

**IMPLEMENTACION DE UNA TECNICA HEURISTICA PARA MINIMIZAR EL
MAKESPAN EN PROBLEMAS DE JOB SHOP CON RESTRICCIONES DE NO-WAIT
(J_m /no – wait/ C_{MAX})**

FELIPE GONZÁLEZ ARISTIZÁBAL

**Trabajo de grado para Optar al
Titulo de Magíster en Ingeniería Industrial**

**Asesor:
GONZALO MEJÍA DELGADILLO, Ph.D.**

**UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA INDUSTRIAL
ÁREA DE PRODUCCION Y LOGISTICA
BOGOTA D.C., DICIEMBRE DE 2006**

TABLA DE CONTENIDO

TABLA DE CONTENIDO.....	2
INDICE DE TABLAS.....	3
INDICE DE FIGURAS.....	4
LISTA DE ANEXOS.....	5
1. INTRODUCCIÓN.....	3
2. MARCO TEÓRICO.....	8
2.1 Descripción del Problema.....	8
2.2 Formulación Matemática.....	8
2.3 Ámbitos de Aplicación.....	9
2.4 Búsqueda Tabú.....	10
3. REVISIÓN BIBLIOGRÁFICA.....	12
3.1 Programación de Operaciones (Scheduling).....	12
3.2 Problemas Tipo Job Shop.....	13
3.3 Técnicas Heurísticas.....	14
3.4 Problemas Tipo Job Shop con restricción de No – wait.....	15
4. PROPUESTA DE SOLUCIÓN.....	20
4.2 Procedimiento General de Solución.....	21
4.2.1.1 Vector de Precedencias de Operación de los Trabajos.....	22
4.2.1.2 Vector de Tiempos de Operación de los Trabajos.....	22
4.2.2 Solución Inicial.....	23
Y la última regla es selección aleatoria (Random Selection,) el orden en que deben ser secuenciados los trabajos para ser procesados en el sistema es generado aleatoriamente..	25
4.2.2 Procedimiento JSSP No - Wait.....	25
4.2.2.1 Matriz MatrixGantt.....	28
4.2.2.2 Vector MenorTI.....	29
4.2.2.3 Vectores VectorTI y VectorTF.....	30
4.2.2.4 Casos Donde se Infrinja la Ubicación de la Operación.....	31
4.2.3 Búsqueda Tabú JSSP No - Wait.....	32
4.2.3.1 Vecindario.....	34
4.3 Ejemplo de Aplicación para el Procedimiento JSSP No – Wait.....	35
5. RESULTADOS.....	40
5.1 Parámetros de la Búsqueda Tabú Para el Procedimiento JSSP No – Wait.....	40
5.2 Resultados obtenidos.....	41
5.2.1 Resultados con SPT como solución inicial.....	42
5.2.2 Resultados con LPT como solución inicial.....	44
5.2.3 Resultados con Rnd como solución inicial.....	45
5.2.4 Mejores Resultados.....	47
5.2.5 Resultados de las Soluciones Iniciales (SPT y LPT) contra la Mejor Heurística de Schuster y Framiñan 2003.....	49
5.2.5 Tiempos computacionales.....	51
6. CONCLUSIONES Y RECOMENDACIONES.....	53
6.1 Conclusiones.....	53
6.2 Recomendaciones.....	54
7. BIBLIOGRAFÍA.....	55
ANEXOS.....	58

ÍNDICE DE TABLAS

Tabla 1: métodos para resolver el problema de scheduling en Job Shop	14
Tabla 2: Matriz de precedencia de las operaciones para cada uno de los trabajos	22
Tabla 3: Matriz de tiempos de proceso para las operaciones de cada uno de los trabajos.	23
Tabla 4: Tiempo total de proceso por trabajo.	24
Tabla 5: Instancias pequeñas	40
Tabla 6: Instancias grandes	40
Tabla 7: Tamaño de lista tabú según número de trabajos	41
Tabla 8: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando SPT.	42
Tabla 9: Gonzáles (2006) utilizando SPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando SPT.	43
Tabla 10: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando SPT por tamaño de instancias y el total de instancias.	43
Tabla 11: Gonzáles (2006) utilizando SPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias.	43
Tabla 12: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando LPT.	44
Tabla 13: Gonzáles (2006) utilizando SPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando LPT.	44
Tabla 14: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando LPT por tamaño de instancias y el total de instancias.	45
Tabla 15: Gonzáles (2006) utilizando LPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) tamaño de instancias y el total de instancias	45
Tabla 16: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando Rnd.	46
Tabla 17: Gonzáles (2006) utilizando Rnd Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando RND.	46
Tabla 18: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo utilizando Rnd por tamaño de instancias y el total de instancias.	47
Tabla 19: Gonzáles (2006) utilizando Rnd Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias.	47
Tabla 20: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo	48
Tabla 21: Gonzáles (2006) Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006)	48
Tabla 22: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo por tamaño de instancias y el total de instancias	49
Tabla 23: Gonzáles (2006) Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias	49
Tabla 24: soluciones iniciales (las mejores) de Schuster y Framiñan 2003 contra las soluciones inidales de FGA – SPT y FGA – LPT.	50
Tabla 25: Tiempos Computacionales Reportados para cada una de las instancias (segundos) Framiñan & Schuster (2003) y (2006) comparados González (2006)	50

ÍNDICE DE FIGURAS

Figura 1: Representación basada en trabajos, ejemplo tomado de Cheng (1996)	21
Figura 2: Representación basada en trabajos, adapta para el problema de job shop con restricciones de no wait	21
Figura 3: Diagrama Procedimiento General de Solución	21
Figura 4 Representación del vector RouteNumber (Precedencia de las operaciones en cada uno de los trabajos)	22
Figura 5: Representación del vector OperationTime (Precedencia de las operaciones en cada uno de los trabajos)	23
Figura 6: Representación del vector SeqJobs según regla SPT	24
Figura 7: Representación del vector SeqJobs según regla LPT	24
Figura 8: Diagrama del Procedimiento JSSP No – Wait	26
Figura 9: MatrixGantt	28
Figura 10: MatrixGantt ingreso de trabajos y sus precedencias	28
Figura 11: Información de los Vectores MenorTI, VectorTF y VectorJob	29
Figura 12: Procedimiento de búsqueda para la mejor ubicación de la operación	29
Figura 13: Datos de ejemplo para la diferencia entre MenorTI y VectorTI	30
Figura 14: Diferencia entre MenorTI y VectorTI	30
Figura 15: Casos donde es posible asignar la operación sin infringir las anteriores de los trabajos ya programados	31
Figura 16: Casos donde no es posible asignar la operación sin infringir las anteriores de los trabajos ya programados	32
Figura 17: Búsqueda Tabú para el Procedimiento JSSP No – Wait	33
Figura 18: Generación de vecinos, mediante el intercambio de trabajos consecutivos	35
Figura 19: Secuencia de inicio para el ejemplo de aplicación	35
Figura 20: Ejemplo de aplicación. Paso 1	35
Figura 21: Ejemplo de aplicación. Paso 2	35
Figura 22: Ejemplo de aplicación. Paso 3	35
Figura 23: Ejemplo de aplicación. Paso 4	37
Figura 24: Ejemplo de aplicación. Paso 5	37
Figura 25: Ejemplo de aplicación. Paso 6	37
Figura 26: Ejemplo de aplicación. Paso 7	37
Figura 27: Ejemplo de aplicación. Paso 8	38
Figura 28: Ejemplo de aplicación. Paso 9	38
Figura 29: Ejemplo de aplicación. Paso 10	38
Figura 30: Ejemplo de aplicación. Paso 11	39
Figura 31: Ejemplo de aplicación. Paso 12	39
Figura 32: Ejemplo de aplicación. Paso 13	39

LISTA DE ANEXOS

Anexo 1: Instancias Usadas para el Procedimiento JSSP No – Wait con SPT Comparadas Contra el Óptimo	58
Anexo 2: Instancias Usadas para el Procedimiento JSSP No – Wait con SPT Comparadas frente a Schuster & Framiñan (2003) y (2006)	61
Anexo 3: Instancias Usadas para el Procedimiento JSSP No – Wait con LPT Comparadas Contra el Óptimo	64
Anexo 4: Instancias Usadas para el Procedimiento JSSP No – Wait con LPT Comparadas frente a Schuster & Framiñan (2003) y (2006)	67
Anexo 5: Instancias Usadas para el Procedimiento JSSP No – Wait con Rnd Comparadas Contra el Óptimo	70
Anexo 6: Instancias Usadas para el Procedimiento JSSP No – Wait con Rnd Comparadas frente a Schuster & Framiñan (2003) y (2006)	73
Anexo 7: Instancias Usadas para el Procedimiento JSSP No – Wait de los Mejores Resultados (FGA Best) Comparadas Contra el Óptimo	76
Anexo 8: Instancias Usadas para el Procedimiento JSSP No – Wait de los Mejores Resultados (FGA Best) Comparadas frente a Schuster & Framiñan (2003) y (2006)	79

1. INTRODUCCIÓN

En la actualidad las empresas manufactureras en su mayoría desarrollan sus procesos bajo un enfoque jerárquico de producción, el cual les permite realizar actividades que se encuentran en cada una de estas como apoyo para el funcionamiento del sistema productivo. Cada una de estas etapas debe ser coherente a las necesidades y capacidades de la empresa y todas trascienden en los otros niveles de la jerarquía del sistema, estas etapas son:

1. Planificación estratégica.
2. Planificación agregada.
3. Programación maestra.
4. Programación de componentes.
5. Ejecución y control de la producción.

La programación de operaciones (scheduling) es la última etapa de este enfoque jerárquico se ejecutan las actividades encaminadas a programar y controlar las operaciones necesarias para dar cumplimiento a los pedidos, y en el caso especial de este trabajo se deben establecer los pasos de los diferentes pedidos por los centros de trabajo o máquinas bajo una configuración productiva tipo Job Shop, buscando minimizar el makespan, siendo este el problema clásico de job shop pero con una restricción adicional de no espera (no - wait), donde un trabajo al ser procesado por una máquina, no se es permitido parar el proceso en su máquina siguiente, es decir, al programar un trabajo este debe secuenciar todas sus operaciones de manera que el fin de una indique inmediatamente el inicio de su posterior operación, hasta que todas queden programadas.

Hay dos tipos principales de usos para los problemas job shop con restricción de no - wait. El primer es cuando esta restricción se adapta en la tecnología de la producción, por ejemplo algunas características de un material (tales como temperatura) donde se requiere que una operación sea procesada inmediatamente después de haber sido procesa en otra.

Sucede por ejemplo en la fabricación de acero cuando diversas operaciones tienen que moldear los lingotes, desmoldarlos y recalentar lo cual se debe hacer sin interrupciones (no - wait) para no afectar el proceso ya realizado. Otros ejemplos ocurren en la industria química (anodización de los productos de aluminio), transformación de los alimentos (la conserva debe ser hecha inmediatamente después de cocinar el producto) donde de igual manera la interrupción del proceso perjudicaría la calidad del producto en proceso.

El segundo tipo de usos para los problemas no - wait es programar en ambientes cuando no hay almacenajes intermedios entre las máquinas entre dos operaciones consecutivas. En este caso la única manera es ejecutar un trabajo mediante ambientes con restricciones de no - wait.

Como se puede observar anteriormente esta restricción es típica en muchos usos de la vida real tales como producción de acero, sistemas informáticos, procesamiento de

alimentos, industria química, producción del concreto o instalaciones para pruebas de semiconductores, industria farmacéutica,

Por lo anterior se puede decir que este tipo de problemas tiene una gran aplicabilidad en la industria actual y es un caso interesante de estudio debido a su complejidad dado que incurren en procesos donde los productos no pueden esperar a ser procesados entre las máquinas que requieren para su producción por las características de los materiales o procesos inherentes a estos, no hay procedimientos de solución estándar en la literatura por la dificultad de programación del problema y la poca atención que se le ha prestado relación al problema job shop clásico

Este trabajo se encuentra dividido en una revisión bibliográfica acerca de problema de job shop en el caso especial con restricciones de no-wait, así como otros elementos relaciones a la solución del mismo. Se continúa con una descripción del problema, donde éste es definido, y se muestra una de las formulaciones matemáticas propuestas en la literatura así como los ámbitos de aplicación. La siguiente parte muestra la propuesta de solución realizada y muestra un ejemplo de aplicación del procedimiento desarrollado. Luego se muestran los resultados obtenidos y sus diferentes comparaciones con la literatura. Y por último se hacen las conclusiones pertinentes del problema de expuesto.

2. MARCO TEÓRICO

Este problema de minimizar el makespan en ambientes job shop con restricciones de no-wait se puede describir como $J/n\text{-wait}/C_{\text{MAX}}(\alpha/\beta/\gamma)$ según la notación de Graham et al. citado por Blazewicz et al. (2002), en el cual α , describe el ambiente de producción, β provee detalles de las características de procesamiento y las restricciones y γ describe la función objetivo.

Al igual que el problema clásico de job shop, este presenta las mismas características, pero además agrega la restricción no-wait, donde no pueden existir almacenamientos o esperas entre operaciones consecutivas de un mismo trabajo, es decir al iniciar un trabajo este debe ser completado totalmente sin interrupciones por todas las máquinas asignadas a las tareas asociadas a éste y además agrega Gröflin & Klinkert (2006) que el tiempo de inicio de la primera operación determina el tiempo de inicio de todas las otras operaciones del trabajo.

2.1 Descripción del Problema

Este problema de minimizar el makespan en ambientes job shop con restricciones de no-wait es descrito por Macchiaroli, Mole & Riemma 1999 de la siguiente manera:

Dado $J = \{J_1, \dots, J_n\}$ y $M = \{M_1, \dots, M_m\}$, describen respectivamente, un conjunto de trabajos a ser procesados en un conjunto de máquinas disponibles. Cada trabajo J_i requiere ser procesado en la máquina m_i , esto es, que cada uno de los trabajos está compuesto por diferentes tareas (o operaciones) o_{ik} . Además t_{ik} y m_{ik} representan respectivamente, el tiempo de proceso y la máquina que requiere la k -ésima operación del trabajo i , donde $i \in \{1, \dots, n\}$ y $k \in \{1, \dots, m_i\}$. Y p_{ik} denota el tiempo de proceso del trabajo i en máquina k , y se define s_{ik} y e_{ik} respectivamente, el tiempo de inicio y fin de la k -ésima operación del trabajo i . Para la restricción de no-wait se impone claramente que $s_{i,k+1} = e_{ik}$ para todo $k \in \{1, m_i-1\}$.

2.2 Formulación Matemática

Para ambientar $J/n\text{-wait}/C_{\text{MAX}}$ existen diferentes formulaciones matemáticas planteada por los diferentes autores consultados, de las cuales se mostrará la formulación matemática planteada por Raaymakers & Hoogeveen (2000) la cual es la siguiente:

Parámetros

N	número de máquinas ($n=1; \dots; N$)
J	número de trabajos ($j=1; \dots; J$)
l_j	número de operaciones del trabajo j ($i=1; \dots; l_j$)
p_{ij}	tiempo de proceso de la operación i del trabajo j
δ_{ij}	tiempo de espera entre el inicio del trabajo j y el tiempo de inicio de la operación i del trabajo j

ε_{ijn} 1 si la operación i puede ser realizada por la máquina n ;
0 de lo contrario

Variables de decisión

T_j tiempo de inicio del trabajo j
 x_{ijn} 1 si la operación i del trabajo j es alistada a la máquina n
 0 de lo contrario

Función objetivo

C_{max} makespan

Minimizar C_{MAX}

Sujeto a:

$$T_j + \delta_{ij} + p_{ij} \leq C_{MAX} \quad \forall i, j, \quad (1)$$

$$\sum_{n=1}^N x_{ijn} = 1 \quad \forall i, j, \quad (2)$$

$$x_{ijn} \leq \varepsilon_{ijn} \quad \forall i, j, n \quad (3)$$

$$T_j \geq 0 \quad \forall j \quad (4)$$

$$x_{ijn} \in \{0,1\} \quad \forall i, j, n \quad (5)$$

La función objetivo es minimizar el tiempo de terminación de ultimo trabajo procesado (makespan) sujeto a estas cinco restricciones, la (1) hace referencia a que el makespan no puede ser menor que el tiempo de terminación de cualquier operación programada. (2) Cada operación es programada exactamente en una máquina. (3) cada operación solo puede ser programada en una maquina si la maquina puede programar la operación. (4) se deben iniciar todos los trabajos (el tiempo de inicio de cada trabajo y operación tiene que se no negativo). (5) la variable x_{ijn} solo puede tomar valores de cero o uno, es decir se programa en esa maquina la operación o no. Y además agrega otra restricción pero no la representa matemáticamente donde cada maquina solo puede realizar una operación a la vez” (Raaymakers & Hoogeveen 2000, p. 134-135).

2.3 Ámbitos de Aplicación

Hay dos tipos principales de usos para los problemas job shop con restricción de no - wait. El primer es cuando esta restricción se adapta en la tecnología de la producción, por ejemplo algunas características de un material (tales como temperatura) donde se requiere que una operación sea procesada inmediatamente después de haber sido procesa en otra.

Sucede por ejemplo en la fabricación de acero cuando diversas operaciones tienen que moldear los lingotes, desmoldarlos y recalentar lo cual se debe hacer sin interrupciones (no – wait) para no afectar el proceso hay realizado. Otros ejemplos ocurren en la

industria química (anodización de los productos de aluminio), transformación de los alimentos (la conserva debe ser hecho inmediatamente después de cocinar el producto) donde de igual manera la interrupción del proceso perjudicaría la calidad del producto en proceso.

El segundo tipo de usos para los problemas no – wait es programar en ambientes cuando no hay almacenajes intermedios entre las máquinas entre dos operaciones consecutivas. En este caso la única manera es ejecutar un trabajo mediante ambientes con restricciones de no - wait.

Como se puede observar anteriormente esta restricción es típica en muchos usos de la vida real y se han realizado investigaciones entorno a estos ambientes, entre ellos Schuster & Framiñan (2003 y 2006) resaltan trabajos realizados por diferentes autores en diferentes ámbitos industriales, tales como producción de acero (Wismer 1972), sistemas informáticos (Reddi 1973), procesamiento de alimentos (Hall & & Sriskandarajah 2000), industria química (Rajendran 1994), producción del concreto (Grabowski & Pempera 2000) o instalaciones para pruebas de semiconductores (Ovacik & Uzsoy 1997), Además Raaymakers & Hoogeveen 2000 hacen una aplicación a la industria farmacéutica,

Además los autores que han tratado este tema hacen hincapié en la gran complejidad del mismo donde Schuster & Framiñan (2003) resaltan que los mejores resultados obtenidos (óptimos para 72 instancias) se han realizado con branch & bound y buenos resultados con métodos de aproximación como búsqueda tabú y recocido simulado, con lo cual no es fácil llegar al óptimo para diferentes instancias, sin embargo se han utilizado los métodos de aproximación para resolver estos problemas en diferentes ambientes de producción entre los cuales se encuentran la producción de acero, industria química, producción de concreto, procesamiento de alimentos, instalación de semiconductores de prueba, proceso metalúrgicos, discos de computadores, y en general ambientes en que los productos intermedios no son muy estables o no pueden ser detenido el proceso debido a su composición o forma de procesar (Ej. Procesos donde se debe mantener cierta temperatura) Raaymakers & Hoogeveen (2000) y Schuster & Framiñan (2003,2006) y siendo Schuster & Framiñan (2003) los únicos autores que han expuesto los resultados utilizando las instancias clásicas de la literatura de los problemas tipo job shop para este tipo de restricción.

2.4 Búsqueda Tabú

Búsqueda tabú es el procedimiento utilizado para resolver el problema del job shop mediante el **Procedimiento JSSP No – Wait**, se hace una descripción de este y sus pasos principales

La búsqueda tabú es un procedimiento de búsqueda local propuesto por Glover en 1986 Gendreau et. al. (2003), en el cual se le proporciona memoria (lista tabú) a la búsqueda con el fin de evitar ciclajes (volver a soluciones anteriormente encontradas), el cual busca el mejor vecino en cada iteración y moverse hacia él, para evitar el ciclaje utiliza una lista tabú donde se almacenan las movidas prohibidas (ya recorridas) que solo se aceptan según algún criterio de aspiración establecido.

Además Pezcella & Merelli (2000) lo describen como método de búsqueda basado en vecindarios, y su funcionamiento se puede describir en los siguientes pasos:

1. **Seleccione una solución inicial $t \in T$; y haga el conjunto $t^*=t$ y $k=0$.**
Aquí se debe seleccionar una solución inicial factible que pertenezca a todos las posibles soluciones; se hace igual la solución inicial a la mejor solución hasta ahora encontrada y se inicializa el contador en cero ($k = 0$).
2. **$k=k+1$ y generé un subconjunto de N de la solución $N(t,k)$ tal que la movida aplicada no este en la lista tabú o cumpla el criterio de aspiración.**
Se comienza a contar las iteraciones realizadas y se genera un vecindario a partir de la solución inicial (o una encontrada en una de las iteraciones) para esa iteración ($k=k+1$). Esta solución no debe pertenecer a la lista tabú a no ser que halla cumplido con el criterio de aspiración (tradicionalmente se usa como criterio el superar la mejor solución hasta ahora encontrada)
3. **Seleccione la mejor solución de $t \in \hat{N}$ de acuerdo a la función objetivo f^* del problema.**
Se selecciona la mejor solución del vecindario según la función objetivo del problema.
4. **Si $f(t) < f(t^*)$ entonces el conjunto $t^*=t$.**
Se evalúa la solución, es decir si la función objetivo de esta supera la función objetivo de la mejor solución hasta ahora encontrada.
5. **Actualice la lista tabú y el criterio de aspiración.**
6. **Si se llega al criterio de parada, se detiene de lo contrario se repite el paso 2.**
Generalmente el criterio de parada es un número máximo de iteraciones ($k = \text{Máximo número de iteraciones}$) donde no se mejora la función objetivo.

3. REVISIÓN BIBLIOGRÁFICA

Esta revisión bibliográfica pretende hacer un recorrido a través de los diferentes autores que han tratado el problema de minimizar el *makespan* para problemas de job shop con restricciones de no-wait, sin embargo antes se hará una breve descripción de la programación de operaciones (scheduling) así como una ambientación de los problema tipo job shop y una definición sobre las técnicas heurísticas en general y las heurísticas de búsqueda local de los métodos de recocido simulado y búsqueda tabú siendo este ultimo el utilizado para el **Procedimiento JSSP No – Wait** el cual se describe ampliamente el capítulo 4.

3.1 Programación de Operaciones (Scheduling)

La programación de operaciones (scheduling) se encuentra en la última etapa del enfoque Jerárquico de producción, el cual se divide en cinco etapas, para Domínguez et. al. (1995):

1. Planificación estratégica.
2. Planificación agregada.
3. Programación maestra.
4. Programación de componentes.
5. Ejecución y control de la producción.

Para Domínguez et. al. (1995), Chase et. al. (2000) y Schroder (1983) la primera etapa fija las metas y planes de la organización, en la segunda se toman las decisiones de producción a mediano plazo, la tercera etapa convierte las cantidades en unidades específicas de producción, la cuarta etapa establece los requerimientos de producción y la quinta etapa ejecuta las actividades encaminadas a programar (scheduling) y controlar las operaciones necesarias para dar cumplimiento a los pedidos.

Dentro de la última etapa del sistema jerárquico de producción juega un rol importante ya que determina la secuencia de actividades a realizar así como la asignación de recursos (máquinas, personas, componentes, etc.) para la culminación de las tareas programadas en las etapas anteriores con el fin de cumplir la promesa al cliente, por ende según Pinedo (2002) se debe procesar los trabajos requeridos en ciertas máquinas (una o más máquinas según el caso) de acuerdo a ciertos parámetros y restricciones del sistema con el fin de optimizar uno o más objetivos.

En General para Pinedo (2002) y Blazewicz et. al (2001) los problemas de scheduling se consideran tres conjuntos, estos son: máquinas (o procesadores), trabajos (o tareas) y recursos, este ultimo no es mencionado por Pinedo y no se tiene en cuenta para el desarrollo de este trabajo de investigación; los trabajos son asignadas a una máquina o máquinas para ser procesados, dependiendo del tipo de máquinas y de la cantidad, existen diferentes ambientes o configuraciones productivas que se pueden caracterizar por máquinas que realizan la misma función (máquinas en paralelo) y máquinas dedicadas, las cuales solo pueden ejecutar una tarea específica.

Dentro de esta última se encuentran las configuraciones tipo flow shop, job shop y open shop. Blazewicz et. al (2001) describe estas configuraciones de la siguiente manera "... las tareas forman n subconjuntos (cadenas en el caso de flow y job shop), cada subconjunto es llamado trabajo. Estos trabajos J_j son divididos en n_j tareas, $T_{1j}, T_{2j}, \dots, T_{n_jj}$, y dos tareas adyacentes son realizadas en diferentes máquinas." (Blazewicz et. al. 2001, p. 58). Para problemas job shop, ambiente productivo en que se enfoca el trabajo, las tareas presentan una restricción de precedencia como se muestra a continuación:

$$T_{ij} < T_{i+1j} \quad \forall j \in J_j$$

La tarea i (operación i) se debe procesar antes que la tarea i+1 (operación i+1) para el trabajo J_j , dada la precedencia predeterminada de las tareas para cada uno de los trabajos que se este realizando, estos pueden tener diferentes números de tareas y pueden seguir diferentes secuencias para ser procesadas por las máquinas.

