

La optimización evolutiva multi objetivo en la confección de equipos de desarrollo de software: una forma de lograr la calidad en el producto final

(Evolutionary multi-objective optimization for software development teams building: a way of obtaining quality in the final product)

Yasnalla Rivero Peña¹, Pavel Novoa Hernández², Yandi Fernández Ochoa³

Resumen:

En la presente investigación se propone un modelo matemático que permite enfocar el proceso de creación de equipos de desarrollo de software como un problema multi objetivo discreto. Los objetivos considerados son el nivel de competencia y el nivel de utilización de los profesionales en el equipo. Dada la complejidad del espacio de búsqueda del problema, se propone la aplicación de un método de optimización aproximado. Específicamente se seleccionó el algoritmo genético NSGA-II basado en el concepto de Dominancia de Pareto. Este método fue aplicado en seis escenarios diferentes con el objetivo de analizar la calidad de soluciones que obtiene. En general se puede afirmar que el método es eficiente y obtiene soluciones (asignaciones) de alta calidad.

Palabras clave: confección de equipos; optimización multiobjetivo; dominancia de Pareto

Abstract:

In this research a mathematical model to approach the process of creating software development teams as a discrete multi-objective problem is proposed. The objectives considered are the level of competition and the level of utilization of professionals in the team. Given the complexity of the problem search space, the application of an approximate optimization method is proposed. Specifically, the genetic algorithm NSGA-II based on the concept of Pareto dominance was selected. This method was applied in six different scenarios in order to analyze the quality of the obtained solutions. In general we can say that the method is efficient and gets solutions (assignments) of high quality.

Keywords: team building; multiobjective optimization; Pareto dominance

¹ Universidad de Holguín "Oscar Lucero Moya", Facultad de Informática y Matemática, Holguín – Cuba (yasnalla@facinf.uho.edu.cu)

² Universidad de Holguín "Oscar Lucero Moya", Facultad de Informática y Matemática, Holguín – Cuba (pnovoa@facinf.uho.edu.cu)

³ Universidad Tecnológica Equinoccial, Facultad de Ciencias de la Ingeniería, Campus Santo Domingo, Quito – Ecuador (yandi.fernandez@ute.edu.ec)

1. Introducción

El aseguramiento de la calidad es un factor clave en el proceso de realización del software. En este contexto la selección adecuada del personal que formará los equipos de desarrollo es crucial e influye directamente en la efectividad del proceso (Llerena, 2004).

En todas las metodologías de desarrollo actuales existen roles de trabajo que deben ser ocupados por profesionales capaces. Por ejemplo, en el caso del Proceso Unificado de Rational (RUP)(Ivar Jacobson, 2000), se pueden contar más de 30 roles agrupados en Analistas, Desarrolladores, Gestores, Apoyo, Especialistas de Prueba, y Otros roles. Obviamente RUP está diseñada para proyectos grandes. A diferencia de esta, las metodologías ágiles, y por tanto dedicadas a proyectos más pequeños, poseen un número de roles considerablemente menor. Tal es el caso de la metodología Programación Extrema (XP)(Beck, 2004), la cual cuenta con un total de 7 roles: Programador, Cliente, Encargado de pruebas, Encargado de seguimiento, Entrenador, Consultor, y Gestor. A excepción del Cliente (que forma parte del desarrollo del software), el resto debe ser seleccionado adecuadamente para lograr los objetivos del proyecto.

Independientemente del tipo de metodología, la tarea de crear equipos de desarrollo a través de la asignación de profesionales a los roles, es compleja. Lo anterior se justifica si se tiene en cuenta que los profesionales poseen distintos niveles de competencias (habilidades, conocimientos, etc.) y que los roles requieren distintos niveles de dichas competencias. Además, es habitual que los mismos profesionales ocupen diversos roles en un mismo proyecto. Esto provoca un nivel de sobreutilización que pudiera ser beneficioso en algunos casos, pero desde el punto de vista de los profesionales constituye un aumento de su contenido de trabajo.

