

The logo for the Revista Cubana de Ciencias Informáticas (RCCI) is displayed in a blue, bold, sans-serif font. It is contained within a white rectangular box with rounded corners. Below the box, there are three faint, light blue curved lines.

Revista Cubana de Ciencias Informáticas

ISSN: 1994-1536

ISSN: 2227-1899

Editorial Ediciones Futuro

Infante Abreu, Ana Lilian

Tratamiento de restricciones en el problema de formación de equipos
de proyectos de software. Técnicas de penalización y preservación.

Revista Cubana de Ciencias Informáticas, vol. 13, núm. 3, 2019, Julio-Septiembre, pp. 13-29

Editorial Ediciones Futuro

Disponible en: <https://www.redalyc.org/articulo.oa?id=378365913002>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

UAEH The logo for the Redalyc system, featuring the text "redalyc.org" in a stylized font with a red dot above the "i".

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 29/10/18 | Aceptado: 08/06/19 | Publicado: 24/09/19

Tratamiento de restricciones en el problema de formación de equipos de proyectos de software. Técnicas de penalización y preservación.

Treatment of restrictions on the problem of forming teams software projects. Penalization and preservation techniques.

Ana Lilian Infante Abreu ^{1*}

¹ Universidad Tecnológica de La Habana José Antonio Echevarría (CUJAE). Calle 114 # 11901 / Ciclovía y Rotonda, Marianao, La Habana. ainfante@ceis.cujae.edu.cu

* Autor para correspondencia: ainfante@ceis.cujae.edu.cu

Resumen

La formación de equipos de proyectos de software es un problema complejo, en tanto debe considerar varios factores, como son, la asignación de las personas con las competencias apropiadas a los roles, la carga de trabajo, las incompatibilidades entre los miembros, entre otros. Este proceso es aún más complejo en organizaciones medianas y grandes teniendo en cuenta la cantidad de combinaciones de asignaciones posibles. El trabajo toma como antecedente un modelo que plantea: maximizar las competencias de los trabajadores, minimizar las incompatibilidades entre los miembros del equipo, balancear la carga de trabajo y minimizar el costo de desarrollar software a distancia. Además, incluye doce tipos de restricciones. El modelo citado responde a un problema combinatorio multiobjetivo por lo que para su solución se utilizaron un conjunto de variantes multiobjetivo de los algoritmos metaheurísticos Escalador de Colinas y Recocido Simulado. En trabajos previos se utilizó la técnica de rechazo para el tratamiento de las restricciones obteniéndose que más del 60% de las soluciones que se generan son no factibles, por lo que son desechadas durante la búsqueda. Este trabajo implementa otras técnicas como son penalización y preservación, con las que se obtienen muy buenos resultados.

Palabras clave: técnicas para tratamiento de restricciones, algoritmos metaheurísticos, problema de formación de equipos

Abstract

Forming teams of software projects is a complex problem, as should consider several factors, such as the allocation of people with the appropriate roles responsibilities, workload, incompatibilities between members, among others. This process is even more complex in medium and large organizations considering the number of combinations of possible assignments. The work takes as a model posing history: maximize the skills of workers, minimize incompatibilities between team members, balancing the workload and minimize the cost of developing software remotely. It also includes twelve types of restrictions. Said model responds to a multi-objective combinatorial problem so for its solution set of

multiobjective metaheuristics algorithms variants Hill Climbing and Simulated Annealing were used. In previous work we were used rejection technique for treating restrictions yielding over 60% of the generated solutions are not feasible, so are discarded during the search was used in previous studies. This work implements other techniques such as penalty and preservation, with very good results.

Keywords: *techniques for treatment of restrictions, metaheuristics algorithms, forming teams of software projects problem*

1. Introducción

En los últimos años la industria de software ha obtenido un desarrollo vertiginoso, sin embargo resulta elevada la cifra de proyectos que no culminan con éxito (StandishGroup 2015). Entre las principales causas del fracaso de los proyectos se encuentran las asociadas a factores humanos, como son la asignación no adecuada del personal y los problemas de trabajo en equipo (Ángeles et al. 2005; Neil 2006; StandishGroup 2015). El proceso de formación de equipos de proyectos se torna complejo en medianas y grandes empresas, teniendo en cuenta la totalidad de roles a cubrir y trabajadores disponibles.