3.2 Problemas Tipo Job Shop

La restricción de precedencia ($T_{ij} < T_{i+1j}$ para $\forall j \in J_j$) descrita en el apartado anterior caracteriza a los problemas en Job Shop, el cual según Pinedo (2002), Blazewicz et. al. (2001) y Jain et. Al. (1999) se puede describir este tipo de problemas en forma general de la siguiente forma:

Cada trabajo esta compuesto por un conjunto de operaciones, que deben ser ejecutadas en un conjunto de máquinas, en un tiempo determinado, las operaciones tienen un orden específico para ser procesadas en cada una de las máquinas, con el fin de encontrar la secuencia en que los trabajos deben ser procesados en las máquinas para cumplir con uno o varios objetivos¹, y además se deben cumplir las siguientes restricciones:

1. Se deben respetar las precedencias de las operaciones de cada trabajo
2. Cada máquina ejecuta una operación a la vez.
3. Las operaciones no se pueden interrumpir.

El problema de scheduling en Job Shop para Jain et al. (1999) se ha tratado ampliamente en la literatura; hace más 40 años, uno de los primeros autores en tratar el tema fueron Fisher y Thompson en 1963 según lo expresa Blazewicz et. al. (2001), y a partir de esto se han ido proponiendo diferentes técnicas de solución para estos problemas, usando métodos de optimización como métodos de aproximación, algunos de estos se pueden ver en la tabla 1.

Métodos de Aproximación	Métodos iterativos	Inteligencia artificial
		Búsquedas locales

¹ Makespan, Retardo, Tardanza, Número de trabajos tardíos, entre otros.

	Métodos constructivos	Heurísticas basadas en cuello de botella
		Algoritmos de inserción
		Reglas de prioridad de despacho
Métodos de optimización	Métodos eficientes	
	Métodos enumerativos	Branch & Bound
		Matemáticos

Tabla 1: métodos para resolver el problema de scheduling en Job Shop.
Tomado de Jain et al. (1999)

3.3 Técnicas Heurísticas

Como se menciona el apartado anterior existen diferentes técnicas para resolver los problemas de scheduling, y en general para el job shop dentro de los métodos de aproximación uno de los más utilizados en la actualidad son las búsquedas locales las cuales son unas técnicas heurísticas, donde se explora una porción de todo el espacio de posibles soluciones (aproximación) en las que se pueden obtener buenos resultados tanto en calidad de respuesta como en tiempo en computo, aunque estas técnicas no garantizan soluciones óptimas como en los métodos exactos, si permiten lograr buenas soluciones dependiendo de su estructura de búsqueda.

Según Melián et. al. (2003) las heurísticas son un conjunto de algoritmos de búsqueda basados en estructuras repetitivas, en la consecución de una meta (función objetivo) que permita encontrar una solución factible, con la cual se espera acercarse a la solución óptima, sin embargo Martí & Moreno (2003) mencionan que las heurísticas suministran soluciones locales que pueden estar alejadas de las soluciones óptimas, para lo cual Melián et. al. (2003) se han desarrollado las técnicas metaheurísticas que permiten explorar un problema bajo un criterio de búsqueda más eficiente.

Estas técnicas según Lourenço et. al. (2003) se pueden dividir en dos grupos, el primero de estos son las metaheurísticas de vecindario (Neighborhood based metaheuristics), entre las cuales se encuentran Recocido simulado y Búsqueda Tabú las cuales realizan exploraciones a través de una heurística de búsqueda local, y el otro grupo son las metaheurísticas de multiarranque (Multi-start based metaheuristics) entre estas se encuentran algoritmos de colonias de hormigas y los algoritmos evolutivos, donde las soluciones se generan de manera iterativa.

Para los problemas de scheduling en job shop, las metaheurísticas de vecindario han tenido una gran acogida, en especial las dos técnicas anteriormente mencionadas (Recocido simulado - Simulated Annealing (SA) - y Búsqueda Tabú – Tabu Search (TS)), para Lee et. al. (1997) han sido muy populares en las últimas décadas, al igual que para el caso especial de no-wait son muy utilizadas obteniendo buenos resultados (Raaymakers & Hoogeveen, 2000), además estas técnicas tienen características comunes como los patrones de búsqueda entre los vecinos (soluciones que parten de una solución inicial dada que varían de esta según un patrón de intercambios), tener un criterio de aceptación (aunque difieren de su uso, mientras que en SA es por probabilidad en TS es a través de una lista tabú) además se han desarrollado técnicas

hibridas utilizando estas dos o combinando ambas. Por lo cual estas dos técnicas son las más viables para el desarrollo del problema propuesto.

Recocido simulado apareció por primera vez por Kirckpatrick, Gelatt y Vecchi en 1983 (Lee et. al. 1997). Handerson et. al. (1997) lo describe como un algoritmo de búsqueda local capaz de escapar de óptimos locales para buscar nuevas soluciones en otros espacios de búsqueda. Su nombre se basa en una analogía con el proceso físico de templado, “en el cual se forma una estructura reticular en un sólido calentándolo en un baño caliente hasta ablandarlo y luego enfriándolo pausadamente hasta solidificarse en un estado de baja energía” (Handerson 2003, p. 288). Las mejores soluciones generadas por los vecinos son aceptadas e incluso algunas de las peores según una probabilidad, la cual va decreciendo en cada iteración del algoritmo.

Además de resolverse por diferentes vías estos problemas también pueden estar sujetos a diferentes restricciones, algunas de estas son tiempos de alistamiento (setup) dependientes de la secuencia, Bloqueos de maquinas (los pedidos deben aguardar que la siguiente máquina este libre para procesarlos y mientras tanto bloquean la máquina actual), recirculación (son procesados mas de una vez por una misma máquina o centro de trabajo), fechas de inicio (release dates), daños de máquinas (Breakdowns), interrupciones de trabajos (preemptions) y no espera (no-wait), siendo esta ultima la restricción utilizada en el presente trabajo.

3.4 Problemas Tipo Job Shop con restricción de No – wait

Esta restricción se basa en que un trabajo no puede esperar entre dos máquinas consecutivas; es decir no se permiten almacenamientos intermedios entre máquinas, ya que al terminar de procesarse un trabajo en una de éstas debe pasar de inmediato a la siguiente máquina, sin embargo estos autores hacen referencia a esta restricción hacia problemas de tipo flow shop, y en la literatura especializada esta restricción se trata principalmente en este ambiente y son pocos los que han propuesto alternativas de solución hacia problemas tipo job shop.

Según lo comentado por Framiñan & Schuster (2003, 2006), Raaymakers & Hoogeveen (2000) y Mascis & Pacciarelli (2002) entre otros es poca la atención que se le ha prestado al problema de scheduling en job shop con restricciones de no-wait en relación al job shop clásico ($J_m // C_{MAX}$), tal vez porque presentan una gran dificultad computacional en relación a los otros problemas de job shop según este ultimo autor. Sin embargo se han ido realizando diferentes trabajos de investigación para abordar este tipo de problema, utilizando métodos de optimización y de aproximación, como Branch & Bound, también la utilización de algoritmos pseudo-polinomiales y así mismo técnicas heurísticas y metahurísticas de búsqueda local.

Entre los trabajos más destacados para Raaymakers & Hoogeveen (2000) se encuentran, los realizados por Sahni & Cho en 1979; Sriskandarajah & Ladet en 1986; Goyal & Sriskandarajah en 1988; Kamoun & Sriskandarajah, 1993; Hall & Sriskandarajah en 1996 los cuales realizan investigaciones sobre la complejidad matemática de estos problemas, también esta el trabajo de Reddi & Ramamoorthy en 1973, el cual presentan un limite inferior (lower bound) para minimizar el makespan, Kubiak en 1989 realiza un su trabajo mediante un algoritmo psedu-polinimial para dos

máquinas, Kravchenko (1998) realiza un algoritmo polinomial para reducir el tiempo de flujo.

Hall & Sriskandarajah (1996) hacen una revisión del estado del arte de los problemas con restricciones de no-wait, donde describe a este problema ($J / \text{no-wait} / C_{\text{MAX}}$) de gran dificultad computacional para encontrar soluciones óptimas, además expone una revisión del estado del arte estos problemas no – wait para diferentes configuraciones (entre estos los tipo job shop) de los cuales tan solo expone cuatro investigaciones realizadas de las 45 que revisa en su investigación, con lo que se puede sustentar el hecho de que este tipo de problema no se ha estudiado de manera amplia en la literatura especializada y además, estos trabajos no logran demostrar soluciones óptimas para el problema de job shop en grandes instancias, sin embargo Macchiaroli et. al. en 1996 realiza un algoritmo utilizando una búsqueda tabú para resolver grandes instancias obteniendo resultados cercanos al óptimo para instancias reales (a nivel industrial).

Algunos de estos trabajos presentan las bases para las investigaciones hechas recientemente que se encuentran en la literatura especializada, en los cuales varios de estos han utilizando técnicas metaheurísticas. Entre estos trabajos se encuentran los siguientes:

Kravchenko (1998) como se mencionó anteriormente utiliza un algoritmo polinomial, donde la clave se encuentra en programar los trabajos en cadenas de tareas para luego procesar los trabajos en sucesión según una lista con el objetivo de minimizar el tiempo medio de flujo, con este procedimiento indica que los problemas $J_3 / \text{no-wait}$, $t_{ij} = 1 / \sum C_i$ son de tipo NP – hardness en sentido fuerte de dificultad y por lo tanto afirma que los problemas $J_2 / \text{no-wait}$, $t_{ij} = 1 / \sum C_i$ son lo máximo que se pueden resolver mediante soluciones polinomiales para la restricción de no – wait, ya que estos ultimo son NP – hard en el sentido fuerte de dificultad.

Raaymakers & Hoogeveen (2000) en su trabajo “*programación multipropósito en procesos industriales por lotes con restricciones de no espera utilizando recocido simulado*”² el cual proponen una heurística de recocido simulado para este caso especial de job shop, además definen la estructura de vecinos que se requieren para este tipo de problemas ya que difiere de la estructura convencional de los problemas clásicos, los cuales no tiene en cuenta la restricción de no-wait, donde en los problemas clásicos se hacen intercambios entre operaciones, mientras que este problema es entre trabajos, también resaltan la importancia de empezar con una buena solución inicial (utilizando una regla de despacho) que permita una mejor búsqueda de vecinos con el recocido simulado. El funcionamiento del algoritmo es evaluado para programar casos industriales de una planta multipropósito en la hornada de productos farmacéuticos, el cual trata 50 instancias reales obtenidas de una planta de de este tipo, obteniendo buenos resultados para reducir el makespan para las diferentes instancias de la producción en la planta y también en comparación con la regla de despacho y logran reducir el tiempo computacional, ya que actualmente la programación requería un día entero de trabajo y con el recocido simulado este tiempo se reduce considerablemente

² Nombre original : Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing.

(en menos de una hora para instancias de 30 a 50 trabajos), además afirman que hasta este momento (año 2000), es poca la atención se ha prestado a los problemas $J_m/\text{no-wait}/C_{\text{MAX}}$, dado que estos problemas son muy difíciles de solucionar debido a la estructura de procesamiento dada por la restricción de no-wait.

Mascis & Pacciarelli (2003) realizan un procedimiento heurístico para el problema scheduling en Job-shop para restricciones de no-wait y bloqueos, los cuales consideran a este tipos de problema en especial para el caso de job shop de gran dificultad, además lo más interesante es la propuesta que realizan de un grafo disyuntivo alternativo aplicado para este tipo de problemas en especial, debido a que con el modelo del grafo disyuntivo tratado tradicionalmente en la literatura no es posible tratar estos problemas, por lo anterior a partir de este grafo alternativo se realizan diferentes heurísticas para resolver el problema incluyendo el algoritmo cuello de botella, sin embargo para estos autores luego de realizar diferentes pruebas, la de calidad de la solución (algoritmo cuello de botella) resulta poco probables para las restricciones de bloque y no-wait en job shop.

También desarrollan un algoritmo branch & bound con el fin de comparar los métodos de optimización y de aproximación con los diferentes resultados, para lo cual se utilizaron 72 instancias, donde se obtuvieron el óptimo con el branch & bound.

Woeginger (2004) trata el problema de job shop con restricciones de no-wait para minimizar el *makespan*, para el caso de de dos máquinas con más de una operación por trabajo y el caso de tres máquinas con al menos tres operaciones por trabajo. Plantean una programación de aproximación polinomial, sin embargo para estos problemas no se pueden realizar por este método tal como lo señala de igual forma Kravchenko (1998) pero el objetivo de éste es minimizar el tiempo medio de flujo.

Valdez et. al. (2005) presentan un caso de aplicaron para un fabrica de vidrio con una gran variedad de productos, la estructura del problema es tipo flexible job shop, donde unas de las operaciones involucradas por el proceso de transformación presentan restricciones de no-wait, ya que las operaciones que involucran liquido caliente para la fabricación del vidrio no se pueden interrumpir ya que se malograría el material en proceso. Estos autores desarrollan una heurística para resolver el problema que produzca soluciones de calidad y en un corto tiempo computacional a través de dos fases, el primero es encontrar una solución inicial factible y la segunda mejorar esta. Los resultados obtenidos fueron evaluados por la compañía la cual decidió implementar la técnica propuesta.

Gröflin & Klinkert (2006) proponen un procedimiento de inserción factible para diferentes escenarios de scheduling para problemas tipo job shop, entre los cuales se encuentran los problemas con restricciones de no-wait. Los autores implementan una heurística de búsqueda local en la cual a través de las soluciones de los vecinos extraen las operaciones y la insertan en otra posición factible (de una tarea no trivial, es decir que afecte el makespan) buscando obtener mejores soluciones para minimizar el makespan del problema planteado.

Además se encuentran trabajos que aunque no tratan directamente el problema de scheduling en job shop con restricciones de no-wait, si realizan aproximaciones a estos problemas, como es el caso de Carotenuto (s.f.) el cual relacionan el problema de job shop con el de rutas alternativas para el problema de ruteo de vehículos, donde la

restricción de no-wait se utiliza cuando se supone que un área no es segura por lo cual cuando los vehículos comienzan a viajar del origen no puede parar hasta que llega al punto final de destino, es decir, se debe realizar una asignación de la ruta del vehículo y el horario del trabajo del mismo que permita cumplir con esta restricción, y el trabajo de Zhou et. al. (2005) el cual realiza un procedimiento para la transmisión de información a través de líneas de comunicación en red, al realizar una transmisión de paquetes de información deben estar bloqueados los otros canales para no interferir con la señal, es decir al enviar una señal esta debe tener el camino libre para no generar interrupciones en la transmisión de la información.

Como se puede observar son pocas las referencias que se encuentran en la literatura, ya que la gran mayoría de los problemas con restricciones de no-wait, cuando son analizados, son trabajos bajo el esquema de operación del flow shop, sin embargo cabe resaltar que en los últimos años se han realizado importantes aportes en pro de buscar mejores soluciones para este problema, utilizando principalmente técnicas metaheurísticas, entre los cuales se destacan los trabajos desarrollados por Schuster & Framiñan (2003 y 2006) pues son los que hasta ahora han publicado resultados de las instancias clásicas de la literatura especializada.

El primero de estos trabajos (Schuster & Framiñan 2003), los autores desarrollan el problema de job shop con restricción de no-wait para minimizar el makespan, a través de un procedimiento que se divide en dos partes, la primera parte desarrollan un algoritmo timetabling con el fin de ordenar los trabajos y posteriormente proponen dos algoritmos de búsqueda local. Una de las principales características de este artículo y del otro propuesto en 2006 por estos mismos autores es la divulgación de los resultados para las diferentes instancias probadas, para este primer artículo los resultados obtenidos tienen resultados favorables en relación a las mejores estrategias existentes. También cabe resaltar que aunque el diseño no es para ambientes tipo flow shop la propuesta desarrollada supera al mejor existente en la literatura.

Además comentan que para este tipo de restricción las heurísticas de búsqueda local como recocido simulado y búsqueda tabú proporcionan una buenas soluciones que permitan desarrollar la estructura especial del problema, y por lo tanto los dos algoritmos desarrollados por estos autores se basan en técnicas de búsqueda local.

El primer algoritmo denominado VNS (Variable Neighborhood Search) es un algoritmo determinístico de búsqueda rápida de vecindades que está diseñado para explotar la estructura especial del problema tanto como sea posible utilizando un procedimiento timetabling, estos procedimientos (timetabling) se definen de manera sencilla por Gunadhi, Anand & Wee Yong (1996, p.243) como “una subclase de los problemas de scheduling, donde el objetivo es asignar las actividades en periodos de tiempo, tal que varias restricciones sean satisfechas”,

El procedimiento en sí mismo consiste en tres módulos. El primer módulo genera una solución inicial, a través de una regla de despacho o prioridad, el segundo módulo conduce el proceso rápido de búsqueda un grado óptimo local y el tercer módulo intenta escaparse del óptimo local para evitar quedarse atrapado en los óptimos locales de manera temprana del módulo 2.

El segundo algoritmo propuesto es llamado GASA, debido a ser un algoritmo híbrido entre algoritmos genéticos (GA por sus siglas en inglés) y recocido simulado (SA por sus siglas en inglés).

Con el algoritmo genético buscan generar la población a partir de la solución inicial obtenida (regla de despacho o prioridad) con lo cual generan los nuevos individuos de la población a través de los elementos clásicos de esta técnica como son el cruce y la mutación. Y el recocido simulado es tiene como propósito desarrollar búsquedas individuales de la población.

En el segundo trabajo (Schuster & Framiñan 2006) desarrollan el problema de job shop con restricción de no – wait para minimizar el makespan sin embargo en este los autores primero se centran en el problema timetabling, aprovechando la simetría del problema para sugerir un nuevo procedimiento timetabling (inverse timetabling), y posteriormente, utilizan timetabling en una búsqueda local con metaheurística con memoria similar a al búsqueda tabú.

También recalcan (de igual forma en 2003) que son pocas las contribuciones en la literatura a este tipo especial del problema de job shop y que son pocos los autores que han tratado las instancias clásicas, entre ellos mencionan que solamente conocen un procedimiento de branch & bound propuesto por Mascis, A. & Pacciarelli en 2000 y 2002 en relación a los métodos exactos y en relación a los métodos de aproximación existen entre otros acercamientos basados en la búsqueda tabú (Macchiarelli, Mole & Riemma 1999), recocido simulado (Raaymakers & Hoogeveen 2000), procedimientos de horizontes rodantes (Ovacik & Uzsoy 1997), Búsqueda de vecinos variables (Schuster & Framiñan 2003) y algoritmos híbridos entre algoritmo genético y recocido simulado (Schuster & Framiñan 2003) siendo estos dos últimos los que tratan las instancias clásicas de la literatura ya que el resto de trabajos se centra en ambientes específicos de la producción como la industria galvánica, la industria farmacéutica, producción de las instalaciones de prueba de semiconductores, etc.

El procedimiento desarrollado por estos autores se centra en el timetabling en el cual partiendo de una solución factible pasan a resolver el timetabling sin embargo utilizan un procedimiento inverso (inverse timetabling) en el cual la programación de la secuencia dada es programada al revés con el fin de realizar una exploración más amplia de la solución, es decir una misma secuencia puede tener dos soluciones, y aplican este procedimiento a una heurística denominada CLM (Complete local search with memory), búsqueda local completa con memoria, donde las soluciones exploradas son almacenadas según el orden en que fueron exploradas. Este concepto es muy similar a la búsqueda tabú, sin embargo ellos manejan tres grupos de soluciones, LIVE la cual almacena soluciones buenas para futuras exploraciones, DEAD almacena las soluciones que ya fueron exploradas en LIVE y un último conjunto NEWGEN que almacena temporalmente las nuevas soluciones que se generan durante la iteración del algoritmo.

Además a diferencia al artículo anterior en éste comparan sus resultados contra los óptimos de las instancias clásicas de la literatura y no contra la mejor heurística propuesta como se mostraron en 2003, donde solamente comparan con el óptimo las instancias de tamaño pequeño.

4. PROPUESTA DE SOLUCIÓN

Para solucionar el problema de minimizar el makespan en ambientes job shop con restricciones de no-wait (J /no-wait/ C_{MAX}) se propone un algoritmo de secuenciación denominado **JSSP No – Wait** (Job Shop Scheduling Problem con restricción de No Espera) el cual encuentra el menor makespan dada una secuencia de trabajos y va evolucionado a través de una heurística de búsqueda local buscando la mejor solución final.

Este trabajo utiliza una técnica heurísticas de búsqueda local debido a que estas aunque no garantizan soluciones optimas si permiten obtener buenos resultado y utilizando un tiempo computacional relativamente razonable en comparación con las técnicas de optimización Raaymakers & Hoogeveen (2000), por su forma de exploración entre vecinos (que son soluciones factibles) las cuales no hacen grandes cambios a la representación del problema por lo cual no afectando severamente soluciones futuras.

Para solucionar este problema, este trabajo selecciona una adecuada representación del mismo dada la restricción sujeta a este y posteriormente se da a lugar un procedimiento general de solución.

4. 1 Representación del Problema

Para la representación de este problema y dadas las restricción de no wait del mismo donde las movidas del vecindario requieren una definición en la cual un trabajo entero es removido, en lugar de una sola operación Raaymakers & Hoogeveen (2000), la estructura del cromosoma a utilizar para la búsqueda local se puede realizar a través de la representación basada en trabajos, esta es una representación directa del problema. Para Cheng (1996) esta representación consiste en una lista de n trabajos, el primer trabajo de la lista es programado con todas sus operaciones y así sucesivamente, estas se programan de la mejor manera posible, un ejemplo de esta representación esta dada por al siguiente lista [2 3 1], donde el trabajo 1 se programa en la máquina 1, 2 y 3, el trabajo 2 se programa en la máquina 1, 3 y 2 y el trabajo 3 se programa en la máquina 2, 1 y 3; programa primero todas las operaciones del trabajo dos, luego las del trabajo tres y por ultimo las de trabajo uno como se muestra en la figura 1, sin embargo dadas las condiciones especiales del problema a tratar esta representación se programaría como se muestra en la figura 2.

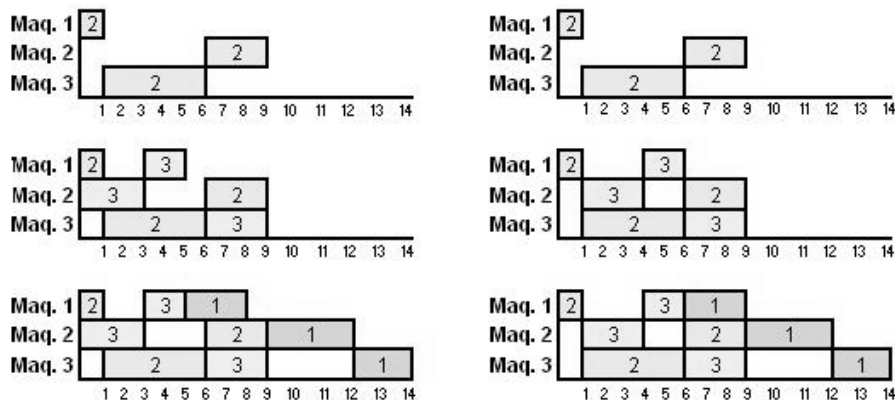


Figura 1: Representación basada en trabajos, ejemplo tomado de Cheng (1996)

Figura 2: Representación basada en trabajos, adapta para el problema de job shop con restricciones de no wait.

Para el JSSP No -Wait se genera un vector denominado SeqJobs (secuencia de trabajos) el cual como se indica anteriormente consiste en una lista de tamaño igual al número de trabajos que almacena el orden en que los trabajos con todas sus operaciones deberán ser procesados en el sistema.

4.2 Procedimiento General de Solución

Para resolver este problema se parte de un procedimiento general de solución, el cual muestra el orden jerárquico de las actividades a realizar para buscar el mejor resultado posible (Minimizar makespan), así como la forma en que se muestran los resultados obtenidos.

Este se divide varias etapas las cuales se irán explicando en los siguientes puntos del trabajo, este procedimiento se puede observar en el diagrama Procedimiento General de Solución:

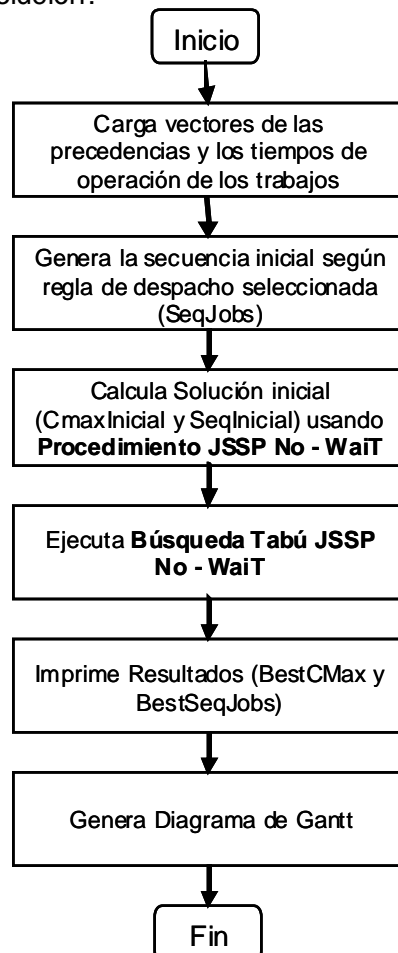


Figura 3: Diagrama Procedimiento General de Solución

4.2.1 Vectores de Precedencias y Tiempos de Operación de los Trabajos

Estos vectores almacena la información concerniente a las precedencias y los tiempos de operación de los trabajos según las instancias del problema cargados inicialmente, estas instancias corresponden a las encontradas en la literatura (con las cuales se hacen las comparaciones de resultados) o ingresadas por el usuario del **JSSP No – Wait**.

