

# HIBRIDIZACIÓN CON BÚSQUEDA LOCAL DE UN ALGORITMO DE ESTIMACIÓN DE DISTRIBUCIÓN PARA LA RESOLUCIÓN DEL PROBLEMA DE SECUENCIAMIENTO DE FLOW SHOP

Daniel Pandolfi<sup>†</sup>, Andrea Villagra<sup>†</sup>, Guillermo Leguizamón<sup>‡</sup>

<sup>†</sup> LabTEm, UACO-UNPA, Caleta Olivia, Argentina.  
{dpandolfi,avillagra}@uaco.unpa.edu.ar  
<sup>‡</sup> LIDIC UNSL, San Luis, Argentina.  
legui@unsl.edu.ar

## Resumen

Los algoritmos de Estimación de Distribuciones, conocidos como EDAs (*Estimation of Distribution Algorithms*) son una clase de algoritmos basados en el paradigma de Computación Evolutiva. Los EDAs, sustituyen los mecanismos de variación (*cruce* y *mutación*) utilizados tradicionalmente por los Algoritmos Evolutivos (AEs) por la generación de individuos obtenidos por simulación de una distribución de probabilidad. La distribución es estimada a partir del proceso iterativo de competencia de los individuos seleccionados en la generación anterior.

Por otra parte, el problema de secuenciamiento de Flow Shop y conocido como FSSP (Flow Shop Sequencing Problem) ha convocado la atención de muchos investigadores en los últimos años. En el caso de *makespan*, se trata de minimizar el tiempo de salida de la última tarea en la última máquina. Para máquinas mayores e iguales a tres el problema se transforma en NP-hard, conforme se incrementa el número de tareas.

Este trabajo, propone aplicar a la resolución del FSSP dos versiones de EDAs, para ser comparados con dos hibridaciones de EDA aplicándole métodos de búsqueda local. Detalles de los algoritmos, experimentos y resultados son presentados en las secciones subsiguientes.

## Palabras Clave

Algoritmo de Estimación de Distribuciones, Computación Evolutiva, Flow Shop Sequencing Problem. Búsqueda Local

## 1. Introducción

Los problemas de planificación abarcan una variedad de problemas de optimización en campos tales como operaciones de producción y despacho en la industria manufacturera, sistemas distribuidos y paralelos, logística y tráfico. Algunos de ellos pueden incluirse dentro de la clase general de problemas de scheduling (Garey, 1979).

En general, un scheduling consiste en la asignación de tareas, a través del tiempo, cuando la disponibilidad de recursos es limitada, donde ciertos objetivos deben ser optimizados y varias restricciones deben ser satisfechas. Los problemas de scheduling son de aplicación en las organizaciones y en la industria y en consecuencia tienen un fuerte impacto económico y social. El estudio de estos problemas data aproximadamente de 1950 donde investigadores de ingeniería industrial, investigación operativa, y administradores desarrollaron nuevos enfoques y algoritmos que tienen como objetivo principal la reducción de los costos de producción en la industria (Leung, 2004).

Muchos algoritmos eficientes han sido desarrollados para encontrar soluciones óptimas, aunque para tamaños pequeños de este tipo de problema. Por ejemplo, se pueden mencionar los trabajos de Jackson (1955), Johnson (1954). Con el advenimiento de la teoría de complejidad (Cook, 1971), muchas investigaciones sobre dicha temática se han desarrollado debido a la inherente dificultad para resolver esta clase de problemas de forma exacta. Muchos de los problemas de scheduling son computacionalmente complejos y el tiempo requerido para calcular una solución óptima se incrementa con el tamaño del problema (Morton y Pentico, 1993; Pinedo, 1995). Además, se ha demostrado, por cierto, que muchos problemas de scheduling pertenecen a la clase de NP-Hard (Brucker, 2004; Lenstra y Kan, 1978).

En el problema de secuenciamiento de Flow Shop existen  $m$  máquinas en serie y  $n$  tareas (jobs) que deben ser procesadas en cada una de las  $m$  máquinas. Todas las tareas deben seguir la misma ruta en cuanto al uso de las máquinas, es decir, primero tienen que procesarse en la máquina 1, luego deben procesarse en la máquina 2 y así sucesivamente. Luego de haberse completado en una máquina, una tarea se pone en cola de la próxima, en general la disciplina de cada cola es FIFO.

La Computación Evolutiva es un campo de investigación emergente que provee nuevas metaheurísticas para la resolución de problemas de optimización donde los enfoques tradicionales hacen al problema computacionalmente intratable. Reflejando la relevancia industrial de estos problemas, se han reportado en la literatura una variedad de métodos basados en AEs para la resolución de problemas de Scheduling (Syswerda, 1991; Branke y Mattfeld, 2000; Burke et al., 2001; Cowling et al., 2002).

Este trabajo está organizado de la siguiente manera. En las próximas dos secciones se presenta respectivamente una breve explicación de los Métodos de Búsqueda Local y los Algoritmos de Estimación de Distribuciones. Seguidamente, se presenta la propuesta general de hibridación de EDAs con Búsqueda Local. En la sección de Experimentos y Resultados se presentan los algoritmos estudiados considerando enfoques con y sin hibridación y se analizan estadísticamente los resultados obtenidos. Finalmente, las conclusiones y trabajos futuros son presentados.

