

# Programación de Tareas (Scheduling)

García Prado, Sergio

`sergio@garciparedes.me`

25 de mayo de 2017

## Resumen

[TODO]

## 1. INTRODUCCIÓN

El problema de la programación (o planificación de tareas) consiste en la selección del orden de un conjunto de tareas o trabajos entre un número prefijado de máquinas o unidades de trabajo de tal manera que se minimice una determinada característica referida a las mismas. En este documento se trata el problema de la planificación de tareas referida a una única máquina, es decir, no se permite la paralelización de trabajo. La característica que se pretende minimizar es la tardanza total ponderada, la cual se define como la suma total de las unidades de tiempo de retraso de las tareas, cada una de las cuales tienen un peso (ponderación) diferente, por lo tanto, unas son más prioritarias que otras.

Para poder resolver este problema es necesario conocer un conjunto de valores referidos a cada tarea. Primero denotaremos cada tarea con el índice  $j \in [1, n]$  siendo  $n$  el número total de tareas a planificar. Para cada una de estas tareas se conoce  $p_j$ , que determina el tiempo de duración de la tarea  $j$ ,  $d_j$  que indica el momento de finalización máximo de la tarea  $j$  y  $w_j$  que se refiere al coste por unidad de tiempo de retraso de cada tarea.

Una vez conocidas dichas características y la medida que se pretende optimizar se está en condiciones de modelizar dicha expresión de manera matemática. Para ello es necesario declarar la variable  $t_j$  que indica las unidades de tiempo de retraso de la tarea  $j$ . Por tanto, el objetivo del problema se muestra en la ecuación (1).

$$\text{Minimizar } \sum_{i=1}^n t_j w_j \quad (1)$$

A continuación se describen distintas formulaciones para resolver este problema. Las dos primeras (modelo disyuntivo y modelo de índices de tiempo) se corresponden con estrategias exactas mientras que las siguientes (heurística de selección aleatoria y heurística de búsqueda local) se corresponden con soluciones aproximadas.

### 1.1 Modelo Disyuntivo

[TODO]

### 1.2 Modelo de Índices de Tiempo

[TODO]

$$\begin{aligned}
 &\text{Minimizar} && \sum_{j=1}^n w_j r_j \\
 &\text{sujeto a} && x_j + p_j + s_j - r_j = d_j, && \forall j \in \{1, \dots, n\} \\
 &&& x_i + p_i - x_j \leq M(1 - y_{ij}), && \forall i, j \in \{1, \dots, n\}, i < j \\
 &&& x_j + p_j - x_i \leq M y_{ij}, && \forall i, j \in \{1, \dots, n\}, i < j \\
 &&& x_j, s_j, r_j \geq 0, && \forall j \in \{1, \dots, n\} \\
 &&& y_{ij} \in \{0, 1\}, && \forall i, j \in \{1, \dots, n\}, i < j
 \end{aligned} \tag{2}$$

**Ecuación 2:** *Formulación Disyuntiva para el problema de programación de tareas (Scheduling).*

$$\begin{aligned}
 &\text{Minimizar} && \sum_{j=1}^n \sum_{t=1}^T w_j x_{jt} \max(\{0, t - 1 + p_j - d_j\}) \\
 &\text{sujeto a} && \sum_{t=1}^{T-p_j+1} x_j = 1, && \forall j \in \{1, \dots, n\} \\
 &&& \sum_{j=1}^n \sum_{s=\max(\{0, t-p_j+1\})}^t x_{js} \leq 1, && \forall t \in \{1, \dots, T\} \\
 &&& x_{jt} \in \{0, 1\}, && \forall j \in \{1, \dots, n\}, \forall t \in \{1, \dots, T\}
 \end{aligned} \tag{3}$$

**Ecuación 3:** *Formulación de Índices de tiempo para el problema de programación de tareas (Scheduling).*

### 1.3 Heurística de Selección Aleatoria

La estrategia de selección aleatoria consiste en la generación de  $N$  soluciones de manera aleatoria para seleccionar la mejor de entre todas ellas. Para ello se basa en la generación de números aleatorios de entre el conjunto de las  $j$  tareas ( $\{1, 2, \dots, n\}$ ) sin repetición. En cada iteración se compara la solución obtenida en la iteración anterior con la actual utilizando como medida la función objetivo descrita en la ecuación (1) para después seleccionar la que minimice dicha ecuación. Los resultados de esta estrategia suelen ser mucho peores que las de otras más sofisticadas, sin embargo su reducido coste computacional la convierte en una estrategia a considerar en algunos casos.