4.2.1.1 Vector de Precedencias de Operación de los Trabajos

Este vector almacena la información relacionada a la precedencia de operaciones de cada uno de los trabajos, es decir indica el orden en que debe ser atendido un trabajo por las diferentes máquinas del sistema. Para **JSSP No – Wait** este vector se llama RouteNumber, el cual es de de tamaño $n \times m$, donde n es el número de trabajos y m la cantidad de máquinas, donde cada segmento de tamaño m es la ruta de por la cual debe pasar un trabajo, y así para los n trabajos. La tabla 2 muestra un problema de 3×3 (trabajos x operaciones) la cual es una matriz de trabajos con su respectivas operaciones, que son el orden en que deben se atendidas y la figura 4 hace una representación del vector RouteNumber.

	Trabajo 1	Trabajo 2	Trabajo 3
Operación 1	2	1	3
Operación 2	1	3	1
Operación 3	3	2	2

Tabla 2: Matriz de precedencia de las operaciones para cada uno de los trabajos.

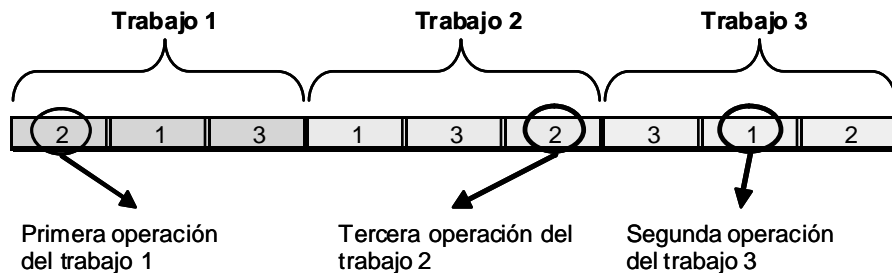


Figura 4 Representación del vector RouteNumber (Precedencia de las operaciones en cada uno de los trabajos)

Como se puede observar en la figura 4, cada trabajo consta de tres operaciones, y se puede leer de la siguiente manera, para el trabajo 1 la primera operación se realiza en la máquina 2, la segunda operación en la máquina 1 y la tercera operación en la máquina 3, y de igual manera para los demás trabajos.

4.2.1.2 Vector de Tiempos de Operación de los Trabajos

Este vector almacena la información relacionada a los tiempos de operación de cada una de las máquinas para cada uno de los trabajos, es decir para cada operación del trabajo indica el tiempo de proceso en la máquina en que este asignada dicha operación. Para **JSSP No – Wait** este vector se llama OperationTime, el cual es de de

tamaño $n \times m$, donde n es el número de trabajos y m la cantidad de máquinas, donde cada segmento de tamaño m indica los tiempos de proceso de cada operación, y así para los n trabajos. La tabla 3 muestra un problema de 3×3 (trabajos \times operaciones) la cual es una matriz de trabajos con su respectivas operaciones, que indica el tiempo de proceso de cada una de estas y la figura 5 hace una representación del vector OperationTime.

	Trabajo 1	Trabajo 2	Trabajo 3
Operación 1	4	3	1
Operación 2	2	5	1
Operación 3	1	4	6

Tabla 3: Matriz de tiempos de proceso para las operaciones de cada uno de los trabajos.

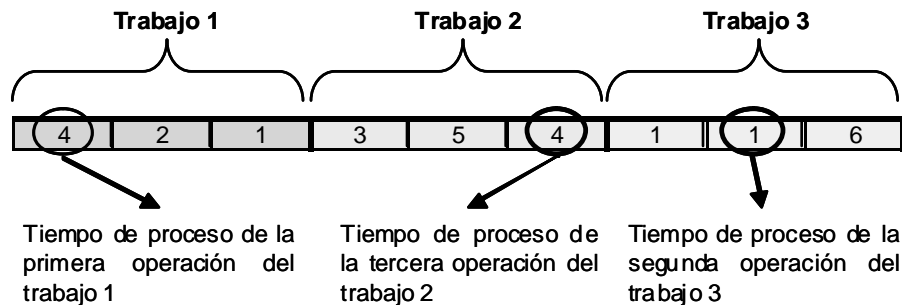


Figura 5: Representación del vector OperationTime (Precedencia de las operaciones en cada uno de los trabajos)

Como se puede observar en la figura 5, cada trabajo consta de tres operaciones, y se puede leer de la siguiente manera (teniendo en cuenta el orden de entrada a las máquinas, como se muestra en la figura 4), para el trabajo 1 la primera operación se realiza en la máquina 2 con una duración de 4 unidades de tiempo, la segunda operación en la máquina 1 con una duración de 2 unidades de tiempo y la tercera operación en la máquina 3 con una duración de 1 unidad de tiempo, y de igual manera para los demás trabajos.

4.2.2 Solución Inicial

Este apartado hace referencia a la generación de la secuencia inicial según regla de despacho seleccionada (SeqJobs) del procedimiento general de solución.

Se debe realizar una secuencia de partida o solución inicial del problema, ya que las técnicas basadas en heurísticas de búsqueda local parten de una solución del mismo, además una buena solución inicial según diversos autores como Matsuo et al., 1988, Van Laarhoven et al., 1992, Dell'Amico & Trubian, 1993, Nowicki & Smutnicki, 1996 citados por Pezzella, F. & Merelli (2000) han notado la importancia de ésta para obtener resultados de calidad de la solución final y en tiempo computacional razonable para los problemas clásicos de Job Shop ($J // C_{MAX}$), la cual para este problema en especial se puede realizar a través de una sencilla heurística como utilizan Schuster & Framiñan (2003 y 2006) seleccionando una regla de despacho para encontrar la solución inicial, con la cual logran muy buenos resultados finales (Con el algoritmo CLM no se desvían

en promedio más del 3% del resultado óptimo para las 62 instancias probadas en Schuster & Framiñan 2006).

Por lo anterior se debe seleccionar un punto de partida, es decir determinar la secuencia de trabajos (vector SeqJobs según apartado 4.1) para el cálculo de la solución inicial a través del **Procedimiento JSSP No – Wait**, este se realiza utilizando una regla de despacho o regla de prioridad, para este problema se seleccionaron tres reglas (las mismas que utilizaron Schuster & Framiñan 2006 para el CLM y CLM–time).

Las reglas de despacho o prioridad utilizadas se describen a continuación:
 La primera regla utilizada es el tiempo de proceso más corto (SPT – rule, Shoertest processing time), esta regla va seleccionando los trabajo con menor tiempo de proceso y los va ordenando de manera ascendente según este criterio, es decir el trabajo en que sus operaciones (la suma de éstas) duren menos que las otras operaciones de los diferentes trabajos será secuenciado de primero y así sucesivamente hasta programar todos los trabajos disponibles. Como ejemplo podemos tomar la tabla 3 que contiene los tiempos de proceso para las operaciones de cada uno de los trabajos para calcular el tiempo total de proceso de éstos, como se puede observar en la tabla 4

	Trabajo 1	Trabajo 2	Trabajo 3
Operación 1	4	3	1
Operación 2	2	5	1
Operación 3	1	4	6
Tiempo total de proceso por trabajo	7	12	8

Tabla 4: Tiempo total de proceso por trabajo.

Con lo cual se secuenciarían primero el trabajo 1, luego el trabajo 3 y por último el trabajo 2, como se muestra en la figura 6, donde el vector SeqJobs es de tamaño 3 por el número total de trabajos.

Vector SeqJobs (3) :

1	3	2
---	---	---

Figura 6: Representación del vector SeqJobs según regla SPT

La segunda regla usada es el tiempo de proceso más largo (LPT – rule, Longest processing time), esta regla va seleccionando los trabajo con mayor tiempo de proceso y los va ordenando de manera descendente según este criterio, es decir el trabajo en que sus operaciones (la suma de éstas) duren más que las otras operaciones de los diferentes trabajos será secuenciado de primero y así sucesivamente hasta programar todos los trabajos disponibles. Tomando el tiempo total de proceso de los trabajos de la tabla 3, se secuenciarían primero el trabajo 2, luego el trabajo 3 y por último el trabajo 1, como se muestra en la figura 7.

Vector SeqJobs (3) :

2	3	1
---	---	---

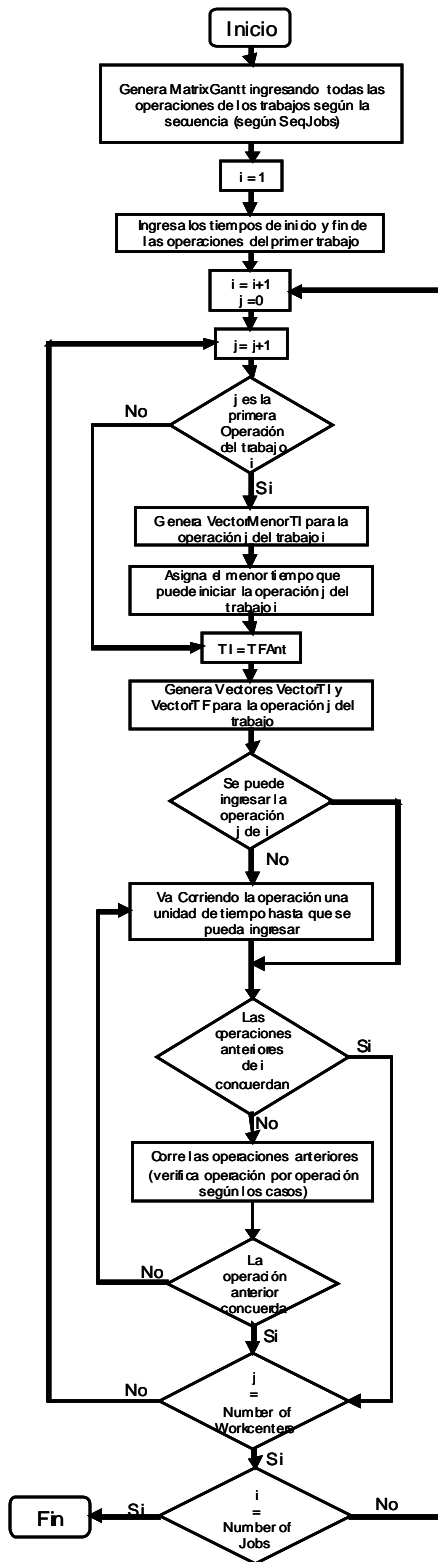
Figura 7: Representación del vector SeqJobs según regla LPT

Y la última regla es selección aleatoria (Random Selection,) el orden en que deben ser secuenciados los trabajos para ser procesados en el sistema es generado aleatoriamente.

4.2.2 Procedimiento JSSP No - Wait

Este apartado hace referencia al Cálculo Solución inicial (C_{max} Inicial y Seq Inicial) usando el **Procedimiento JSSP No – Wait** del procedimiento general de solución, ya que con este procedimiento se decodifica la secuencia de trabajos ($SeqJobs$) obtenida inicialmente a través de las reglas de despacho (STP, LPT o Random selection) y se halla la solución inicial (C_{Max} y Seq Inicial) para la creación de los vecindarios que se utilizarán posteriormente para el desarrollo de la heurística de búsqueda local, además este procedimiento es el mismo que se utiliza a lo largo de la heurística para los cálculos pertinentes en búsqueda de la mejor solución posible (Minimizar C_{max}) de las instancias a tratar.

Este procedimiento es el algoritmo principal desarrollado en este trabajo, pues con éste se calcula el makespan del problema, generando el mejor valor con la secuencia asignada, ya que busca y acomoda de la mejor forma posible cada uno de los trabajos y sus respectivas operaciones respetando la restricción de no – wait, este procedimiento se puede observar de forma general la figura 8, los conceptos ahí utilizados se explicarán después de la figura.



i	trabajos
j	operaciones
TI	tiempo de inicio de la operación actual
TFAnt	tiempo de fin de la operación anterior

Figura 8: Diagrama del Procedimiento JSSP No - Wait

A continuación se mostrara el funcionamiento del **Procedimiento JSSP No – Wait** mostrado en el diagrama anterior (figura 7), sin embargo para un mayor entendimiento del mismo posteriormente se explicara detalladamente los elementos principales del mismo.

Al iniciar el procedimiento ya se han almacenado con anterioridad las precedencias de los trabajos (vector RouteNumber como se explica en el apartado 4.2.1.1) y los tiempos de operación de los mismos (vector OperationTime como se explica en el apartado 4.2.1.2) así como la secuencia en que deberán ser procesados en el sistema (vector SeqJobs) según la regla seleccionada (como se indica en el apartado 4.2.2).

Se genera una matriz denominada MatrixGantt la cual almacena la información de los trabajos, sus operaciones, así como los tiempos de inicio y fin de cada una de las éstas según se vaya generando en el transcurso del algoritmo; como ya se conoce la secuencia en que serán procesados los trabajos y las precedencias de los mismos se ingresan estos datos y se inicializa el primer trabajo ($i = 1$), donde i es el trabajo y NumberOfJobs (número de trabajos) es el número total de trabajos, se ingresa el primer trabajo de la mejor manera posible, este no tiene ninguna complicación puesto que las máquinas están inicializadas en tiempo cero (0) y no se han ingresado trabajos anteriores que impidan la programación de estos sin violar la restricción de no – wait.

Posteriormente se inicia el ingreso del siguiente trabajo ($i = i+1$) y se inicializa $j = 0$, para el caso en que ya se hallan realizado varios trabajos y permita ingresar las operaciones del nuevo, donde j es operación (o máquina y NumberOfWorkcenters es la cantidad total de operaciones o máquinas) ha realizar (según el vector RouteNumber del trabajo almacenado en SeqJobs (i)) y se da inicio a ingresar la operación j del trabajo ($j = j+1$),

Se verifica si j es la primera operación del trabajo i , de ser así, se genera el vector MenorTI (menor tiempo de inicio) el cual almacena los tiempo de inicio de los trabajos ya procesados en la máquina j , ordenándolos de menor a mayor, con lo cual busca asignar la operación en la mejor manera posible (menor tiempo que puede iniciar la operación j del trabajo i), en caso contrario o luego se realizar los pasos descritos anteriormente se asigna el tiempo en que debería iniciar la siguiente operación del trabajo para respetar la restricción de no – wait, esto se hace asignando el tiempo de inicio de la siguiente operación igual al tiempo de terminación de la operación anterior del trabajo i .

Ahora se procede a generar los vectores VectorTI (vector tiempo de inicio) y VectorTF (vector tiempo de fin), los cuales almacenan respectivamente los tiempos de inicio y fin de los trabajos ya procesados, con lo cual se verifica si se puede ingresar la operación j del trabajo i (en el caso de la asignación de $TI = TFAnt$ o al haber corrido la operación) revisando cada uno de los casos posibles (casos donde se infrinja la ubicación de la operación por estar ya ocupada por un trabajo ya programado), si no se puede ingresar la operación, esta se va corriendo una unidad de tiempo hasta lograr ingresarla sin ocupar espacios ya asignados a otras operaciones; si se puede ingresar de igual manera se realiza el paso descrito a continuación.

El siguiente paso es uno de los mas importantes, ya que es que permite el cumplimiento de la restricción de no – wait, este verifica si las operaciones anteriores de i concuerdan con la restricción, en este paso se van acomodando las operaciones para el cumplimiento de la ésta, y a la vez se verifica que no interfieran con operaciones ya

programadas de los trabajos anteriores revisando los casos posibles (como se explica anteriormente), de interferir se corre una unidad de tiempo la operación j , y así hasta lograr que todas las operaciones actuales queden programadas, sin realizar interferencias con las de los otros trabajos y cumpliendo la restricción de no –wait.

Se revisa si se han programado todas las operaciones del trabajo i ($j = \text{NumberOfWorkcenters}$), si no se han programado se procede a realizar la siguiente operación ($j = j+1$) realizando los pasos descritos anteriormente hasta programar la totalidad de las operaciones del trabajo i , si ya se han realizado, se verifica si ya se han programado todos los trabajos, si no se ha hecho, se programa un nuevo trabajo ($i = i+1$ y $j = 0$) hasta realizar la totalidad de estos ($i = \text{NumberOfJobs}$).

4.2.2.1 Matriz MatrixGantt

Como se menciono anteriormente esta matriz almacena la información de los trabajos, sus operaciones, así como los tiempos de inicio y fin de cada una de las éstas según se vaya generando en el transcurso del algoritmo, esta matriz es de 4 por el número de trabajos multiplicado por la cantidad de máquinas ($4 \times (\text{NumberOfJobs} \times \text{NumberOfWorkcenters})$). Para explicar mejor esta matriz, se realizara un ejemplo (figura 8) de esta usando las instancias de las tablas 2 y 3, el cual representan un problema de 3×3 .

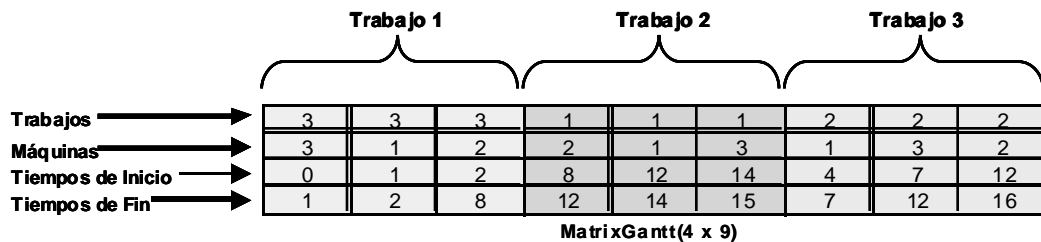


Figura 9: MatrixGantt

El Funcionamiento de la matriz es el siguiente, como ya se conoce la secuencia en que serán procesados los trabajos y las precedencias de los mismos se ingresan estos datos y se inicializa el primer trabajo (SeqJobs (1) = 3), este se ingresa de la mejor manera posible como ilustra la figura 10.

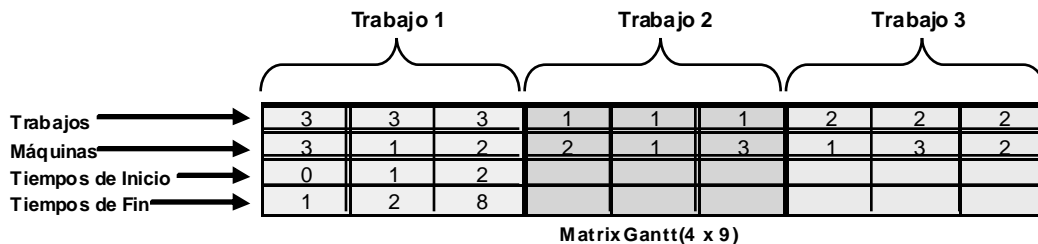


Figura 10: MatrixGantt ingreso de trabajos y sus precedencias

Y así sucesivamente se van ingresando los tiempos de inicio y fin de las operaciones para cada una de los trabajos según se vayan generando en el transcurso del **Procedimiento JSSP No – Wait** hasta completar la totalidad de estos (como se muestra en la figura 9).

4.2.2.2 Vector MenorTI

El vector MenorTI (menor tiempo de inicio), almacena los tiempos de inicio de los trabajos procesados anteriormente en la máquina actual (en la que se desea programar la operación del trabajo en que se encuentra), este vector es de tamaño variable, pues va aumentando según el número de trabajos que han sido atendidos anteriormente hasta el número de trabajos menos uno ($\text{NumberOfJobs} - 1$), este vector ordena las operaciones de los trabajos (ya programados en la misma máquina) de menor a mayor (este orden no es necesariamente el mismo en el que entran al sistema), con el fin de encontrar el menor tiempo en que pueda iniciar la primera operación (únicamente se usa para la primera operación de cada trabajo exceptuando el primer trabajo) del trabajo a procesar, además este vector se apoya en otros, el primero denominado VectorTF (vector tiempo de fin) el cual almacena el tiempo de culminación de las operaciones relacionadas en MenorTI y el segundo llamado vector VectorJob (el cual sirve de referencia para saber en que trabajo se encuentra la información de los tiempos) para realizar el procedimiento anteriormente descrito, además cabe resaltar que antes de realizar el procedimiento se revisa si la operación se puede realizar en tiempo cero (0).

Esto procedimiento se explica con el siguiente ejemplo; suponiendo que ya se han programado cuatro trabajos, se desea programar la primera operación del quinto trabajo en la máquina J el cual tiene una duración de 4 unidades de tiempo, por lo tanto se almacena la información requerida en los diferentes vectores (figura 10) y se proceda a buscar la mejor ubicación (figura 11) como se muestra a continuación:

MenoTI	→	1	9	16	17
VectorTF	→	6	12	17	18
VectorJob	→	2	1	4	3

Figura 11: Información de los Vectores MenorTI, VectorTF y VectorJob.

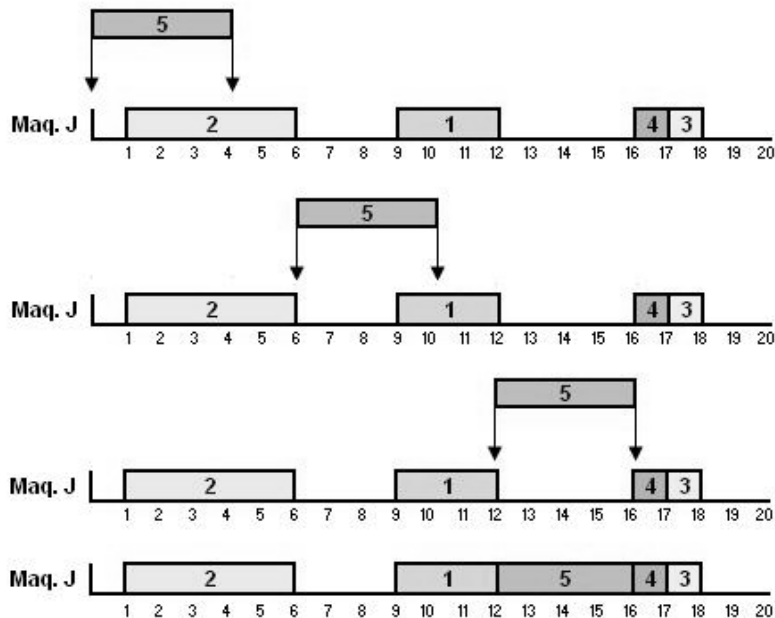


Figura 12: Procedimiento de búsqueda para la mejor ubicación de la operación.

4.2.2.3 Vectores VectorTI y VectorTF

El VectorTI (Vector tiempo de inicio) es similar al MenorTI, sin embargo este no ordena los tiempos de inicio de las operaciones de menor a mayor, sino que los almacena según van entrando al sistema (en el orden de la MatrixGantt), y el VectorTF (vector tiempo de fin) el cual almacena el tiempo de culminación de las operaciones e igualmente se utiliza un tercer vector de apoyo llamado vector VectorJob (indica a que trabajo corresponde los tiempos de los dos vectores anteriores), aunque son similares (almacenan la misma información y poseen el mismo tamaño) a los vectores anteriores (apartado 4.2.2.2), estos cumplen funciones diferentes, mientras que los vectores del 4.2.2.2 buscan acomodar la operación con el menor tiempo de inicio, estos vectores son usados para verificar que no se interfieran con las otras operaciones realizadas en la misma máquina por los otros trabajos. Para ver más claramente esta diferencia, ésta se ilustra en el siguiente ejemplo, donde la figura 12 muestra una MatrixGantt y su correspondiente diagrama de gantt para un problema de 4 x 3.

