

IMPLEMENTACIÓN DE UNA APLICACIÓN PARA SERVER CALCULATION & FACILITY LAYOUT PLANNING (SC-FLP)

Oscar Javier Parra Ortega¹. oparraor@poligran.edu.co.

Resumen – El presente artículo describe la formulación y metodología para la implementación de un aplicativo que permite el cálculo del número de servidores, junto con la disposición de planta óptima para una instancia FLP (Aplicativo SC-FLP). Se incluyen conceptos de estimación de porcentajes de utilización efectivos, cálculo de superficies parciales como complemento a la implementación del Algoritmo Genético.

Abstract: This paper describes the design and implementation methodology to optimize the distribution layout, along the calculation of the number of servers required in each Workstation (SC-FLP Application). Concepts related to bottleneck rates, utilization and partial surfaces calculation (evolution, gravitation and static surfaces) are included with the Genetic Algorithm implementation for FLP.

1. INTRODUCCIÓN

La disposición de planta (Facility Layout) constituye junto con la localización de plantas, una de las actividades principales en el proceso de planeación de instalaciones [1]. Específicamente, el problema de determinar la mejor ubicación posible de las distintas áreas que componen una instalación, se conoce como Facility Layout Problem (FLP). En este contexto, un área de la planta o “facility”, es cualquier entidad que permita desempeñar un trabajo. Puede ser una máquina, una estación de trabajo, una celda de manufactura, una bodega o un departamento, entre otros [2]. Es claro que la ubicación de las áreas de trabajo que hacen parte de una planta de producción, tiene un impacto significativo en la productividad, el nivel de inventario en proceso, los costos de manufactura, y los tiempos de ciclo. Se considera que una apropiada localización de planta permite reducir hasta en un 50% los costos de operación total [1]. La naturaleza combinatoria de este tipo de problemas y su complejidad, hacen que los FLP sean generalmente NP-Hard [3].

La formulación original del FLP se remonta a Armour y Buffa [4] tal como se enuncia a continuación: Sean i una colección de n departamentos de área rectangular, con dimensiones fijas y conocidas h_i y w_i , estos deben ser ubicados al interior de una instalación de área $H \times W$. Entre cada par de departamentos (i, j) hay un flujo de material $F(i, j)$, así como un costo unitario asociado $C(i, j)$ al transportar el material entre ambos departamentos. El objetivo del FLP es particionar la instalación de dimensiones $H \times W$, en n áreas que representan a cada uno de los departamentos, de forma que la mejor partición τ obtenida minimice el costo por tráfico total al interior de la instalación:

$$T = \sum_{i=1}^n \sum_{j=1}^n C_{ij} \cdot F_{ij} d_{ij}^{\tau} \quad (1)$$

Donde d_{ij}^{τ} corresponde a la distancia obtenida entre los centroides de los departamentos (i, j) sujetos a la distribución de una partición τ específica.

Algunos artículos de revisión bibliográfica sobre FLP, hacen hincapié en el tipo de formulación, el tipo de sistema de manufactura y la forma de las áreas de trabajo [5]. Cabe resaltar entre las revisiones más recientes, la realizada por Driera et al. [6] la cual hace énfasis en los métodos de resolución. Entre los artículos que abordan el problema de localizar n áreas rectangulares en una instalación de área R , se resalta entre los métodos exactos, la formulación de Montreuil reseñada por por parte de Meller et al. [7]. En dicho artículo, se resalta que aunque este enfoque de

¹ Ingeniero Industrial. Docente Facultad de Ingeniería y Ciencias Básicas – Institución Universitaria Politécnico Grancolombiano. El proyecto de investigación del cual es producto este documento (2011-FICB-CB-TC), ha sido financiado por la Institución Universitaria Politécnico Grancolombiano.

programación entera mixta permite la inclusión de restricciones adicionales de tipo lineal, su efectividad para alcanzar soluciones óptimas se limita a instancias con 6 departamentos o menos. En general, los enfoques de resolución por métodos exactos, no son apropiados para el caso de problemas de gran escala (con un mayor número de departamentos a localizar). Por esta razón diversos autores han optado por aplicar heurísticas y meta-heurísticas como método de solución. Entre estos métodos, se pueden identificar los enfoques de búsqueda global (como recocido simulado y búsqueda Tabú), y métodos evolucionarios (como colonias de hormigas o algoritmos genéticos). Entre los últimos métodos, los algoritmos genéticos (AG) han sido ampliamente utilizados en problemas de tipo combinatorio, por lo que han ganado popularidad en la resolución de FLP.