La presente investigación trata de resolver esta dificultad mediante la automatización de este proceso. En esencia, se propone un modelo matemático que permite enfocar esta situación como un problema de optimización multiobjetivo discreto. Dada la complejidad del espacio de búsqueda de dicho problema, se propone en consecuencia la aplicación de un método aproximado moderno. Específicamente se trata del algoritmo genético NSGA-II, basado en el concepto de dominancia de Pareto. Dicho método fue analizado a través de 6 casos de estudio (escenarios) con diferentes número de roles y profesionales.

2. Metodología computacional

A continuación se exponen los principales pasos seguidos en la metodología computacional empleada para darle solución al problema de la confección de equipos de desarrollo de software. El problema que se pretende resolver en esta investigación queda definido como sigue:

Realizar la mejor asignación de un grupo de profesionales al conjunto de roles de la metodología RUP, maximizando el nivel de competencias requerido en cada rol, y minimizando la sobreutilización general de los profesionales empleados.

Para resolver este problema se hace necesario modelar los aspectos que lo describen. Por ejemplo, una cuestión importante es cómo medir el grado de competencias de un profesional o un rol determinado. Asimismo, resulta necesario definir el grado de utilización óptima de los profesionales en los flujos de trabajo. Finalmente, aunque no queda de manera explícita en la definición dada anteriormente, este problema incluye casos especiales como por ejemplo, cuando el número de profesionales es inferior a la cantidad de roles, en cuyo caso habría que asignar un mismo profesional a varios roles. En la Sección siguiente se describe el modelo matemático propuesto que incluye estas y otras cuestiones.

Modelo del problema

A partir de la descripción dada en la sección anterior, el modelo matemático que se propone es el siguiente:

Datos:

$R = \{1, 2, \dots, n\}$ Roles a ocupar por los profesionales.

$C = \{1, 2, \dots, m\}$ Competencias a considerar en la asignación de los roles.

$P = \{1, 2, \dots, q\}$ Profesionales disponibles para ocupar los roles.

$M \in [0, 1]^{(n \times m)}$ Matriz de roles vs. competencias.

$M^* \in [0, 1]^{(p \times m)}$ Matriz de profesionales vs. competencias.

VARIABLES DE DECISIÓN:

$X = (x_1, x_2, \dots, x_n)^T$ donde $x_j \in P (j = 1, \dots, n)$

Funciones objetivo:

- 1) Error del nivel de competencia requerida:

$$\min F_{comp}(X) = \sum_{j=1}^n (m_{ji}^* - m_{ji})$$

Donde $m_{ji} \in M$ es el nivel de competencia del profesional x_j en la competencia i , mientras que $m_{ji}^* \in M^*$ es el nivel de competencia i requerido por el rol j .

- 2) Error del nivel de sobre-utilización de los profesionales:

$$\min F_{utilid}(X) = \sum_{j=1}^n (u(x_j) - 1)$$

Donde $u(x_j)$ es una función que devuelve la cantidad de veces que el profesional es empleado en los roles.

Como se aprecia en este modelo, el objetivo es obtener soluciones que posean valores mínimos de estos errores. En particular, las soluciones del problema son vectores de combinaciones de números enteros, los cuales a su vez son índices correspondientes a los profesionales. Habitualmente, en estos tipos de problemas la cantidad de soluciones (tamaño del espacio de búsqueda), puede determinarse mediante fórmulas conocidas del análisis combinatorio. En efecto, sea Ω el espacio de búsqueda, es fácil ver que su tamaño depende del número de profesionales disponibles (p) y el número de roles a ocupar (r). Más formalmente, la cantidad de soluciones viene dada por el número total de r - arreglos que se puede formar a partir de p elementos, en (este caso de profesionales):

$$|\Omega| = p^r$$

Para tener una idea de cuánto aumenta el tamaño de este espacio de búsqueda en función de p y r , suponga el caso en que se tienen que asignar 20 profesionales a 10 roles. En este caso $|\Omega| = 20^{10} = 1.24 \times 10^{13}$, sin dudas un número grande. Claro está, entre estas soluciones también se encuentran algunas que resultarían poco prácticas y por tanto no deseables, tales como asignar a todos los roles el mismo profesional, esto es, cuando las componentes del vector solución cumplen que $x_j = a (j = 1, \dots, n)$. En la literatura existen varias investigaciones acerca de este tipo problema, conocido frecuentemente como problemas de construcción de equipos (*team building problems*), de entre los que se destacan los de (Wegener, 2005);(Novoa, 2013);(Ahmed, 2013).

