

# Hibridización entre un Algoritmo Evolutivo y un Algoritmo de Estimación de Distribuciones para la solución de FSSP

Daniel Pandolfi, Andrea Villagra\* y Guillermo Leguizamón\*\*

## Abstract

The Evolutionary Algorithms (EAs) are one of the broadly studied metaheuristics. They can be improved in their design in order to carry out a more efficient exploration of the search space. In turn, the Estimation of Distribution Algorithms (EDAs) are a kind of algorithms based on the Evolutionary Computation paradigm that substitute the variation mechanisms used in EAs by the generation of individuals generated through the produced information of the simulation of a probability distribution.

The Flow Shop Scheduling Problem (FSSP) has summoned the attention of many researchers in the last years. For the resolution of the FSSP and with the objective of improving the efficiency of the search as well as the computational effort required, this work proposes a hybrid algorithm among these two approaches. Detail of the implementation as well as the obtained improvements will be discussed.

**Keyword:** Evolutionary Algorithms, Estimation of Distribution Algorithms, Flow Shop Scheduling Problems.

---

• Universidad Nacional de la Patagonia Austral, Unidad Académica Caleta Olivia, Laboratorio de Tecnologías Emergentes (LabTEm) Caleta Olivia, Santa Cruz, Argentina, (9011). {dpandolfi, avillagra}@uaco.unpa.edu.ar

•• Universidad Nacional de San Luis, Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC). Av. Ejército de Los Andes, 950. San Luis, Argentina, 5700. legui@unsl.edu.ar

## Resumen

Los Algoritmos Evolutivos (AEs) son una de las metaheurísticas más ampliamente estudiadas. Éstas, pueden ser mejoradas en su diseño a fin de realizar una exploración más eficiente del espacio de búsqueda. A su vez, los algoritmos de Estimación de Distribuciones (EDAs) son una clase de algoritmos basados en el paradigma de Computación Evolutiva que sustituyen los mecanismos de variación, utilizados en AEs, por la generación de individuos generados a través de la información producida de la simulación de una distribución de probabilidad.

El problema de secuenciamiento de Flow Shop (FSSP) ha convocado la atención de muchos investigadores en los últimos años. Para la resolución del FSSP y con el objetivo de mejorar la eficiencia de la búsqueda como así el esfuerzo computacional requerido, este trabajo propone un algoritmo híbrido entre estos dos enfoques. Detalles de la implementación como así las mejoras obtenidas serán discutidas.

**Palabras Claves:** Algoritmos Evolutivos, Algoritmo de Estimación de Distribuciones, Problema de secuenciamiento de Flow Shop.

## 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 [12]. En general, el 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 tiene 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 [20].

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 [16], Johnson [17]. Con el advenimiento de la teoría de complejidad [8], 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 [22], [29]. Además, se ha demostrado, por cierto, que muchos problemas de scheduling pertenecen a la clase de NP-Hard [5], [19].

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. Todos los jobs tienen el mismo routing, es decir, primero tienen que procesarse en la máquina 1, luego debe procesarse en la siguiente máquina 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 de se han reportado en la literatura una variedad métodos basados en algoritmos evolutivos de resolución de problemas de Scheduling [4], [6], [31], [9].

## 2. Scheduling Evolutivo

### 2.1. Algoritmos Evolutivos

Los Algoritmos Evolutivos (AEs) son una de las metaheurísticas más ampliamente difundidas y estudiadas [1]. Éstas, como muchas otras metaheurísticas poblacionales, pueden ser mejoradas en su diseño a fin de realizar una exploración más eficiente del espacio de búsqueda. En el caso de los AEs, un adecuado desempeño de los mismos, depende en gran medida de los operadores y/o mecanismos de exploración involucrados y que adecuadamente implementados, pueden dar lugar a versiones más eficientes.

El enfoque denominado canónico de un AE, está basado en representación binaria y el proceso de recombinación es operado sobre una par de soluciones que son seleccionadas desde una población. Tal enfoque es conocido como SCPC (Single Crossover per Couple).

Este enfoque, ha sido exitosamente aplicado para resolver problemas de Flow Shop, sustituyendo la representación binaria por una representación basada en permutaciones. Tsujimura et al. [33] muestra la calidad de los AEs en contraste con otras heurísticas convencionales, aplicando operadores de cruce ampliamente conocidos en representaciones basadas en permutaciones, tales como Partially-Mapped Crossover -PMX- [13], Order Crossover -OX- [10] y Cycle Crossover -CX- [26].