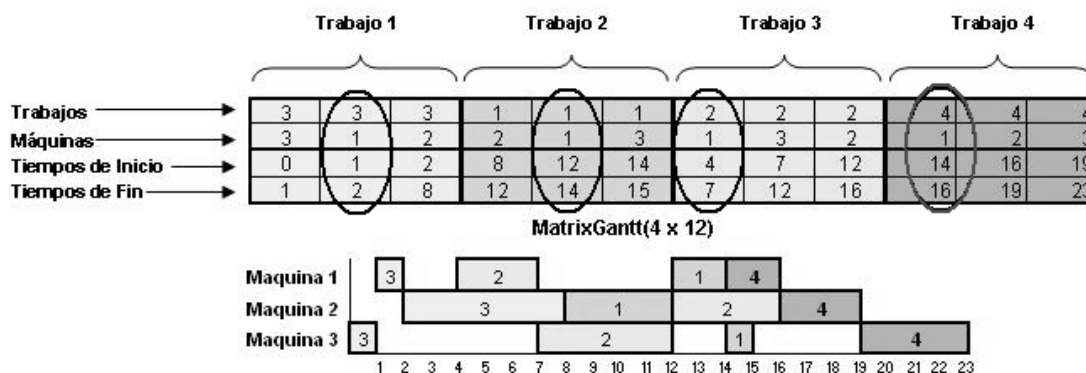


Figura 13: Datos de ejemplo para la diferencia entre MenorTI y VectorTI

Ahora antes de iniciar a procesar el trabajo 4 en su primera operación (máquina 1) los vectores de MenorTI y VectorTI son diferentes como se muestra en la figura 13:

VectorTI	1	12	4	MenorTI	1	4	12
VectorTF	2	14	7	VectorTF	2	7	14
VectorJob	3	1	2	VectorJob	3	2	1

Figura 14: Diferencia entre MenorTI y VectorTI

Como se observa estos vectores son distintos ya que cumplen diferentes funciones, mientras que los vectores relacionados al MenorTI busca poder acomodar de la menor manera posible la operación, el otro grupo de vectores tiene como misión evitar que la operación se interponga en las operaciones ya programadas de los trabajos anteriores según los diferentes casos donde se infrinja la ubicación de las operación, los cuales se mostraran en el siguiente apartado.

4.2.2.4 Casos Donde se Infrinjan la Ubicación de la Operación

Este apartado hace referencia a los posibles casos donde al correr una operación o las operaciones de un trabajo se pueda interrumpir o no operaciones ya programadas para los trabajos anteriores. Existen dos tipos de casos, el primero cuando no existen problemas para acomodar las operaciones, con lo cual se ingresaría sin ningún problema y el segundo caso es cuando al ingresar la operación se interfiere con una operación ya ubicada en esa posición de tiempo.

Para explicar mejor estos dos tipos de casos se ilustraran con los ejemplos de las posibles causas por las cuales se puedan o no acomodar las operaciones, para ello a que tener en cuenta la siguiente información:

TIP: Tiempo de inicio de la operación ya programada en la máquina J.

TFP: Tiempo de fin de la operación ya programada en la máquina J.

TI: Tiempo de inicio de la operación a programar en la máquina J.

TF: Tiempo de inicio de la operación a programar en la máquina J

Para el primer caso existen dos situaciones posibles que se ilustraran en la figura 15:

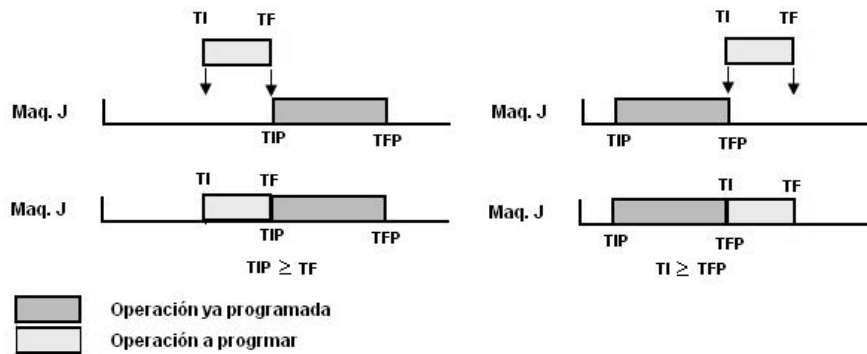


Figura 15: Casos donde es posible asignar la operación sin infringir las anteriores de los trabajos ya programados

Para el segundo caso existen cuatro posibles situaciones (para una mejor explicación se incluyen dos más de manera grafica) donde se interrumpen las operaciones anteriormente programadas, las cuales se ilustran en la figura 16:

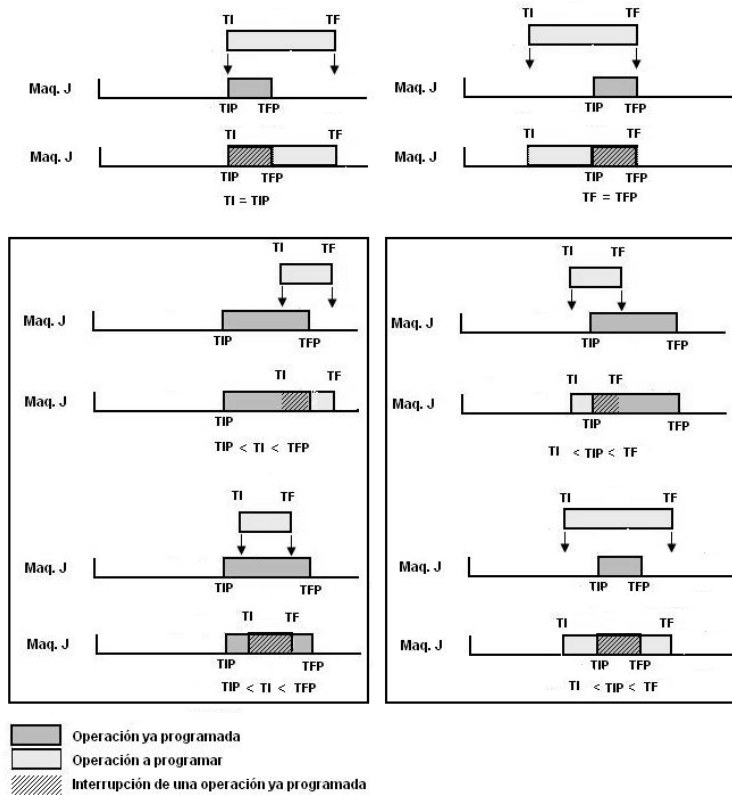


Figura 16: Casos donde no es posible asignar la operación si n infrin gir las anteriores de los trabajos ya programados.

4.2.3 Búsqueda Tabú JSSP No - Wait

Para solucionar el problema de minimizar el makespan para ambientes tipo job shop con restricción de no - wait ($J_m / no - wait / C_{MAX}$) se proceda a aplicar una técnica heurística de búsqueda local, ya que estas tienen una gran acogida y han sido muy populares en las últimas décadas Lee et. al. (1997) para los problemas clásicos de Job Shop ($J_m // C_{MAX}$) y además también son las utilizadas por los autores consultados y con los que se hacen las comparaciones de resultados Schuster & Framiñan (2003 y 2006).

Para este trabajo se utiliza la búsqueda tabú, este método proporciona buenas soluciones (cercanas al óptimo) ya que ofrecen un buen procedimiento de búsqueda local, su fácil implementación para diferentes tipos de problemas y que ofrecen la oportunidad de escapar de óptimos locales (lista tabú para evitar movidas ya visitas, evitando entrar en ciclos que puedan incurrir en generar óptimos locales). La búsqueda tabú utilizada aplicada al **Procedimiento JSSP No Wait** se puede observar en el siguiente diagrama (figura 17):

LT	lista tabú
SeqInicial	Secuencia inicial
BestSeq	Mejor secuencia
MejorSeq	Mejor secuencia encontrada en la iteración
CMaxInicial	Makespan inicial
BestCMax	Mejor makes pan
MejorCMax	Mejor makes pan encontrado en la iteración
MejorCmaxNoT	Mejor makes pan encontrado en la iteración no tabú
ConIter	Contador del numero de iteraciones realizadas
MaxIter	Máximo numero de iteraciones a realizar

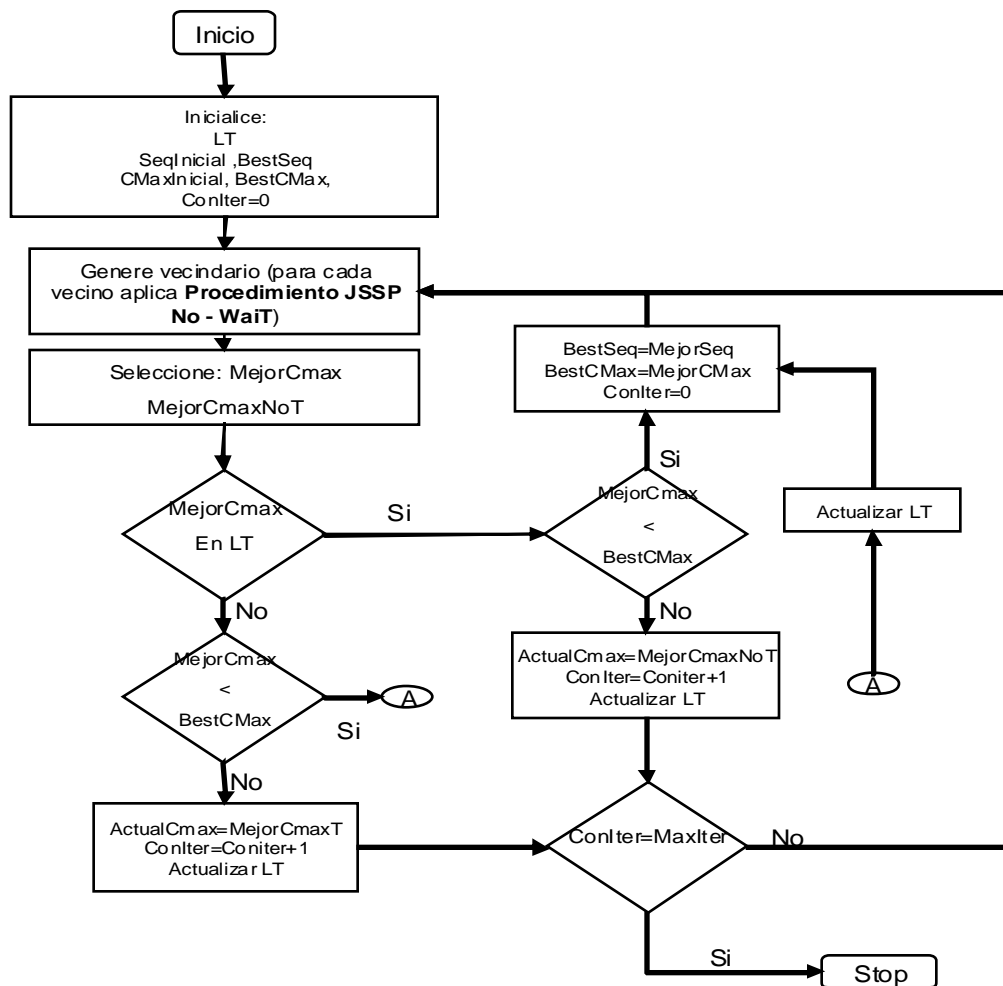


Figura 17: Búsqueda Tabú para el **Procedimiento JSSP No - Wait**

A continuación se mostrara el funcionamiento de la búsqueda tabú para el **Procedimiento JSSP No – Wait** mostrado en el diagrama anterior (figura 17).

Se inicializan los valores necesarios para la ejecución de la búsqueda tabú, como el tamaño de lista (La lista tabú le proporciona memoria a la búsqueda con el fin de evitar volver a soluciones anteriormente encontradas), el cual permitirá que no entre en ciclos (óptimos locales), así como la secuencia de inicio y los valores del makespan que permitirán el inicio y evolución del mismo en procura de buscar la mejor solución posible, así como el contador de iteraciones realizadas (ConIter), el cual determina el numero de vecindarios que han sido explorados y el criterio de parada, para es caso un número máximo de iteraciones (MaxIter) de no haber mejorado la solución almacena como BestCmax,

La solución inicial como se explico anteriormente en el apartado 4.2.2 se realiza a través de una regla de despacho o prioridad la cual determina la secuencia de inicio (SeqJobs) y para el cálculo del Cmax se utilizando el Procedimiento JSSP No – Wait, estos mismos valores parten como los mejores valores hasta ahora encontrados (BestCMax y BestSeqJobs).

Ya con la solución inicial se genera el vecindario, el cual es un conjunto de vectores (SeqJobs) que parten de la secuencia original y se hace un cambio en la representación del mismo (el cual se explicará en el siguiente apartado) que permite la exploración de posibles mejores soluciones al problema.

Después de hacer la exploración de los vecinos se almacena la mejor solución actual (correspondiente al vecindario explorado) y la mejor solución no tabú (es decir una solución que no se encuentra en la lista tabú).

Se verifica si la mejor solución actual se encuentra en la lista tabú, de no encontrarse se pasa a la siguiente etapa, la cual es verificar si esta solución es la mejor hasta ahora encontrada, de ser así se actualiza el mejor Cmax y la mejor secuencia (BestCMax y BestSeqJobs), se actualiza la lista tabú y se reinicia el contador de las iteraciones (ConIter = 0) y se procede a realizar nuevamente la generación del vecindario a partir de esta solución, si no es la mejor solución, de igual manera se actualiza la lista, pero se aumenta el contador de las iteraciones (ConIter = ConIter + 1) y se utiliza este resultado para realizar la nueva exploración de vecindarios siempre y cuando ConIter (contador de iteraciones) sea menor que MaxIter (máximo número de iteraciones permitidas) de ser igual el procedimiento ha llegado a su fin.

Pero si la mejor solución actual se encuentra en la lista tabú, se revisa si esta es mejor que la mejor solución encontrada hasta ahora, si esto es verdadero (criterio de aspiración) se actualiza el mejor Cmax y la mejor secuencia (BestCMax y BestSeqJobs), pero no se actualiza la lista tabú y se reinicia el contador de las iteraciones (ConIter = 0) y se procede a realizar nuevamente la generación del vecindario a partir de esta solución.

Pero si esta solución se encuentra en la lista tabú y no es la mejor solución encontrada hasta el momento, entonces se selecciona la mejor solución actual no tabú (solución que aun no se encuentra en la lista) y se procede a actualizar la lista, se aumenta el contador de las iteraciones (ConIter = ConIter + 1) y se realiza la siguiente exploración de vecindario siempre y cuando ConIter (contador de iteraciones) sea menor que MaxIter (máximo número de iteraciones permitidas) de ser igual el procedimiento ha llegado a su fin.

Este procedimiento se repite hasta el criterio de detención se haya cumplido (ConIter = MaxIter).

4.2.3.1 Vecindario

La selección del vecindario es un punto importante de la heurística, ya que este será el encargado de explorar y evolucionar en la consecución de las soluciones que permitan alcanzar el objetivo buscado, el cual es minimizar el makespan para el problema $J_m / no - w ait / C_{MAX}$.

Sin embargo debido a la representación del problema (representación por trabajos) no se utilizaron procedimientos sofisticados como los expuestos en la literatura por autores como Zhang et. al. (2002) y Blazewicz et. al. (2002), ya que estas propuestas hacen relación principalmente a representaciones por operaciones, además también por la

restricción misma del problema (no – wait) la cual no permite la utilización de los mismos.

Por lo anterior y para este trabajo se realizó la generación de vecinos a través de un procedimiento de intercambio entre trabajos consecutivos los cuales no hacen grandes cambios a la representación del problema por lo cual no afectan severamente la búsqueda de soluciones futuras pero permiten hacer una buena exploración de vecindarios para este tipo de problemas.

Con este procedimiento cada vecindario generar $\text{NumberOfJobs} - 1$ vecinos (es decir el numero total de trabajos menos uno), esto se puede observar con el ejemplo ilustrado en la figura 187, en este caso existen seis trabajos que parten de un secuencia original generando así cinco vecinos para ser explorados.

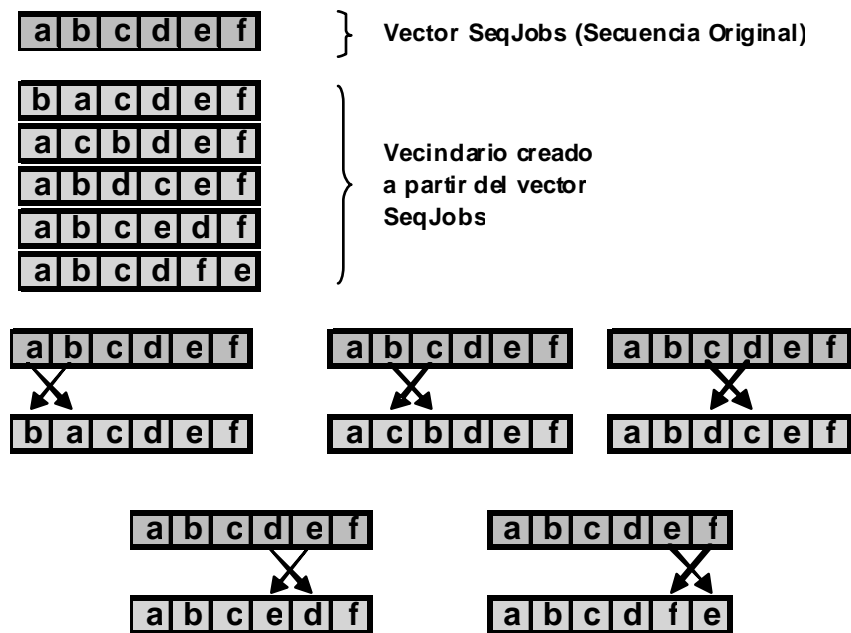


Figura 18: Generación de vecinos, mediante el intercambio de trabajos consecutivos.

4.3 Ejemplo de Aplicación para el Procedimiento JSSP No – Wait

Para tener una idea más clara del procedimiento utilizado para el **JSSP No – Wait** que permita minimizar el makespan, se utiliza un ejemplo de aplicación del mismo, el cual muestra el funcionamiento y desarrollo del procedimiento propuesto, este ejemplo comienza desde que se inicializan las máquinas (en tiempo cero) hasta que se hallan podido programar todas las operaciones de los trabajos que intervienen en el problema.

Para este ejemplo se utilizó un pequeño problema de tres trabajos por tres máquinas, el cual es el mismo que utilizan Schuster & Framiñan (2006) para la explicación del procedimiento propuesto por ellos.

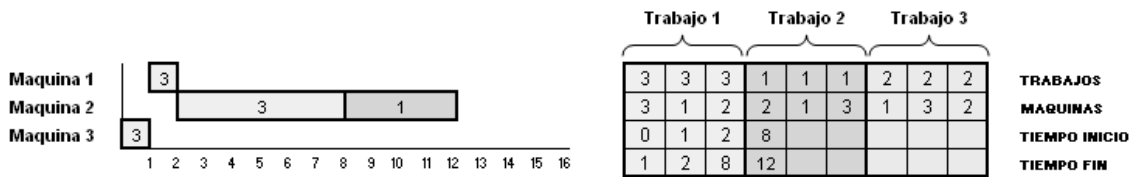


Figura 23: Ejemplo de aplicación. Paso 4.

Paso 5: Continuando en el mismo trabajo del paso anterior, se toma como referencia el tiempo de fin de la operación anterior y se toma como el tiempo de inicio para la segunda operación y se verifica si es posible programar en ese intervalo de tiempo como se muestra en la figura 24.

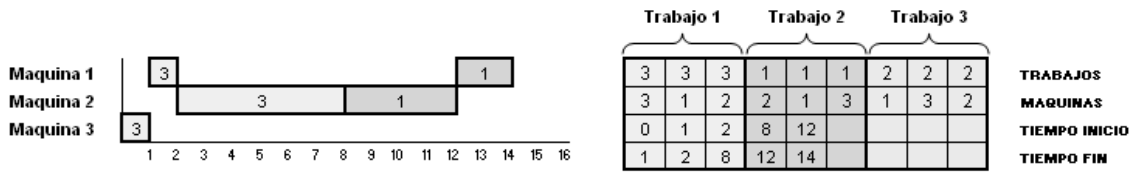


Figura 24: Ejemplo de aplicación. Paso 5.

Paso 6: Falta por programar la última operación del trabajo 1 ($SeqJobs(2) = 1$), de igual manera se toma el tiempo de inicio de esta operación igual al tiempo de fin de la operación anterior, esto con el fin de respetar la restricción de no – wait del problema y se verifica si se puede ingresar en el sistema (figura 25).

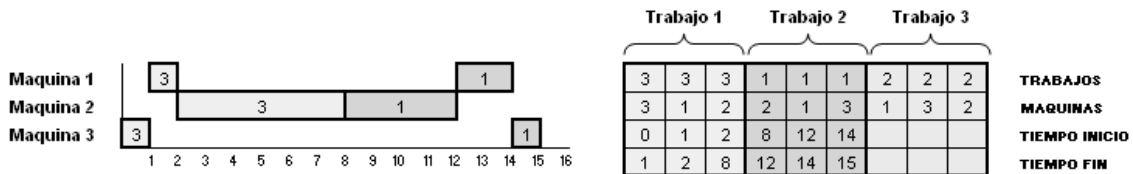


Figura 25: Ejemplo de aplicación. Paso 6.

Paso 7: Ya se han programado los dos primeros trabajos de la secuencia dada inicialmente. Se procede a programar el último trabajo ($SeqJobs(3) = 2$) empezando por la primera operación de este (en la máquina 1), como se ha mencionado anteriormente se genera el vector MenorTI y se encuentra un espacio disponible entre el tiempo 0 al 1, sin embargo el tiempo de proceso de la operación es mayor al espacio que se encuentra disponible como se ilustra en la figura 26.

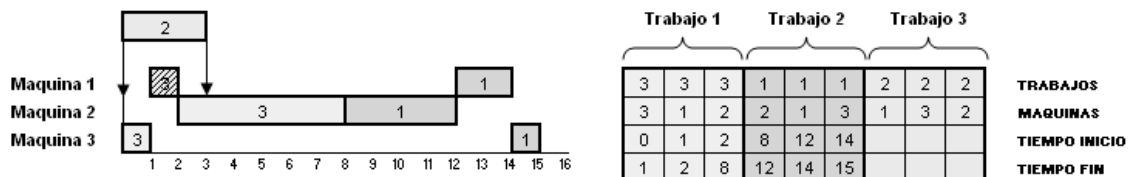


Figura 26: Ejemplo de aplicación. Paso 7.

Paso 8: Por lo descrito en el paso anterior se procede a buscar la mejor ubicación posible (menor tiempo en el que puede empezar) utilizando el vector MenorTI, dando como resultado iniciar la operación en el tiempo 2 (figura 27).

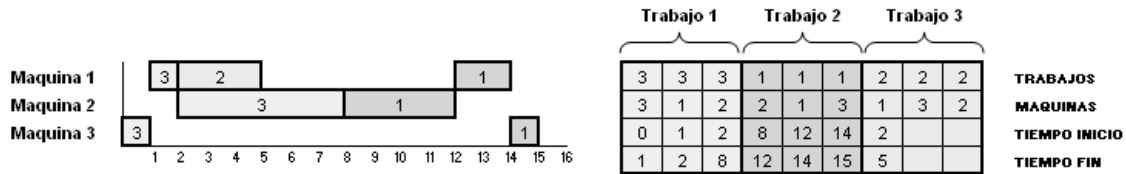


Figura 27: Ejemplo de aplicación. Paso 8.

Paso 9: Ahora se programará la segunda operación del trabajo 2 (SeqJobs (3) = 2), el cual podría iniciar en el tiempo 5 (tiempo determinación de la primera operación de este trabajo) si es posible como se muestra en la figura 28.

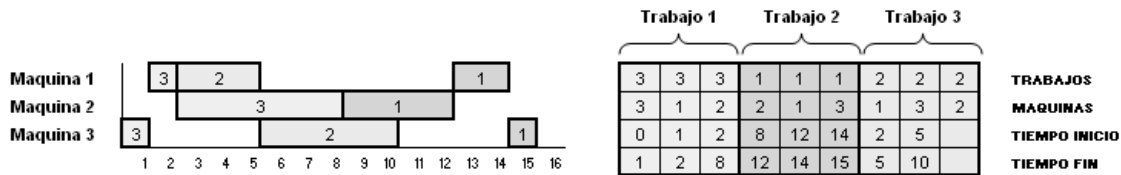


Figura 28: Ejemplo de aplicación. Paso 9.

Paso 10: Por último se procede ingresar la tercera operación del trabajo, sin embargo al asignarle el tiempo de inicio de esta operación igual al tiempo de fin de la operación anterior esta interrumpiría la operación del trabajo 1 (SeqJobs (2) = 1) que se realiza en la máquina 2 (figura 29).

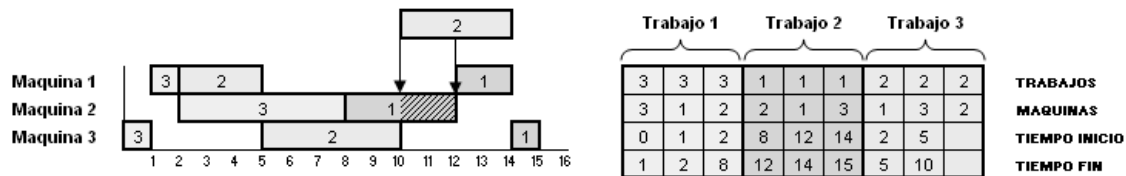


Figura 29: Ejemplo de aplicación. Paso 10.

Paso 11: En el paso anterior sucede uno de los casos donde se infrinjan la ubicación de la operación (apartado 4.2.2.4), esta infracción se detecta al comparar los tiempos de inicio y fin asignados preliminarmente a la operación contra la información almacenada en los vectores VectorTI y VectorTF (apartado 4.2.2.3) por lo cual no es posible asignar la operación en esta posición, por tanto se corre la operación actual una unidad de tiempo hacia adelante y de igual manera se corren todas las operaciones anteriores de ese trabajo una unidad de tiempo con el fin de respetar la restricción de no – wait, y se compara usando los vectores VectorTI y VectorTF que cada una de estas operaciones (que pertenecen al trabajo 2 (SeqJobs (3) = 2)) al desplazarse no interfieran con las operaciones de los otros trabajos ya programados en sus respectivas máquinas (figura 30).