Entre otros, se cuentan el trabajo de Misola & Navarro [8], en el que se propone una formulación mono-objetivo de minimización de costos, así como los resultados obtenidos por Mihajlovic et al. [9] en los cuales se hace una evaluación comparativa del desempeño de los AGs versus métodos exactos, en términos de la calidad de la solución obtenida, y del número de iteraciones requeridos para alcanzar un óptimo en instancias conocidas. Los resultados de dicha comparación indican que para instancias de mayor número de departamentos, los algoritmos genéticos presentan una alta probabilidad de obtener soluciones óptimas para el FLP, con menor número de iteraciones que los métodos exactos.

Dado que el FLP es un problema que tiene un alto impacto en la reducción de costos para una planta de manufactura, el objetivo del presente artículo es proponer e implementar un algoritmo genético que pueda ser aplicado para instancias relacionadas con micro, pequeñas y medianas empresas (miPyMEs), que no siempre tienen acceso directo a un proceso de implementación de este tipo de herramientas. Al considerar que de acuerdo con un estudio reciente de la cámara de Comercio de Bogotá [10], el número de miPyMEs pertenecientes a la categoría CIU-D (Industrias manufactureras) asciende a 29.525 empresas registradas, el impacto de implementar una aplicación que resuelva FLP y a la vez calcule el número requerido de servidores por estación de trabajo (de acuerdo con la formulación propuesta por Spearman [11]), tendría un impacto positivo en la reducción de costos y tiempos de ciclo en este tipo de empresas.

2. METODOLOGÍA

Para implementar una solución al FLP, se optó por la implementación de AGs en Visual Basic for Applications. La principal razón para elegir este lenguaje de programación, fue la integración con Microsoft Excel, la cual permitiría una mayor capacidad de distribución de la aplicación al no requerir configuración o instalación de software adicional para su ejecución.

Para la representación del individuo solución al problema FLP, se aplicó la representación del problema de bahías flexibles (flexible-bay layout) [12]. De acuerdo con esta configuración, los departamentos son ubicados en columnas verticales (conocidas como bahías), de forma que cada departamento es ubicado solamente en una bahía. El área principal de la instalación, de dimensiones HxW, se divide en un número de bahías de ancho variable. El número de bahías es un número entero entre 1 y n cuyo valor óptimo se determina al resolver el problema. Cada una de las bahías se divide a su vez en departamentos rectangulares de igual ancho pero de longitud variable, tal como se aprecia en la figura 1.

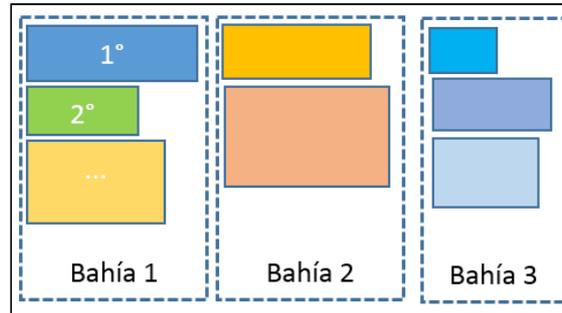


Figura 1. Representación de un Flexible-bay layout.

Cada individuo solución, es codificado en la representación compacta de un vector de tamaño $[1 \times n]$, el cual representa una permutación de tamaño n :

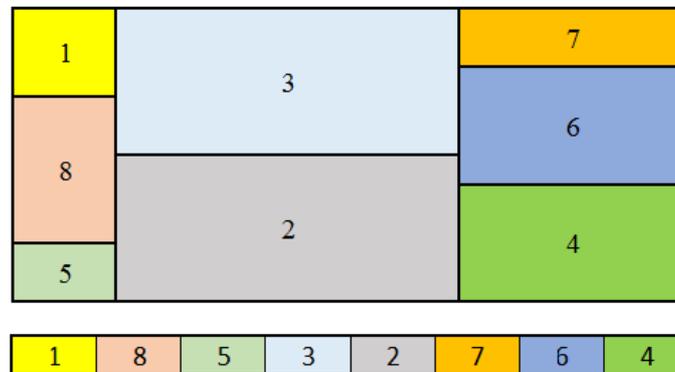


Figura 2. Codificación de un individuo solución.