### 1.4 Heurística de Búsqueda Local

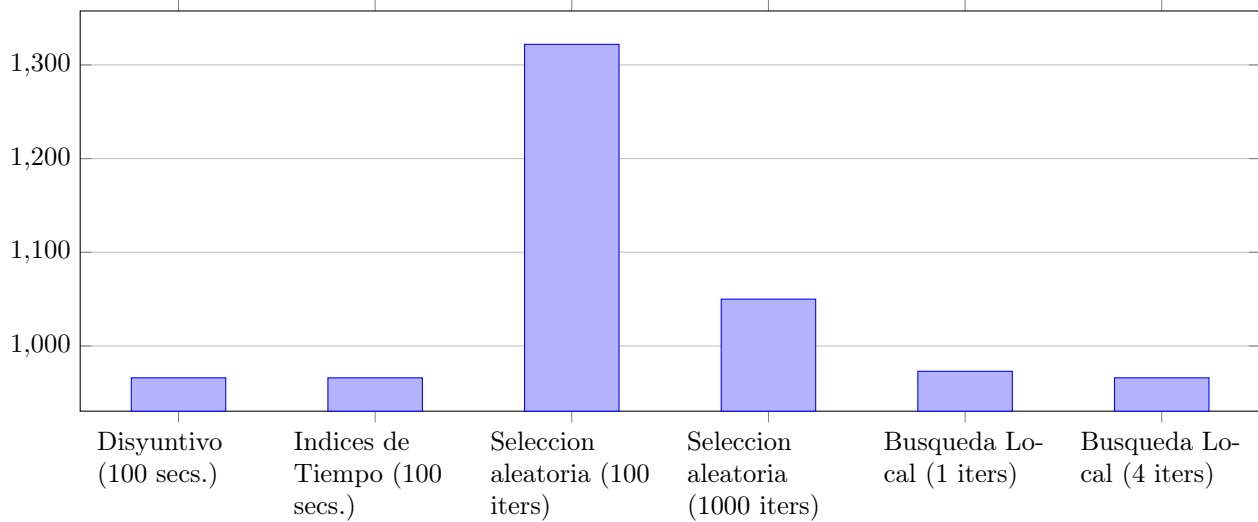
La heurística de búsqueda local se basa en la mejora de una solución inicial calculada previamente a partir de otra estrategia como selección aleatoria u otras heurísticas más sofisticadas para después tratar de mejorarla. La estrategia que utiliza para ello es la búsqueda del intercambio entre el orden de dos tareas que más minimice la ecuación (1) para después realizar fijar dicho intercambio y buscar un nuevo intercambio de índices a partir de dicha solución parcial. El algoritmo termina cuando no se encuentran nuevos intercambios que minimicen la ecuación (1). Nótese que esta estrategia encontrará soluciones óptimas en aquellos casos en que no entre en mínimos locales de los cuales no pueda salir para encontrar el mínimo global.

## 2. RESOLUCIÓN DE PROBLEMAS

En esta sección se presentan los resultados obtenidos tras resolver el *problema de planificación de tareas* (Scheduling) con el objetivo de minimizar la tardanza ponderada sobre distintos conjuntos de datos de entrada. Dichos resultados se han agrupado por problema en lugar de por estrategia de resolución, lo cual permite comparar de manera más simple cada una de ellas. Las implementaciones se han llevado a cabo utilizando el entorno de desarrollo *Xpress-Mosel* [FIC] utilizando el servidor *Neos Server* [NEO] para ejecutar las mismas. Puesto que las estrategias exactas presentan un elevado coste computacional se ha añadido la restricción de tiempo de cómputo de *100 segundos*.

| Modelo                           | Retraso | Programación         |
|----------------------------------|---------|----------------------|
| Disyuntivo (100 secs.)           | 966     | 4 3 10 6 1 9 7 2 5 8 |
| Indices de Tiempo (100 secs.)    | 966     | 4 3 10 6 1 9 7 2 5 8 |
| Selección aleatoria (100 iters)  | 1322    | 4 1 6 7 10 2 9 8 3 5 |
| Selección aleatoria (1000 iters) | 1050    | 4 3 10 6 1 7 2 9 5 8 |
| Busqueda Local (1 iters)         | 973     | 4 10 6 1 3 7 9 2 5 8 |
| Busqueda Local (4 iters)         | 966     | 4 3 6 10 1 9 7 2 5 8 |

**Tabla 1:** Resultados Obtenidos para el problema *sched\_10\_1*



**Figura 1:** Resultados Obtenidos para el problema *sched\_10\_1*

## 2.1 sched\_10\_1

El conjunto de datos está formado por los tiempos de procesamiento  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se deben planificar **10 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )

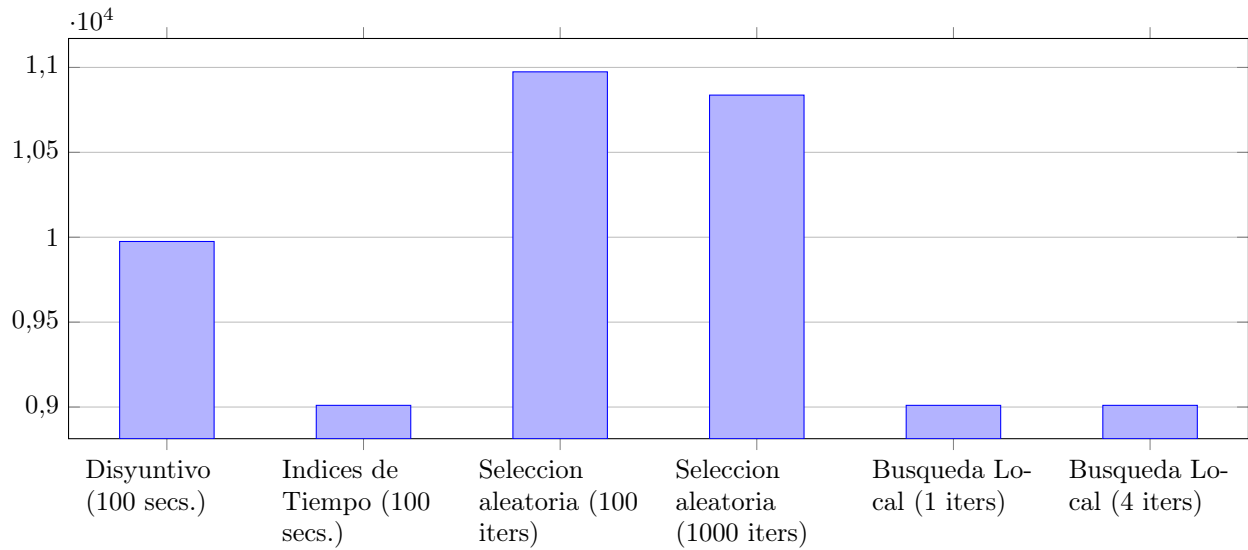
Los resultados se muestran en la tabla 1. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 1.

## 2.2 sched\_20\_1

El conjunto de datos está formado por los tiempos de procesamiento  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se deben planificar **20 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )

| Modelo                           | Retraso | Programación                                       |
|----------------------------------|---------|--|
| Disyuntivo (100 secs.)           | 9975    | 2 6 1 17 11 10 16 3 12 4 20 14 15 13 19 5 18 7 9 8 |
| Indices de Tiempo (100 secs.)    | 9010    | 2 6 10 1 11 3 12 4 13 7 18 20 17 14 19 16 15 9 5 8 |
| Selección aleatoria (100 iters)  | 10974   | 4 11 12 13 2 6 1 5 3 10 17 16 19 20 15 8 9 18 7 14 |
| Selección aleatoria (1000 iters) | 10837   | 2 18 6 12 3 13 7 19 11 1 14 9 15 10 17 4 5 16 20 8 |
| Busqueda Local (1 iters)         | 9010    | 6 2 10 1 11 3 12 4 13 18 7 17 20 14 19 16 15 9 5 8 |
| Busqueda Local (4 iters)         | 9010    | 6 2 10 1 11 12 3 4 13 18 7 20 17 14 19 16 9 15 5 8 |

**Tabla 2:** Resultados Obtenidos para el problema *sched\_20\_1*



**Figura 2:** Resultados Obtenidos para el problema *sched\_20\_1*

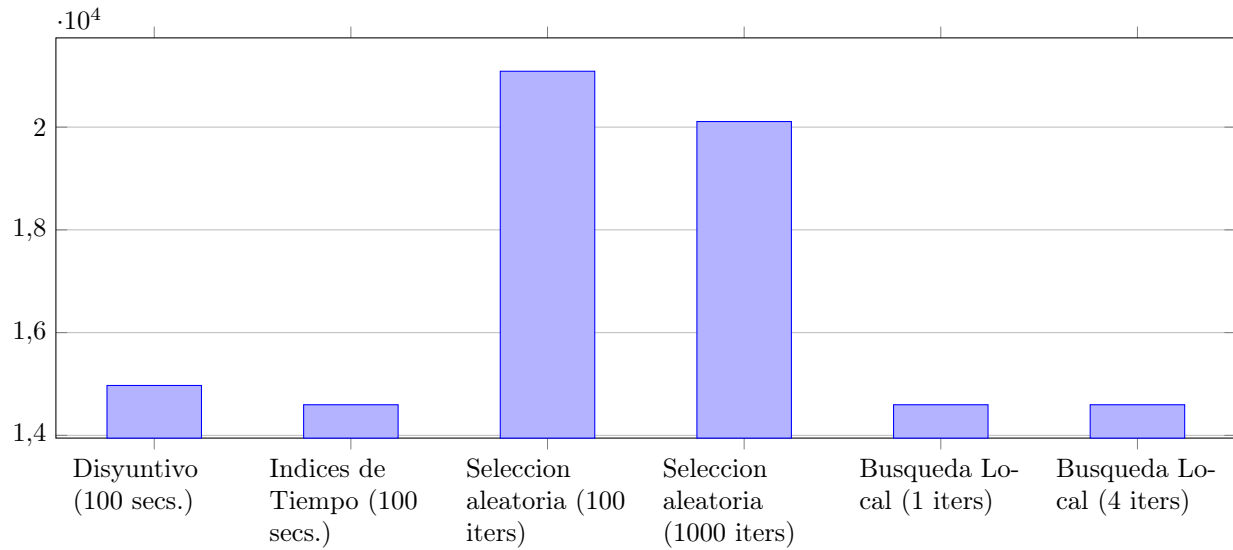
Los resultados se muestran en la tabla 2. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 2.