Dada su complejidad, la aplicación de técnicas exactas (Beveridge Gordon S. , 1970; Griva, 2009; Kaufmann, 1978) resulta en algunos casos poco práctica y en otros simplemente imposible. En este contexto, los métodos numéricos aproximados justifican su aplicación, en especial las denominadas metaheurísticas (Edmund K. Burke, 2005; Melián, 2003) que han resultado ser eficientes en el tratamiento de este tipo de problemas. En la próxima sección se detalla el método de solución empleado. Las metaheurísticas son procedimientos que se caracterizan por obtener soluciones de alta calidad en un tiempo razonable.

3. Resultados y discusión

En la actualidad existen diversos métodos para solucionar el modelo anterior. Sin embargo, dada la complejidad del problema, resulta adecuada la aplicación de métodos pertenecientes al campo de la Computación Inteligente (o Soft Computing). Estos métodos, denominados metaheurísticas, encuentran soluciones muy cercanas a la óptima en un tiempo razonable. Dentro de las metaheurísticas que pueden resolver eficientemente el modelo anterior se encuentran las basadas en el concepto de dominancia de Pareto (K. Deb, 2005; Knowles, 2008). Básicamente, este

concepto permite definir el conjunto de soluciones óptimas teniendo en cuenta el valor de cada función objetivo por separado. Como consecuencia se puede decir que una solución A domina a otra B, si A posee mejores valores en las funciones objetivo en relación a B. En particular, al conjunto de soluciones que no son dominadas por ninguna otra se le llama Conjunto de Pareto, mientras que a los valores correspondientes de las funciones objetivo se le denomina Frente de Pareto. Entonces, bajo estas consideraciones, resolver un problema multiobjetivo consiste en encontrar el conjunto de soluciones no dominadas más cercanas al Conjunto de Pareto. Las ventajas de aplicar un método que se base en estas consideraciones es que la posibilidad de contar al final con varias soluciones igual de buenas. Queda por parte del decisor (o usuario final), seleccionar las soluciones que le resulten útiles en su contexto.

Una de las metaheurísticas más estudiadas (y de excelente rendimiento) en problemas de este tipo es la denominada NSGA-II (e. Deb, 2002). Este método está basado en el paradigma computacional Algoritmos Genéticos (GA), y su principal característica es la presencia de un novedoso procedimiento de ordenamiento de las soluciones no dominadas, el cual posee una baja complejidad computacional. En el trabajo original, NSGA-II fue probado en problemas multiobjetivo continuos, mientras que el problema que nos ocupa posee un espacio de búsqueda discreto. De manera que se necesita modificar los operadores de variación del algoritmo con el objetivo de que tengan sentido en el contexto discreto. Con respecto a esta cuestión, existen varias posibilidades. En el caso concreto de esta investigación seleccionamos los siguientes:

- Operador de mutación: Uniforme con tasa de mutación 0.01.
- Operador de cruzamiento: Scattered.
- Operador de selección: Tournament.

El resto de los pasos del algoritmo NSGA-II se dejaron sin variación. Como entorno de experimentación se empleó el software Matlab versión 2012b, en particular la aplicación Global Optimization perteneciente a la caja de herramienta de igual nombre. Esta aplicación brinda una interfaz gráfica al usuario que posibilita el establecimiento de los elementos del modelo, así como los operadores mencionados anteriormente al algoritmo NSGA-II.

Ejemplos ilustrativos

Con el objetivo de mostrar el funcionamiento del método descrito anteriormente en la solución del modelo propuesto, hemos diseñado tres casos de estudio. Sus características se muestran en la **Error! Reference source not found.**, como se aprecia, se tratan de tres escenarios posibles en la creación de equipos en metodologías de desarrollo como RUP. Cada escenario varía el número de roles, profesionales y competencias, con la consecuente repercusión en el tamaño del espacio de búsqueda (No. Soluciones, última columna).