Cabe anotar respecto al enfoque de solución elegido, que los AGs fueron descritos inicialmente por Holland [13] como algoritmos que replican los mecanismos de selección natural existentes en los sistemas biológicos, y que hacen parte del conjunto de heurísticas de optimización global. Los componentes distintivos de los AGs son, la codificación del individuo solución, la existencia de un mecanismo de selección a partir de una población inicial de individuos, el cual se basa en la evaluación a partir de una función de ajuste, un mecanismo de cruce y un mecanismo de mutación [14]. Para el mecanismo de cruce, se tuvieron en cuenta los diferentes mecanismos de cruce disponibles para para codificación de tipo combinatoria reseñados por Chan & Tansri [15], y se implementó el algoritmo PMX (Algoritmo de cruce basado en una correspondencia parcial – Partially Matched Crossover). Respecto al algoritmo de mutación, se propuso el intercambio de dos genes del vector solución (Swap mutation), de acuerdo con una probabilidad de mutación parametrizable al inicio de la ejecución del algoritmo.

En la implementación actual, se tiene una función mono-objetivo equivalente a la ecuación (1), pero con la posibilidad de extender dicha función a multi-objetivo incluyendo una penalización por restricciones particulares de proximidad entre departamentos. Para reducir la complejidad de la formulación, se asume que los puntos de Entrada/Salida están localizados en el centroide de cada departamento. También se asume que para una partición específica z , la distancia entre dos departamentos (i, j) con coordenadas de centroides (x_i^z, y_i^z) y (x_j^z, y_j^z) , será calculada como distancia rectilínea, así:

$$d_{ij}^z = [x_i^z - x_j^z] + [y_i^z - y_j^z] \quad (2)$$

Finalmente, se incluyó como apoyo adicional para el usuario de la aplicación implementada, el cálculo previo del número de servidores requeridos para cada departamento, con base en la formulación propuesta por Spearman [11], en la cual, sea λ la tasa global esperada del sistema de manufactura, y μ_i la tasa individual de proceso de un servidor en el departamento o estación de trabajo i , que tiene una eficiencia de operación ef_i , entonces:

$$k_i = \frac{\lambda}{\mu_i(1-ef_i)} \quad (3)$$

es el número promedio de servidores ocupados en la estación de trabajo i . De acuerdo con lo anterior, el número requerido de servidores para la estación i es igual a:

$$m_i = [k_i] \quad (4)$$

y el porcentaje de ocupación de cada departamento es igual a:

$$\rho_i = \frac{k_i}{m_i} \quad (5)$$

3. RESULTADOS

En la implementación del AG en VBA, junto con las ecuaciones de balanceo de estaciones de trabajo y cálculo de número de servidores, se utilizó una instancia de prueba con 8 estaciones de trabajo, con la siguiente información:

Activa/Inactiva		Nombre estación	Nombre servidor	Largo (m)	Ancho (m)	Accesibilidad*
Activa		Estación torneado	Torno	2.00	1.50	3 lado(s)
Activa		Estación pulido	Pulidora	3.00	2.00	2 lado(s)
Activa		Estación fresado	Fresadora	1.50	1.30	2 lado(s)
Activa		Estación limado	Limadora	2.00	1.80	2 lado(s)
Activa		Estación empaque	Empacador	1.50	1.50	1 lado(s)
Activa		Bodega materia prima	Bodega A	5.00	4.00	0 lado(s)
Activa		Bodega producto terminado	Bodega B	5.00	3.50	0 lado(s)
Activa		Estación inspección	Servidor inspección	2.00	2.00	2 lado(s)

Figura 3. Información básica para configuración de estaciones de trabajo (departamentos)

Respecto al tráfico de productos entre las distintas estaciones de trabajo, esta fue la información mínima requerida para el cálculo de la función de ajuste planteada en la ecuación (1):