### 2.3 sched\_30\_1

El conjunto de datos está formado por los tiempos de procesado  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se deben planificar **30 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )

Los resultados se muestran en la tabla 3. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 3.

| Modelo                           | Retraso | Programación  |
|----------------------------------|---------|---|
| Disyuntivo (100 secs.)           | 14971   | 8 2 9 25 15 17 7 12 5 1 24 23 14 21 19 28 27 13 6 16 10 20 11 4 30<br>26 29 18 3 22 |
| Indices de Tiempo (100 secs.)    | 14596   | 25 9 15 2 8 17 7 5 1 24 12 23 14 21 19 27 28 13 6 16 10 29 30 26<br>3 20 11 4 18 22 |
| Seleccion aleatoria (100 iters)  | 21088   | 9 6 18 8 24 12 1 20 7 10 17 15 14 23 19 26 27 2 25 30 11 28 22 5<br>21 16 29 3 13 4 |
| Seleccion aleatoria (1000 iters) | 20109   | 19 21 25 1 15 7 24 2 22 27 8 12 18 17 28 6 9 4 14 23 10 16 30 29<br>3 5 13 11 20 26 |
| Busqueda Local (1 iters)         | 14596   | 15 25 9 2 8 17 7 5 1 24 12 14 23 21 19 27 28 13 6 16 10 29 26 30<br>3 11 20 4 18 22 |
| Busqueda Local (4 iters)         | 14596   | 15 25 9 2 8 17 7 5 1 24 12 23 14 21 19 28 27 13 6 16 10 30 29 26<br>3 20 11 4 22 18 |

**Tabla 3:** Resultados Obtenidos para el problema *sched\_30\_1***Figura 3:** Resultados Obtenidos para el problema *sched\_30\_1*

| Modelo                           | Retraso | Programación  |
|----------------------------------|---------|---|
| Disyuntivo (100 secs.)           | 51095   | 34 16 31 14 40 39 10 24 17 3 23 15 35 7 6 1 2 29 11 28 26 9 27 32<br>38 19 33 8 25 18 22 21 37 20 13 12 5 4 36 30 |
| Indices de Tiempo (100 secs.)    | 40385   | 3 14 23 8 34 21 1 7 10 11 24 35 9 39 28 32 31 25 2 27 29 33 36 22<br>38 6 17 15 5 18 20 37 19 16 12 4 40 13 30 26 |
| Seleccion aleatoria (100 iters)  | 51502   | 8 23 34 35 5 7 14 32 24 11 33 40 12 31 1 13 39 9 3 17 37 18 27 36<br>28 30 2 21 16 22 19 6 4 15 26 29 10 20 38 25 |
| Seleccion aleatoria (1000 iters) | 51502   | 8 23 34 35 5 7 14 32 24 11 33 40 12 31 1 13 39 9 3 17 37 18 27 36<br>28 30 2 21 16 22 19 6 4 15 26 29 10 20 38 25 |
| Busqueda Local (1 iters)         | 40388   | 3 14 8 34 23 21 1 7 10 11 35 24 9 39 32 28 31 25 2 27 29 33 36 22<br>38 6 17 15 18 5 20 37 19 16 12 4 13 30 40 26 |
| Busqueda Local (4 iters)         | 40385   | 3 14 23 8 34 21 1 7 10 11 24 35 9 39 28 32 31 25 2 27 29 36 33 22<br>38 6 17 15 5 18 37 20 19 16 12 4 13 30 40 26 |

**Tabla 4:** Resultados Obtenidos para el problema *sched\_40\_1*

#### 2.4 sched\_40\_1

El conjunto de datos está formado por los tiempos de procesado  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se debn planificar **40 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )

Los resultados se muestran en la tabla 4. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 4.