En la bibliografía consultada se reportan varios trabajos que proponen modelos que afrontan la asignación de personas a proyectos de software, entre los que se encuentran (Barreto 2003; DeCarvalho 2003; Ngo-The and Ruhe 2008). Este trabajo toma como antecedente el modelo presentado en (André et al. 2011) por considerarse el más completo, ya que toma en cuenta no solo factores individuales para la asignación de personas a roles sino factores que contribuyen a la formación del equipo como un todo.

El modelo propuesto en (André, Baldoquín and Acuña 2011) responde a un problema combinatorio multiobjetivo en tanto considera tres objetivos: maximizar las competencias de los trabajadores, minimizar las incompatibilidades entre los miembros del equipo y balancear la carga de trabajo. Igualmente, el modelo incluye doce tipos de restricciones, que restringe la asignación de personas a roles definidos como incompatibles, la cantidad máxima de roles que pueden ser desempeñados por un individuo, así como la carga máxima de trabajo que este puede tener, entre otras.

Para dar solución al modelo se han utilizado las variantes monobjetivo y multiobjetivo de los algoritmos de trayectoria Escalador de Colinas, Recocido Simulado y Búsqueda Tabú (Infante 2012) y la técnica de rechazo para el tratamiento de las restricciones. Sin embargo, de acuerdo a un conjunto de pruebas realizadas teniendo en cuenta un tamaño de problema de 500 trabajadores y cuatro roles a cubrir, se comprobó que la técnica de rechazo para el tratamiento de restricciones presenta inconvenientes ya que, más del 60% de las soluciones generadas son no factibles (Escalera et al. 2015).

Existen otras técnicas para el tratamiento de las restricciones como son la penalización (acepta una solución no factible penalizando la función objetivo), la reparación (aplica un operador de reparación a cada solución no factible que se genere con el objeto de transformarla en otra factible) y la preservación (obtiene una representación del problema o un operador que siempre garantice obtener soluciones factibles) (Talbi 1965).

Existen numerosos trabajos donde se utilizan técnicas para el tratamiento de restricciones como: penalización, reparación y preservación (Centeno et al. 2012; Coello and Cortés 2004; Coello 2003; Gupta et al. 2009a; Jadaan et al. 2009a; Lukasiewicz et al. 2008a; Lukasiewicz et al. 2008b; Mezura and Coello 2003; Rasjido et al. 2011; Valdez et al. 2011; Villagra et al. 2010a; Villagra et al. 2010b; Vose 1990). El estudio de las fuentes analizadas permite concluir que restringir el área de búsqueda a la región factible con la técnica de preservación, es una buena manera de tratar problemas con restricciones, pero esta técnica resulta difícil de implementar ya que no es sencillo, para un problema específico encontrar una representación del problema u operadores que garanticen la factibilidad de la solución (Michalewicz and Koziel 1998).

Teniendo en cuenta lo planteado anteriormente este trabajo tiene como objetivo implementar nuevas técnicas para el tratamiento de restricciones al problema de formación de equipos de proyectos de software para disminuir el tiempo de respuesta de los algoritmos.

2. Materiales y métodos

2.1 Modelo de formación de equipos de proyectos de software

El problema de formación de equipos de proyecto de software plantea la asignación de n personas a m roles, donde una persona puede desempeñar más de un rol y un rol puede ser desempeñado por más de una persona. El modelo propuesto en (André, Baldoquín and Acuña 2011) plantea:

- Maximizar las competencias de los trabajadores en el rol o los roles asignados.
$$\max \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}, \text{ siendo } x_{ij} = 1 \text{ si el empleado } i \text{ es asignado al rol } j \text{ y } 0 \text{ en caso contrario, y } c_{ij} \text{ la competencia neta del empleado } i \text{ para desempeñar el rol } j.$$
- Minimizar las incompatibilidades entre los miembros de un equipo de proyecto.
$$\min \sum_{h=1}^{n-1} \sum_{i>h}^n s_{hi} u_h u_i, \text{ siendo } s_{hi} \text{ la incompatibilidad existente entre los trabajadores } h \text{ e } i \text{ y } u_i = 1 \text{ si el empleado } i \text{ es asignado al menos a un rol y } 0 \text{ en caso contrario.}$$
- Balancear la carga del personal del equipo.

$\min \sum_{i=1}^n [(L_i + \sum_{j=1}^m b_j x_{ij}) - ME]^2$, donde $L_i = \sum_{j=1}^m g_{ij}$ y $ME = \frac{\sum_{j=1}^m [(\sum_{i=1}^n g_{ij}) + b_j]}{n}$, g_{ij} es la carga de trabajo del empleado i en el rol j según los proyectos a los que está asignado y b_j es la carga de trabajo que implica asumir el rol j en un proyecto determinado.