## 2. Métodos de Búsqueda Local

Los Algoritmos de Búsqueda Local (ABL), son métodos muy veloces que parten a menudo de una solución inicial aleatoria a la que iterativamente reemplazan por otra solución mejor que resulta de aplicar algún mecanismo de variación en un espacio de vecindad predeterminado. Una estructura de vecindad junto con una instancia de problema definen el espacio de búsqueda, donde las soluciones pueden ser visualizadas como un grafo donde los nodos son las soluciones y los arcos representan la relación en dos vecinas (Stadler P. F, 1996) .

---

```
ABL  
s ← Generación_de_la_Solución_Inicial ;  
While  $\exists s' \in N(s) / f(s') < f(s)$  do  
    s ← Mejor_de_la_vecindad( N(s) );  
EndWhile  
Output: s
```

---

**Figura 1:** Pseudocódigo de un ABL

El algoritmo de la Figura 1 muestra un esquema general del proceso de búsqueda local para un proceso de minimización., donde  $N(s)$  representa  $h$  soluciones nuevas generadas del vecindario de  $s$  y  $f$  es la función objetivo. Así, el ABL itera mientras mejore la posición de inicio de la vecindad y se detiene cuando alcanza un mínimo local.

Existen al menos dos formas de implementar la función *Mejor\_de\_la\_vecindad(.)*. La primera, consiste en buscar hasta que una solución mejor sea encontrada en el vecindario, mientras que la segunda consiste en retornar el mejor valor de la búsqueda exhaustiva en el vecindario. La performance de ambas funciones depende fuertemente de la definición de la estructura de su vecindad  $N(s)$ .

## 3. Algoritmos basados en Estimación de Distribuciones

Los EDAs fueron introducidos originalmente por Mühlenbein y PaaB (1996) como una extensión de EAs, donde no se requiere de los operadores de variación tradicionales de cruce y mutación para la generación de nuevas soluciones. La nueva población de soluciones se obtiene de la simulación de una distribución de probabilidad, la cual es estimada a partir de la información generada por las soluciones generadas en poblaciones en iteraciones pasadas. La motivación en la utilización de este enfoque está dada por dos aspectos principales. La primera, se refiere a la dificultad de los EAs en trabajar con problemas de decepción y no separabilidad, y segundo, que a la búsqueda implícita llevada a cabo por los operadores de variación de los EAs, se le puede añadir información acerca de la correlación entre las variables del problema (Mühlenbein et al., 1999).

---

```
EDA  
t ← 1;  
Generar  $N \gg 0$  individuos aleatoriamente;  
While (no se cumpla la condición de terminación) do  
    Seleccionar  $M \leq N$  individuos;  
    Estimar distribución  $p^s(x,t)$  a partir de  $M$  individuos seleccionados;  
    Generar  $N$  nuevos individuos de acuerdo a la distribución  $p^s(x,t)$ ;  
    t ← t + 1;  
EndWhile
```

---

**Figura 2:** Pseudocódigo de un EDA

La Figura 2 muestra un pseudo-código de un EDA. Como puede observarse, comienza generando en forma aleatoria la población inicial de soluciones potenciales y luego el algoritmo comienza, iterativamente, a evolucionar la población actual hasta que se cumpla una condición de terminación. Esta condición de terminación podría ser, entre otras: encontrar una solución dada o realizar una cantidad máxima de evaluaciones. La manera en que las soluciones van mejorando a través de las iteraciones (evolución) es la principal diferencia entre los EDAs y los AEs. Las nuevas soluciones son el producto de una memoria evolutiva que permite construir un modelo de distribución de probabilidad  $p^s(x,t)$ . A partir de un conjunto de N soluciones se seleccionan M soluciones para actualizar el modelo probabilístico  $p^s(x,t)$ . Usando este modelo actualizado, se generan nuevamente N soluciones y así siguiendo a través del proceso evolutivo.

Los EDAs pueden ser clasificados de acuerdo al tipo de interacción entre las variables permitidas en el modelo de distribución de probabilidad (Larrañaga y Lozano, 2002). Así, podemos distinguir distintos enfoques en los EDAs, sin dependencias entre variables, con interacciones de a pares, o con múltiples interacciones.

Los EDAs sin dependencias, es la forma más simple de estimar una distribución, donde se asume que las variables son independientes unas de otras. Algunos ejemplos de estos algoritmos son Univariate Marginal Distribution Algorithm -UMDA- (Mühlenbein y Voigt, 1996) y Population Based Incremental Learning algorithm -PBIL- (Bajula, 1994).

Para el caso de interdependencias múltiples, en la literatura se han propuesto varios algoritmos EDAs los cuales requieren estadísticos de orden superior a dos. Baluja y Davies (1997) son los primeros en proponer dichos modelos, aunque no presentan implementación de éstos. La mayoría de estos algoritmos utilizan redes Bayesianas para la implementación de interdependencias múltiples, Castillo et al. (1999) presentan una amplia introducción de las dependencias multivariadas. Algunos de estos algoritmos son: EBNA (Estimation of Bayesian Network Algorithm) propuesto por Larrañaga et al. (2000), BIC (Bayesian Information Criterion) propuesto por Schwarz (1978) y BOA (Bayesian Optimization Algorithm) por Pelikan y Goldberg (2000).