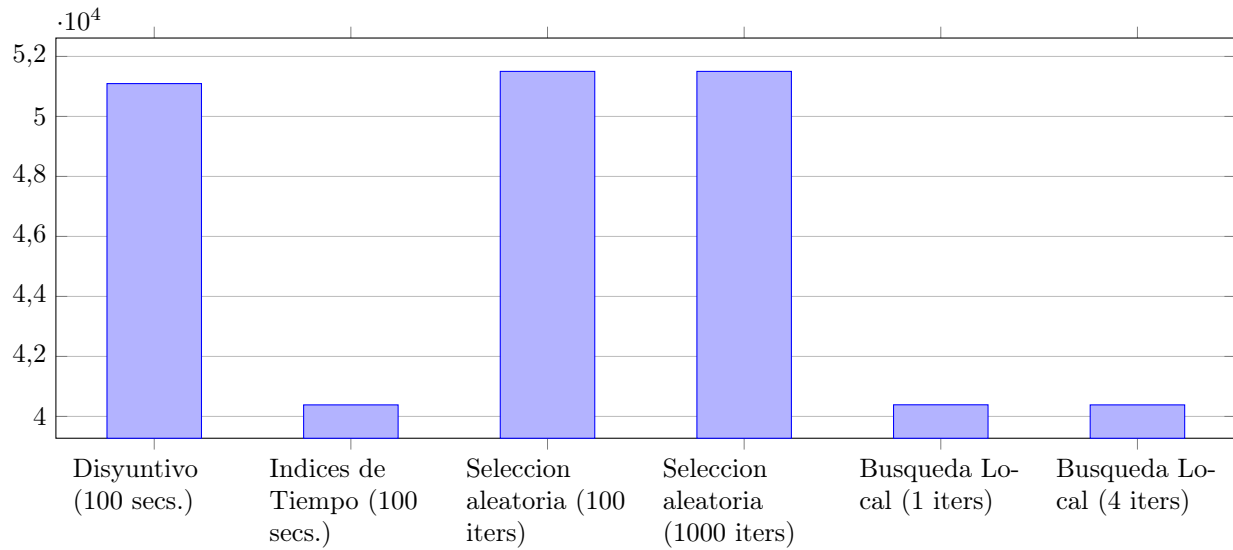
#### 2.5 sched\_100\_1

El conjunto de datos está formado por los tiempos de procesado  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se debn planificar **100 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )

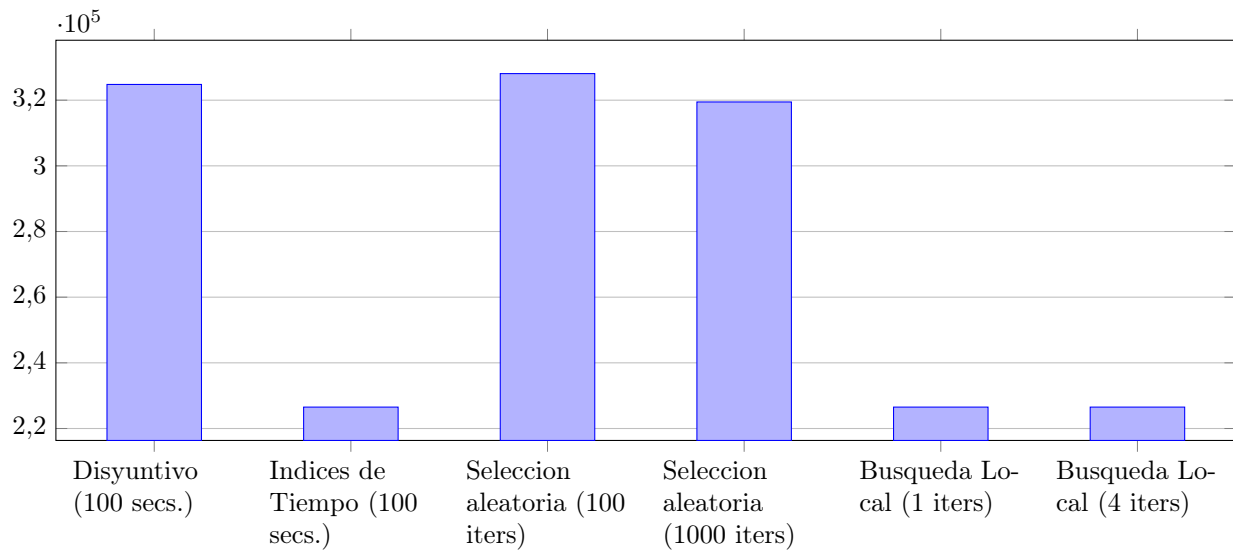
Los resultados se muestran en la tabla 5. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 5.

#### 2.6 sched\_200\_1

El conjunto de datos está formado por los tiempos de procesado  $p_j$ , los costes de retraso  $w_j$  por unidad de tiempo y el momento máximo de terminación  $d_j$  a partir de los cuales se debn planificar **200 tareas** de tal manera que se minimice la tardanza ponderada total, es decir, la suma de todos los costes de retraso. El problema ha sido resuelto a partir de las estrategias descritas anteriormente (*Modelo Disyuntivo* en la sección 1.1, *Modelo de Índices de Tiempo* en la sección 1.2, *Heurística de Selección Aleatoria* con 100 y 1000 iteraciones en la sección 1.3 y *Heurística de Búsqueda Local* con 1 y 4 iteraciones en la sección 1.4 )