Debido a que el problema de Flow Shop es esencialmente un problema de secuenciamiento de permutaciones, una permutación puede ser usada como un esquema de representación de los cromosomas, lo cual representa una forma natural de expresar la solución del problema. La representación, también llamada representación por orden, puede producir un descendiente ilegal si se opera con un cruce tradicional como OPX (One Point Crossover). Consecuentemente, se han propuesto varios operadores de cruce que permiten a partir de dos soluciones codificadas como permutaciones producir otras soluciones válidas, tales como los mencionados anteriormente PMX, OX, y CX entre otros.

Reeves [30] propone un enfoque híbrido, el cual inserta una semilla en la población inicial, generada por la heurística NEH [22]. También ha mostrado en su implementación un operador de cruce llamado one-cut-point crossover (OCPX) que es capaz de operar con soluciones codificadas como permutaciones. Luego de unas pocas experimentaciones Reeves observó que el enfoque evolutivo propuesto producía resultados comparables a los producidos por Simulated Annealing (SA) para FSSP en los benchmarks de Taillard [32]. Los resultados reportados mostraron que el enfoque evolutivo se comportaba sensiblemente mejor para problemas de gran tamaño, produciendo muy buenas soluciones y de forma rápida.

## 2.2. Algoritmos de Estimación de Distribuciones

Los EDAs fueron introducidos originalmente por [24] como una extensión de AEs, 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 iteraciones pasadas (ver Fig 1).

La motivación en la utilización de este enfoque esta dada por dos aspectos principales. El primero, se refiere a la dificultad de los AEs en trabajar con problemas de decepción y no separabilidad, y el segundo se refiere a que a la búsqueda implícita llevada a cabo por los operadores de variación de los AEs, se le puede añadir información acerca de la correlación entre las variables del problema [18], [23].

Esta nueva clase de algoritmos es conocida también en la literatura como *Probabilistic Model Building Genetic Algorithms –PMBGA-* e *Iterated Density Estimators Iterated Density Estimators Evolutionary Algorithms –IDEAS-* [3].

Los EDAs pueden ser clasificados de acuerdo al tipo de interacción entre las variables permitidas en el modelo de distribución de probabilidad [18]. 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-* [25] y *Population Based Incremental Learning algorithm –PBIL-* [2].

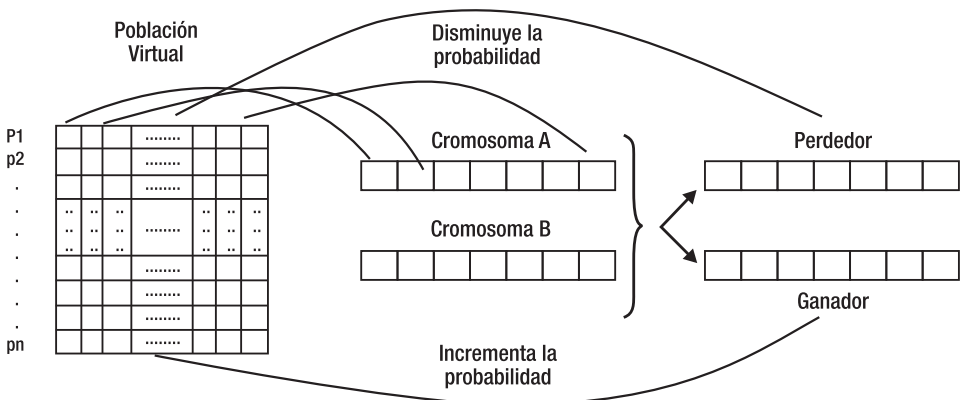


Fig. 1: Esquema global del Algoritmo de Estimación de Distribuciones

Los EDAs con dependencias bivariantes asumen que las variables en un problema no son independientes entre sí. Estos algoritmos simulan la distribución de probabilidad considerando que existe una dependencia entre dos variables. En este tipo de algoritmos podemos mencionar, *Mutual Information Maximization Algorithm for Input Clustering* -MIMIC- [11] y *Bivariate Marginal Distribution Algorithm* -BMDA- [22].

### 2.3. H-EDA: Algoritmo Híbrido de Estimación de Distribuciones.

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 [18]. Sin embargo, los EDAs presentan su principal debilidad en el costo computacional que insumen a comparación de algoritmos genéticos más clásicos.