La aplicación los EDAs a problemas de optimización han reportado en muchos de los resultados mejoras a la aplicación de los enfoques evolutivos, tales como algoritmos genéticos (Larrañaga y Lozano 2002). Sin embargo, los EDAs presentan su principal debilidad en el costo computacional que insumen en comparación a los AEs más clásicos.

#### **4. HBL-EDA: Algoritmo Híbrido de Estimación de Distribuciones con Búsqueda Local.**

La aplicación de los EDAs a problemas de optimización han reportado en muchos de los resultados mejoras a la aplicación de los enfoques evolutivos, tales como algoritmos genéticos (Mühlenbein y PaaB, 1996). Sin embargo, como se mencionaba más arriba, los EDAs presentan su principal debilidad en el costo computacional, principalmente cuando se los compara con implementaciones tradicionales de AEs.

En esta sección, presentamos nuestra propuesta general para disminuir el costo computacional de un EDA clásico y al mismo tiempo, mantener y/o mejorar la calidad de las soluciones encontradas.

HBL-EDA, tal cual lo denominamos aquí, es un algoritmo híbrido basado en el algoritmo de Estimación de Distribuciones que incorpora un proceso de búsqueda local con el objetivo de mejorar la explotación de los subespacios (vecindarios) de las soluciones muestreadas por EDA, así como la velocidad de convergencia sin pérdida la calidad de resultados y al mismo tiempo mitigando el costo computacional.

La Figura 3 muestra un pseudocódigo de un HBL-EDA. Como puede observarse, comienza generando en forma aleatoria la población inicial de soluciones potenciales y luego el algoritmo comienza, iterativamente, a evolucionar la población actual hasta que se cumpla una condición de terminación. Esta condición de terminación podría ser, entre otras: encontrar una solución dada o realizar una cantidad máxima de evaluaciones. La manera en que las soluciones van mejorando a través de las iteraciones (evolución) es la principal diferencia entre los EDAs y los AEs. Las nuevas soluciones son el producto de una memoria evolutiva que permite construir un modelo de distribución de probabilidad  $p^s(x,t)$ . A partir de un conjunto de N soluciones se selecciona una solución  $s$  a la que se le es aplicado un proceso de búsqueda local. Finalmente, las mejores soluciones son utilizadas para actualizar el modelo probabilístico  $p^s(x,t)$ . El proceso se itera, se generan nuevamente N soluciones y así siguiendo a través del proceso evolutivo.

---

#### **H-EDA**

$t \leftarrow 1$ ;

Generar  $N \gg 0$  individuos aleatoriamente;

**While** (no se cumpla la condición de terminación) **do**

    Seleccionar  $M \leq N$  individuos;

    Estimar distribución  $p^s(x,t)$  a partir de M individuos seleccionados;

    Generar N nuevos individuos de acuerdo a la distribución  $p^s(x,t)$ ;

    Seleccionar individuo  $s$  del conjunto de los N generados en el paso previo

**While**  $\exists s' \in N(s) / f(s') < f(s)$  **do**

$s \leftarrow \text{Mejor\_de\_la\_vecindad}(N(s))$ ;

**EndWhile**

    Seleccionar N individuos de los N+1 generados

$t \leftarrow t + 1$ ;

**EndWhile**

---

**Figura 3:** Pseudocódigo de un HBL-EDA

## **5. Experimentos y Resultados.**

Taillard (1993) ha construido muchos casos de prueba (benchmarks) para tres clases de problemas de scheduling que han sido ampliamente estudiados en la literatura. El primero de ellos corresponde a Flow Shop Sequencing Problem y tienen varios tamaños que van desde 20 tareas y 5 máquinas hasta 500 tareas y 20 máquinas. El número de tareas determina la dimensionalidad del espacio de búsqueda, mientras que el número de máquinas influye en el comportamiento, caótico, de la función de fitness. En particular, este trabajo estudia las diez instancias para problemas de 100 tareas y 5 máquinas.

Este trabajo, muestra los resultados de dos experimentos que analizan los resultados de dos algoritmos EDA y dos algoritmos EDA hibridizados y un ABL. Ambos enfoques se implementan con dos tipos de representaciones univariable (UV) y multivariable (MV). En la representación univariable, se implementa la preferencia de cada tarea sobre cada posición de la secuencia de ejecución de las mismas. En la representación multivariable

se implementa la preferencia de una tarea sobre todas las demás independientemente de la posición que ocupen en la secuencia.

Inicialmente, se comparan dos algoritmos EDAs ( EDA-UV, EDA-MV) con un algoritmo de búsqueda local (ABL). Finalmente, se compara los dos algoritmos híbridos HBL-EDA-UV y HBL-EDA-MV con EDA-MV (este último por ser el que mostró mejor desempeño en la parte inicial del estudio experimental). La Tabla 1 muestra los valores de los parámetros escogidos para los algoritmos EDA implementados. El término *virtual* en el tamaño de la población identifica a las soluciones que son muestreadas para identificar aquellas que van a actualizar el modelo de aprendizaje.