**Figura 4:** Resultados Obtenidos para el problema *sched\_40\_1*



**Figura 5:** Resultados Obtenidos para el problema *sched\_100\_1*

| Modelo                           | Retraso | Programación  |
|----------------------------------|---------|---|
| Disyuntivo (100 secs.)           | 324791  | 20 78 21 29 28 60 5 73 43 8 71 96 36 59 85 69 7 45 6 80 97 98 27<br>47 99 62 10 55 54 66 25 87 41 53 93 35 14 89 65 51 9 91 74 24 64<br>18 19 77 57 82 95 58 13 72 1 52 86 40 3 23 48 50 70 33 83 63 92<br>2 37 16 68 22 61 31 39 67 49 75 11 44 42 17 56 12 94 76 81 38 79<br>34 88 32 4 26 15 90 46 30 100 84 |
| Indices de Tiempo (100 secs.)    | 226520  | 91 92 13 21 89 5 27 23 78 8 60 45 77 73 85 74 80 1 3 62 68 69 35<br>26 34 82 50 66 42 86 54 40 75 30 38 9 96 53 39 19 76 10 14 20 52<br>11 44 99 97 88 67 48 17 98 47 71 32 24 93 59 61 70 84 81 16 46<br>64 28 7 22 33 57 100 36 12 6 56 51 65 49 79 18 29 63 90 4 43 95<br>83 2 31 25 37 55 94 72 87 41 58 15 |
| Seleccion aleatoria (100 iters)  | 328096  | 85 67 75 17 29 13 69 23 80 90 77 84 42 86 65 97 46 44 53 18 34<br>25 73 12 56 14 54 38 63 92 59 33 40 39 21 88 1 66 26 81 60 74 43<br>47 24 3 2 51 100 11 91 57 7 45 50 89 31 93 41 70 30 79 62 55 76<br>96 78 27 32 68 22 98 72 4 6 20 8 61 10 52 58 99 36 49 48 82 64 9<br>5 71 95 94 19 15 35 37 83 87 28 16 |
| Seleccion aleatoria (1000 iters) | 319456  | 4 62 70 92 61 99 98 85 59 13 68 20 23 80 82 42 50 48 69 6 8 27 67<br>96 60 9 100 88 41 78 1 15 51 90 86 21 14 97 72 65 87 22 37 38 45<br>24 5 66 35 81 10 83 30 7 73 49 3 16 40 32 46 39 25 71 53 52 89 36<br>33 77 26 75 34 12 74 17 11 47 63 94 18 57 54 2 19 31 55 43 91 84<br>64 93 95 28 76 79 56 29 58 44 |
| Busqueda Local (1 iters)         | 226520  | 92 91 13 21 89 5 27 23 60 8 78 45 77 73 85 74 80 1 3 62 68 35 69<br>26 50 82 34 42 66 86 54 40 75 30 38 96 9 53 19 39 76 14 10 20 52<br>44 11 88 99 97 67 48 17 98 47 71 32 24 93 61 59 70 84 81 16 46<br>64 28 7 22 33 57 100 36 6 12 56 65 51 49 79 18 29 63 90 4 95 43<br>83 31 25 2 37 94 55 87 72 41 15 58 |
| Busqueda Local (4 iters)         | 226520  | 91 92 13 21 89 5 23 27 78 8 60 45 77 73 80 85 74 1 3 62 68 69 35<br>26 34 50 82 66 42 54 86 40 75 30 38 96 9 19 39 53 76 10 14 20 11<br>52 44 97 99 88 17 48 67 98 47 71 32 24 93 61 59 70 84 81 16 64 7<br>46 28 22 33 57 100 36 6 12 56 65 51 49 79 18 29 63 4 90 43 95 83<br>2 31 25 37 55 94 72 87 41 15 58 |