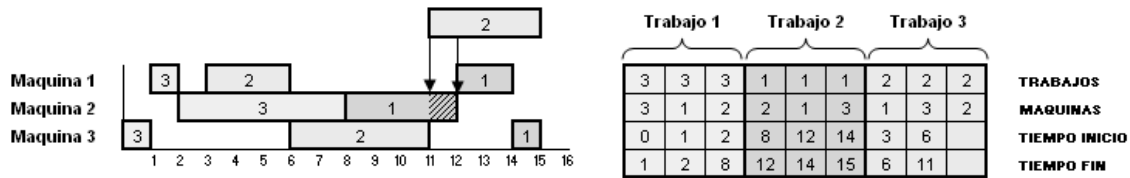


Figura 30: Ejemplo de aplicación. Paso 11.

Paso 12: Sin embargo al desplazarse una unidad de tiempo como se observa en la figura 27, aun la operación del trabajo sigue interfiriendo con la operación del trabajo 1 (SeqJobs (2) =1) que se realiza en la máquina 2, por lo cual el procedimiento descrito en el paso anterior (Paso 11) se repite corriendo nuevamente la operación y las operaciones antecesoras (operaciones de SeqJobs (3) =2 programadas anteriormente) como se observar en la figura 31.

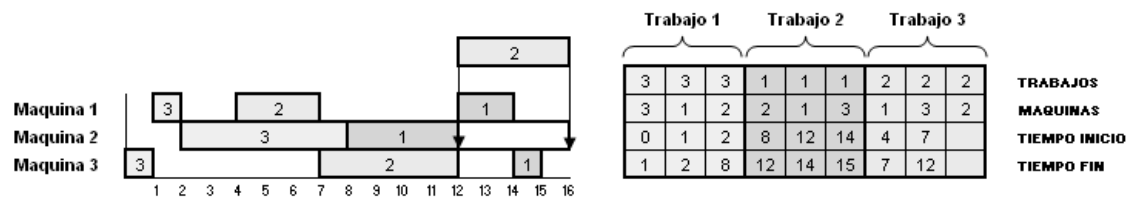


Figura 31: Ejemplo de aplicación. Paso 12.

Paso 13: al realizar el desplazamiento descrito anteriormente se observa (figura 31) que ya se puede asignar la operación al inicio del tiempo 12 respetando la restricción de no wait como se muestra en la figura 32 finalizando así la programación de todos los trabajos de esta caso en especial del scheduling en ambientes job shop.

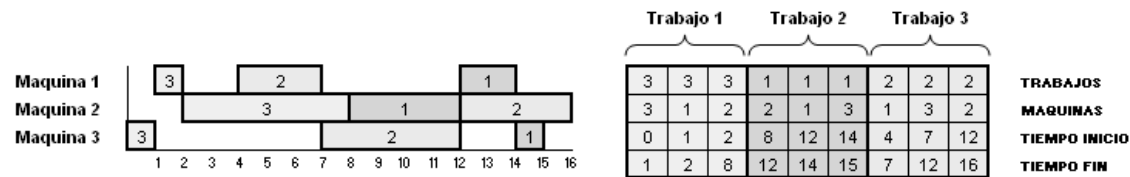


Figura 32: Ejemplo de aplicación. Paso 13.

Como se muestra en los pasos anteriores el **Procedimiento JSSP No -Wait** permite realizar la mejor secuenciación posible según una secuencia dada, para el caso de este ejemplo se logro un makespan igual a 16 unidades de tiempo.

5. RESULTADOS

Después de realizar el recorrido bibliográfico pertinente y haber explicado el funcionamiento del **Procedimiento JSSP No – Wait**. Se muestran los resultados del mismo probados contra 62 instancias clásicas de la literatura en lo que refiera a los problemas tipo job shop tomadas de <http://people.brunel.ac.uk/~mastjib/jeb/info.html>.

Estas instancias se dividen en dos grupos principales, instancias de tamaño pequeño e instancias de tamaño grande.

Las instancias pequeñas son:

Subgrupo	Trabajos (n)	Máquinas (m)
la01 - la5	10	5
la06 - la10	15	5
la16 - la20	10	10
orb1 - orb10	10	10
ft06	6	6
ft10	10	10

Tabla 5: Instancias pequeñas

Y las instancias grandes son:

Subgrupo	Trabajos (n)	Máquinas (m)
la11 - la 15	20	5
la21 - la25	15	10
la26 - la30	20	10
la31 - la35	30	10
la36 - la 40	15	15
swv01 - swv05	20	10
swv06 - swv10	20	15

Tabla 6: Instancias grandes

Además procedimiento descrito anteriormente fue implementado en VBA (Visual Basic for Applications) en el programa Excel del Microsoft Office y ejecutado en un computador personal Pentium IV de 2.2 Mhz y 256 MB de memoria RAM.

Los resultados que se mostraran se realizaron a través de una heurística de búsqueda local, para este caso búsqueda tabú los cuales se realizaron partiendo de tres tipos de solución inicial, SPT - rule Shoertest processing time (apartado 4.2.2.1), LPT - rule Longest processing time (apartado 4.2.2.1) y Random Selection (apartado 4.2.2.3) y los parámetros al aplicar el **Procedimiento JSSP No – Wait** se comentan el siguiente apartado.

5.1 Parámetros de la Búsqueda Tabú Para el Procedimiento JSSP No – Wait

Los parámetros utilizados se describen de la siguiente forma. Para el tamaño del vecindario se exploran $n-1$ vecinos (numero de trabajos menos uno) dada la

representación del problema, y el intercambio es entre trabajos consecutivos para evitar así grandes cambios en la representación original y afectar posibles exploraciones futuras que afectan la solución final.

Para criterio de aspiración se utiliza la mejor solución hasta el momento encontrada, pues es la más utilizada en la literatura (debido que solamente se debe omitir la función de lasita si realmente si logra un resultado mejor) y respecto al tamaño a la lista, según Pezcella & Merelli (2000) esta permite que el procedimiento busque la mejor solución entre vecinos no prohibidos, los vecinos están limitados por una lista (lista tabú la cual no debe ser muy larga ni muy corta) que permita una exploración hacia la evolución en búsqueda de la mejor solución, por lo anterior para este trabajo el tamaño de la lista se selecciono de manera proporcional al número de trabajos dada la representación del trabajo. Estos valores muestran en la tabla 7.

Número de trabajos	Tamaño de lista
6	2
10	4
15	6
20	8
30	10

Tabla 7: Tamaño de lista tabú según número de trabajos

Y para el criterio de parada se utilizo un valor constante para todas las instancias probadas, el cual después de 100 iteraciones sino se mejora la mejor solución la búsqueda se detiene (Max Iter = 100).

5.2 Resultados obtenidos

Para los resultados obtenidos con las diferentes soluciones iniciales, se realizaron comparaciones frente a la solución óptima, así como contra los publicados en la literatura especializada (siendo los únicos que han publicado Schuster & Framiñan 2003 y 2006) para los tres casos, y se seleccionaron el mejor para cada instancia para posteriormente comparar este frente al óptimo y la literatura, y por ultimo un caso especial de comparación, dado que en Schuster & Framiñan 2003 estos en su momento compararon los resultados de las grandes instancias contra la mejor heurística encontrada por ellos mismo en ese artículo, con lo cual en este trabajo se compara la solución inicial (solo para SPT y LPT) contra la heurística que ellos realizaron sus comparaciones para estas instancias.

La nomenclatura encontrada en las tablas de resultados para su interpretación es la siguiente:

n: Trabajos

m: Máquinas

Mejor: Mejor resultado obtenido del Algoritmo para esa instancia.

Subgrupo: agrupa las instancias del mismo tamaño (trabajos y máquinas).

VNS: Variable Neighborhood Search, procedimiento propuesto en Schuster & Framiñan 2003

GASA: A hybrid approximation strategy (Genetic Algorithms Simulated Annealing), procedimiento propuesto en Schuster & Framiñan 2003.

CLM: Complete local search with memory, procedimiento propuesto en Schuster & Framiñan 2006.

CLM: Complete local search with memory – time, procedimiento propuesto en Schuster & Framiñan 2006.

FGA – SPT: Procedimiento JSSP No – Wait usando como solución inicial la regla STP.

FGA – LPT: Procedimiento JSSP No – Wait usando como solución inicial la regla STP.

FGA – Rnd: Procedimiento JSSP No – Wait usando como solución Random Selection (selección aleatoria).

FGA Best: Selecciona el mejor de los resultados obtenidos para el **Procedimiento JSSP No – Wait** de las tres reglas de despacho.

PDR: Porcentaje de Desviación relativo frente al óptimo.

Para calcular el porcentaje de desviación relativo frente al óptimo se utilizó la siguiente fórmula:

$$PDR = \left(\frac{CMax(E) - CMax(Opt.)}{CMax(Opt.)} \right) \times 100$$

5.2.1 Resultados con SPT como solución inicial

Utilizando la regla SPT para obtener la solución inicial se obtuvieron los resultados de la tabla 8, donde se compara el porcentaje de desviación relativo de los cuatro algoritmos encontrados en la literatura (Schuster & Framiñan 2003 y 2006) y el propuesto en este trabajo (FGA – SPT) contra el óptimo (en el Anexo 1 se encuentran los resultados instancia por instancia)

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados (%) con el Optimo							
Subgrupo	n	m	GASSA	VNS	CLM - time	CLM	FGA - SPT
la01 - la5	10	5	6,297	12,388	2,603	2,088	6,500
la06 - la10	15	5	5,558	15,678	7,635	4,530	14,522
la11 - la15	20	5	5,355	15,497	4,371	0,218	10,667
la16 - la20	10	10	7,512	15,749	5,298	4,920	14,487
la21 - la25	15	10	3,629	17,182	7,857	1,930	11,243
la26 - la30	20	10	4,430	13,684	6,864	4,532	14,976
la31 - la35	30	10	11,683	10,939	8,292	2,561	16,618
la36 - la40	15	15	4,307	16,912	7,774	1,297	10,454
orb1 - orb10	10	10	2,421	13,251	2,458	1,439	7,877
swv01 - swv05	20	10	4,446	9,310	3,769	0,786	16,627
swv06 - swv10	20	15	6,549	9,600	5,910	1,221	18,459
ft06	6	6	0,000	0,000	0,000	0,000	0,000
ft10	10	10	0,809	12,632	0,747	0,747	7,094

Tabla 8: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando SPT.

Por ejemplo para el subgrupo de las instancias la01 a la05 el algoritmo GASA esta 6,297% en promedio por encima del óptimo, el VNS 12,338% CLM – time 2,603%, CLM 2,088% y el FGA – SPT 6,500%, así para cada uno de los subgrupos de instancias analizadas. Y comparando contra los propuestos en la literatura se obtuvieron los siguientes resultados:

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Subgrupo</u>	<u>n</u>	<u>m</u>	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>
la01 - la5	10	5	0,230	-5,102	3,810	4,297
la06 - la10	15	5	8,536	-0,953	6,436	9,667
la11 - la 15	20	5	5,060	-3,733	6,076	10,427
la16 - la20	10	10	6,677	-0,700	8,864	9,262
la21 - la25	15	10	7,306	-5,076	3,193	9,167
la26 - la30	20	10	10,121	1,176	7,586	13,279
la31 - la35	30	10	4,415	5,309	7,697	13,689
la36 - la 40	15	15	5,878	-5,472	2,555	9,030
orb1 - orb10	10	10	6,060	-3,884	6,199	6,693
swv01 - swv05	20	10	11,649	6,681	12,345	15,744
swv06 - swv10	20	15	11,214	8,149	11,880	17,036
ft06	6	6	0,000	0,000	0,000	0,000
ft10	10	10	6.235	-4.917	6.300	6.300

Tabla 9: Gonzáles (2006) utilizando SPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando SPT.

Los resultados de la tabla 9 se pueden interpretar de la siguiente manera, el FGA – SPT comparado frente al VNS lo supera en promedio 5,102% en las instancias la01 a la05, mientras que con el algoritmo GASSA, CLM– time y CLM superan en promedio al FGA – SPT en 0,230%, 3,810% y 4,297% respectivamente, y así para cada subgrupo de instancias (las comparaciones instancia por instancia se encuentra en el anexo 2)

También se compararon los resultados según el tamaño de la instancia y para el total de las instancias analizadas utilizando SPT en primer lugar contra el óptimo.

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo</u>					
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA - SPT</u>
Pequeñas instancias	4,513	13,490	3,815	2,697	9,756
Grandes Instancias	5,771	13,303	6,405	1,364	14,149
Total Instancias	5,223	13,384	5,277	1,944	12,236

Tabla 10: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando SPT por tamaño de instancias y el total de instancias.

Donde solamente las instancias pequeñas con el FGA – SPT los resultados están por debajo del 10% en promedio del óptimo y en general con éste procedimiento usando como solución inicial SPT en promedio las instancias están 12,236% por encima del optimo. Y en segundo lugar comparando por tamaño y el total de las instancias contra la literatura se muestran los resultados de la tabla 11, globalmente el GASSA, CLM – time y CLM superan en promedio al FGA – SPT en 6,422%, 6,357% y 9,841% respectivamente, mientras que el procedimiento propuesto supera en promedio al VNS 1,032%.

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>				
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>
Pequeñas instancias	4,443	-3,673	5,091	6,232
Grandes Instancias	7,949	1,005	7,333	12,625
Total Instancias	6,422	-1,032	6,357	9,841

Tabla 11: Gonzales (2006) utilizando SPT Comparado porcentualmente (%) frente a Framinan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias.

5.2.2 Resultados con LPT como solución inicial

Con la regla LPT se compara el porcentaje de desviación relativo de los cuatro algoritmos encontrados en la literatura (Schuster & Framiñan 2003 y 2006) y el propuesto en este trabajo (FGA – LPT) contra el óptimo como se muestran en la tabla 12 (en el Anexo 3 se encuentran los resultados instancia por instancia).

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados (%) con el Optimo							
Subgrupo	n	m	GASSA	VNS	CLM - time	CLM	FGA - LPT
la01 - la5	10	5	6,297	12,388	2,603	2,088	10,230
la06 - la10	15	5	5,558	15,678	7,635	4,530	14,647
la11 - la 15	20	5	5,355	15,497	4,371	0,218	12,920
la16 - la20	10	10	7,512	15,749	5,298	4,920	12,371
la21 - la25	15	10	3,629	17,182	7,857	1,930	9,267
la26 - la30	20	10	4,430	13,684	6,864	4,532	11,408
la31 - la35	30	10	11,683	10,939	8,292	2,561	9,913
la36 - la 40	15	15	4,307	16,912	7,774	1,297	8,286
orb1 - orb10	10	10	2,421	13,251	2,458	1,439	8,261
swv01 - swv05	20	10	4,446	9,310	3,769	0,786	11,396
swv06 - swv10	20	15	6,549	9,600	5,910	1,221	14,553
ft06	6	6	0,000	0,000	0,000	0,000	0,000
ft10	10	10	0,809	12,632	0,747	0,747	5,476

Tabla 12: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando LPT.

Para el subgrupo de las instancias la01 a la05 para FGA – LPT están 10,230% en promedio por encima del óptimo, y así para cada uno de los subgrupos de instancias analizadas. Las comparaciones de los procedimientos de la literatura contra el óptimo se tratan de la misma forma.

Comparando FGA – LPT contra los propuestos en la literatura se muestran los resultados de la tabla 13 y las comparaciones instancia por instancia se encuentra en el anexo 4.

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)							
Subgrupo	n	m	GASSA	VNS	CLM - time	CLM	
la01 - la5	10	5	3,279	0,658	3,880	3,880	
la06 - la10	15	5	8,636	-0,839	6,559	9,767	
la11 - la 15	20	5	7,205	-1,707	8,290	12,679	
la16 - la20	10	10	4,491	-2,849	6,808	7,168	
la21 - la25	15	10	5,496	-6,749	1,353	7,223	
la26 - la30	20	10	6,713	-1,971	4,226	9,739	
la31 - la35	30	10	-1,566	-0,813	1,572	7,200	
la36 - la 40	15	15	3,852	-7,172	0,463	6,976	
orb1 - orb10	10	10	6,717	-3,100	6,842	7,327	
swv01 - swv05	20	10	6,655	1,900	7,443	10,509	
swv06 - swv10	20	15	7,518	4,541	8,246	13,194	
ft06	6	6	0,000	0,000	0,000	0,000	
ft10	10	10	4,630	-6,354	4,694	4,694	

Tabla 13: Gonzáles (2006) utilizando SPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando LPT.

Con los resultados anteriores (tabla 13), el FGA – SPT comparado frente al GASSA y VNS, CLM – time y CLM es superado en promedio 3,279%, 0,658%, 3,880% y 3,880% respectivamente. Ahora compararon los resultados según el tamaño de la instancia y para el total de las instancias analizadas usando LPT contra el óptimo se encontró que para las instancias pequeñas la desviación sobre el óptimo en promedio es de 10,160%, para instancias grandes es de 11,106% y para el total de las instancias la desviación promedio es de 10,694% como se muestra en la tabla 14, también se muestra la comparación de los algoritmos de Shuster y Framinian 2003 y 2006.

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo</u>					
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA - LPT</u>
Pequeñas instancias	10.160	13.490	3.815	2.697	10.160
Grandes instancias	11.106	13.303	6.405	1.364	11.106
Total Instancias	5.223	13.384	5.277	1.944	10.694

Tabla 14: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando LPT por tamaño de instancias y el total de instancias.

Comparando por tamaño y el total de las instancias contra la literatura se muestran los resultados de la tabla 15, al igual que con LPT globalmente pero en menor proporción, de la siguiente forma GASSA, CLM – time y CLM superan en promedio al FGA – LPT en 4,707%, 4,650% y 8,043% respectivamente, mientras que el procedimiento propuesto supera en promedio al VNS 2,670%.

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>				
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>
Pequeñas instancias	4.166	-3.915	4.826	5.966
Grandes instancias	5.125	-1.710	4.513	9.646
Total Instancias	4.707	-2.670	4.650	8.043

Tabla 15: González (2006) utilizando LPT Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) tamaño de instancias y el total de instancias.

5.2.3 Resultados con Rnd como solución inicial

Para la regla Rnd contra el óptimo se obtuvieron los resultados de la tabla 16, así como para los otros procedimientos, en el Anexo 5 se encuentran los resultados instancia por instancia.

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados (%) con el Optimo</u>							
<u>Subgrupo</u>	<u>n</u>	<u>m</u>	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA - Rnd</u>
la01 - la5	10	5	6,297	12,388	2,603	2,088	2,870
la06 - la10	15	5	5,558	15,678	7,635	4,530	9,814
la11 - la 15	20	5	5,355	15,497	4,371	0,218	6,542
la16 - la20	10	10	7,512	15,749	5,298	4,920	8,709
la21 - la25	15	10	3,629	17,182	7,857	1,930	6,854
la26 - la30	20	10	4,430	13,684	6,864	4,532	6,381
la31 - la35	30	10	11,683	10,939	8,292	2,561	8,867
la36 - la 40	15	15	4,307	16,912	7,774	1,297	10,363
orb1 - orb10	10	10	2,421	13,251	2,458	1,439	4,269
swv01 - swv05	20	10	4,446	9,310	3,769	0,786	7,807
swv06 - swv10	20	15	6,549	9,600	5,910	1,221	9,218
ft06	6	6	0,000	0,000	0,000	0,000	0,000
ft10	10	10	0,809	12,632	0,747	0,747	3,671

Tabla 16: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando Rnd.

Para las instancias la01 a la05 usando FGA – Rnd, este se desvía en promedio 2,870%, de igual forma para cada uno de los subgrupos de instancias analizadas. Y comparando contra los propuestos en la literatura se obtuvieron los siguientes resultados:

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>							
<u>Subgrupo</u>	<u>n</u>	<u>m</u>	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	
la01 - la5	10	5	-3,089	-8,323	0,348	0,830	
la06 - la10	15	5	8,536	-0,953	6,436	9,667	
la11 - la 15	20	5	5,060	-3,733	6,076	10,427	
la16 - la20	10	10	6,677	-0,700	8,864	9,262	
la21 - la25	15	10	7,306	-5,076	3,193	9,167	
la26 - la30	20	10	10,121	1,176	7,586	13,279	
la31 - la35	30	10	4,415	5,309	7,697	13,689	
la36 - la 40	15	15	5,878	-5,472	2,555	9,030	
orb1 - orb10	10	10	6,060	-3,884	6,199	6,693	
swv01 - swv05	20	10	11,649	6,681	12,345	15,744	
swv06 - swv10	20	15	11,214	8,149	11,880	17,036	
ft06	6	6	0,000	0,000	0,000	0,000	
ft10	10	10	6,235	-4,917	6,300	6,300	

Tabla 17: Gonzáles (2006) utilizando Rnd Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) utilizando RND.

La tabla 17 muestra el FGA – Rnd comparado frente al GASSA lo supera en promedio 3,089% en las instancias la01 a la05 y de igual manera supera al VNS 8,323%, mientras que con el algoritmo CLM– time y CLM superan en promedio al FGA – Rnd en 0,348% y 0,830% respectivamente, y así para cada subgrupo de instancias (las comparaciones instancia por instancia se encuentra en el anexo 6).

Compararon los resultados según el tamaño de la instancia y para el total de las instancias contra el óptimo usando Rnd se obtuvieron los siguientes valores (tabla 18):

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo</u>					
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA - Rnd</u>
Pequeñas instancias	10.160	13.490	3.815	2.697	5.679
Grandes instancias	11.106	13.303	6.405	1.364	8.005
Total Instancias	5.223	13.384	5.277	1.944	6.992

Tabla 18: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo utilizando Rnd por tamaño de instancias y el total de instancias.

En comparación con las dos soluciones anteriores (SPT y LPT) con el FGA – Rnd los resultados en promedio son mejores y en comparación sobre el óptimo en promedio las instancias pequeñas están 6,679% encima de éstas, instancias grandes 8,005% y del total de instancias 6,992%.

Y en comparación contra la literatura se muestran los resultados de la tabla 19.

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>				
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>
Pequeñas instancias	1.168	-6.666	1.831	2.934
Grandes instancias	2.182	-4.496	1.566	6.577
Total Instancias	1.740	-5.441	1.681	4.990

Tabla 19: González (2006) utilizando Rnd Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias.

Globalmente el GASSA, CLM – time y CLM superan en promedio al FGA – Rnd en 1,740%, 1,681% y 4,990% respectivamente, mientras que el procedimiento propuesto supera en promedio al VNS 5,441%, siendo con esta regla los mejores resultados obtenidos en promedio utilizando el **Procedimiento JSSP No – Wait**.

5.2.4 Mejores Resultados

Con los resultados anteriormente descritos (apartados 5.2.1, 5.2.2 y 5.2.3) se seleccionaron los mejores de éstos obtenidos para cada una de las instancias y se compara contra el porcentaje de desviación relativo contra el óptimo y los propuesto en la literatura (Schuster & Framiñan 2003 y 2006)

En comparación con el óptimo estos resultados se muestran en la tabla 20, y la comparación instancia por instancia se encuentran en el anexo 7.

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados (%) con el Optimo</u>							
<u>Subgrupo</u>	<u>n</u>	<u>m</u>	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA Best</u>
la01 - la5	10	5	6,297	12,388	2,603	2,088	2,870
la06 - la10	15	5	5,558	15,678	7,635	4,530	9,012
la11 - la 15	20	5	5,355	15,497	4,371	0,218	6,133
la16 - la20	10	10	7,512	15,749	5,298	4,920	8,140
la21 - la25	15	10	3,629	17,182	7,857	1,930	6,095
la26 - la30	20	10	4,430	13,684	6,864	4,532	6,054
la31 - la35	30	10	11,683	10,939	8,292	2,561	7,533
la36 - la 40	15	15	4,307	16,912	7,774	1,297	6,530
orb1 - orb10	10	10	2,421	13,251	2,458	1,439	3,413
swv01 - swv05	20	10	4,446	9,310	3,769	0,786	6,416
swv06 - swv10	20	15	6,549	9,600	5,910	1,221	8,813
ft06	6	6	0,000	0,000	0,000	0,000	0,000
ft10	10	10	0,809	12,632	0,747	0,747	3,671

Tabla 20: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Óptimo.

