	C_i	F_i	D_i	q_{i1}	q_{i2}	q_{i3}
1	1	6	3	2	0	1
2	3	10	4	1	2	1
3	4	11	5	2	1	0
4	6	10	2	2	2	0
5	9	14	3	2	1	1
6	12	15	2	1	0	1
7	7	13	4	0	1	1
8	8	14	3	1	1	0
9	11	15	2	2	0	1
10	13	17	3	1	2	1

Tabla 4.3.

q_{i1} = El costo por el uso del recurso 1 para el desarrollo de cada tarea es = 80.

q_{i2} = El costo por el uso del recurso 2 para el desarrollo de cada tarea es = 170.

q_{i3} = El costo por el uso del recurso 3 para el desarrollo de cada tarea es = 375.

Utilizando como método de optimización de Wagner-Whitin se obtiene la siguiente matriz M solución.

Mi,j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	535	1070	1865	2455	3045	3545	3795	3295	2625	3250	2990	2480	1785	795	795	16525
2			E	E	E	4115	4615	4865	4365	3695	4320	4060	3550	2855	1865	1865	1070
3				E	E	5185	5685	5935	5435	4765	5390	5130	4620	3925	2935	2935	2140
4					E	E	E	E	E	6355	6980	6720	6210	5515	4525	4525	3730
5						E	E	E	E	E	8160	7900	7390	6695	5705	5705	4910
6							E	E	E	E	11480	11220	10710	10015	9025	9025	8230
7								E	E	E	12480	12220	11710	11015	10025	10025	9230
8									E	E	E	E	12210	11515	10525	10525	9730
9										E	E	E	E	10515	9525	9525	8730
10											E	E	E	10765	9775	9775	8980
11												E	E	17515	16525	16525	15730
12													E	E	16005	16005	15210
13														E	15485	15485	14690
14															E	E	19300
15																E	17320
16																	17320
P	0	535	1070	1865	2455	5185	5685	5935	5435	6355	12480	12220	12210	17515	16525	16525	19300
J*	1	1	1	1	1	3	3	3	3	4	7	7	8	11	11	11	14

Tabla 4.4.

A partir de la matriz M se calculan los vectores P y J^* para este ejemplo se obtiene como descomposición del horizonte de planeación para el valor máximo del margen de autonomía es:

$$E(\pi) = [1, 3, 7, 11, 14, 17].$$

Con un margen de carga máximo de **19300**, y con un número de referenciales de **6**.

Diagrama de Gantt para el ejemplo se ilustra a continuación:

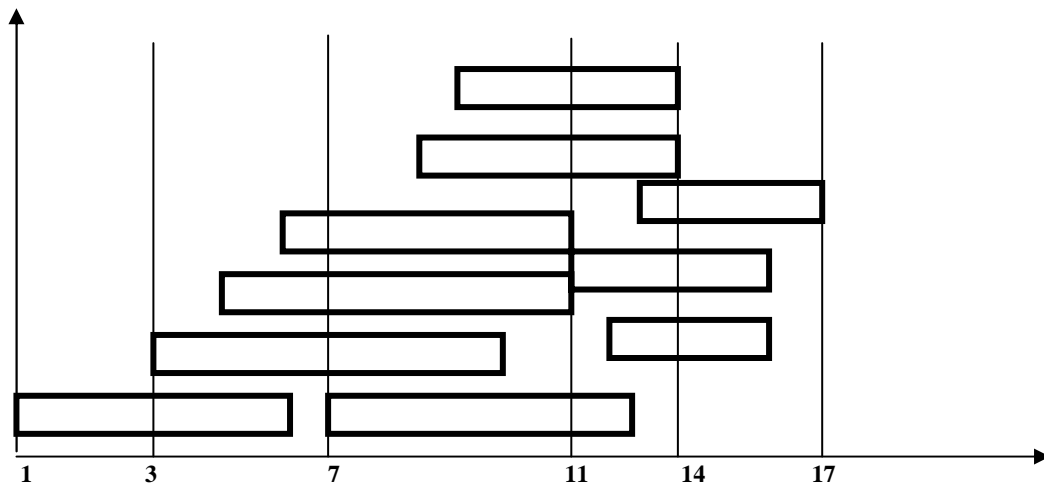


Figura 4.3. Diagrama de Gantt para el ejemplo 2.