Tabla 1. Casos de estudio

Caso de estudio	No. Roles	No. Profesionales	No. Competencias	No. Soluciones
CE1	10	10	5	1.00×10^{10}
CE2	10	20	10	1.02×10^{13}
CE3	10	40	15	1.05×10^{16}
CE4	20	10	5	1.00×10^{20}
CE5	20	20	10	1.05×10^{26}
CE6	20	40	15	1.10×10^{32}

En relación a las respectivas matrices de roles vs. Competencias y profesionales vs. Competencias, se han generado con valores aleatorios en el intervalo [0,1], siguiendo una distribución uniforme. En general se realizaron 5 ejecuciones por cada escenario.

Los resultados se muestran en la Figura 1 y Figura 2, las cuales corresponden al Frente de Pareto (de acuerdo a los objetivos del modelo: error del nivel de competencias y error del nivel de utilización), y al Conjunto de Pareto. En este último caso hemos representado a los profesionales a través de sus índices, los cuales a su vez han sido representados por colores. De este modo se puede obtener una vista sobre la diversidad de las soluciones obtenidas en cada escenario.

A través de las gráficas de ambas Figuras, se puede ver que el algoritmo NSGA-II obtiene soluciones con bajos niveles de ambos objetivos, lo cual permite seleccionar las mejores asignaciones (Fig. 1). Asimismo, nótese que la diversidad de las soluciones se ve claramente afectada cuando el número de profesionales es menor que el número de roles. Tal es el caso del caso de estudio CE4, donde se deben asignar 10 profesionales a 20 roles (Fig. 2-d). En este caso, obsérvese la prevalencia del color azul, el cual indica la asignación de un mismo profesional (representado por ese color) a varios roles. Evidentemente, estas soluciones tienen asociado un error alto del nivel de utilización. No obstante, el algoritmo es capaz de encontrar otras con mejor error de utilización en este mismo escenario, como por ejemplo las soluciones desde la 20 a la 25.

En sentido general, se puede concluir que el algoritmo NSGA-II resulta efectivo en el modelo considerado.