- Minimizar el costo de trabajar a distancia. Este es un objetivo que no está incluido en el modelo por defecto, sino en una versión ampliada ya que solo es aplicable a organizaciones que enfrentan esta variante de desarrollo.

$\min \sum_{i=1}^n \sum_{j=1}^m l_{ij} x_{ij}$, donde l_{ij} es el costo del empleado i según la lejanía que tenga del proyecto y el rol j que va a desempeñar.

y doce tipos de restricciones, seis de las cuales plantean que:

- Los roles deben ser cubiertos en función de la cantidad necesaria de personas a desempeñarlo.
- Una persona no puede desempeñar al mismo tiempo roles que se consideren incompatibles entre sí.
- Restringir el número máximo de roles que puede desempeñar cualquier trabajador en el proyecto a una cantidad fijada por el usuario.
- Para que una persona desempeñe un rol debe cumplir los requisitos mínimos de nivel de competencia para desempeñar dicho rol.
- La carga de trabajo total asignada a un empleado no debe ser mayor que un valor máximo.
- Garantizar que la variable u_i tome valor 1 ó 0.

Otro conjunto de restricciones refleja la relación que existe entre los roles de Belbin¹, los tipos psicológicos de Myers Briggs² y los roles a desempeñar en un equipo de proyecto de software (André, Baldoquín and Acuña 2011):

- Un conjunto de restricciones garantiza que en el equipo de desarrollo se representen las tres categorías de roles propuestas por Belbin (roles de acción, roles mentales y roles sociales).
- En el equipo de trabajo la preferencia de desempeñar roles de acción debe sobrepasar la preferencia por desempeñar los roles mentales.
- En el equipo de trabajo la preferencia de desempeñar roles mentales debe sobrepasar la preferencia por desempeñar los roles sociales.

¹Meredith Belbin es el creador del test que identifica la preferencia de las personas por desempeñar algunos de los nueve roles que define están presentes en un equipo. La metodología establece que en un equipo debe haber presencia de las tres categorías de roles (mentales, sociales y de acción) donde deben predominar los roles de acción y no debe existir una alta presencia de roles sociales ni mentales.

²Test que mide cuatro dimensiones diferentes de las preferencias humanas: Extroversión (E)-Introversión (I), Intuición (N)-Sentidos (S), Emoción (F)-Pensamiento (T), y Juicio (J)-Percepción (P). A partir de los valores de cada dimensión se identifica el tipo psicológico de la persona entre los 16 tipos posibles.

- La persona que desarrolla el rol de Jefe de Proyecto debe tener como preferido los roles de Belbin: Impulsor o Coordinador.
- En el equipo al menos una persona debe tener como preferido el rol mental Cerebro.
- La persona que desarrolla el rol Jefe de Proyecto debe ser extrovertida y planificada (subtipo E_ _J) según el test de Myers-Briggs.

En los problemas de optimización que incluyen restricciones, el espacio de soluciones se divide en dos subconjuntos (el espacio de soluciones factibles y el espacio de soluciones no factibles) (Rasjido, Pandolfi and Villagra 2011). Los algoritmos de búsqueda generan soluciones en ambos espacios. Por lo tanto, el problema consiste en cómo tratar las soluciones no factibles generadas. Este problema tiene la peculiaridad de que debe satisfacer tanto restricciones individuales como restricciones de equipo, estas últimas solo se pueden chequear una vez conformado el equipo, por lo que son más difíciles de garantizar en el momento de conformar la solución.

2.2 Representación del problema de formación de equipos de proyectos de software

Para el problema planteado, la representación de una solución se muestra en la **¡Error! No se encuentra el origen de la referencia.1**. Dado que el problema consiste en asignar trabajadores a los roles definidos en un equipo de proyecto y que se pueden asignar varios trabajadores a un mismo rol, se define como solución, una lista de elementos, donde cada elemento está compuesto por un rol y la lista de trabajadores que juegan ese rol.

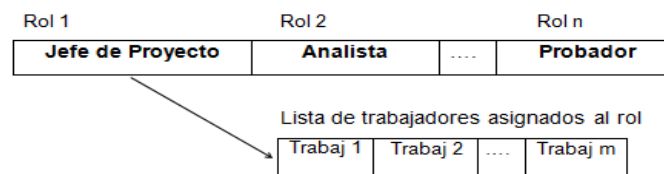


Figura 1: Representación de una solución.