Solución por SA:

Para la solución con la heurística de SA, se modifico el referente 14, el resultado fue el siguiente

```
C:\angela>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
3
7
11
14
17
REFRENTE MODIFICADO: 18805.0
REFRENTE MODIFICADO: 18508.0
REFRENTE MODIFICADO: 18211.0
REFRENTE MODIFICADO: 17914.0
REFRENTE MODIFICADO: 17617.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17320.0
REFRENTE MODIFICADO: 17240.5
REFRENTE MODIFICADO: 17002.0
REFRENTE MODIFICADO: 16763.5
REFRENTE MODIFICADO: 16525.0
REFRENTE MODIFICADO: 16286.500000000002
REFRENTE MODIFICADO: 16048.0
REFRENTE MODIFICADO: 15809.5
iteraciones: 377.9784970029197
Funcion objetivo FINAL: 19300.0
```

Se obtiene el mismo margen de carga, que el obtenido por el método de programación dinámica si solamente se modifica uno de los referenciales, por lo tanto tienen la misma estructura adyacente que la obtenida por el método de Programación dinámica. Retirando uno de los referenciales, basándose en la figura 4.3, para el ejemplo 2 se cancela el referencial 11 y se modifica el 7. Se obtienen los siguientes resultados:

```
CA Símbolo del sistema - java Principal
C:\programaref>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
3
7
11
14
17
¿NUEVO CAMINO?
1
3
7
14
17
REFERENTE MODIFICADO: 12800.0
```

```
Posicion final del referencial modificado: 7.9
Funcion objetivo FINAL: 13250.0
```

La nueva estructura adyacente estaría formada de la siguiente manera:

$$E(\pi) = [1, 3, 7, 14, 17].$$

Con un margen de carga de **12800**.

De manera similar el margen de carga aumenta de **12800**, sin tener el referencial 11, a **13250** solucionándolo con el algoritmo de SA, comparado con el horizonte de planeación que contiene todos los referenciales disminuye el margen de carga. La posición nueva posición del referencial es **7,9**.

- c. El tercer ejemplo donde se ilustra la aplicación del modelo planteado para procesos que requieren el uso de múltiples recursos es:

i	Ci	Fi	Di	qi1	qi2	qi3	qi4	qi5
1	1	5	3	3	2	0	0	1
2	3	7	2	1	2	1	0	0
3	5	9	1	2	0	2	0	0
4	6	10	3	3	2	2	0	1
5	9	14	4	2	1	2	1	1
6	12	14	1	1	0	1	1	1
7	15	20	3	3	0	2	1	0
8	18	26	5	3	0	1	0	0
9	20	27	2	2	2	1	0	0
10	27	30	2	3	1	1	0	1
11	26	33	3	1	2	2	1	1
12	27	32	3	3	2	0	0	2
13	30	40	6	1	2	1	0	2
14	34	40	5	3	1	1	1	2
15	38	44	3	3	2	2	0	2

Tabla 4.5.

Costo q_{i1} = 120

Costo q_{i2} = 232

Costo q_{i3} = 300

Costo q_{i4} = 410

Costo q_{i5} = 259

Para este ejemplo se presenta los vectores solución P y j*.

P	0	1083	1083	1967	3934	3890	8225	8225	7385	12489	12489	12489	13578	16408	16408
j*	1	1	1	1	2	2	5	5	5	7	7	7	7	13	13

17778	19148	19148	18438	23208	24872	25876	25876	25216	24556	25532	29864	32357	35552	36254
13	13	13	13	17	17	17	17	17	17	20	22	22	22	22

36314	39507	46850	48252	50072	50072	48670	47268	47808	61384	63326	61384	59442	69152
22	28	30	30	30	30	30	30	30	35	35	35	35	41

Tabla 4.6.

A partir de la matriz M se calculan los vectores P y J* para este ejemplo se obtiene como descomposición del horizonte de planeación para el valor máximo del margen de autonomía es:

$$E(\pi) = [1, 2, 5, 7, 13, 17, 22, 30, 35, 41, 44].$$

Con un margen de carga máximo de **69152**, y con un número de referenciales de **11**.