Para los EDAs, se implementó una versión, basada en una representación univariable (que representa la frecuencia de las tareas sobre una determinada posición del cromosoma) y una versión multivariable donde cada posición acumulan las preferencias un job respecto a los demás. El objetivo de las mejoras propuestas los algoritmos HBL-EDAs es mejorar la calidad de los resultados obtenidos por los EDAs en un tiempo razonable, para ello se implementaron dos versiones con diferentes tipos de representaciones que utilizan mecanismos de BL para acelerar la convergencia. Detalles de los parámetros para de los algoritmos se detallan en la tabla 2.

**Tabla 1:** Parámetros para las distintas versiones de los EDAs estudiados

|                                | EDA-UV       | EDA -MV        | HBL-EDA -UV  | HBL-EDA -MV    |
|--------------------------------|--------------|----------------|--------------|----------------|
| Tamaño población (virtual)     | 100          | 100            | 100          | 100            |
| Criterio parada (iteraciones)  | 4000         | 4000           | 1000         | 1000           |
| Representación                 | Uni-variable | Multi-variable | Uni-variable | Multi-variable |
| Tamaño de la vecindad en la BL | -            |                | 350          | 350            |

Los algoritmos propuestos se ejecutaron en 30 corridas independientes, para las 10 instancias seleccionadas y se analizaron las siguientes variables de performance:

**Ebest:** Error porcentual promedio del mejor individuo sobre upper bound publicado.

**Evals:** Cantidad de evaluaciones (en miles) promedio efectuada por el algoritmo para encontrar el mínimo makespan.

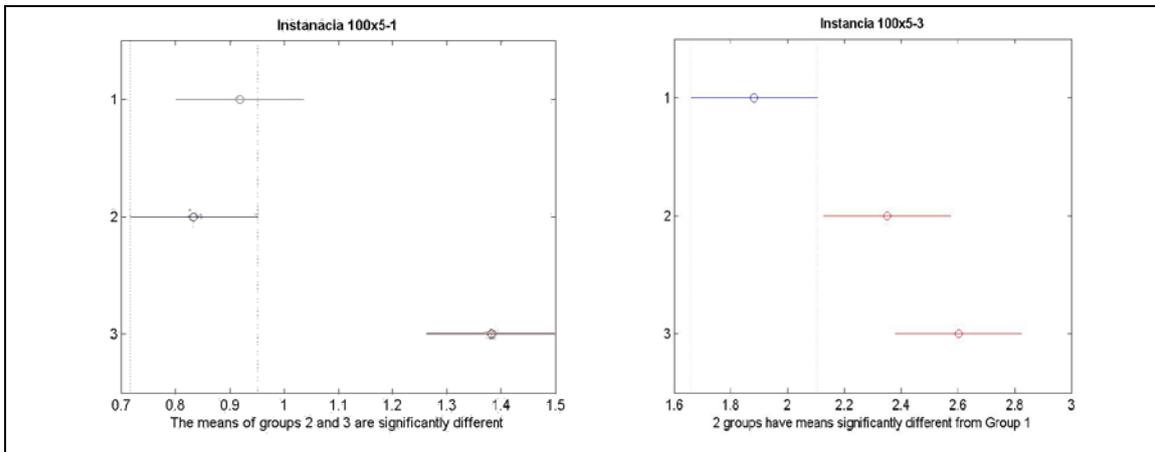
**Esfuerzo Computacional:** Tiempo en segundos que demora un algoritmo para efectuar 1000000 evaluaciones.

|          | Media  |        |       | Mediana |        |       | ANOVA |
|----------|--------|--------|-------|---------|--------|-------|-------|
|          | EDA-UV | EDA-MV | ABL   | EDA-UV  | EDA-MV | ABL   |       |
| 100x5-1  | 0,918  | 0,833  | 1,382 | 0,947   | 0,837  | 1,302 | 0     |
| 100x5-2  | 1,746  | 1,765  | 1,794 | 1,951   | 1,784  | 1,863 | 0,4   |
| 100x5-3  | 1,883  | 2,351  | 2,602 | 1,883   | 2,351  | 2,602 | 0     |
| 100x5-4  | 2,046  | 1,444  | 1,455 | 2,046   | 1,444  | 1,455 | 0     |
| 100x5-5  | 2,094  | 1,526  | 1,356 | 1,687   | 2,048  | 2,235 | 0     |
| 100x5-6  | 1,687  | 2,048  | 2,235 | 2,158   | 1,238  | 2,12  | 0     |
| 100x5-7  | 2,158  | 1,238  | 2,12  | 1,876   | 1,758  | 2,105 | 0     |
| 100x5-8  | 2,457  | 2,088  | 2,029 | 2,457   | 2,088  | 2,029 | 0     |
| 100x5-9  | 2,015  | 1,945  | 2,31  | 2,015   | 1,945  | 2,31  | 0     |
| 100x5-10 | 1,988  | 1,894  | 1,895 | 1,988   | 1,894  | 1,895 | 0,8   |
| Avg      | 1,90   | 1,71   | 1,92  | 1,90    | 1,74   | 1,99  |       |
| Min      | 0,92   | 0,83   | 1,36  | 0,95    | 0,84   | 1,3   |       |
| Max      | 2,46   | 2,35   | 2,60  | 2,46    | 2,35   | 2,60  |       |

**Tabla 2:** Resultados media y mediana para Ebest

Para asegurar la significancia de los resultados estadísticos, es decir si los errores estadísticos medios menores de un algoritmo pueden asegurar la ventaja de este contra otros algoritmos (Hervás C., 2004) se realizó un análisis de ANOVA (Analisis Of Variance).