El operador definido es el de sustitución (Infante 2012), el cual escoge uno de los roles de la solución y sustituye uno de los trabajadores que juegan ese rol por otro del espacio de soluciones, de forma aleatoria. Esto trae como consecuencia que en muchas ocasiones los trabajadores seleccionados no cumplan con los requisitos mínimos de competencias para desempeñar el rol. Por lo que el operador utilizado no garantiza la factibilidad de la solución.

2.3 Técnicas para el tratamiento de restricciones en problemas de optimización

Muchos de los problemas de optimización están sujetos a restricciones, por lo que existen técnicas para su manejo que actúan principalmente en la representación de la solución o de la función objetivo. Estas técnicas se pueden clasificar como: rechazo, penalización, reparación, preservación y decodificación (Talbi 1965). En la Tabla 1 se muestran las

técnicas con sus variantes y una breve descripción de estas. En la Tabla 2 se pueden observar las referencias a los 35 trabajos más relevantes de los analizados, relacionados con el empleo de técnicas para el tratamiento de restricciones en la solución de problemas de optimización. Este análisis permite identificar que las técnicas más utilizadas son: penalización, reparación y preservación.

Tabla 1: Técnicas para el tratamiento de restricciones en problemas de optimización

Técnica		Descripción
Rechazo		Acepta solo soluciones factibles durante la búsqueda, descartando las soluciones no factibles que son generadas.
Reparación	Enfoque Lamarkiano	Modifica genéticamente una solución no factible y la transforma en una solución factible (Rasjido, Pandolfi and Villagra 2011).
	Enfoque Baldwiniano	Permite combinar el aprendizaje y la evolución. La solución no factible solo es reparada para su evaluación, pero en la población, la solución no es modificada (permanece no factible) (Whitley et al. 1994).
	Enfoque Annealing	Está basado en el concepto de Recocido Simulado (Kirkpatrick et al. 1983). Las soluciones no factibles son aceptadas con cierta probabilidad. En las primeras generaciones las soluciones no factibles tienen una alta probabilidad de ser aceptadas para formar parte o contribuir con la siguiente generación, pero esta probabilidad va disminuyendo conforme pasan las generaciones. Cuando una solución no factible no es aceptada, entonces se procede a reparar dicha solución.
Penalización	Penalidad estática	Aplica una penalidad constante a aquellas soluciones que violan la factibilidad (Rasjido, Pandolfi and Villagra 2011).
	Penalidad Dinámica	Incorpora un aspecto dinámico que incrementa la rigurosidad de las penalidades para una distancia dada a medida que avanza la búsqueda. Permite, con alta probabilidad, soluciones no factibles muy tempranamente en la búsqueda, pero continuamente incrementa la penalidad impuesta hasta que se mueva la solución final hacia la región factible.
	Penalidad Adaptativa	Permiten un incremento o un decremento de la penalidad impuesta durante la evolución, por lo tanto, la función estará condicionada por un multiplicador que se actualizará adaptativamente. Por ejemplo, a medida que la búsqueda avanza, el multiplicador de la función de penalidad es actualizado cada cierta cantidad de generaciones, basado en si la mejor solución fue o no factible durante ese intervalo (Rasjido, Pandolfi and Villagra 2011).
	Restricciones violadas	Cuenta el número de restricciones que se incumplen y agrega a la función objetivo una función de penalización.
Preservación		Obtiene una representación específica y operadores que garanticen la generación de soluciones factibles. Esta técnica es adaptada a problemas específicos y por lo tanto, no puede ser generalizada al tratamiento de restricciones de todo problema de optimización (Talbi 1965).
Decodificación		Utiliza codificaciones indirectas. La función de decodificación debe tener las siguientes propiedades (Talbi 1965): A cada solución $r \in R$, le corresponde una solución factible $s \in S$; para cada solución factible $s \in S$, existe una representación $r \in R$ que le corresponde; la complejidad computacional del decodificador debe ser reducida; las soluciones factibles en S deben tener el mismo número de correspondientes soluciones en R ; la representación del espacio debe tener propiedades locales en el sentido de que la distancia entre soluciones en R debe ser positivamente correlacionada con la distancia entre soluciones factibles en S .