Estos resultados (tabla 20) muestran que para las instancias la01 a la05 los FGA Best están únicamente en promedio 2,870% encima del óptimo, y de igual manera para cada uno de los subgrupos de instancias analizadas con los mejores resultados. Y comparando contra los propuestos en la literatura se obtuvieron los resultados de la tabla 21:

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>							
<u>Subgrupo</u>	<u>n</u>	<u>m</u>	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	
la01 - la5	10	5	-3,134	-8,363	0,303	0,785	
la06 - la10	15	5	3,276	-5,733	1,341	4,403	
la11 - la 15	20	5	0,779	-7,644	1,718	5,904	
la16 - la20	10	10	0,638	-6,397	2,794	3,152	
la21 - la25	15	10	2,394	-9,451	-1,548	4,110	
la26 - la30	20	10	1,611	-6,653	-0,755	4,512	
la31 - la35	30	10	-3,695	-3,000	-0,653	4,863	
la36 - la 40	15	15	2,149	-8,741	-1,129	5,220	
orb1 - orb10	10	10	0,585	-8,734	0,712	1,174	
swv01 - swv05	20	10	1,890	-2,643	2,608	5,593	
swv06 - swv10	20	15	2,147	-0,705	2,774	7,523	
ft06	6	6	0,000	0,000	0,000	0,000	
ft10	10	10	2,840	-7,956	2,903	2,903	

Tabla 21: Gonzáles (2006) Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006)

En comparación con la literatura se obtienen resultados buenos, pues en el peor de los casos en promedio superan a los mejores resultados (FGA Best) en un 6% aproximadamente como se muestra en la tabla 21 (en el anexo 8 se encuentran las comparaciones de cada una de las instancias).

Compararon los resultados según el tamaño de la instancia y para el total de las instancias analizadas utilizando SPT en primer lugar contra el óptimo.

<u>Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo</u>					
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>	<u>FGA - Rnd</u>
Pequeñas instancias	10.160	13.490	3.815	2.697	5.108
Grandes instancias	11.106	13.303	6.405	1.364	6.797
Total Instancias	5.223	13.384	5.277	1.944	6.061

Tabla 22: González (2006) y Framiñan & Schuster (2003) y (2006) Comparado porcentualmente (%) contra el Optimo por tamaño de instancias y el total de instancias.

Los resultados de la tabla 22 muestran que en promedio la desviación frente al óptimo es de 5,108%, 6,797% para las instancias de tamaño pequeño y grande respectivamente, y para el total de las instancias la desviación promedio es 6,061%.

Ahora comparando los mejores resultados (FGA Best) contra los expuestos por Schuster & Framiñan 2003 y 2006, los resultados obtenidos en la tabla 23 son buenos, dado que en términos generales contra el GASSA y CLM – time la diferencia en que superan los FGA Best no es mayor al 1% en promedio además se obtienen mejores resultados frente al VNS, ya que los FGA Best son mejores un 6,256% en promedio para el total de las instancias, y la mayor diferencia esta contra el CLM, el cual es mejor tan solo en 4,078% en promedio.

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>				
	<u>GASSA</u>	<u>VNS</u>	<u>CLM - time</u>	<u>CLM</u>
Pequeñas instancias	0.617	-7.173	1.283	2.378
Grandes instancias	1.039	-5.548	0.431	5.389
Total Instancias	0.855	-6.256	0.802	4.078

Tabla 23: Gonzales (2006) Comparado porcentualmente (%) frente a Framiñan & Schuster (2003) y (2006) por tamaño de instancias y el total de instancias.

5.2.5 Resultados de las Soluciones Iniciales (SPT y LPT) contra la Mejor Heurística de Schuster y Framiñan 2003

Como se comento en el apartado 5.2 en Schuster y Framiñan 2003, estos autores hacen las comparaciones de las instancias grandes a través de una heurística utilizando una regla de despacho (con el procedimiento propuesto, puesto esta solución debe ser no - wait), con esto generan la solución inicial y comparan los resultados. Por lo anterior se comparan las soluciones iniciales (las mejores) de Schuster y Framiñan 2003 contra las soluciones iniciales de FGA – SPT y FGA – LPT.

Estos resultados se muestran en la tabla 24, la cual se puede interpretar de la siguiente manera, para la instancia la06 la mejor solución inicial (Cmax) es de 1863, y la solución inicial con FGA – SPT es de 1804 y con FGA – LPT es de 1770, superando a Schuster y Framiñan en 3,17% y 4,99% respectivamente, y así para todas las instancias grandes.

SPT y LPT contra la Mejor Heurística de Schuster y Framiñan 2003
 Gonzalez (2006) y Framiñan & Schuster (2003) Solucion Inicial

Instancia Nombre	Mejor		FGA - SPT		FGA - LPT		
	n	m	Heurística Sol.	(%)	Sol.	(%)	
la06	15	5	1863	1804	3.17	1770	4.99
la07	15	5	1965	1964	0.05	1646	16.23
la08	15	5	1910	1670	12.57	1640	14.14
la09	15	5	2189	2125	2.92	1871	14.53
la10	15	5	2045	1805	11.74	1666	18.53
la11	20	5	2821	2360	16.34	2323	17.65
la12	20	5	2434	2050	15.78	2089	14.17
la13	20	5	2650	2512	5.21	2278	14.04
la14	20	5	2662	2604	2.18	2179	18.14
la15	20	5	2765	2164	21.74	2318	16.17
la21	15	10	3574	2856	20.09	2748	23.11
la22	15	10	2879	2912	-1.15	3015	-4.72
la23	15	10	3228	3101	3.93	2972	7.93
la24	15	10	3075	3051	0.78	2905	5.53
la25	15	10	2985	2842	4.79	2797	6.30
la26	20	10	4668	3591	23.07	4037	13.52
la27	20	10	4674	4523	3.23	3730	20.20
la28	20	10	4478	4533	-1.23	3663	18.20
la29	20	10	4117	3571	13.26	3519	14.53
la30	20	10	4097	4325	-5.57	3673	10.35
la31	30	10	6519	6232	4.40	4881	25.13
la32	30	10	6912	6370	7.84	5115	26.00
la33	30	10	6445	5394	16.31	4928	23.54
la34	30	10	6380	5738	10.06	5352	16.11
la35	30	10	6563	5789	11.79	5228	20.34
la36	15	15	4526	4109	9.21	3807	15.89
la37	15	15	4766	5164	-8.35	3948	17.16
la38	15	15	4426	4524	-2.21	3732	15.68
la39	15	15	4295	4058	5.52	3790	11.76
la40	15	15	4099	3499	14.64	3598	12.22
swv01	20	10	3824	3479	9.02	3339	12.68
swv02	20	10	3800	3881	-2.13	3273	13.87
swv03	20	10	3797	3713	2.21	3684	2.98
swv04	20	10	3895	4039	-3.70	3453	11.35
swv05	20	10	3836	3879	-1.12	3093	19.37
swv06	20	15	5163	5008	3.00	4777	7.48
swv07	20	15	5076	4333	14.64	4213	17.00
swv08	20	15	5547	5732	-3.34	4720	14.91
swv09	20	15	5117	4625	9.62	4489	12.27
swv10	20	15	5347	5223	2.32	5205	2.66
Mejoras			6.32		14.05		

Tabla 24: soluciones iniciales (las mejores) de Schuster y Framiñan 2003 contra las soluciones iniciales de FGA – SPT y FGA – LPT.

Con los resultados obtenidos anteriormente para la solución inicial usando SPT, esta supera en promedio al las soluciones con la mejor heurística de Schuster Framiñan (2003) para las instancias grandes en 6,32% y usando la LPT, esta la supera en un 14,05%. Para este caso la mejor solución de inicio es usando la regla LPT, y además de las 40 instancias grandes solamente una de las propuestas de Schuster y Framiñan 2003 es mejor, es decir el 97,5% de las soluciones con LPT son mejores que las de artículo, mientras que con la regla SPT estas superan en un 77,5% a la de estos autores.

5.2.5 Tiempos computacionales

En relación a los tiempos computacionales, comparando con los de Schuster & Framiñan 2003 y 2006, los expuestos en este trabajo son ampliamente superados por estos autores, dado la estructura de programación y el funcionamiento del **JSSP No – WaiT**, sin embargo vale la pena decir que el procedimiento propuesto ordena de la mejor manera posible una secuencia dada (Mejor CMax para esa secuencia), mientras que las propuestas de los autores esta solución puede variar.

Además también cabe destacar que en algunos casos para secuencias del mismo tamaño (número de trabajos y cantidad de máquinas) hay grandes variaciones en los tiempos de computo, y esto es dado por la búsqueda tabú, pues se puede dar el caso que una instancia encuentre una mejor solución cuando se esta aproximando al numero máximo de iteraciones, con lo cual se reiniciaría el contador de estas en cero, mientras que para otras instancias (iguales en tamaño) esto no suceda. Los tiempos obtenidos por las tres formas de abordar el **JSSP No – WaiT** y los expuestos por Schuster & Framiñan 2003 y 2006 se muestran a continuación en la tabla 25:

Tiempos Computacionales Reportados para cada una de las instancias (segundos)
Gonzalez (2006) y Framiñan & Schuster (2003) y (2006)

Instancia Nombre	n	m	GASSA*	VNS*	CLM	Rnd	FGA	
							SPT	LPT
la01	10	5	0.2	0.0	1.5	202.3	247.7	33.0
la02	10	5	0.2	0.0	1.5	196.5	38.7	71.9
la03	10	5	0.2	0.0	1.9	123.5	75.8	32.3
la04	10	5	0.2	0.0	1.6	236.2	46.5	52.1
la05	10	5	0.2	0.0	2.2	101.6	81.4	73.2
la06	15	5	0.7	0.0	6.4	424.1	649.0	669.5
la07	15	5	0.7	0.0	8.1	330.5	1266.1	634.9
la08	15	5	0.7	0.0	5.7	314.9	575.9	485.5
la09	15	5	0.7	0.0	5.6	383.0	1812.6	766.0
la10	15	5	0.7	0.0	12.7	335.8	525.6	449.3
la16	10	10	0.4	0.0	4.4	622.7	406.6	412.3
la17	10	10	0.3	0.0	3.0	415.0	401.3	465.8
la18	10	10	0.4	0.0	0.1	477.0	404.0	411.2
la19	10	10	0.3	0.0	3.9	599.8	356.1	458.9
la20	10	10	0.4	0.0	4.0	1130.9	455.3	501.3
orb1	10	10	0.3	0.0	3.0	689.2	385.3	385.3
orb2	10	10	0.4	0.0	2.4	798.4	349.0	349.0
orb3	10	10	0.3	0.0	3.5	556.0	392.6	392.6
orb4	10	10	0.3	0.0	2.5	974.1	514.2	514.2
orb5	10	10	0.4	0.0	4.0	699.3	379.2	379.2
orb6	10	10	0.4	0.0	3.4	548.0	600.5	600.5
orb7	10	10	0.4	0.0	2.5	602.3	153.5	153.5
orb8	10	10	0.4	0.0	3.7	513.6	536.9	536.9
orb9	10	10	0.3	0.0	2.4	625.3	315.2	315.2
orb10	10	10	0.3	0.0	2.5	663.8	426.2	426.2

Tiempos Computacionales Reportados para cada una de las instancias (segundos)
Gonzalez (2006) y Framiñan & Schuster (2003) y (2006)

Instancia Nombre	n m		GASSA*	VNS*	CLM	FGA		
	n	m				Rnd	SPT	LPT
ft06	6	6	0.1	0.0	0.1	8.8	4.8	6.7
ft10	10	10	0.4	0.0	2.3	643.5	317.6	926.0
la11	20	5	1.8	0.0	21.7	1475.5	1335.1	1295.6
la12	20	5	1.6	0.0	28.3	1158.1	1702.9	1365.7
la13	20	5	1.8	0.0	24.5	641.5	1696.7	1302.6
la14	20	5	1.7	0.0	20.6	1071.6	2030.2	2654.7
la15	20	5	1.6	0.0	25.1	838.8	867.5	810.9
la21	15	10	1.3	0.0	20.0	2559.9	3125.6	2378.6
la22	15	10	1.2	0.0	10.5	2045.3	2987.2	3088.7
la23	15	10	1.3	0.0	19.5	2494.5	4587.0	5985.5
la24	15	10	1.2	0.0	14.5	1653.3	3331.6	2754.8
la25	15	10	1.3	0.0	16.6	2020.7	4189.7	2259.4
la26	20	10	2.9	0.0	49.5	5464.6	4565.3	4123.8
la27	20	10	3.0	0.0	66.4	5897.4	4406.2	14244.4
la28	20	10	3.0	0.0	53.2	6596.4	5512.2	5209.1
la29	20	10	2.9	0.0	51.0	6582.4	4918.4	4025.7
la30	20	10	3.1	0.0	82.7	8574.1	4598.3	4388.8
la31	30	10	8.8	0.0	348.8	30120.3	27117.3	30256.9
la32	30	10	10.3	0.0	134.4	27862.2	23537.4	22354.1
la33	30	10	9.6	0.0	293.1	32668.2	23751.7	23569.0
la34	30	10	9.6	0.0	166.3	30216.9	24574.6	24472.3
la35	30	10	8.7	0.0	301.2	32331.8	27492.3	26894.2
la36	15	15	1.8	0.0	25.6	3198.75	7475.5	6190.7
la37	15	15	1.8	0.0	19.8	7442.328	3427.3	9914.4
la38	15	15	2.1	0.0	29.0	2843.9	5096.0	7179.2
la39	15	15	1.9	0.0	28.2	2856.5	4235.9	7963.6
la40	15	15	1.9	0.0	21.6	2821.2	3849.0	4581.6
swv01	20	10	3.2	0.0	29.6	6243.1	8044.1	8279.5
swv02	20	10	3.4	0.0	61.6	6707.8	7665.0	5411.7
swv03	20	10	3.2	0.0	30.4	3932.4	13009.6	7782.0
swv04	20	10	3.3	0.0	33.4	4842.4	4537.6	6293.1
swv05	20	10	3.2	0.3	53.3	3818.4	5087.1	8618.3
swv06	20	15	5.0	0.0	84.3	10512.9	6887.9	5505.7
swv07	20	15	4.1	0.0	45.0	12455.1	8072.7	9366.7
swv08	20	15	4.9	0.0	122.8	14161.7	4735.9	5874.3
swv09	20	15	4.5	0.0	69.0	10140.5	8028.7	5749.4
swv10	20	15	4.8	0.0	59.6	14627.4	4209.9	12350.0

Tabla 25: Tiempos Computacionales Reportados para cada una de las instancias (segundos) Framiñan & Schuster (2003) y (2006) comparados González (2006)

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El Procedimiento JSSP No – Wait presentado en este trabajo de investigación, es un avance más hacia la solución de los problemas tipo job shop con restricción de no – wait, pues este genera el mejor makespan de una secuencia dada, el cual es un aporte a esta clase de problemas que no ha sido tratado a profundidad en la literatura dada su estructura especial de secuenciación y programación.

Diversos autores hacen hincapié en la importancia de poder iniciar los procedimientos de búsqueda local con una buena solución de inicio, y dentro de este tipo de problemas los autores que lo han tratado coinciden en que una buena solución se puede generar partiendo de una regla de despacho o prioridad, ya que estas son de rápida implementación y generan a partir de estos buenos resultados finales, este postulado se cumple en este trabajo pues utilizando estas reglas, los mejores resultados se encuentran dentro del 6% en promedio para el total de las instancias probadas.

Dada la estructura especial del problema, la representación por trabajos, permiten realizar una representación adecuada del mismo, y no se utilizaron procedimientos sofisticados como los expuestos en la literatura por autores como Zhang et. Al. (2002) y Blazewicz et. al. (2002), ya que estas propuestas hacen relación principalmente a representaciones por operaciones, y dada la restricción misma del no – wait no se permite la utilización de estas representaciones expuestas en la literatura.

En relación a los resultados computacionales se puede decir que, de las tres soluciones iniciales tomadas para desarrollar el procedimiento los mejores resultados comparados en relación al óptimo (porcentaje de desviación relativo) se presentan en el siguiente orden, Rnd, LPT y SPT con los siguientes valores 6,992%, 10,694% y 12,236% respectivamente,

Comparando contra la literatura utilizando SPT los resultados obtenidos sobre el total de instancias son los siguientes, contra el GASSA, CLM – time y CLM superan en promedio al FGA – SPT en 6,422%, 6,357% y 9,841% respectivamente, mientras que el FGA - SPT supera en promedio al VNS 1,032%, usando LPT GASSA, CLM – time y CLM superan en promedio al FGA – LPT en 4,707%, 4,650% y 8,043% respectivamente, mientras que el procedimiento supera en promedio al VNS 2,670% y usando FGA – Rnd el GASSA, CLM – time y CLM lo superan en promedio en 1,740%, 1,681% y 4,990% respectivamente, mientras que este supera en promedio al VNS 5,441%, siendo con esta regla los mejores resultados obtenidos en promedio.

Al seleccionar los mejores resultados de cada instancia la desviación promedio frente al óptimo es de 5,108%, 6,797% para las instancias de tamaño pequeño y grande respectivamente, y para el total de las instancias la desviación promedio es 6,061%.

Comparando los mejores resultados (FGA Best) contra los expuestos por Schuster & Framiñan 2003 y 2006, en términos generales contra el GASSA y CLM – time la

diferencia en que superan los FGA Best no es mayor al 1% (0,855 y 0,802 respectivamente) en promedio además se obtienen mejores resultados frente al VNS, ya que los FGA Best son mejores un 6,256% en promedio para el total de las instancias, y la mayor diferencia esta contra el CLM, el cual es mejor tan solo en 4,078% en promedio.

Otros resultados que se compararon son las soluciones iniciales (los mejores resultados) expuestas utilizadas por Schuster & Framiñan (2003) con los cuales estos comparan los procedimientos VNS y GASSA para las instancias de tamaño grande, este trabajo comparo las soluciones iniciales expuestas por estos autores contra las desarrolladas para el **JSSP No – WaiT** con lo cual se concluye que usando SPT, esta supera en promedio a las soluciones con la mejor heurística de Schuster Framiñan (2003) para las instancias grandes en 6,32% y usando la LPT, esta la supera en un 14,05%. Además la mejor solución de inicio es usando la regla LPT, y de las 40 instancias grandes solamente una de las propuestas de Schuster y Framiñan 2003 es mejor, es decir el 97,5% de las soluciones con LPT son mejores que las de artículo, mientras que con la regla SPT estas superan en un 77,5% a la de estos autores.

En relación a los tiempos computacionales, comparando con los de Schuster & Framiñan 2003 y 2006, los expuestos en este trabajo son ampliamente superados por estos autores, dado la estructura de programación y el funcionamiento del **JSSP No – WaiT**, sin embargo vale la pena decir que el procedimiento propuesto ordena de la mejor manera posible una secuencia dada (Mejor CMax para esa secuencia), mientras que las propuestas de los autores esta solución puede variar.

6.2 Recomendaciones

Dado que el tema de investigación tratado en este trabajo de investigación ha sido poco estudiado en la literatura, aun son muchos los aportes que se pueden hacer al tema, tales como desarrollar nuevos procedimientos que permitan explorar mejores soluciones para este, así como adaptar procedimientos éxitos utilizados en los problemas de job shop clásicos.

En relación directa a este trabajo es necesario desarrollar un procedimiento inicial que permita realizar secuenciaciones más rápidas para mejorar los tiempos de computo a los mostrados anteriormente (apartado 5.2.5), y utilizar el **Procedimiento JSSP No – WaiT** como herramientas para probar si las mejores soluciones obtenidas se pueden mejorar, ya que como se ha mencionado este programa de la mejor manera una secuencia dada.

También se pueden desarrollar trabajos futuros usando el **Procedimiento JSSP No – WaiT** a través de diferentes heurísticas de búsqueda local, como recocido simulado, pues esta podría explorar nuevas soluciones por tener un componente de aleatoriedad, al aceptar o no una solución dado un cambio de temperatura.

7. BIBLIOGRAFÍA

1. Blazewicz, J. Ecker, K. H., Pesch, E., Schmidt, G., Weglarz, J. Scheduling Computer and Manufacturing Processes. Segunda Edición, Springer. Berlín, Alemania. 2001.
2. Carotenuto, P., Giordani, S., Ricciardelli, S., Rismondo, R. A tabu search approach for scheduling hazmat shipments. Computers & Operations Research (Article in Press).
3. Chase, R., Aquilano, N. & Jacobs, F. Administración de Producción y Operaciones, Manufactura y servicios. Octava Edición, Ed. McGraw Hill. Santa Fe de Bogotá, Colombia. 2000.
4. Cheng R., Gen, M., Tsujimura, Y. A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms: I. Representation. Computer and Industrial Engineering, Vol. 30, No. 4, (1996) p. 983-997.
5. Domínguez M., J. A., Garcia G., S., Ruiz J., A., Domínguez M., M. A., Alvarez G. M. J. Dirección de Operaciones: Aspectos Tácticos y Operativos en la Producción y Servicios. Ed. McGraw Hill. Madrid, España. 1995.
6. Framinan, J.M. & Schuster, C. An enhanced timetabling procedure for the no-wait job shop problem: a complete local search approach. Computers & Operations Research, 331 (2006) p. 1200–1213.
7. Gendreau, M. An Introduction to Tabu Search .En: Handbook of metaheuristics. Kluwer academic publisher, 2003.
8. Gröflin, H. & Klinkert, A. Feasible insertions in job shop scheduling, short cycles and stable sets. European Journal of Operational Research (2006, Article in press) p. 1 – 23.
9. Gunadhi, H. Anand, V. J. & Wee Yong, Y. Automated Timetabling Using an Object-oriented Scheduler. Expert Systems With Applications, Vol. 10, No. 2 (1996) p. 243-256.
10. Henderson, D., Jacobson, S. Johnson, A. The theory and practice of Simulated Annealing. En: Handbook of metaheuristics. Kluwer academic publisher, 2003.
11. Jain, A.S. Meeran, S. Deterministic job-shop scheduling: Past, present and future. European Journal of Operational Research, 113 (1999) p. 390-434.
12. Kravchenko, S. A polynomial algorithm for a two-machine no-wait job-shop scheduling problem. European Journal of Operational Research, 106 (1998) p. 101-107.
13. Martí, R., Moreno, M. Métodos Multiarranque. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No.19 (2003) p. 49-60.

14. Mascis, A. & Pacciarelli, D. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143 (2002) p. 498–517.
15. Melián, B., Moreno, J., Moreno, M. Metaheurísticas: una visión global. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*. No.19 (2003) p. 7-28.
16. Lee, C.Y., Lei, L., Pinedo, M. Current trends in deterministic scheduling. *Annals of Operations Research*, 70 (1997) p.1 – 41.
17. Lourenço, H., Martin, O., Stützle, T. Iterated local search. En: *Handbook of metaheuristics*. Kluwer academic publisher, 2003.
18. Pezzella, F., Merelli, E. A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, 120 (2000) p. 297-310.
19. Pinedo, M. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, New Jersey, USA. 2002.
20. Schroder, R [1983]: *Administración de Operaciones, Toma de Decisiones en la Función de Operaciones*. Ed. McGraw Hill. México D.F., México. 1983
21. Schuster, C. & Framinan, J.M. Approximative procedures for no-wait job shop scheduling. *Operations Research Letters*. 31, (2003) p. 308 – 318.
22. Raaymakers & Hoogeveen. Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. *European Journal of Operational Research*, 126 (2000) p. 131-151.
23. Valdés, R., Fuertes, A., Tamarit, J.M., Giménez, G. & Ramos, R. A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, 165 (2005) p. 525 – 534.
24. Woeginger, G. Inapproximability results for no-wait job shop scheduling. *Operations Research Letters*, 32 (2004) p. 320 – 325.
25. Zhou, F., Xu, Y. Chen, G. No-wait scheduling in single-hop multi-channel LANs. *Information Processing Letters*, 93 (2005) p.19–24.
26. Hall G., N. & Sriskandarajah, C. A survey of machine scheduling problems with blocking and no – wait in process. *Operations Research Letters*. 44, (1996) p. 510 – 525.
27. Macchiarolo, R., Mole, S. & Riemma, S. Modeling and optimization of industrial manufacturing processes subject to no – wait constraints. *International Journal of Production Research*, 37 (1999) p. 2585 – 2607.

28. Zhang, C., Li, P., Guan, Z., Rao, Y. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. En: Computers & Operations Research. Disponible en www.sciencedirect.com desde 2006.