Solución por SA:

Para la solución con la heurística de SA, se modifico el referente 13, el resultado fue el siguiente

```
C:\angela>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
2
5
7
13
17
22
30
35
41
44
REFRENTE MODIFICADO: 67737.0
REFRENTE MODIFICADO: 66888.0
REFRENTE MODIFICADO: 66039.0
REFRENTE MODIFICADO: 65190.0
REFRENTE MODIFICADO: 64341.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 63492.0
REFRENTE MODIFICADO: 30820.5
REFRENTE MODIFICADO: 31026.0
REFRENTE MODIFICADO: 31231.5
REFRENTE MODIFICADO: 31437.0
REFRENTE MODIFICADO: 31642.5
REFRENTE MODIFICADO: 31848.0
REFRENTE MODIFICADO: 32053.5
REFRENTE MODIFICADO: 32258.999999999996
REFRENTE MODIFICADO: 32464.499999999996
REFRENTE MODIFICADO: 32670.0
REFRENTE MODIFICADO: 32875.5
REFRENTE MODIFICADO: 33081.0
REFRENTE MODIFICADO: 33286.5
REFRENTE MODIFICADO: 33492.0
iteraciones: 377.9784970029197
Funcion objetivo FINAL: 69152.0
```

Se obtiene el mismo margen de carga, que el obtenido por el método de programación dinámica si solamente se modifica uno de los referenciales, por lo tanto tienen la misma estructura adyacente que la obtenida por el método de Programación dinámica. Retirando uno de los referenciales, de manera aleatoria, para el ejemplo 3 se cancela el referencial 22 y se modifica el 7. Se obtienen los siguientes resultados:

```
C:\> Símbolo del sistema - java Principal
C:\programaref>java Principal
numeroIt: -11.512925464970229
epsilon: -0.030459207484708574
1
2
5
7
13
17
22
30
35
41
44
¡NUEVO CAMINO!
1
2
5
7
13
17
30
35
41
44
REFERENTE MODIFICADO: 61176.0
```

```
Posicion final del referencial modificado: 7.0
Funcion objetivo FINAL: 61176.0
```

Para este ejemplo el margen de carga se disminuyo del obtenido por el método de programación dinámica donde tiene todos los referenciales a cuando se le es retirado el referencial 22. Pasa de 69152 a 61176.

Después de modificar el referencial 7 aplicando el método de SA, no se obtiene ninguna mejoría en el margen de carga y varía la posición de este referencial. Esto puede deberse a que el referencial retirado y el referencial modificado se escogieron de manera aleatoria sin hacer un previo estudio del diagrama de gantt para el horizonte de planeación.

6. ANÁLISIS DE RESULTADOS

Para el algoritmo de programación dinámica se obtiene como resultado el número máximo de referenciales posibles para maximizar el Margen de Carga. Estos referenciales se encuentran posicionados en los enteros positivos del horizonte de planeación.

Aplicando el algoritmo de Simulated Annealing al resultado obtenido a través de la programación dinámica y dándole como opción de modificación uno de los referenciales respuesta no se evidencia ninguna mejoría.

Si uno de los referenciales es retirado se disminuye el margen de carga para todos los ejemplos evaluados:

5.1 RESULTADOS DE LOS EJEMPLOS

Ejemplo 1:

$E(\pi) = [1, 4, 8, 14, 20]$.

Margen de carga máximo inicial = **5688**.

Referencial cancelado = 14

$E(\pi)^* = [1, 4, 8, 20]$.

Margen de carga sin el referencial = **3244**.

Referencial a modificar = 8

Margen de carga modificando el referencial = **4866**.

Nueva posición del referencial = 12,049

Ejemplo 2:

$E(\pi) = [1, 3, 7, 11, 14, 17]$.

Margen de carga máximo inicial = **19300**.

Referencial cancelado = 11

$E(\pi)^* = [1, 3, 7, 14, 17]$.

Margen de carga sin el referencial = **12800**.

Referencial a modificar = 7

Margen de carga modificando el referencial = **13250**.

Nueva posición del referencial = 7,9

Ejemplo 3:

$E(\pi) = [1, 2, 5, 7, 13, 17, 22, 30, 35, 41, 44]$.

Margen de carga máximo inicial = **69152**.

Referencial cancelado = 22

$E(\pi)^* = [1, 2, 5, 7, 13, 17, 30, 35, 41, 44]$.

Margen de carga sin el referencial = **61176**.

Referencial a modificar = 13

Margen de carga modificando el referencial = **61176**.

Nueva posición del referencial = 13

Tabla comparativa de resultados:

RESULTADOS	EJEMPLO 1	EJEMPLO 2	EJEMPLO 3
Margen de carga máximo inicial	5688.	19300.	69152.
Referencial cancelado	14	11	22
Margen de carga sin el referencial	3244.	12800.	61176.
% de disminución de la carga	42,97	33,68	11,53
Referencial a modificar	8	7	13
Margen de carga modificando el referencial	4866.	13250.	61176.
% de aumento de la carga aplicando S.A	50,00	3,52	0,00
Nueva posición del referencial	12,049	7,9	13

Tabla 6.1

5.2 ANÁLISIS DE RESULTADOS

Es evidente que el Margen de Carga disminuye si es retirado alguno de los referenciales obtenidos en el horizonte de planeación.