**Tabla 5:** Resultados Obtenidos para el problema *sched\_100\_1*



| Modelo                           | Retraso  | Programación  |
|----------------------------------|----------|---|
| Disyuntivo (100 secs.)           | 14280000 | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25<br>26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47<br>48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69<br>70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90<br>91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108<br>109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124<br>125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140<br>141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156<br>157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172<br>173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188<br>189 190 191 192 193 194 195 196 197 198 199 200 |
| Indices de Tiempo (100 secs.)    | 916507   | 51 50 24 5 133 170 56 99 128 10 126 35 96 199 16 190 110 37 29 83<br>105 11 122 58 175 159 127 134 158 194 86 120 142 81 44 178 146<br>89 151 48 187 60 78 34 193 183 21 42 107 57 43 106 162 79 26 179<br>197 19 53 32 174 130 36 102 65 168 192 31 135 7 196 121 12 23 73<br>152 52 153 1 184 80 169 59 82 155 46 25 41 18 97 71 141 180 88<br>163 76 139 144 182 55 91 136 62 9 49 172 198 77 156 94 118 116<br>119 115 191 140 3 131 68 54 8 171 69 28 157 2 64 45 113 38 70 72<br>47 154 33 145 147 40 124 109 138 117 150 186 137 85 176 101 93<br>143 66 173 167 125 100 111 98 84 123 90 87 13 161 103 200 195<br>188 92 15 165 30 20 75 61 181 166 149 114 129 22 6 67 74 4 17<br>132 177 164 189 14 95 112 63 160 104 39 148 185 108 27 |
| Seleccion aleatoria (100 iters)  | 13573400 | 178 154 121 21 99 42 87 39 128 19 77 162 134 165 49 100 23 125<br>94 138 199 144 115 89 120 176 181 152 60 135 109 59 8 140 88 111<br>5 29 189 58 86 194 191 48 68 147 168 90 149 50 96 18 4 12 76 51<br>22 91 124 126 133 179 146 104 30 32 10 183 192 103 101 107 151<br>82 79 153 123 159 116 62 136 166 106 84 24 83 85 127 122 71 186<br>141 139 43 1 13 114 167 155 105 112 36 92 80 27 28 113 25 34 196<br>31 102 70 2 177 197 56 172 97 132 131 163 195 184 175 142 110<br>38 198 93 78 160 57 193 156 143 188 67 69 47 61 118 130 37 180<br>55 182 185 75 33 95 16 108 11 65 171 200 119 66 169 20 14 41 81<br>174 52 187 98 161 73 44 15 9 53 117 64 7 6 54 46 45 3 26 72 35<br>148 145 74 173 157 158 137 17 150 129 164 170 63 190 40 |
| Seleccion aleatoria (1000 iters) | 13417900 | 161 72 54 164 77 56 35 141 103 109 105 66 10 61 124 18 7 111 37<br>74 99 51 41 83 16 6 79 102 179 53 27 173 34 133 122 144 9 73 196<br>192 142 140 134 135 68 154 5 178 3 90 91 175 60 52 49 42 182 110<br>139 181 184 19 128 195 89 85 106 190 76 117 24 193 120 130 46<br>149 84 113 48 186 23 75 115 127 188 44 191 29 129 92 80 88 93<br>125 153 170 172 20 87 187 180 40 21 189 81 165 96 57 197 1 12 33<br>30 163 17 156 116 26 119 28 69 158 150 50 108 107 198 138 131<br>169 168 101 45 4 94 78 200 13 43 8 159 100 38 65 95 112 160 32<br>118 176 22 47 147 64 137 151 14 63 58 86 59 39 167 145 171 136<br>157 183 143 194 177 97 104 2 126 36 62 174 98 166 67 71 155 146<br>162 31 82 25 132 199 152 148 123 185 55 70 15 114 121 11 |