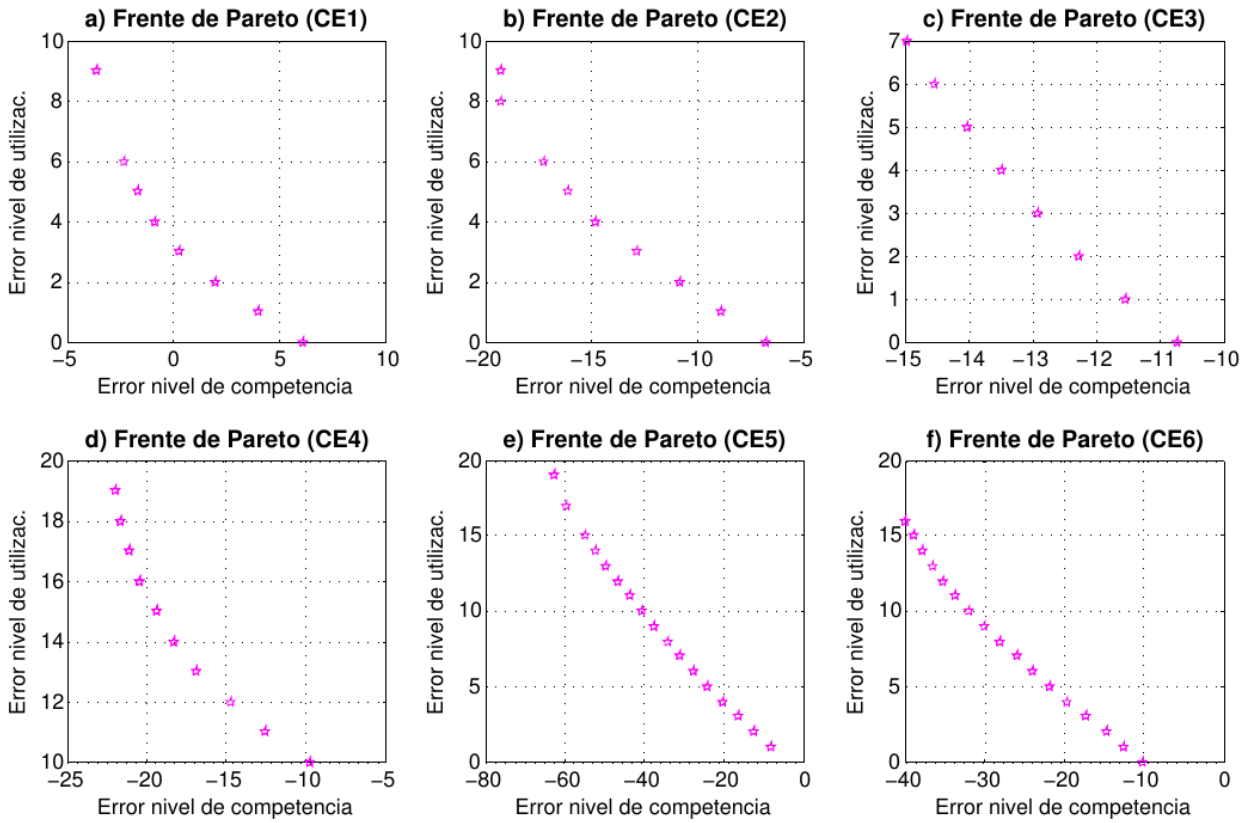


Figura 1. Frentes de Pareto correspondientes a los casos de estudio considerados.

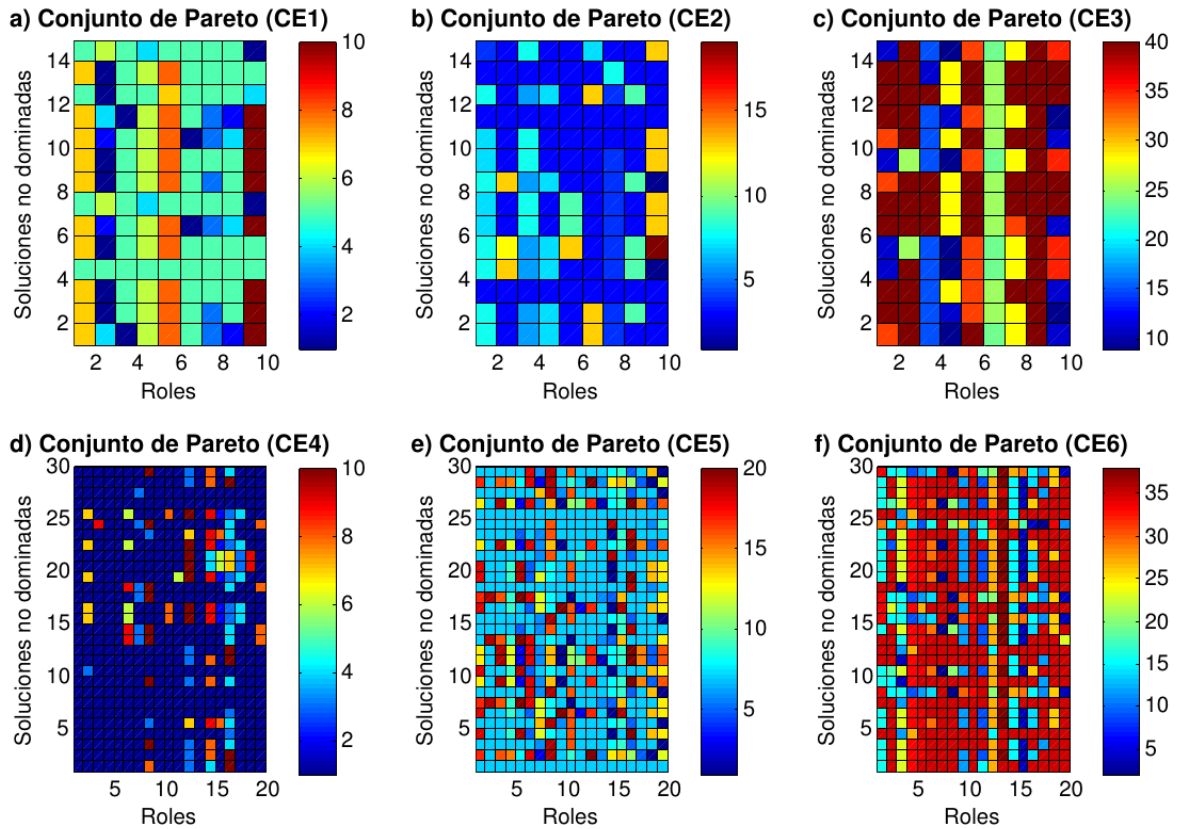


Figura 2. Conjunto de Pareto (soluciones no dominadas) correspondientes a los casos de estudio considerados. Los colores representan los índices de los profesionales asignados a cada rol.