H-EDA, es un algoritmo híbrido basado en el algoritmo de Estimación de Distribuciones que incorpora el operador de crossover propio de los enfoques recombinativos de los Algoritmos Evolutivos. Asociado al objetivo de mejora de la velocidad de convergencia sin pérdida la calidad de resultados, H-EDA intenta reducir el costo computacional.

La Figura 2 muestra un pseudocódigo de un H-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 a las cuales se les aplica mecanismos de recombinación propio de los AEs. 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)

  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  $p^s(x,t)$  aplicando crossover;   $t \leftarrow t + 1;$ 

EndWhile

Fig. 2: Pseudocódigo de un H-EDA

**3. Experimentos y Resultados**

Taillard [32] 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 influencia en la conducta caótica de la función de *fitness*. En particular, este trabajo estudia las diez instancias para problemas de 100 tareas y 5 máquinas.

Para el Algoritmo Evolutivo, se implementó una versión del enfoque *Single Crossover per Couple*, basada en una representación basada en permutaciones; detalles de los parámetros para cada algoritmo se detallan en la Tabla 1.

**Tabla 1:** Parámetros para AE

	<b>AE</b>
Representación	Permutaciones
Tamaño población	250
Tamaño cromosoma	100
Criterio parada (generación)	6000
Recombinación	Partial Mapped Crossover
Mutación	Swapping Mutation
Probabilidad Recombinación	0,65
Probabilidad Mutación	0,05

Para el EDA, se implementó una versión, basada en una representación univariable (que representa la frecuencia de las tareas sobre una determinada posición del cromosoma); detalles de los parámetros para el algoritmo se detallan en la Tabla 2.

**Tabla 2:** Parámetros para EDA

	<b>EDA Uni -Variable</b>
Tamaño población (virtual)	100
Tamaño cromosoma	100
Criterio parada (iteraciones)	8000
Representación	Uni-variable

Para el H-EDA, se implementó una versión basada en una representación univariable, donde los parámetros son descriptos en la tabla 2 (EDA) y los mecanismos de recombinación son los utilizados por AE son los detallados en la Tabla 1 (AE).

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

**Mínimo Makespan:** representa la mejor solución obtenida por el algoritmo en las 30 corridas efectuadas.

**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.

**Tabla 3:** Mínimo Makespan

<b>Inst.</b>	<b>Upper Bound</b>	<b>AE</b>	<b>EDA-UP</b>	<b>H-EDA</b>
100-1	5493	5540	5517	5508
100-2	5268	5368	5317	5316
100-3	5175	5309	5237	5233
100-4	5014	5114	5059	5040
100-5	5250	5366	5311	5311
100-6	5135	5222	5220	5209
100-7	5246	5336	5305	5322
100-8	5106	5227	5178	5174
100-9	5454	5567	5546	5535
100-10	5328	5452	5407	5384



Para la variable *Mínimo Makespan*, tal como lo observado (ver Tabla 3) los algoritmos **H-EDA** obtienen los mejores mínimos para 9 de las 10 instancias probadas, y solo una corresponde a **EDA** (5).

**Tabla 4:** Resultados Ebest

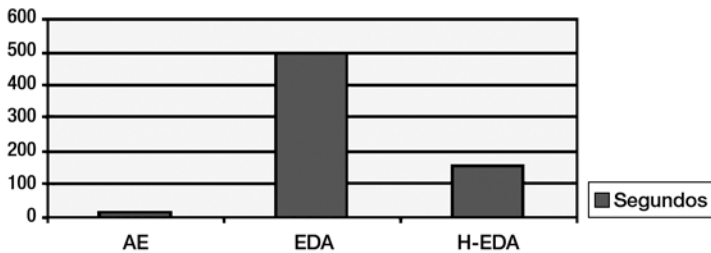
<b>Inst.</b>	<b>Upper Bound</b>	<b>AE</b>	<b>EDA</b>	<b>H-EDA</b>
100-1	5493	1,90	0,92	0,78
100-2	5268	2,65	1,95	1,44
100-3	5175	3,25	1,88	1,76
100-4	5014	2,76	1,77	1,51
100-5	5250	3,22	1,69	1,54
100-6	5135	2,14	2,16	2,05
100-7	5246	2,70	1,88	1,83
100-8	5106	3,26	2,46	2,35
100-9	5454	2,71	2,02	2,07
100-10	5328	3,12	1,99	1,59
	<b>Avg</b>	<b>2,77</b>	<b>1,87</b>	<b>1,69</b>
	<b>Min</b>	<b>1,90</b>	<b>0,92</b>	<b>0,78</b>
	<b>Max</b>	<b>3,26</b>	<b>2,46</b>	<b>2,35</b>