Anexo 1: Instancias Usadas para el Procedimiento JSSP No – Wait con SPT Comparadas Contra el Óptimo

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
la01	10	5	971	1037	6,797	1064	9,578	1031	6,179	1031	6,179	1109	14,212
la02	10	5	937	990	5,656	1011	7,898	937	0,000	937	0,000	1000	6,724
la03	10	5	820	932	13,659	973	18,659	832	1,463	832	1,463	897	9,390
la04	10	5	887	889	0,225	1020	14,994	889	0,225	889	0,225	888	0,113
la05	10	5	777	817	5,148	861	10,811	817	5,148	797	2,574	793	2,059
Promedio				6,297		12,388		2,603		2,088		6,500	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
la06	15	5	1248	1339	7,292	1431	14,663	1299	4,0865	1256	0,641	1438	15,224
la07	15	5	1172	1240	5,802	1366	16,553	1282	9,3857	1253	6,9113	1292	10,239
la08	15	5	1244	1296	4,180	1390	11,736	1342	7,8778	1307	5,0643	1454	16,881
la09	15	5	1358	1447	6,554	1586	16,789	1463	7,732	1451	6,8483	1501	10,530
la10	15	5	1287	1338	3,963	1527	18,648	1404	9,0909	1328	3,1857	1541	19,736
Promedio				5,558		15,678		7,635		4,530		14,522	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
la11	20	5	1714	1825	6,476	1915	11,727	1895	10,560	1714	0,000	1955	14,061
la12	20	5	1507	1631	8,228	1694	12,409	1548	2,721	1507	0,000	1706	13,205
la13	20	5	1665	1766	6,066	1907	14,535	1718	3,183	1668	0,180	1776	6,667
la14	20	5	1757	1805	2,732	2313	31,645	1819	3,529	1773	0,911	1956	11,326
la15	20	5	1771	1829	3,275	1898	7,171	1804	1,863	1771	0,000	1914	8,075
Promedio				5,355		15,497		4,371		0,218		10,667	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
la16	10	10	1575	1637	3,937	1673	6,222	1659	5,3333	1637	3,9365	1893	20,190
la17	10	10	1371	1430	4,303	1556	13,494	1389	1,3129	1389	1,3129	1642	19,767
la18	10	10	1417	1555	9,739	1713	20,889	1562	10,233	1555	9,7389	1578	11,362
la19	10	10	1482	1610	8,637	1786	20,513	1572	6,0729	1572	6,0729	1657	11,808
la20	10	10	1526	1693	10,94	1795	17,628	1580	3,5387	1580	3,5387	1668	9,305
Promedio				7,512		15,749		5,298		4,920		14,487	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

<u>Instancia</u>				<u>GASSA*</u>		<u>VNS*</u>		<u>CLM - time*</u>		<u>CLM*</u>		<u>FGA - SPT</u>	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la21	15	10	2092	2182	4,302	2496	19,312	2387	14,101	2149	2,725	2364	13,002
la22	15	10	1928	1965	1,919	2210	14,627	1979	2,645	1979	2,645	2121	10,010
la23	15	10	2038	2193	7,605	2397	17,615	2224	9,127	2038	0,000	2414	18,449
la24	15	10	2061	2150	4,318	2455	19,117	2227	8,054	2133	3,493	2282	10,723
la25	15	10	2034	2034	0,000	2344	15,241	2143	5,359	2050	0,787	2116	4,031
Promedio					3,629		17,182		7,857		1,930		11,243

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

<u>Instancia</u>				<u>GASSA*</u>		<u>VNS*</u>		<u>CLM - time*</u>		<u>CLM*</u>		<u>FGA - SPT</u>	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la26	20	10	2664	2945	10,548	3194	19,895	2912	9,3093	2868	7,658	3178	19,294
la27	20	10	2933	3036	3,512	3286	12,035	3062	4,3982	2933	0,000	3396	15,786
la28	20	10	2789	2902	4,052	3134	12,370	3034	8,7845	2789	0,000	3264	17,031
la29	20	10	2565	2617	2,027	2838	10,643	2739	6,7836	2565	0,000	2921	13,879
la30	20	10	2835	2892	2,011	3217	13,474	2978	5,0441	2835	0,000	3087	8,889
Promedio					4,430		13,684		6,864		1,532		14,976

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

<u>Instancia</u>				<u>GASSA*</u>		<u>VNS*</u>		<u>CLM - time*</u>		<u>CLM*</u>		<u>FGA - SPT</u>	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la31	30	10	3822	4298	12,454	4076	6,646	4226	10,570	3989	4,369	4681	22,475
la32	30	10	4186	4686	11,945	4476	6,928	4512	7,788	4297	2,652	4924	17,630
la33	30	10	3754	4214	12,254	4361	16,169	4180	11,348	3879	3,330	4329	15,317
la34	30	10	3938	4401	11,757	4447	12,925	4144	5,231	3938	0,000	4430	12,494
la35	30	10	3908	4299	10,005	4378	12,027	4163	6,525	4004	2,456	4501	15,174
Promedio					11,683		10,939		8,292		2,561		16,618

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

<u>Instancia</u>				<u>GASSA*</u>		<u>VNS*</u>		<u>CLM - time*</u>		<u>CLM*</u>		<u>FGA - SPT</u>	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la36	15	15	2810	2949	4,947	3199	13,843	3154	12,242	2810	0,000	3085	9,786
la37	15	15	3044	3216	5,650	3665	20,401	3290	8,081	3161	3,844	3495	14,816
la38	15	15	2726	2762	1,321	3002	10,125	2957	8,474	2726	0,000	2907	6,640
la39	15	15	2752	2908	5,669	3347	21,621	2971	7,958	2784	1,163	3129	13,699
la40	15	15	2838	2950	3,946	3365	18,569	2898	2,114	2880	1,480	3046	7,329
Promedio					4,307		16,912		7,774		1,297		10,454

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

<u>Instancia</u>				<u>GASSA*</u>		<u>VNS*</u>		<u>CLM - time*</u>		<u>CLM*</u>		<u>FGA - SPT</u>	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
orb1	10	10	1615	1663	2,972	1927	19,319	1646	1,920	1646	1,920	1754	8,607
orb2	10	10	1485	1555	4,714	1814	22,155	1536	3,434	1518	2,222	1579	6,330
orb3	10	10	1599	1603	0,250	1603	0,250	1603	0,250	1603	0,250	1708	6,817
orb4	10	10	1653	1653	0,000	1898	14,822	1763	6,655	1653	0,000	1738	5,142
orb5	10	10	1365	1415	3,663	1490	9,158	1371	0,440	1371	0,440	1473	7,912
orb6	10	10	1555	1555	0,000	1790	15,113	1555	0,000	1555	0,000	1621	4,244
orb7	10	10	689	706	2,467	777	12,772	722	4,790	706	2,467	787	14,224
orb8	10	10	1319	1319	0,000	1319	0,000	1319	0,000	1319	0,000	1319	0,000
orb9	10	10	1445	1535	6,228	1742	20,554	1515	4,844	1515	4,844	1663	15,087
orb10	10	10	1557	1618	3,918	1843	18,369	1592	2,248	1592	2,248	1719	10,405
Promedio					2,421		13,251		2,458		1,439		7,877

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
swv01	20	10	2396	2510	4,758	2631	9,808	2441	1,878	2441	1,878	2765	15,401
swv02	20	10	2485	2605	4,829	2705	8,853	2569	3,380	2485	0,000	2959	19,074
swv03	20	10	2489	2582	3,736	2703	8,598	2500	0,442	2520	1,245	2648	6,388
swv04	20	10	2506	2609	4,110	2735	9,138	2689	7,302	2506	0,000	3015	20,311
swv05	20	10	2482	2601	4,795	2734	10,153	2627	5,842	2502	0,806	3027	21,958
Promedio				4.446		9.310		3.769		0.786		16.627	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
swv06	20	15	3291	3639	10,574	3757	14,160	3608	9,632	3291	0,000	3946	19,903
swv07	20	15	3331	3569	7,145	3505	5,224	3518	5,614	3398	2,011	3963	18,973
swv08	20	15	3611	3744	3,683	3943	9,194	3884	7,560	3648	1,025	4300	19,081
swv09	20	15	3378	3551	5,121	3705	9,680	3446	2,013	3378	0,000	3877	14,772
swv10	20	15	3552	3773	6,222	3898	9,741	3720	4,730	3661	3,069	4247	19,566
Promedio				6.549		9.600		5.910		1.221		18.459	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - SPT	
Nombre	n	m	Opt.	Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR		Mejor PDR	
ft06	6	6	73	73	0,000	73	0,000	73	0,000	73	0,000	73	0,000
ft10	10	10	1607	1620	0,809	1810	12,632	1619	0,747	1619	0,747	1721	7,094

**Anexo 2: Instancias Usadas para el Procedimiento JSSP No – Wait con SPT
Comparadas frente a Schuster & Framiñan (2003) y (2006)**

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la01	10	5	-0,964	-3,477	-0,388	-0,388
la02	10	5	-2,929	-4,946	2,561	2,561
la03	10	5	-7,511	-11,408	3,606	3,606
la04	10	5	0,000	-12,843	0,000	0,000
la05	10	5	-4,039	-8,943	-4,039	-1,631
Promedio			-3,089	-8,323	0,348	0,830

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la06	15	5	7,394	0,489	10,701	14,490
la07	15	5	4,194	-5,417	0,780	3,113
la08	15	5	12,191	4,604	8,346	11,247
la09	15	5	3,732	-5,359	2,597	3,446
la10	15	5	15,172	0,917	9,758	16,039
Promedio			8,536	-0,953	6,436	9,667

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la11	20	5	7,123	2,089	3,166	14,061
la12	20	5	4,598	0,708	10,207	13,205
la13	20	5	0,566	-6,869	3,376	6,475
la14	20	5	8,366	-15,435	7,532	10,321
la15	20	5	4,647	0,843	6,098	8,075
Promedio			5,060	-3,733	6,076	10,427

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la16	10	10	15,638	13,150	14,105	15,638
la17	10	10	14,825	5,527	18,215	18,215
la18	10	10	1,479	-7,881	1,024	1,479
la19	10	10	2,919	-7,223	5,407	5,407
la20	10	10	-1,477	-7,075	5,570	5,570
Promedio			6,677	-0,700	8,864	9,262

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
la21	15 10	8,341	-5,288	-0,964	10,005
la22	15 10	7,939	-4,027	7,175	7,175
la23	15 10	10,078	0,709	8,543	18,449
la24	15 10	6,140	-7,047	2,470	6,985
la25	15 10	4,031	-9,727	-1,260	3,220
Promedio		7,306	-5,076	3,193	9,167

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
la26	20 10	7,912	-0,501	9,135	10,809
la27	20 10	11,858	3,348	10,908	15,786
la28	20 10	12,474	4,148	7,581	17,031
la29	20 10	11,616	2,925	6,645	13,879
la30	20 10	6,743	-4,041	3,660	8,889
Promedio		10,121	1,176	7,586	13,279

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
la31	30 10	8,911	14,843	10,767	17,348
la32	30 10	5,079	10,009	9,131	14,592
la33	30 10	2,729	-0,734	3,565	11,601
la34	30 10	0,659	-0,382	6,902	12,494
la35	30 10	4,699	2,810	8,119	12,413
Promedio		4,415	5,309	7,697	13,689

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
la36	15 15	4,612	-3,564	-2,188	9,786
la37	15 15	8,675	-4,638	6,231	10,566
la38	15 15	5,250	-3,165	-1,691	6,640
la39	15 15	7,600	-6,513	5,318	12,392
la40	15 15	3,254	-9,480	5,107	5,764
Promedio		5,878	-5,472	2,555	9,030

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
orb1	10 10	5,472	-8,978	6,561	6,561
orb2	10 10	1,543	-12,955	2,799	4,018
orb3	10 10	6,550	6,550	6,550	6,550
orb4	10 10	5,142	-8,430	-1,418	5,142
orb5	10 10	4,099	-1,141	7,440	7,440
orb6	10 10	4,244	-9,441	4,244	4,244
orb7	10 10	11,473	1,287	9,003	11,473
orb8	10 10	0,000	0,000	0,000	0,000
orb9	10 10	8,339	-4,535	9,769	9,769
orb10	10 10	6,242	-6,728	7,977	7,977
Promedio		6,060	-3,884	6,199	6,693

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv01	20 10	10,159	5,093	13,273	13,273
swv02	20 10	13,589	9,390	15,181	19,074
swv03	20 10	2,556	-2,035	5,920	5,079
swv04	20 10	15,562	10,238	12,123	20,311
swv05	20 10	16,378	10,717	15,226	20,983
Promedio		11,649	6,681	12,345	15,744

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv06	20 15	8,436	5,031	9,368	19,903
swv07	20 15	11,040	13,067	12,649	16,627
swv08	20 15	14,850	9,054	10,711	17,873
swv09	20 15	9,181	4,642	12,507	14,772
swv10	20 15	12,563	8,953	14,167	16,007
Promedio		11,214	8,149	11,880	17,036

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv06	20 15	0,000	0,000	0,000	0,000
swv07	20 15	6,235	-4,917	6,300	6,300
Promedio		3,117	-2,459	3,150	3,150

Anexo 3: Instancias Usadas para el Procedimiento JSSP No – Wait con LPT Comparadas Contra el Óptimo

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la01	10	5	971	1037	6,797	1064	9,578	1031	6,179	1031	6,179	1071	10,299
la02	10	5	937	990	5,656	1011	7,898	937	0,000	937	0,000	997	6,403
la03	10	5	820	932	13,659	973	18,659	832	1,463	832	1,463	945	15,244
la04	10	5	887	889	0,225	1020	14,994	889	0,225	889	0,225	974	9,808
la05	10	5	777	817	5,148	861	10,811	817	5,148	797	2,574	850	9,395
Promedio				6.297		12.388		2.603		2.088		10.230	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la06	15	5	1248	1339	7,292	1431	14,663	1299	4,0865	1256	0,641	1449	16,106
la07	15	5	1172	1240	5,802	1366	16,553	1282	9,3857	1253	6,9113	1323	12,884
la08	15	5	1244	1296	4,180	1390	11,736	1342	7,8778	1307	5,0643	1445	16,158
la09	15	5	1358	1447	6,554	1586	16,789	1463	7,732	1451	6,8483	1521	12,003
la10	15	5	1287	1338	3,963	1527	18,648	1404	9,0909	1328	3,1857	1494	16,084
Promedio				5.558		15.678		7.635		4.530		14.647	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la11	20	5	1714	1825	6,476	1915	11,727	1895	10,560	1714	0,000	1932	12,719
la12	20	5	1507	1631	8,228	1694	12,409	1548	2,721	1507	0,000	1724	14,399
la13	20	5	1665	1766	6,066	1907	14,535	1718	3,183	1668	0,180	1889	13,453
la14	20	5	1757	1805	2,732	2313	31,645	1819	3,529	1773	0,911	1954	11,212
la15	20	5	1771	1829	3,275	1898	7,171	1804	1,863	1771	0,000	1998	12,818
Promedio				5.355		15.497		4.371		0.218		12.920	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la16	10	10	1575	1637	3,937	1673	6,222	1659	5,3333	1637	3,9365	1673	6,222
la17	10	10	1371	1430	4,303	1556	13,494	1389	1,3129	1389	1,3129	1481	8,023
la18	10	10	1417	1555	9,739	1713	20,889	1562	10,233	1555	9,7389	1544	8,963
la19	10	10	1482	1610	8,637	1786	20,513	1572	6,0729	1572	6,0729	1776	19,838
la20	10	10	1526	1693	10,94	1795	17,628	1580	3,5387	1580	3,5387	1813	18,807
Promedio				7.512		15.749		5.298		4.920		12.371	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
la21	15	10	2092	2182	4,302	2496	19,312	2387	14,101	2149	2,725	2394	14,436
la22	15	10	1928	1965	1,919	2210	14,627	1979	2,645	1979	2,645	2037	5,654
la23	15	10	2038	2193	7,605	2397	17,615	2224	9,127	2038	0,000	2229	9,372
la24	15	10	2061	2150	4,318	2455	19,117	2227	8,054	2133	3,493	2199	6,696
la25	15	10	2034	2034	0,000	2344	15,241	2143	5,359	2050	0,787	2241	10,177
Promedio					3,629		17,182		7,857		1,930		9,267

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
la26	20	10	2664	2945	10,548	3194	19,895	2912	9,3093	2868	7,658	3125	17,305
la27	20	10	2933	3036	3,512	3286	12,035	3062	4,3982	2933	0,000	3065	4,501
la28	20	10	2789	2902	4,052	3134	12,370	3034	8,7845	2789	0,000	3066	9,932
la29	20	10	2565	2617	2,027	2838	10,643	2739	6,7836	2565	0,000	3015	17,544
la30	20	10	2835	2892	2,011	3217	13,474	2978	5,0441	2835	0,000	3055	7,760
Promedio					4,430		13,684		6,864		1,532		11,408

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
la31	30	10	3822	4298	12,454	4076	6,646	4226	10,570	3989	4,369	4098	7,221
la32	30	10	4186	4686	11,945	4476	6,928	4512	7,788	4297	2,652	4679	11,777
la33	30	10	3754	4214	12,254	4361	16,169	4180	11,348	3879	3,330	4058	8,098
la34	30	10	3938	4401	11,757	4447	12,925	4144	5,231	3938	0,000	4316	9,599
la35	30	10	3908	4299	10,005	4378	12,027	4163	6,525	4004	2,456	4411	12,871
Promedio					11,683		10,939		8,292		2,561		9,913

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
la36	15	15	2810	2949	4,947	3199	13,843	3154	12,242	2810	0,000	3207	14,128
la37	15	15	3044	3216	5,650	3665	20,401	3290	8,081	3161	3,844	3101	1,873
la38	15	15	2726	2762	1,321	3002	10,125	2957	8,474	2726	0,000	3028	11,079
la39	15	15	2752	2908	5,669	3347	21,621	2971	7,958	2784	1,163	3083	12,028
la40	15	15	2838	2950	3,946	3365	18,569	2898	2,114	2880	1,480	2904	2,326
Promedio					4,307		16,912		7,774		1,297		8,286

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
orb1	10	10	1615	1663	2,972	1927	19,319	1646	1,920	1646	1,920	1733	7,307
orb2	10	10	1485	1555	4,714	1814	22,155	1536	3,434	1518	2,222	1518	2,222
orb3	10	10	1599	1603	0,250	1603	0,250	1603	0,250	1603	0,250	1692	5,816
orb4	10	10	1653	1653	0,000	1898	14,822	1763	6,655	1653	0,000	1833	10,889
orb5	10	10	1365	1415	3,663	1490	9,158	1371	0,440	1371	0,440	1497	9,670
orb6	10	10	1555	1555	0,000	1790	15,113	1555	0,000	1555	0,000	1689	8,617
orb7	10	10	689	706	2,467	777	12,772	722	4,790	706	2,467	772	12,046
orb8	10	10	1319	1319	0,000	1319	0,000	1319	0,000	1319	0,000	1444	9,477
orb9	10	10	1445	1535	6,228	1742	20,554	1515	4,844	1515	4,844	1561	8,028
orb10	10	10	1557	1618	3,918	1843	18,369	1592	2,248	1592	2,248	1690	8,542
Promedio					2,421		13,251		2,458		1,439		8,261

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia	GASSA*			VNS*		CLM - time*		CLM*		FGA - LPT			
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	
swv01	20	10	2396	2510	4,758	2631	9,808	2441	1,878	2441	1,878	2763	15,317
swv02	20	10	2485	2605	4,829	2705	8,853	2569	3,380	2485	0,000	2638	6,157
swv03	20	10	2489	2582	3,736	2703	8,598	2500	0,442	2520	1,245	2808	12,816
swv04	20	10	2506	2609	4,110	2735	9,138	2689	7,302	2506	0,000	2695	7,542
swv05	20	10	2482	2601	4,795	2734	10,153	2627	5,842	2502	0,806	2858	15,149
Promedio					4,446		9,310		3,769		0,786		11,396

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
swv06	20	15	3291	3639	10,574	3757	14,160	3608	9,632	3291	0,000	3881	17,928
swv07	20	15	3331	3569	7,145	3505	5,224	3518	5,614	3398	2,011	3733	12,068
swv08	20	15	3611	3744	3,683	3943	9,194	3884	7,560	3648	1,025	3911	8,308
swv09	20	15	3378	3551	5,121	3705	9,680	3446	2,013	3378	0,000	3963	17,318
swv10	20	15	3552	3773	6,222	3898	9,741	3720	4,730	3661	3,069	4161	17,145
Promedio				6,549		9,600		5,910		1,221		14,553	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - LPT		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
ft06	6	6	73	73	0,000	73	0,000	73	0,000	73	0,000	73	0,000
ft10	10	10	1607	1620	0,809	1810	12,632	1619	0,747	1619	0,747	1695	5,476
Promedio				0,404		6,316		0,373		0,373		2,738	

Anexo 4: Instancias Usadas para el Procedimiento JSSP No – Wait con LPT Comparadas frente a Schuster & Framiñan (2003) y (2006)

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la01	10	5	3,279	0,658	3,880	3,880
la02	10	5	0,707	-1,385	6,403	6,403
la03	10	5	1,395	-2,878	13,582	13,582
la04	10	5	9,561	-4,510	9,561	9,561
la05	10	5	4,039	-1,278	4,039	6,650
Promedio			3,796	-1,878	7,493	8,015

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la06	15	5	8,215	1,258	11,547	15,366
la07	15	5	6,694	-3,148	3,198	5,587
la08	15	5	11,497	3,957	7,675	10,559
la09	15	5	5,114	-4,098	3,964	4,824
la10	15	5	11,659	-2,161	6,410	12,500
Promedio			8,636	-0,839	6,559	9,767

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la11	20	5	5,863	0,888	1,953	12,719
la12	20	5	5,702	1,771	11,370	14,399
la13	20	5	6,965	-0,944	9,953	13,249
la14	20	5	8,255	-15,521	7,422	10,209
la15	20	5	9,240	5,269	10,754	12,818
Promedio			7,205	-1,707	8,290	12,679

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la16	10	10	2,199	0,000	0,844	2,199
la17	10	10	3,566	-4,820	6,623	6,623
la18	10	10	-0,707	-9,866	-1,152	-0,707
la19	10	10	10,311	-0,560	12,977	12,977
la20	10	10	7,088	1,003	14,747	14,747
Promedio			4,491	-2,849	6,808	7,168

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la21	15 10	9,716	-4,087	0,293	11,401	
la22	15 10	3,664	-7,828	2,931	2,931	
la23	15 10	1,642	-7,009	0,225	9,372	
la24	15 10	2,279	-10,428	-1,257	3,094	
la25	15 10	10,177	-4,394	4,573	9,317	
<u>Promedio</u>		<u>5,496</u>	<u>-6,749</u>	<u>1,353</u>	<u>7,223</u>	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la26	20 10	6,112	-2,160	7,315	8,961	
la27	20 10	0,955	-6,726	0,098	4,501	
la28	20 10	5,651	-2,170	1,055	9,932	
la29	20 10	15,208	6,237	10,077	17,544	
la30	20 10	5,636	-5,036	2,586	7,760	
<u>Promedio</u>		<u>6,713</u>	<u>-1,971</u>	<u>4,226</u>	<u>9,739</u>	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la31	30 10	-4,653	0,540	-3,029	2,733	
la32	30 10	-0,149	4,535	3,701	8,890	
la33	30 10	-3,702	-6,948	-2,919	4,615	
la34	30 10	-1,931	-2,946	4,151	9,599	
la35	30 10	2,605	0,754	5,957	10,165	
<u>Promedio</u>		<u>-1,566</u>	<u>-0,813</u>	<u>1,572</u>	<u>7,200</u>	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la36	15 15	8,749	0,250	1,680	14,128	
la37	15 15	-3,576	-15,389	-5,745	-1,898	
la38	15 15	9,631	0,866	2,401	11,079	
la39	15 15	6,018	-7,888	3,770	10,740	
la40	15 15	-1,559	-13,700	0,207	0,833	
<u>Promedio</u>		<u>3,852</u>	<u>-7,172</u>	<u>0,463</u>	<u>6,976</u>	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
orb1	10 10	4,209	-10,067	5,286	5,286	
orb2	10 10	-2,379	-16,318	-1,172	0,000	
orb3	10 10	5,552	5,552	5,552	5,552	
orb4	10 10	10,889	-3,425	3,971	10,889	
orb5	10 10	5,795	0,470	9,190	9,190	
orb6	10 10	8,617	-5,642	8,617	8,617	
orb7	10 10	9,348	-0,644	6,925	9,348	
orb8	10 10	9,477	9,477	9,477	9,477	
orb9	10 10	1,694	-10,390	3,036	3,036	
orb10	10 10	4,450	-8,302	6,156	6,156	
<u>Promedio</u>		<u>6,717</u>	<u>-3,100</u>	<u>6,842</u>	<u>7,327</u>	

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv01	20 10	10,080	5,017	13,191	13,191
swv02	20 10	1,267	-2,477	2,686	6,157
swv03	20 10	8,753	3,885	12,320	11,429
swv04	20 10	3,296	-1,463	0,223	7,542
swv05	20 10	9,881	4,535	8,793	14,229
Promedio		6,655	1,900	7,443	10,509

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv06	20 15	6,650	3,301	7,567	17,928
swv07	20 15	4,595	6,505	6,111	9,859
swv08	20 15	4,460	-0,812	0,695	7,209
swv09	20 15	11,602	6,964	15,003	17,318
swv10	20 15	10,284	6,747	11,855	13,657
Promedio		7,518	4,541	8,246	13,194

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
FT06	20 15	0,000	0,000	0,000	0,000
FT10	20 15	4,630	-6,354	4,694	4,694
Promedio		2,315	-3,177	2,347	2,347

Anexo 5: Instancias Usadas para el Procedimiento JSSP No – Wait con Rnd Comparadas Contra el Óptimo