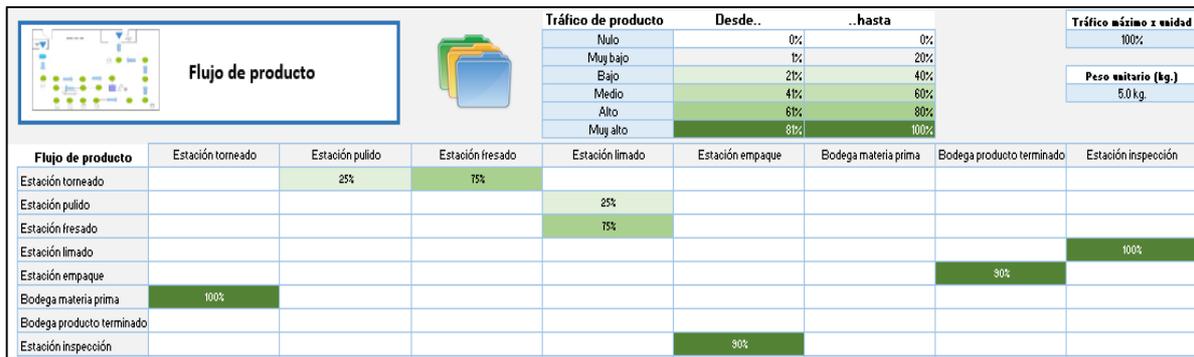


Figura 4. Matriz de tráfico entre estaciones.

Adicionalmente, se incluyó la posibilidad de parametrizar los siguientes parámetros globales de la planta: dimensiones, tipo de industria, throughput esperado, y costo unitario por unidad de distancia recorrida.



Figura 5. Dimensiones layout planta (Instancia con 8 estaciones de trabajo).

Con base en los parámetros individuales de tasas de proceso de cada estación, se calcula el número mínimo de servidores requeridos en cada estación, teniendo en cuenta la superficie estática, de gravitación y de evolución de cada estación de trabajo, en función del tipo de industria seleccionada, y del grado de accesibilidad requerido por cada servidor.

Para estos cálculos se aplicó el método propuesto por P.F. Guerchet y citado por López et al. [16] para la estimación de superficies parciales:

$$SG = SE * l \quad (6)$$

Donde SE corresponde a la superficie realmente ocupada por los servidores (superficie estática), SG corresponde a la superficie reservada para los operarios que trabajan en cada máquina, y *l* corresponde al número de lados del servidor o máquina, por los que es accesible. Adicionalmente, la superficie necesaria para los desplazamientos entre estaciones de trabajo y para labores de mantenimiento (superficie de evolución) se calculó con base en:

$$SV = (SE + SG) * f \quad (7)$$

Donde *f* es un factor que varía dependiendo del tipo de industria.

4. CONCLUSIONES

En este artículo, se describe la metodología y formulación utilizada para implementar una aplicación en VBA para el cálculo del número mínimo de servidores y la disposición de planta (SC-FLP). El algoritmo propuesto converge con

rapidez a la solución óptima, lo cual se refleja tanto en el valor promedio como en el valor mínimo de la función de adaptación (función objetivo o fitness function).

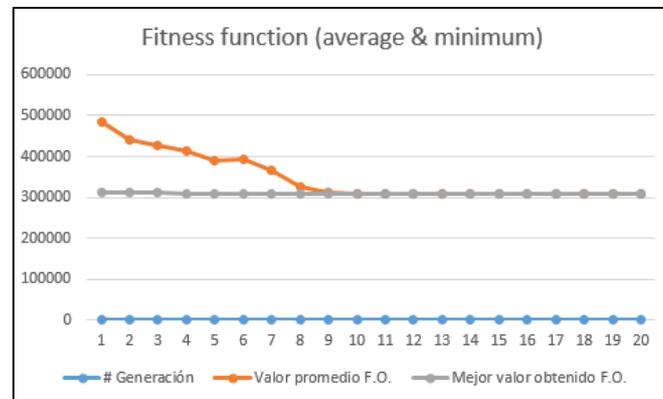


Figura 6. Convergencia a la solución óptima para la instancia propuesta.

En la generación 4 se obtuvo el óptimo global (\$309.202), mientras que en la generación 11, se obtuvo convergencia del valor promedio de la función objetivo (como criterio de parada opcional).

El AG implementado permite modificar parámetros de ejecución tales como el tamaño de la población inicial, el número de generaciones como criterio de parada, la tasa de cruce y la tasa de mutación.