4. Conclusiones y Recomendaciones

En este trabajo se presentó un modelo multiobjetivo para resolver el problema de la creación de equipos de desarrollo de software de manera óptima. Dicho modelo establece dos objetivos a tener en cuenta en la asignación de profesionales a los roles: el primero está relacionado con el nivel de competencia que poseen los profesionales en cada rol, mientras que el segundo está dedicado a cuantificar la sobreutilización de los profesionales en los equipos.

La idea en general es obtener equipos de trabajo formado por los profesionales más capaces en cada rol, y al mismo tiempo, que no se sobre utilicen dichos profesionales, esto es, no se asigne el mismo profesional a varios roles.

Adicionalmente, se estudió la complejidad del espacio de búsqueda de dicho modelo, la cual resultó ser alta teniendo en cuenta que el número de soluciones aumenta conforme el número de profesionales y roles crecen. En consecuencia, se aplicó como método de solución, el algoritmo NSGA-II de excelentes resultados en diversos contextos reales. Dicho algoritmo fue modificado para tratar el tipo de solución (discreta en este caso) que presenta el modelo propuesto. Con el objetivo de estudiar como el algoritmo seleccionado resuelve el modelo propuesto, se diseñaron 6 casos de estudio. Los resultados obtenidos confirman la efectividad del algoritmo.

Como trabajos futuros proponemos el estudio de otros algoritmos y enfoques que resuelvan este problema, así como la aplicación de estos a otros escenarios similares en el contexto de la creación automatizada de equipos.

Bibliografía

- Ahmed, F., Deb, K., Jindal, A. (2013). Multi-objective optimization and decision making approaches to cricket team selection *Applied Soft Computing*. *Applied Soft Computing Journal*, 13(1), 402-414.
- Beck, K. (2004). *Extreme Programming Explained: Embrace Change*. Vol 2, 58-71.
- Beveridge Gordon S. , G. R. S. S. (1970). *Optimization: Theory and Practice*. 153-227.
- Deb, e. (2002). *A fast and elitist multi-objective genetic algorithm: Nsga-II* ,*IEEE Transactions in Evolutionary Computation*,6(2),182-197.
- Deb, K. (2005). *Search methodologies. Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science-Business Media, LLC, 273-316.
- Edmund K. Burke, G. K. (2005). *Search methodologies. Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science-Business Media, LLC, 620.
- Griva, I., Stephen G. Nash, Ariela Sofer. (2009). *Linear and nonlinear optimization*. Paper presented at the Society for Industrial and Applied Mathematics.

- Ivar Jacobson, G. B., James Rumbaugh. (2000). El Proceso Unificado de Desarrollo de Software.135-289.
- Kaufmann, A., A. Henry Labordere. (1978). Métodos y modelos de la Investigación de Operaciones. In E. Continental (Eds.),478.
- Knowles, J., David Corne, y Kalyanmoy Deb. (2008). Multiobjective Problem Solving from Nature.From Concepts to Applications. In S.-V. B. Heidelberg (Eds.)pp. 1-28).
- Llerena, G. M. G. (2004). *Experiencias en la implantación de un sistema de gestión de la calidad para el proceso de producción de software*. Paper presented at the Convención Internacional.1-4.
- Melián, B., J.A. Moreno Pérez, y J.M. Moreno Vega. (2003). Metaheurísticas: Una visión global. *Revista Iberoamericana de Inteligencia Artificial*, 19, 7-28.
- Novoa, P.,M.A Novoa Hernández y Y. Rivero Peña (2013). Propuesta de técnicas evolutivas para la confección automática de tribunales de trabajo de diplomas. *Revista Cubana de Ciencias Informáticas*, 7(4), 90-99.
- Wegener, I. (2005). Complexity Theory. Exploring the Limits of Efficient Algorithms. *Springer-Verlat*, 308.