La Tabla 2, muestra EDA-MV que obtuvo, en general, los mejores valores de media y mediana para las instancias seleccionadas. Del análisis de ANOVA, surge que para dos instancias (2 y 10) no existen diferencias significativas en los tres algoritmos, mientras que en el resto sí, aunque no puede afirmarse estadísticamente una diferencia entre ellos. Así, en las instancias 1 y 3 puede observarse (ver Figura 4) que para EDA-MV observan diferencias estadísticamente significativas con ABL, aunque con EDA-MV sólo puede comprobarse en la instancia 3. Detalles sobre la significancia del menor valor estadístico de un algoritmo sobre otro, son resaltadas en la tabla de medias.



**Figura 4:** Análisis de significancia de Medias (Anova)

La Tabla 3, muestra claramente ventajas claras de los algoritmo EDA hibridizado sobre el EDA-MV tanto en los valores de media y mediana. Del análisis de ANOVA, surge que en sólo una instancia (9) HBL-EDA-MV sobre HBL-EDA-UV, ambos algoritmos tienen diferencias estadísticamente significativas sobre EDA-MV. Detalles sobre la significancia del menor valor estadístico de un algoritmo sobre otro, son resaltadas en la tabla de medias.

|          | Media      |            |        | Mediana    |            |        | Anova |
|----------|------------|------------|--------|------------|------------|--------|-------|
|          | HBL-EDA-UV | HBL-EDA-MV | EDA-MV | HBL-EDA-UV | HBL-EDA-MV | EDA-MV |       |
| 100x5-1  | 0,056      | 0,038      | 0,833  | 0,036      | 0,036      | 0,837  | 0,0   |
| 100x5-2  | 0,382      | 0,361      | 1,765  | 0,418      | 0,389      | 1,784  | 0,0   |
| 100x5-3  | 0,788      | 0,719      | 2,351  | 0,744      | 0,734      | 2,351  | 0,0   |
| 100x5-4  | 0,242      | 0,261      | 1,444  | 0,179      | 0,299      | 2,094  | 0,0   |
| 100x5-5  | 0,273      | 0,220      | 1,526  | 0,286      | 0,286      | 2,048  | 0,0   |
| 100x5-6  | 0,327      | 0,264      | 2,048  | 0,331      | 0,214      | 1,238  | 0,0   |
| 100x5-7  | 0,527      | 0,454      | 1,238  | 0,543      | 0,467      | 1,758  | 0,0   |
| 100x5-8  | 0,546      | 0,513      | 2,088  | 0,607      | 0,529      | 2,088  | 0,0   |
| 100x5-9  | 0,814      | 0,705      | 1,945  | 0,898      | 0,697      | 1,945  | 0,0   |
| 100x5-10 | 0,284      | 0,264      | 1,894  | 0,263      | 0,263      | 1,894  | 0,0   |
| Avg      | 0,42       | 0,38       | 1,71   | 0,43       | 0,39       | 1,80   |       |
| Min      | 0,06       | 0,04       | 0,83   | 0,04       | 0,04       | 0,84   |       |
| Max      | 0,81       | 0,72       | 2,35   | 0,90       | 0,73       | 2,35   |       |

**Tabla 3:** Resultados media y mediana.

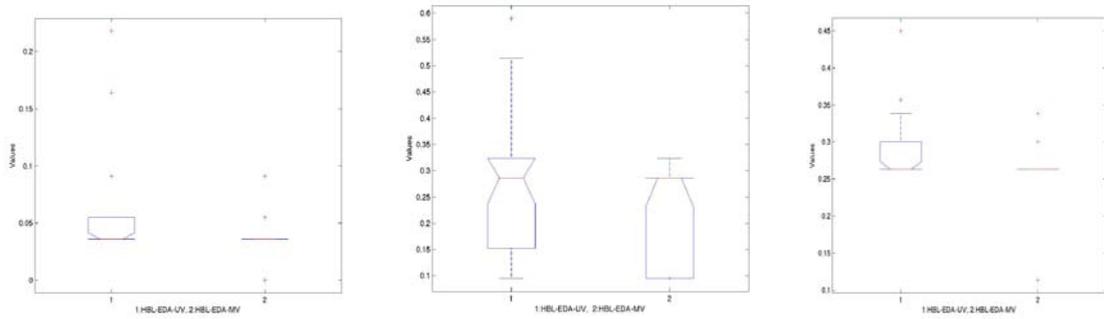


Figura 6: Boxplot Instancias 100x5-1, 100x5-5, 100x5-10

Para un análisis mas detallado de los resultados, se muestran gráficos de boxplot para los algoritmos hibridizados. En la Figura 6, se puede observar en tres instancias que ambos algoritmos poseen medias iguales, aunque HBL-EDA-MV tiene una distribución más achatada sobre los cuartiles. La Figura 7, muestra HBL-EDA-UV muestra una media menor. Finalmente en la Figura 8, HBL-EDA-MV muestra en seis instancias una media menor sobre HBL-EDA-UV.