Para futuros trabajos se sugiere explorar la posibilidad de modelaje de áreas no rectangulares en cada estación de trabajo, así como la inclusión de una función multi-objetivo para la evaluación de soluciones óptimas. También es posible agregar restricciones de no proximidad para áreas específicas, áreas no rectangulares para el layout de la instalación general, y la implementación de algoritmos de cruce y mutación diferentes a los ya aplicados.

Dado el alto impacto que puede generar una mejor distribución de estaciones de trabajo al interior de una planta de manufactura, en términos de tiempo de ciclo y costos totales, se recomienda validar la utilidad de la implementación en casos reales al interior de miPyMEs, para generar una versión más robusta con base en la retroalimentación obtenida de la utilización de esta aplicación para optimizar layouts en plantas existentes.

REFERENCIAS

- [1] J. A. Tompkins y J. A. White, Facilities Planning, New York: John Wiley, 1996.
- [2] S. S. Heragu, Facilities design, Boston: PWS Publishing Company, 1997.
- [3] M. R. Garey y D. S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, New York: WH Freeman, 1979.

- [4] G. Armour y E. Buffa, «A heuristic algorithm and simulation approach to relative allocation of facilities,» *Management Science*, vol. 9, pp. 294-309, 1963.
- [5] A. Kusiak y S. S. Heragu, «The facility layout problem,» *European Journal of Operational Research*, vol. 29, nº 3, pp. 229-251, 1987.
- [6] A. Drira, H. Pierreval y S. Hajri-Gabouj, «Facility layout problems: A survey,» *Annual Reviews in Control*, vol. 31, pp. 255-267, 2007.
- [7] R. D. Meller y K.-Y. Gau, «The facility layout problem: recent and emerging trends and perspectives,» *Journal of Manufacturing Systems*, vol. 15, nº 5, pp. 351-366, 1996.
- [8] M. G. Misola y B. B. Navarro, «Optimal Facility Layout Problem Solution Using Genetic Algorithm,» *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, vol. 7, nº 8, pp. 622-627, 2013.
- [9] I. Mihajlovic, Z. Zivkovic, N. Strbac, D. Zivkovic y A. Jovanovic, «Using Genetic Algorithms to Resolve Facility Layout Problem,» *Serbian Journal of Management*, vol. 2, nº 1, pp. 35-46, 2007.
- [10] Cámara de Comercio de Bogotá, «Estadísticas del número de empresas en Bogotá y 59 municipios de Cundinamarca,» 10 07 2012. [En línea]. Available: http://www.ccb.org.co/documentos/10512_estadisticasempresasbd.xls. [Último acceso: 28 07 2014].
- [11] M. L. Spearman, *Factory Physics*, Waveland Press, Incorporated, 2011.
- [12] D. M. Tate y A. E. Smith, «Unequal-area facility layout by genetic search,» *IIE Transactions*, vol. 27, pp. 465-472, 1995.
- [13] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press, 1975.
- [14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, Boston, MA: Addison Wesley, 1989.
- [15] K. C. Chan y H. Tansri, «A study of genetic crossover operations on the facilities layout problem,» *Computers and Industrial Engineering*, vol. 26, nº 3, pp. 537-550, 1994.
- [16] E. López Méndez, J. A. Estrada Beltrán, A. Ramírez Leyva y J. I. Ruiz Ibarra, «Plant Re-Distribution to Optimize Production using Mathematical Models,» *Revista Aristas: Ciencia e Ingeniería. Facultad de Ciencias Químicas e Ingeniería. UABC*, vol. 1, nº 2, pp. 13-26, 2012.
- [17] K. Y. Tam, «Genetic algorithms, function optimization, and facility layout design,» *European Journal of Operational Research*, vol. 63, 1992.

- [18] M. Ficko, M. Brezocnik y J. Balic, «Designing the layout-and multiple-rows flexible manufacturing system by genetic algorithms,» *Journal of Materials Processing Technology*, pp. 150-158, 2004.
- [19] M. Adel El-Baz, «A genetic algorithm for facility layout problems of different manufacturing environments,» *Computers & Industrial Engineering*, vol. 47, pp. 233-246, 2004.
- [20] Y. S. Wong, F. T. S. Chan y K. L. Mak, «A genetic algorithm for facility layout problems,» *Journal of Computer Integrated Manufacturing Systems*, vol. 1, nº 1-2, pp. 113-123, 1998.