Para el ejemplo 1 la disminución es del 42,96%; para el ejemplo 2 la disminución del margen de carga fue del 33,68% y para el ejemplo 3 fue del 11,53%.

Aplicando el algoritmo de Simulated Annealing y haciendo un análisis preliminar en el diagrama de Gantt para los ejemplos 1 y 2 se escogieron los referenciales a cancelar y los referenciales a modificar o alterar. En ellos el Margen de Carga aumento para el ejemplo 1 en un 50% y para el ejemplo 2 mejoró en un 3,5%. Para el ejemplo 3 se escogió de manera aleatoria los referenciales a cancelar y modificar y no se presentó ninguna aumento en el Margen de Carga.

El algoritmo de SA disminuye la eficiencia en los ejemplos planteados cuando el número de tareas aumenta a lo largo del horizonte de planeación. Así mismo el porcentaje de disminución del margen de carga también disminuye a medida que el número de tareas aumenta en el horizonte de planeación.

7. CONCLUSIONES

- El modelo propuesto para la solución del horizonte de planeación en el caso donde se requieren múltiples tareas y múltiples recursos se basa en un método de solución para funciones que poseen múltiples objetivos, el método aplicado es el de Suma de Pesos. Para próximos trabajos puede pensarse en aplicar otros métodos de solución para múltiples objetivos y generar diferentes funciones para evaluar y comparar resultados.
- Dependiendo del costo o peso que se le de a cada recurso variará el margen de carga para el plan de operaciones planteado, por este motivo se debe conocer muy bien el proceso para tener obtener resultados que se acerquen mas a la realidad del problema.
- El algoritmo de solución por medio de Programación Dinámica da los mejores resultados para el Margen de Carga. Este algoritmo no tiene restricciones en el número de referenciales que obtenga el problema para un Horizonte de Planeación dado.
- El algoritmo de Simulated Annealing es efectivo para problemas donde se tiene una limitación en el número de referenciales, sin embargo su desempeño depende en gran medida del plan de operaciones y que tan bien se conozca el problema.
- Los Horizontes de Fabricación que tienen menor número de tareas favorecen la consecución de mejoras en el Margen de Carga aplicando el

método de optimización de Simulated Annealing, para Horizontes donde se tiene una limitación en el número de referenciales.

- El empeoramiento del Margen de Carga al retirar uno de los referenciales obtenidos por el método de programación dinámica disminuye a medida que el número de tareas aumenta en el Horizonte de Planeación para los ejemplos evaluados.

8. BIBLIOGRAFÍA

1. **COELLO, C. A.** Optimización evolutiva con objetivos múltiples: Estado del arte y tendencias futuras. Col Isleta. <http://mailweb.udlap.mx/>
2. **NAHMIA, S.** Análisis de la producción de las operaciones. México: Compañía Editorial Continental, c1999.
3. **SÁNCHEZ, C. Y.** Programación óptima de tareas sobre máquinas idénticas, por medio de algoritmos genéticos. Universidad de los Andes, 2001.
4. **SARDI D. L.** Algoritmos Genéticos y Optimización en la Investigación de Operaciones. Caso: Problema de Optimización con Objetivos Múltiples Expresado como un Modelo de Programación No Lineal por Metas. Departamento de Computación, Facultad de Ciencias y Tecnología, Universidad de Carabobo, Valencia, Venezuela.
5. **RONALD R.** Optimization in Operations Research. Prentice Hall, Inc. 1998.
6. **TORRES, J. F.** Descomposición robusta de horizontes temporales en planificación de la producción. Universidad de los Andes, Departamento de Ingeniería Industrial.
7. **TORRES, J. F.** Regulación robusta de cargas de trabajo en planificación de la producción. Universidad de los Andes.

8. **TORRES R.** Problema de Secuenciamiento y Extracción usando Simulated Annealing. <http://www.inf.utfsm.cl/~romina/INVESTIGACION/IAAA/iaaa.pdf>.
16 de agosto de 2002

9. **VÉLEZ, M. C.** Descomposición robusta de fabricación en problemas de regulación agregada de cargas. Universidad de los Andes, 1999.