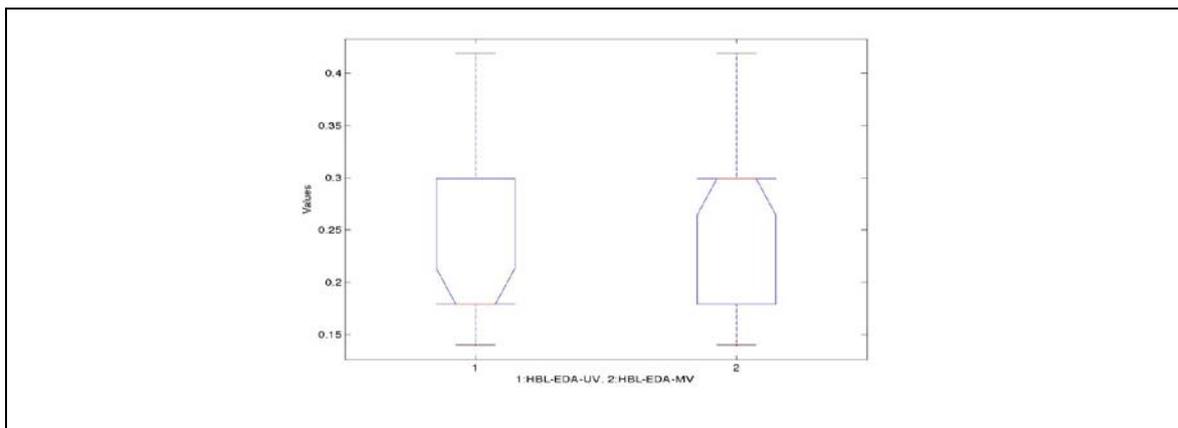
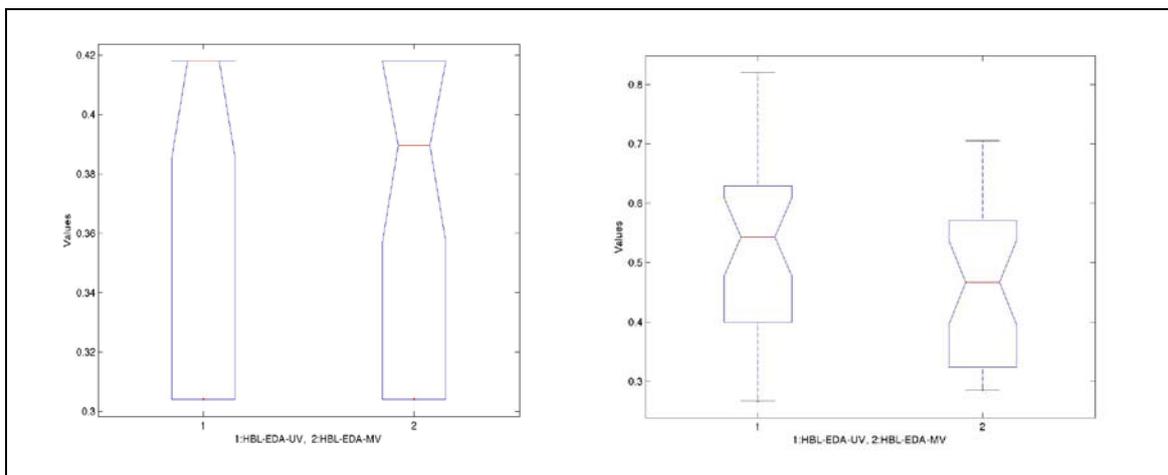
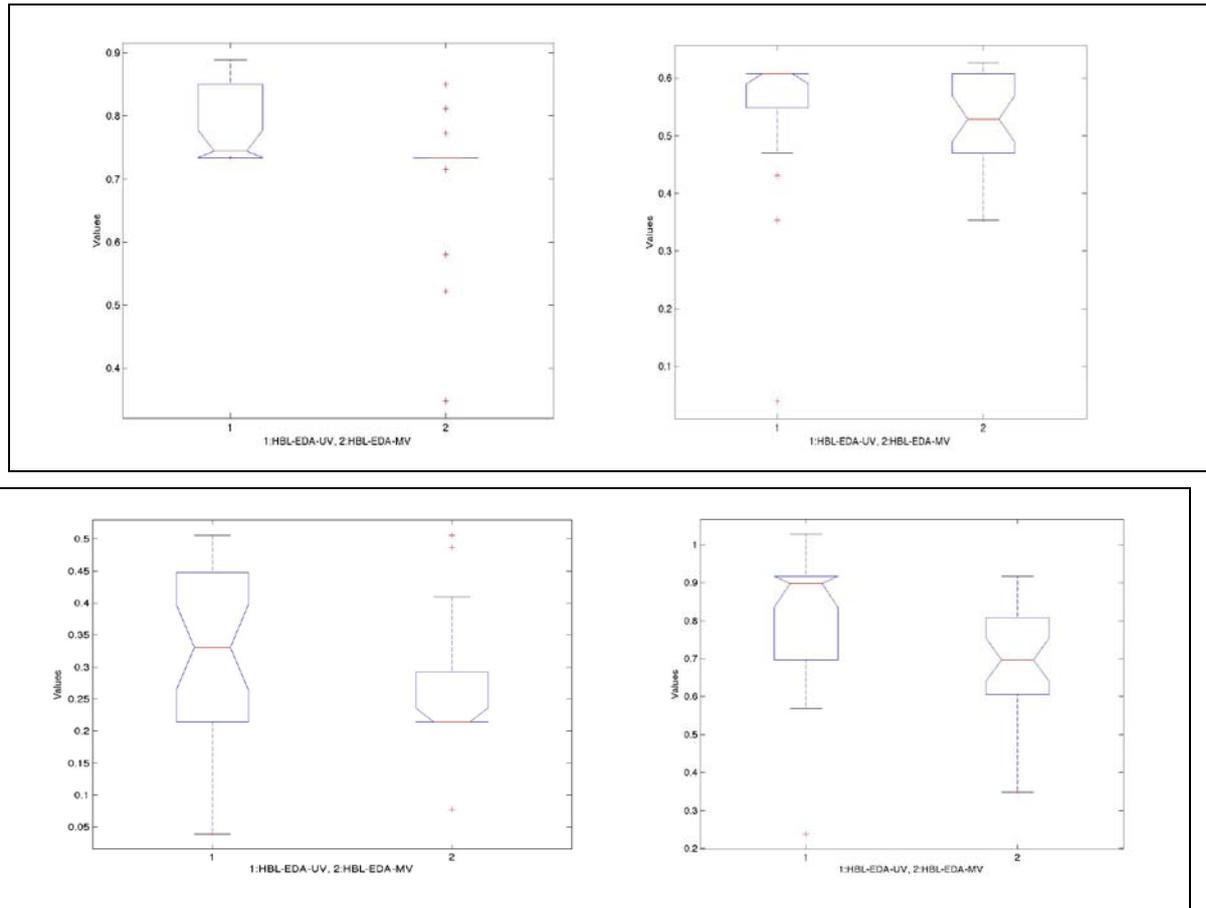