**Tabla 5:** Resultados Evals

<b>Inst.</b>	<b>AE</b>	<b>EDA</b>	<b>H-EDA</b>
100-1	1229	427	1008
100-2	1083	310	849
100-3	971	405	940
100-4	886	419	998
100-5	805	410	992
100-6	1180	217	734
100-7	928	435	1000
100-8	1356	333	731
100-9	971	341	817
100-10	929	420	1245
<b>Avg</b>	1034	372	931
<b>Min</b>	805	217	731
<b>Max</b>	1356	435	1245

En cuanto a la variable de performance *Ebest* (ver Tabla 4) los menores errores promedio corresponden al algoritmo de H-EDA, sobre los enfoques basados en AE y EDA, observándose una mejora entre el 0,2% y 1%.

Respecto al comportamiento del esfuerzo requerido por los algoritmos -en evaluaciones promedio (*Evals*)-, los algoritmos EDA requieren (ver Tabla 5) una cantidad significativamente menor de evaluaciones, y su mejora está entre 60% y 55% aproximadamente.



**Fig. 3:** Esfuerzo Computacional (segundos)

Finalmente y tal como puede verse en el Figura 3 y la Tabla 6 (tiempo de ejecución promedio para 1 millón de evaluaciones); los algoritmos EDA requieren de un mayor costo computacional para producir las mismas cantidad de evaluaciones e independientemente de la calidad de las mismas. El algoritmo híbrido H-EDA propuesto disminuye el costo computacional de EDA, aunque sin lograr los tiempos de AE.

**Tabla 6:** Esfuerzo Computacional

Algoritmo	Segundos
AE	11
EDA	500
H-EDA	152

#### 4. Conclusiones

Muchos de los problemas *scheduling* representan un duro escollo a resolver. Dentro de esta categoría puede considerarse los problemas de secuenciamiento de *Flow Shop*.

Los Algoritmos Evolutivos son metaheurísticas poblacionales, probadamente eficientes en este tipo de problemas, ya que realizan un buen balance entre una exploración del espacio de búsqueda y la explotación de subesquemas que se encuentran codificados dentro las soluciones.

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 a partir de modelos de datos que se producen por retroalimentación de información. H-EDA es un novel enfoque que hibrida un EDA incorporándole los mecanismos de recombinación propios de los AEs. Los resultados mejoran las principales variables analizadas (mínimo Makespan, Ebest) en las instancias analizadas. En cuanto a la velocidad de convergencia, el enfoque híbrido reduce las evaluaciones requeridas respecto AE y reduce el costo en términos de cómputo con respecto a EDA, la que es la principal debilidad de este método.

## **5. Trabajos Futuros**

Esta línea de trabajo tiene como principal objetivo el estudio, desarrollo e implementación de Técnicas en Computación Evolutiva básicas y avanzadas (Algoritmos Genéticos, y Algoritmos Basados en Estimación de Distribuciones) y su aplicación a la resolución de problemas de secuenciamiento 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.

## **Agradecimientos**

Los autores agradecen a la Universidad Nacional de la Patagonia Austral, de la cual recibe financiamiento y apoyo continuo y además, a la cooperación de los integrantes del LabTEm 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

- [1] Bäck T. “Evolutionary Algorithms in theory and practice”. New York:Oxford University Press, (1996).
- [2] Baluja, “Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning”. *Technical Report CMU-CS*, pp. 94-163, Carnegie Mellon University, (1994).
- [3] Bosman A.N. and Thierens. D.; “Linkage information processing in distribution estimation algorithms”. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, 1, pp 60–67. Morgan Kaufmann Publishers, San Francisco, LA. (1999).
- [4] Branke J., Mattfeld D.; “Anticipation in Dynamic Optimization: The Scheduling Case”. *Proceedings of VI PPSN*, pp 253-262, (2000).
- [5] Brucker P., “Scheduling Algorithms”, 3<sup>rd</sup> ed. *Springer-Verlag New York*, (2004).
- [6] 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).
- [7] Campbell H., Dudek R., Smith M., “A heuristic algorithm for the n job m machine sequencing problem”, *Management Science* 16, pp. 630-637 (1970).
- [8] 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).
- [9] 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).
- [10] Davis L., “Handbook of Genetic Algorithms”, New York: Van Nostrand Reinhold Computer Library, (1991).
- [11] De Bonet J. S., Isbell C. L., and Viola P.; “MIMIC: Finding optima by estimating probability densities”. *Advances in Neural Information Processing Systems*, 9. (1997).
- [12] Garey R., Johnson D.; “Computers and Intractability: A Guide to the Theory of NP-Completeness”. *Freemann & Co., San Francisco, CA*, (1979).