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Óptimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la01	10	5	971	1037	6,797	1064	9,578	1031	6,179	1031	6,179	1027	5,767
la02	10	5	937	990	5,656	1011	7,898	937	0,000	937	0,000	961	2,561
la03	10	5	820	932	13,659	973	18,659	832	1,463	832	1,463	862	5,122
la04	10	5	887	889	0,225	1020	14,994	889	0,225	889	0,225	887	0,000
la05	10	5	777	817	5,148	861	10,811	817	5,148	797	2,574	784	0,901
Promedio				6.297		12.388		2.603		2.088		2.870	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Óptimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la06	15	5	1248	1339	7,292	1431	14,663	1299	4,0865	1256	0,641	1427	14,343
la07	15	5	1172	1240	5,802	1366	16,553	1282	9,3857	1253	6,9113	1339	14,249
la08	15	5	1244	1296	4,180	1390	11,736	1342	7,8778	1307	5,0643	1318	5,949
la09	15	5	1358	1447	6,554	1586	16,789	1463	7,732	1451	6,8483	1415	4,197
la10	15	5	1287	1338	3,963	1527	18,648	1404	9,0909	1328	3,1857	1420	10,334
Promedio				5.558		15.678		7.635		4.530		9.814	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Óptimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la11	20	5	1714	1825	6,476	1915	11,727	1895	10,560	1714	0,000	1891	10,327
la12	20	5	1507	1631	8,228	1694	12,409	1548	2,721	1507	0,000	1549	2,787
la13	20	5	1665	1766	6,066	1907	14,535	1718	3,183	1668	0,180	1810	8,709
la14	20	5	1757	1805	2,732	2313	31,645	1819	3,529	1773	0,911	1855	5,578
la15	20	5	1771	1829	3,275	1898	7,171	1804	1,863	1771	0,000	1865	5,308
Promedio				5.355		15.497		4.371		0.218		6.542	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Óptimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la16	10	10	1575	1637	3,937	1673	6,222	1659	5,3333	1637	3,9365	1688	7,175
la17	10	10	1371	1430	4,303	1556	13,494	1389	1,3129	1389	1,3129	1498	9,263
la18	10	10	1417	1555	9,739	1713	20,889	1562	10,233	1555	9,7389	1507	6,351
la19	10	10	1482	1610	8,637	1786	20,513	1572	6,0729	1572	6,0729	1642	10,796
la20	10	10	1526	1693	10,94	1795	17,628	1580	3,5387	1580	3,5387	1678	9,961
Promedio				7.512		15.749		5.298		4.920		8.709	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la21		15	10	2092	2182	4,302	2496	19,312	2387	14,101	2149	2,725	2246	7,361
la22		15	10	1928	1965	1,919	2210	14,627	1979	2,645	1979	2,645	2059	6,795
la23		15	10	2038	2193	7,605	2397	17,615	2224	9,127	2038	0,000	2283	12,022
la24		15	10	2061	2150	4,318	2455	19,117	2227	8,054	2133	3,493	2171	5,337
la25		15	10	2034	2034	0,000	2344	15,241	2143	5,359	2050	0,787	2090	2,753
Promedio						3,629		17,182		7,857		1,930		6,854

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la26		20	10	2664	2945	10,548	3194	19,895	2912	9,3093	2868	7,658	2888	8,408
la27		20	10	2933	3036	3,512	3286	12,035	3062	4,3982	2933	0,000	3113	6,137
la28		20	10	2789	2902	4,052	3134	12,370	3034	8,7845	2789	0,000	2976	6,705
la29		20	10	2565	2617	2,027	2838	10,643	2739	6,7836	2565	0,000	2775	8,187
la30		20	10	2835	2892	2,011	3217	13,474	2978	5,0441	2835	0,000	2905	2,469
Promedio						4,430		13,684		6,864		1,532		6,381

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la31		30	10	3822	4298	12,454	4076	6,646	4226	10,570	3989	4,369	4128	8,006
la32		30	10	4186	4686	11,945	4476	6,928	4512	7,788	4297	2,652	4350	3,918
la33		30	10	3754	4214	12,254	4361	16,169	4180	11,348	3879	3,330	4279	13,985
la34		30	10	3938	4401	11,757	4447	12,925	4144	5,231	3938	0,000	4179	6,120
la35		30	10	3908	4299	10,005	4378	12,027	4163	6,525	4004	2,456	4389	12,308
Promedio						11,683		10,939		8,292		2,561		8,867

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la36		15	15	2810	2949	4,947	3199	13,843	3154	12,242	2810	0,000	3161	12,491
la37		15	15	3044	3216	5,650	3665	20,401	3290	8,081	3161	3,844	3346	9,921
la38		15	15	2726	2762	1,321	3002	10,125	2957	8,474	2726	0,000	2970	8,951
la39		15	15	2752	2908	5,669	3347	21,621	2971	7,958	2784	1,163	3085	12,100
la40		15	15	2838	2950	3,946	3365	18,569	2898	2,114	2880	1,480	3075	8,351
Promedio						4,307		16,912		7,774		1,297		10,363

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
	Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
orb1		10	10	1615	1663	2,972	1927	19,319	1646	1,920	1646	1,920	1681	4,087
orb2		10	10	1485	1555	4,714	1814	22,155	1536	3,434	1518	2,222	1575	6,061
orb3		10	10	1599	1603	0,250	1603	0,250	1603	0,250	1603	0,250	1599	0,000
orb4		10	10	1653	1653	0,000	1898	14,822	1763	6,655	1653	0,000	1761	6,534
orb5		10	10	1365	1415	3,663	1490	9,158	1371	0,440	1371	0,440	1463	7,179
orb6		10	10	1555	1555	0,000	1790	15,113	1555	0,000	1555	0,000	1555	0,000
orb7		10	10	689	706	2,467	777	12,772	722	4,790	706	2,467	736	6,821
orb8		10	10	1319	1319	0,000	1319	0,000	1319	0,000	1319	0,000	1363	3,336
orb9		10	10	1445	1535	6,228	1742	20,554	1515	4,844	1515	4,844	1472	1,869
orb10		10	10	1557	1618	3,918	1843	18,369	1592	2,248	1592	2,248	1663	6,808
Promedio						2,421		13,251		2,458		1,439		4,269

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
swv01	20	10	2396	2510	4,758	2631	9,808	2441	1,878	2441	1,878	2555	6,636
swv02	20	10	2485	2605	4,829	2705	8,853	2569	3,380	2485	0,000	2713	9,175
swv03	20	10	2489	2582	3,736	2703	8,598	2500	0,442	2520	1,245	2741	10,125
swv04	20	10	2506	2609	4,110	2735	9,138	2689	7,302	2506	0,000	2700	7,741
swv05	20	10	2482	2601	4,795	2734	10,153	2627	5,842	2502	0,806	2615	5,359
Promedio				4.446		9.310		3.769		0.786		7.807	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
swv06	20	15	3291	3639	10,574	3757	14,160	3608	9,632	3291	0,000	3703	12,519
swv07	20	15	3331	3569	7,145	3505	5,224	3518	5,614	3398	2,011	3510	5,374
swv08	20	15	3611	3744	3,683	3943	9,194	3884	7,560	3648	1,025	3984	10,330
swv09	20	15	3378	3551	5,121	3705	9,680	3446	2,013	3378	0,000	3642	7,815
swv10	20	15	3552	3773	6,222	3898	9,741	3720	4,730	3661	3,069	3909	10,051
Promedio				6.549		9.600		5.910		1.221		9.218	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
ft06	6	6	73	73	0,000	73	0,000	73	0,000	73	0,000	73	0,000
ft10	10	10	1607	1620	0,809	1810	12,632	1619	0,747	1619	0,747	1666	3,671
Promedio				0.404		6.316		0.373		0.373		1.836	

Anexo 6: Instancias Usadas para el Procedimiento JSSP No – Wait con Rnd Comparadas frente a Schuster & Framiñan (2003) y (2006)

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la01	10	5	-0,964	-3,477	-0,388	-0,388
la02	10	5	-2,929	-4,946	2,561	2,561
la03	10	5	-7,511	-11,408	3,606	3,606
la04	10	5	-0,225	-12,843	0,000	0,000
la05	10	5	-4,039	-8,943	-4,039	-1,631
Promedio			-3,134	-8,323	0,348	0,830

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la06	15	5	6,572	-0,280	9,854	13,615
la07	15	5	7,984	-1,977	4,446	6,864
la08	15	5	1,698	-5,180	-1,788	0,842
la09	15	5	-2,211	-10,782	-3,281	-2,481
la10	15	5	6,129	-7,007	1,140	6,928
Promedio			4,034	-5,045	2,074	5,153

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la11	20	5	3,616	-1,253	-0,211	10,327
la12	20	5	-5,028	-8,560	0,065	2,787
la13	20	5	2,492	-5,087	5,355	8,513
la14	20	5	2,770	-19,801	1,979	4,625
la15	20	5	1,968	-1,739	3,381	5,308
Promedio			1,164	-7,288	2,114	6,312

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>			<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n	m				
la16	10	10	3,115	0,897	1,748	3,115
la17	10	10	4,755	-3,728	7,847	7,847
la18	10	10	-3,087	-12,026	-3,521	-3,087
la19	10	10	1,988	-8,063	4,453	4,453
la20	10	10	-0,886	-6,518	6,203	6,203
Promedio			1,177	-5,887	3,346	3,706

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la21	15 10	2,933	-10,016	-5,907	4,514	
la22	15 10	4,784	-6,833	4,042	4,042	
la23	15 10	4,104	-4,756	2,653	12,022	
la24	15 10	0,977	-11,568	-2,515	1,782	
la25	15 10	2,753	-10,836	-2,473	1,951	
Promedio		3,110	-8,802	-0,840	4,862	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la26	20 10	-1,935	-9,580	-0,824	0,697	
la27	20 10	2,536	-5,265	1,666	6,137	
la28	20 10	2,550	-5,041	-1,912	6,705	
la29	20 10	6,037	-2,220	1,314	8,187	
la30	20 10	0,450	-9,698	-2,451	2,469	
Promedio		1,928	-6,361	-0,441	4,839	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la31	30 10	-3,955	1,276	-2,319	3,485	
la32	30 10	-7,170	-2,815	-3,590	1,233	
la33	30 10	1,542	-1,880	2,368	10,312	
la34	30 10	-5,044	-6,027	0,845	6,120	
la35	30 10	2,094	0,251	5,429	9,615	
Promedio		-2,507	-1,839	0,546	6,153	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
la36	15 15	7,189	-1,188	0,222	12,491	
la37	15 15	4,042	-8,704	1,702	5,853	
la38	15 15	7,531	-1,066	0,440	8,951	
la39	15 15	6,087	-7,828	3,837	10,812	
la40	15 15	4,237	-8,618	6,108	6,771	
Promedio		5,817	-5,481	2,462	8,975	

<u>Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)</u>						
<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>	
Nombre	n m					
orb1	10 10	1,082	-12,766	2,126	2,126	
orb2	10 10	1,286	-13,175	2,539	3,755	
orb3	10 10	-0,250	-0,250	-0,250	-0,250	
orb4	10 10	6,534	-7,218	-0,113	6,534	
orb5	10 10	3,392	-1,812	6,710	6,710	
orb6	10 10	0,000	-13,128	0,000	0,000	
orb7	10 10	4,249	-5,277	1,939	4,249	
orb8	10 10	3,336	3,336	3,336	3,336	
orb9	10 10	-4,104	-15,499	-2,838	-2,838	
orb10	10 10	2,781	-9,767	4,460	4,460	
Promedio		1,252	-8,067	1,379	1,841	

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
swv01	20 10	1,793	-2,889	4,670	4,670
swv02	20 10	4,146	0,296	5,605	9,175
swv03	20 10	6,158	1,406	9,640	8,770
swv04	20 10	3,488	-1,280	0,409	7,741
swv05	20 10	0,538	-4,353	-0,457	4,516
Promedio		3,225	-1,364	3,974	6,975

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
swv06	20 15	1,759	-1,437	2,633	12,519
swv07	20 15	-1,653	0,143	-0,227	3,296
swv08	20 15	6,410	1,040	2,575	9,211
swv09	20 15	2,563	-1,700	5,688	7,815
swv10	20 15	3,605	0,282	5,081	6,774
Promedio		2,537	-0,335	3,150	7,923

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
swv06	20 15	0,000	0,000	0,000	0,000
swv07	20 15	2,840	-7,956	2,903	2,903
Promedio		1,420	-3,978	1,452	1,452

Anexo 7: Instancias Usadas para el Procedimiento JSSP No – Wait de los Mejores Resultados (FGA Best) Comparadas Contra el Óptimo

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA Best	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la01	10	5	971	1037	6.797	1064	9.578	1031	6.179	1031	6.179	1027	5.767
la02	10	5	937	990	5.656	1011	7.898	937	0.000	937	0.000	961	2.561
la03	10	5	820	932	13.659	973	18.659	832	1.463	832	1.463	862	5.122
la04	10	5	887	889	0.225	1020	14.994	889	0.225	889	0.225	887	0.000
la05	10	5	777	817	5.148	861	10.811	817	5.148	797	2.574	784	0.901
Promedio				6.297		12.388		2.603		2.088		2.870	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA Best	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la06	15	5	1248	1339	7.292	1431	14.663	1299	4.0865	1256	0.641	1427	14.343
la07	15	5	1172	1240	5.802	1366	16.553	1282	9.3857	1253	6.9113	1292	10.239
la08	15	5	1244	1296	4.180	1390	11.736	1342	7.8778	1307	5.0643	1318	5.949
la09	15	5	1358	1447	6.554	1586	16.789	1463	7.732	1451	6.8483	1415	4.197
la10	15	5	1287	1338	3.963	1527	18.648	1404	9.0909	1328	3.1857	1420	10.334
Promedio				5.558		15.678		7.635		4.530		9.012	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA Best	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la11	20	5	1714	1825	6.476	1915	11.727	1895	10.560	1714	0.000	1891	10.327
la12	20	5	1507	1631	8.228	1694	12.409	1548	2.721	1507	0.000	1549	2.787
la13	20	5	1665	1766	6.066	1907	14.535	1718	3.183	1668	0.180	1776	6.667
la14	20	5	1757	1805	2.732	2313	31.645	1819	3.529	1773	0.911	1855	5.578
la15	20	5	1771	1829	3.275	1898	7.171	1804	1.863	1771	0.000	1865	5.308
Promedio				5.355		15.497		4.371		0.218		6.133	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia				GASSA*		VNS*		CLM - time*		CLM*		FGA Best	
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la16	10	10	1575	1637	3.937	1673	6.222	1659	5.3333	1637	3.9365	1673	6.222
la17	10	10	1371	1430	4.303	1556	13.494	1389	1.3129	1389	1.3129	1481	8.023
la18	10	10	1417	1555	9.739	1713	20.889	1562	10.233	1555	9.7389	1507	6.351
la19	10	10	1482	1610	8.637	1786	20.513	1572	6.0729	1572	6.0729	1642	10.796
la20	10	10	1526	1693	10.94	1795	17.628	1580	3.5387	1580	3.5387	1668	9.305
Promedio				7.512		15.749		5.298		4.920		8.140	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la21	15	10	2092	2182	4.302	2496	19.312	2387	14.101	2149	2.725	2246	7.361
la22	15	10	1928	1965	1.919	2210	14.627	1979	2.645	1979	2.645	2037	5.654
la23	15	10	2038	2193	7.605	2397	17.615	2224	9.127	2038	0.000	2229	9.372
la24	15	10	2061	2150	4.318	2455	19.117	2227	8.054	2133	3.493	2171	5.337
la25	15	10	2034	2034	0.000	2344	15.241	2143	5.359	2050	0.787	2090	2.753
Promedio				3.629		17.182		7.857		1.930		6.095	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la26	20	10	2664	2945	10.548	3194	19.895	2912	9.3093	2868	7.658	2888	8.408
la27	20	10	2933	3036	3.512	3286	12.035	3062	4.3982	2933	0.000	3065	4.501
la28	20	10	2789	2902	4.052	3134	12.370	3034	8.7845	2789	0.000	2976	6.705
la29	20	10	2565	2617	2.027	2838	10.643	2739	6.7836	2565	0.000	2775	8.187
la30	20	10	2835	2892	2.011	3217	13.474	2978	5.0441	2835	0.000	2905	2.469
Promedio				4.430		13.684		6.864		1.532		6.054	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la31	30	10	3822	4298	12.454	4076	6.646	4226	10.570	3989	4.369	4098	7.221
la32	30	10	4186	4686	11.945	4476	6.928	4512	7.788	4297	2.652	4350	3.918
la33	30	10	3754	4214	12.254	4361	16.169	4180	11.348	3879	3.330	4058	8.098
la34	30	10	3938	4401	11.757	4447	12.925	4144	5.231	3938	0.000	4179	6.120
la35	30	10	3908	4299	10.005	4378	12.027	4163	6.525	4004	2.456	4389	12.308
Promedio				11.683		10.939		8.292		2.561		7.533	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
la36	15	15	2810	2949	4.947	3199	13.843	3154	12.242	2810	0.000	3085	9.786
la37	15	15	3044	3216	5.650	3665	20.401	3290	8.081	3161	3.844	3101	1.873
la38	15	15	2726	2762	1.321	3002	10.125	2957	8.474	2726	0.000	2907	6.640
la39	15	15	2752	2908	5.669	3347	21.621	2971	7.958	2784	1.163	3083	12.028
la40	15	15	2838	2950	3.946	3365	18.569	2898	2.114	2880	1.480	2904	2.326
Promedio				4.307		16.912		7.774		1.297		6.530	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia		GASSA*			VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
orb1	10	10	1615	1663	2.972	1927	19.319	1646	1.920	1646	1.920	1681	4.087
orb2	10	10	1485	1555	4.714	1814	22.155	1536	3.434	1518	2.222	1518	2.222
orb3	10	10	1599	1603	0.250	1603	0.250	1603	0.250	1603	0.250	1599	0.000
orb4	10	10	1653	1653	0.000	1898	14.822	1763	6.655	1653	0.000	1738	5.142
orb5	10	10	1365	1415	3.663	1490	9.158	1371	0.440	1371	0.440	1463	7.179
orb6	10	10	1555	1555	0.000	1790	15.113	1555	0.000	1555	0.000	1555	0.000
orb7	10	10	689	706	2.467	777	12.772	722	4.790	706	2.467	736	6.821
orb8	10	10	1319	1319	0.000	1319	0.000	1319	0.000	1319	0.000	1319	0.000
orb9	10	10	1445	1535	6.228	1742	20.554	1515	4.844	1515	4.844	1472	1.869
orb10	10	10	1557	1618	3.918	1843	18.369	1592	2.248	1592	2.248	1663	6.808
Promedio				2.421		13.251		2.458		1.439		3.413	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA Best		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
swv01	20	10	2396	2510	4.758	2631	9.808	2441	1.878	2441	1.878	2555	6.636
swv02	20	10	2485	2605	4.829	2705	8.853	2569	3.380	2485	0.000	2638	6.157
swv03	20	10	2489	2582	3.736	2703	8.598	2500	0.442	2520	1.245	2648	6.388
swv04	20	10	2506	2609	4.110	2735	9.138	2689	7.302	2506	0.000	2695	7.542
swv05	20	10	2482	2601	4.795	2734	10.153	2627	5.842	2502	0.806	2615	5.359
Promedio				4.446		9.310		3.769		0.786		6.416	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
swv06	20	15	3291	3639	10.574	3757	14.160	3608	9.632	3291	0.000	3703	12.519
swv07	20	15	3331	3569	7.145	3505	5.224	3518	5.614	3398	2.011	3510	5.374
swv08	20	15	3611	3744	3.683	3943	9.194	3884	7.560	3648	1.025	3911	8.308
swv09	20	15	3378	3551	5.121	3705	9.680	3446	2.013	3378	0.000	3642	7.815
swv10	20	15	3552	3773	6.222	3898	9.741	3720	4.730	3661	3.069	3909	10.051
Promedio				6.549		9.600		5.910		1.221		8.813	

Gonzalez (2006) y Framiñan & Schuster (2003) y (2006) Comparados con el Optimo

Instancia			GASSA*		VNS*		CLM - time*		CLM*		FGA - Rnd		
Nombre	n	m	Opt.	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR	Mejor	PDR
ft06	6	6	73	73	0.000	73	0.000	73	0.000	73	0.000	73	0.000
ft10	10	10	1607	1620	0.809	1810	12.632	1619	0.747	1619	0.747	1666	3.671
Promedio				0.404		6.316		0.373		0.373		1.836	

Anexo 8: Instancias Usadas para el Procedimiento JSSP No – Wait de los Mejores Resultados (FGA Best) Comparadas frente a Schuster & Framiñan (2003) y (2006)

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia			GASSA*	VNS*	CLM - time*	CLM*
Nombre	n	m				
la01	10	5	-0.964	-3.477	-0.388	-0.388
la02	10	5	-2.929	-4.946	2.561	2.561
la03	10	5	-7.511	-11.408	3.606	3.606
la04	10	5	-0.225	-13.039	-0.225	-0.225
la05	10	5	-4.039	-8.943	-4.039	-1.631
Promedio			-3.134	-8.363	0.303	0.785

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia			GASSA*	VNS*	CLM - time*	CLM*
Nombre	n	m				
la06	15	5	6.572	-0.280	9.854	13.615
la07	15	5	4.194	-5.417	0.780	3.113
la08	15	5	1.698	-5.180	-1.788	0.842
la09	15	5	-2.211	-10.782	-3.281	-2.481
la10	15	5	6.129	-7.007	1.140	6.928
Promedio			3.276	-5.733	1.341	4.403

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia			GASSA*	VNS*	CLM - time*	CLM*
Nombre	n	m				
la11	20	5	3.616	-1.253	-0.211	10.327
la12	20	5	-5.028	-8.560	0.065	2.787
la13	20	5	0.566	-6.869	3.376	6.475
la14	20	5	2.770	-19.801	1.979	4.625
la15	20	5	1.968	-1.739	3.381	5.308
Promedio			0.779	-7.644	1.718	5.904

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia			GASSA*	VNS*	CLM - time*	CLM*
Nombre	n	m				
la16	10	10	2.199	0.000	0.844	2.199
la17	10	10	3.566	-4.820	6.623	6.623
la18	10	10	-3.087	-12.026	-3.521	-3.087
la19	10	10	1.988	-8.063	4.453	4.453
la20	10	10	-1.477	-7.075	5.570	5.570
Promedio			0.638	-6.397	2.794	3.152

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
la21	15 10	2.933	-10.016	-5.907	4.514
la22	15 10	3.664	-7.828	2.931	2.931
la23	15 10	1.642	-7.009	0.225	9.372
la24	15 10	0.977	-11.568	-2.515	1.782
la25	15 10	2.753	-10.836	-2.473	1.951
Promedio		2.394	-9.451	-1.548	4.110

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
la26	20 10	-1.935	-9.580	-0.824	0.697
la27	20 10	0.955	-6.726	0.098	4.501
la28	20 10	2.550	-5.041	-1.912	6.705
la29	20 10	6.037	-2.220	1.314	8.187
la30	20 10	0.450	-9.698	-2.451	2.469
Promedio		1.611	-6.653	-0.755	4.512

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
la31	30 10	-4.653	0.540	-3.029	2.733
la32	30 10	-7.170	-2.815	-3.590	1.233
la33	30 10	-3.702	-6.948	-2.919	4.615
la34	30 10	-5.044	-6.027	0.845	6.120
la35	30 10	2.094	0.251	5.429	9.615
Promedio		-3.695	-3.000	-0.653	4.863

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
la36	15 15	4.612	-3.564	-2.188	9.786
la37	15 15	-3.576	-15.389	-5.745	-1.898
la38	15 15	5.250	-3.165	-1.691	6.640
la39	15 15	6.018	-7.888	3.770	10.740
la40	15 15	-1.559	-13.700	0.207	0.833
Promedio		2.149	-8.741	-1.129	5.220

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

Instancia		GASSA*	VNS*	CLM - time*	CLM*
Nombre	n m				
orb1	10 10	1.082	-12.766	2.126	2.126
orb2	10 10	-2.379	-16.318	-1.172	0.000
orb3	10 10	-0.250	-0.250	-0.250	-0.250
orb4	10 10	5.142	-8.430	-1.418	5.142
orb5	10 10	3.392	-1.812	6.710	6.710
orb6	10 10	0.000	-13.128	0.000	0.000
orb7	10 10	4.249	-5.277	1.939	4.249
orb8	10 10	0.000	0.000	0.000	0.000
orb9	10 10	-4.104	-15.499	-2.838	-2.838
orb10	10 10	2.781	-9.767	4.460	4.460
Promedio		0.585	-8.734	0.712	1.174

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv01	20 10	1.793	-2.889	4.670	4.670
swv02	20 10	1.267	-2.477	2.686	6.157
swv03	20 10	2.556	-2.035	5.920	5.079
swv04	20 10	3.296	-1.463	0.223	7.542
swv05	20 10	0.538	-4.353	-0.457	4.516
<u>Promedio</u>		<u>1.890</u>	<u>-2.643</u>	<u>2.608</u>	<u>5.593</u>

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
swv06	20 15	1.759	-1.437	2.633	12.519
swv07	20 15	-1.653	0.143	-0.227	3.296
swv08	20 15	4.460	-0.812	0.695	7.209
swv09	20 15	2.563	-1.700	5.688	7.815
swv10	20 15	3.605	0.282	5.081	6.774
<u>Promedio</u>		<u>2.147</u>	<u>-0.705</u>	<u>2.774</u>	<u>7.523</u>

Gonzalez (2006) Comparado (%) Framiñan & Schuster (2003) y (2006)

<u>Instancia</u>		<u>GASSA*</u>	<u>VNS*</u>	<u>CLM - time*</u>	<u>CLM*</u>
Nombre	n m				
ft06	6 6	0.000	0.000	0.000	0.000
ft10	10 10	2.840	-7.956	2.903	2.903
<u>Promedio</u>		<u>1.420</u>	<u>-3.978</u>	<u>1.452</u>	<u>1.452</u>