Figura 7: Boxplot Instancia 100x5-4





**Figura 8:** Boxplot Instancias 100x5-8, 100x5-8, 100x5-8, 100x5-8, 100x5-8, 100x5-8

La tabla 4, detalla los resultados de la cantidad de evaluaciones medias que requieren cada algoritmo para la variable de performance Evals. Puede observarse claramente que los algoritmos que usan hibridización con búsqueda local realizan una mayor cantidad de evaluaciones, aunque HBL-EDA-MV converge más rápidamente en todas las instancias que HBL-EDA-UV.

| Instancia  | Evals      |            |            |
|------------|------------|------------|------------|
|            | HBL-EDA UV | HBL-EDA-MV | EDA-MV     |
| 100x5-1    | 683        | 405        | 310        |
| 100x5-2    | 708        | 486        | 251        |
| 100x5-3    | 748        | 511        | 249        |
| 100x5-4    | 813        | 526        | 256        |
| 100x5-5    | 795        | 537        | 232        |
| 100x5-6    | 640        | 392        | 257        |
| 100x5-7    | 658        | 426        | 216        |
| 100x5-8    | 728        | 501        | 249        |
| 100x5-9    | 590        | 399        | 225        |
| 100x5-10   | 826        | 535        | 278        |
| <b>Avg</b> | <b>719</b> | <b>472</b> | <b>252</b> |
| <b>Min</b> | <b>590</b> | <b>392</b> | <b>216</b> |
| <b>Max</b> | <b>826</b> | <b>537</b> | <b>310</b> |

**Tabla 4:** Resultados Evals.

La Tabla 5 muestra el esfuerzo computacional que demora en promedio un algoritmo para efectuar 1000000 evaluaciones (todos ellos ejecutados sobre un mismo procesador). En dicha tabla se puede observar que los algoritmos hibridizados requieren menor tiempo computacional. Este resultado es coherente con la afirmación de que unos de los principales problemas de los EDA es el costo de muestreo de nuevas soluciones, contra la velocidad de un mecanismo de alteración para producir una solución vecina de un ABL.

**Tabla 5:** Esfuerzo Computacional

| Algoritmo  | Segundos |
|------------|----------|
| EDA-MV     | 520      |
| HBL-EDA-UV | 220      |
| HBL-EDA-MV | 270      |

## Conclusiones

Los Algoritmos de Estimación de Distribución son enfoques derivados de Computación Evolutiva, que a diferencia de aquellos simulan una distribución de probabilidad y no requieren de mecanismos de variación.

Estimar una distribución de probabilidad requiere de un proceso de aprendizaje costoso a partir de modelos de datos que se producen por retroalimentación de información. Los enfoques hibridos incorporan procesos de búsqueda local con el objetivo de mejorar la velocidad de convergencia, calidad de resultados y el costo en términos de cómputo con respecto a EDA, principal debilidad de este método

La sinergia producida en los HBL-EDA, entre la exploración del espacio de búsqueda de los algoritmos EDA y alta explotación de los algoritmos de búsqueda local (ABL) permiten afirmar en términos estadísticos que tienen un mejor comportamiento que cualquiera de éstos individualmente.