- [13] Goldberg, D. and Lingle R.; “Alleles, loci and the traveling salesman problem”, in *Proceeding of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, pp. 154-159, Hillsdale, NJ, (1987).
- [14] Grefenstette J. J., Gopal R., Rosmaita B., Van Gutch D.; “Genetic Algorithm for the TSP”; *Proceedings of the 1st Int. Conf. on Genetic Algorithms*, Pittsburgh, PA. (1991).
- [15] Gupta J., “A functional heuristic algorithm for the flowshop scheduling problem”, *Operational Research Quarterly* 22, pp. 39-48 (1971).
- [16] Jackson J. R.; “Scheduling a production line to minimize maximum tardiness”, *Research Report 43, Management Science Research Project, University of California*; Los Angeles, (1955).
- [17] Johnson S. M. “Optimal two and three stage production”; *Naval Research Logistics Quarterly*, 1, pp 61-67, (1954).
- [18] Larrañaga P. and Lozano J.A.; “Estimation of Distribution Algorithms”. A New Tool for Evolutionary Computation. *Kluwer Academic Publishers*, (2002).
- [19] Lenstra J. K., Rinnooy Kan A. H., “Computational complexity of scheduling under precedence constraints”, *Operations Research*, 26, pp 22-35, (1978).
- [20] Leung Joseph. “Handbook of Scheduling: Algorithms, Models and Performance Analysis”, *Chapman & Hall/CCR Computer and Information Sciences Series* (2004).
- [21] Madera J., Dorronosoro B.; “Estimation of distribution algorithms, Metaheuristic procedures for training neural networks”; *Springer Science and Business Media*, (2006).
- [22] Morton T., Pentico D., “Heuristic scheduling systems”, *Wiley series in Engineering and technology management. John Wiley and Sons, INC* (1993).
- [23] Mühlenbein H., Mahnig T., and Ochoa A.; “Schemata, distributions and graphical models in evolutionary optimization”. *Journal of Heuristics*, 5, pp 215–247, (1999).
- [24] 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).
- [25] Mühlenbein H. and Voigt H.M.; “Gene pool recombination in genetic algorithms”. *Metaheuristics: Theory and applications*, pp 53–62, (1996).
- [26] Nawaz M., Enscore E., Ham I., “A heuristic algorithm for the m-machine n-job flow shop sequencing problem”. *Omega* vol II, pp 11-95 (1983).

- [27] Oliver, I., Smith D., and Holland J.; “A study of permutation crossover operators on the traveling salesman problem”, in *European Journal of Operational Research*, pp. 224-230 (1986).
- [28] Palmer D., “Sequencing jobs through a multistage process in the minimum total time- A quick method of obtaining a near optimum”, *Operational Research Quarterly* 16, pp 101-107 (1965).
- [29] Pinedo M.; “Scheduling- Theory, Algorithms, and Systems. *Prentice Hall International in Industrial and System Engineering* (1995).
- [30] Reeves C., “A genetic algorithm for flow shop sequencing”, *Computers and Operations Research*, 22, pp 5-13 (1995).
- [31] Syswerda G.; “Schedule optimization using genetic algorithms”, *Handbook of Genetic Algorithms* , Van Nostrand Reinhold, New York, 21, pp 332-349. 1991.
- [32] Taillard, E. “Benchmarks for basic scheduling problems”, *European Journal of Operational Research*, 64, pp 278-285 (1993).
- [33] Tsujimura Y., Gen M., Kubota E., “Flow shop scheduling with fuzzy processing time using genetic algorithms”. *The 11th Fuzzy Systems Symposium* pp 248-252. Okinawa (1995).