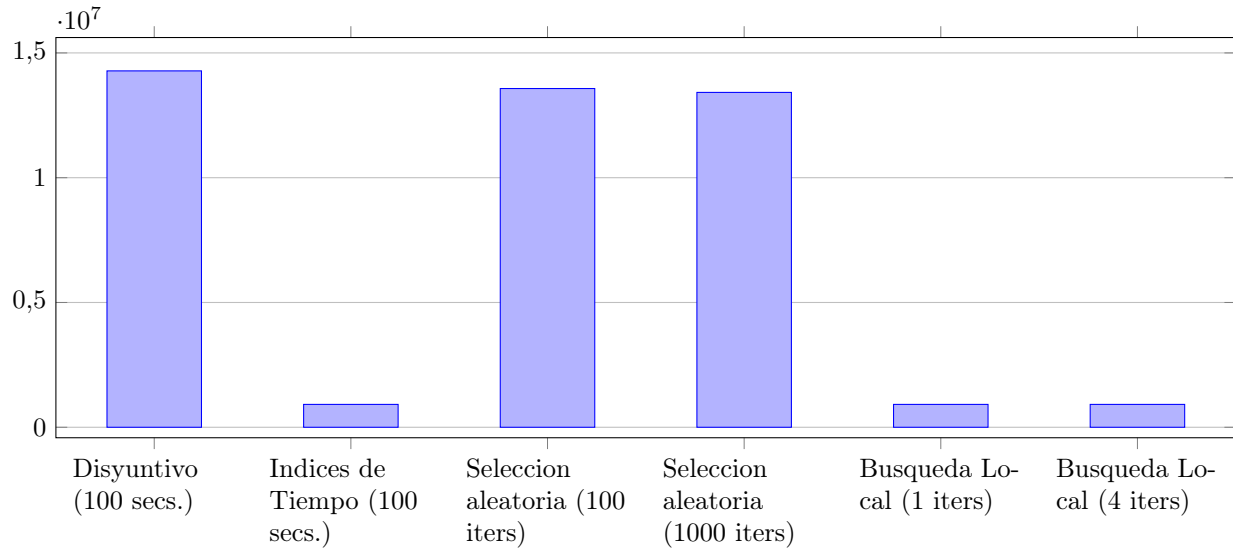


Figura 6: Resultados Obtenidos para el problema sched\_200\_1

| Modelo                   | Retraso | Programación  |
|--------------------------|---------|---|
| Busqueda Local (1 iters) | 916507  | 51 50 24 133 5 99 170 56 10 128 35 96 126 199 110 190 16 37 29 105 83 11 122 159 134 86 175 194 127 120 58 158 81 142 44 178 146 89 151 48 187 60 78 34 21 193 183 43 26 179 57 106 107 79 42 162 197 53 19 32 174 130 36 102 65 192 168 31 135 7 196 23 12 121 73 153 152 52 1 82 169 184 155 59 80 46 41 25 18 97 71 141 180 88 163 76 144 139 182 136 55 91 9 62 49 172 77 198 156 94 118 116 119 115 140 3 191 54 68 131 8 28 171 69 2 45 157 64 113 38 70 72 47 154 33 145 124 147 40 109 117 138 150 186 85 137 93 176 101 143 173 66 167 98 100 125 111 84 123 200 87 90 161 13 103 195 188 15 92 61 165 20 30 75 166 181 22 129 114 149 74 4 67 6 17 132 177 164 14 63 148 189 160 104 95 112 108 39 27 185 |
| Busqueda Local (4 iters) | 916507  | 51 50 24 133 5 99 170 56 128 10 96 35 126 199 190 110 16 37 29 105 83 11 122 120 194 175 159 86 58 158 127 134 44 81 142 178 146 89 151 48 187 60 78 34 183 193 21 26 57 42 79 43 107 179 162 106 53 197 19 32 174 102 130 36 65 192 168 31 135 7 196 12 73 121 23 52 1 152 153 169 80 59 82 155 184 41 18 25 46 97 141 71 163 76 180 88 144 139 182 55 62 136 91 9 49 172 156 77 198 118 94 116 119 115 3 140 191 68 54 131 8 28 171 69 157 45 2 64 72 113 38 70 47 154 33 145 147 40 124 109 117 138 150 186 137 85 93 101 176 143 173 66 167 98 111 100 125 84 123 161 87 200 90 13 103 195 15 188 92 61 20 165 75 30 166 22 181 149 114 129 6 67 74 4 17 132 177 164 148 95 63 112 189 104 39 160 14 27 108 185 |

Tabla 6: Resultados Obtenidos para el problema sched\_200\_1

Los resultados se muestran en la tabla 6. Para facilitar la comprensión a nivel de Tardanza Ponderada se ha incluido un diagrama de barras que muestra la relación entre las distintas estrategias. Dicho diagrama se muestra en la figura 6.

### 3. CONCLUSIONES

Tal y como se puede apreciar en los resultados que se muestran en la sección anterior, las distintas estrategias presentan una clara tendencia. En el caso de las estrategias exactas (y teniendo en cuenta la restricción fijada a priori de tiempo de cómputo de 100 segundos), la formulación de índices de tiempo encuentra la solución en tiempos mucho más reducidos (en problemas grandes la formulación disyuntiva no consigue encontrar soluciones aceptables).

En cuanto a las estrategias basadas en heurísticas, la selección aleatoria presenta malos resultados en todos los casos (a pesar de ello es la más simple a nivel conceptual) aunque se puede apreciar una clara tendencia: cuando se aumenta el número de iteraciones los resultados mejoran, lo cual es un suceso esperable si el generador de números aleatorios genera los valores siguiendo una distribución uniforme. La estrategia de búsqueda local por contra, presenta resultados muy cercanos al óptimo (en un gran número de ocasiones el óptimo) a partir de una única iteración (que cuando se repite 4 iteraciones siempre encuentra el óptimo para los conjuntos de datos utilizados). Por contra, esta estrategia presenta un elevado coste computacional con respecto del tamaño del problema, el cual deja de ser asumible para problemas con un elevado número de tareas.

En cuanto a la comparación entre las estrategias exactas (teniendo en cuenta la restricción de tiempo de cómputo) y las heurísticas, no es posible llevar a cabo una comparación apropiada puesto que los resultados de las heurísticas no han sido calculados teniendo en cuenta restricciones de tiempo. Sin embargo, para problemas de tamaño reducido en los cuales ambos tipos de estrategias terminan antes del tiempo acotado de 100 segundos, la estrategia basada en búsqueda local presenta soluciones muy cercanas al óptimo encontrado por el modelo de índices de tiempo utilizando tiempos de cómputo mucho menores.

### REFERENCIAS

- [Agu17] Jesús Sáez Aguado. Modelos de Investigación Operativa, 2016/17.
- [FIC] FICO Xpress. Xpress-Mosel. [http://www.maths.ed.ac.uk/hall/Xpress/FICO\\_Docs/mosel/mosel\\_lang/dhtml/moselref.html/](http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/mosel/mosel_lang/dhtml/moselref.html/).
- [GP17] Sergio García Prado. Mosel Examples, 2017. <https://github.com/garciparedes/mosel-examples>.
- [NEO] NEOS Server. Solver for Numerical Optimization. <https://neos-server.org/neos/>.