## Trabajos Futuros

Esta línea de trabajo tiene como principal objetivo el estudio, desarrollo e implementación de Técnicas en Computación Evolutiva básicas y avanzadas (Algoritmos Evolutivos y Algoritmos Basados en Estimación de Distribuciones) y su aplicación a la resolución de problemas de secuenciamiento como el Flow Shop. Ambos enfoques tratados en este trabajo, son lo suficientemente flexibles para incorporar conocimiento desde distintas fuentes de información.

Trabajos futuros, tienen como objetivo explorar y combinar distintos mecanismos propios de los algoritmos AEs y EDAs para lograr una mejora importante respecto de la eficiencia en la exploración del espacio de búsqueda en problemas de secuenciamiento, como así demostrar la calidad de los mismos en la escalabilidad del tamaño del problema.

## Agradecimientos

El primer y el segundo autor agradecen a la Universidad Nacional de la Patagonia Austral, de la cual continuamente reciben financiamiento y apoyo. Además, a la cooperación de los integrantes de su proyecto que continuamente proveen de nuevas ideas y críticas constructivas. El tercer autor agradece el constante apoyo brindado por

la Universidad Nacional de San Luis y la ANPCyT que financian sus actuales investigaciones, así como de los integrantes del LIDIC del cual recibe continuo apoyo.

## REFERENCIAS

- Branke J., Mattfeld D., Anticipation in Dynamic Optimization: The Scheduling Case. *Proceedings of VI PPSN*, pp 253-262, (2000).
- Brucker P., Scheduling Algorithms, 3<sup>rd</sup> ed. *Springer-Verlag New York*, (2004).
- Burke E.K., De Causmaecker P., Petrovic S., Vanden Berghe G., Fitness Evaluation for Nurse Scheduling Problems, *Proc Congress on Evolutionary Computation, CEC2001, Seoul, IEEE Press*, pp 1139-1146, (2001).
- Cook S.A. The complexity of theorem-proving procedures, *Proceedings of 3rd Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York*, pp 151-158 (1971).
- Cowling P. Kendall G. Han L., An Investigation of a Hyperheuristic Genetic Algorithm Applied to a Trainer Scheduling Problem, *Proc Congress on Evolutionary Computation, CEC2002, Hawaii, IEEE Press*, pp 1185-1190, (2002).
- Garey R., Johnson D.; Computers and Intractability: A Guide to the Theory of NP-Completeness. *Freemann & Co., San Francisco, CA*, (1979).
- Jackson J. R.; Scheduling a production line to minimize maximum tardiness, *Research Report 43, Management Science Research Project, University of California; Los Angeles*, (1955).
- Johnson S. M. Optimal two and three stage production; *Naval Research Logistics Quarterly*, **1**, pp 61-67, (1954).
- Larrañaga P. and Lozano J.A.; Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. *Kluwer Academic Publishers*, (2002).
- Lenstra J. K., Rinnooy Kan A. H., Computational complexity of scheduling under precedence constraints, *Operations Research*, **26**, pp 22-35, (1978).
- Leung Joseph. Handbook of Scheduling: Algorithms, Models and Performance Analysis, *Chapman & Hall/CCR Computer and Information Sciences Series* (2004).
- Morton T., Pentico D., "Heuristic scheduling systems", *Wiley series in Engineering and technology management. John Wiley and Sons, INC* (1993).
- Mühlenbein H., Mahnig T., and Ochoa A.. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, **5**, pp 215–247, (1999).
- Mühlenbein H. and Paaß G.; From recombination of genes to the estimation of distributions I. Binary parameters. *In Lecture Notes in Computer Science 1411:Parallel Problem Solving from Nature - PPSN IV*, pp 178–187 (1996).
- Pinedo Michael – "Scheduling- Theory, Algorithms, and Systems. *Prentice Hall International in Industrial and System Engineering* (1995).
- Syswerda G.; Schedule optimization using genetic algorithms, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, **21**, pp 332-349. 1991.
- Taillard, E. Benchmarks for basic scheduling problems, *European Journal of Operational Research*, **64**, pp 278-285 (1993).
- Hervás C. Análisis estadístico de comparación de algoritmo o heurísticas. Comunicación Personal, Universidad de Córdoba, España 2004.
- P. F. Stadler Landscapes and their correlations functions. *Journal of Mathematical Chemistry*. 20:1-45, 1996. Also available as SFI preprint 95-07-067.
- P. Larrañaga, R. Etxebarria, J. A. Lozano, and J. M. Peña. Combinatorial optimization by learning and simulation of Bayesian networks. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence, UAI 2000*, pages 343–352, Stanford, CA, USA, 2000.
- G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 7(2):461–464, 1978.

- S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. Technical Report CMU-CS-97-107, Carnegie Mellon University, 1997.
- E. Castillo, J. M. Gutiérrez, and A. S. Hadi. Expert Systems and Probabilistic Network Models. Springer-Verlag, New York, 1997.
- M. Pelikan and D. E. Goldberg. Hierarchical problem solving and the Bayesian optimization algorithm. In D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, editors, Proceedings of the Genetic and Evolutionary Computation Conference, pages 267–274. Morgan Kaufmann, 2000.