Tabla 2: Aplicación de técnicas para el tratamiento de restricciones

Técnica de tratamiento de restricciones	Trabajos
Penalización	
No se menciona el tipo utilizado	(Michalewicz 1995), (Gupta et al. 2009b), (Villagra, Lasso, Pandolfi, Pedro, Villagra, Rasjido, Seron, Orozco, Montenegro and Cossio 2010b), (Valdez, Pandolfi and Villagra 2011)
Estática	(Michalewicz and Schoenauer 1996), (Coello and Cruz 2004; Coello and Cortés 2004), (Aliay et al. 2014), (Villagra, Lasso, Pandolfi, Pedro, Villagra, Rasjido, Serón and Orozco 2010a)
Adaptativa	(Jadaan et al. 2009b), (Villagra, Lasso, Pandolfi, Pedro, Villagra, Rasjido, Serón and Orozco 2010a)
Dinámica	(Rasjido, Pandolfi and Villagra 2011), (Aliay, Golalikhanib and Zhuangby 2014), (Villagra, Lasso, Pandolfi, Pedro, Villagra, Rasjido, Serón and Orozco 2010a)
Reparación	
Baldwiniano	(Liepins and Vose 1990), (Ishibuchi et al. 2005), (Rasjido, Pandolfi and Villagra 2011)
Lamarkiano	(Nakano 1991), (Michalewicz and Nazhiyath 1995), (Zitzler and Thiele 1999), (Ishibuchi, Kaige and Narukawa 2005), (Rasjido, Pandolfi and Villagra 2011)
Annealing	(Orvosh and Davis 1994), (Michalewicz et al. 1995), (Michalewicz et al. 1996), (Rasjido, Pandolfi and Villagra 2011)
Preservación	(Michalewicz and Janikow 1991), (Michalewicz and Schoenauer 1996), (Kowalczyk 1997), (Michalewicz and Koziel 1998), (Coello 2002), (Hu and Eberhart 2002), (Paquet and Engelbrecht 2003), (Lukasiewicz, Glaß and Teich 2008b), (Lukasiewicz, Glaß, Haubelt and Teich 2008a), (Michalewicz 2013), (Padhye et al. 2015).

2.4 Análisis del cumplimiento de las restricciones del problema de formación de equipos de proyectos de software

Con el propósito de analizar el problema, determinar el por ciento de soluciones no factibles obtenidas y saber cuáles son las restricciones que más se violan utilizando la técnica de rechazo, se realiza un estudio del problema teniendo en cuenta tres escenarios. Los datos utilizados para realizar las pruebas fueron generados a partir de encuestas realizadas a un grupo de especialistas de software, debido a que no se dispone de datos reales y tampoco se conoce de bases de datos de pruebas a nivel internacional para este tipo de problema. Los datos necesarios para aplicar el modelo y que fueron recogidos en la encuesta incluyó: Niveles de competencia de los trabajadores, experiencia de los trabajadores en el desempeño de roles, carga de trabajo, incompatibilidades entre los trabajadores, características psicológicas a través de los test de Myers-Briggs y Belbin.

En pruebas realizadas para un tamaño de problema de 500 trabajadores y cuatro roles a cubrir se obtuvo que el 60% de las soluciones generadas son no factibles. Con el propósito de identificar cómo se comporta el por ciento de soluciones no factibles en otros escenarios y cuáles son las restricciones que más se incumplen, se realizaron otro conjunto de pruebas en tres escenarios, tal como se muestra en la Tabla 3. Para los tres escenarios diseñados se tuvieron en cuenta

los algoritmos: Escalador de Colinas (EC), Recocido Simulado (RS), Escalador de Colinas Estocástico Multiobjetivo (ECEMO) y Recocido Simulado Multiobjetivo Multicaso (RSMCMC). Todos los algoritmos toman como condición de parada un número máximo de evaluaciones de las funciones objetivo, definida como 100000 y se realizaron 20 ejecuciones de cada algoritmo.

En la **Error! No se encuentra el origen de la referencia.** 3 se muestra para cada uno de los escenarios diseñados el por ciento de soluciones no factibles que como promedio se obtienen por cada algoritmo. Para los escenarios diseñados aproximadamente el 60% de las soluciones generadas con la técnica de rechazo son no factibles. Esto provoca que se pierda un tiempo considerable en la generación de soluciones no factibles que luego serán rechazadas.

Tabla 3: Por ciento de soluciones no factibles utilizando la técnica de rechazo por algoritmo

Algoritmo	% de soluciones no factibles		
	Escenario 1	Escenario 2	Escenario 3
	(250 trabajadores y 4 roles a	(500 trabajadores y 6 roles a cubrir)	(1500 trabajadores y 6 roles a
EC	53	45.8	70.6
RS	54	47.2	63
ECEMO	47.6	50.6	70
RSMCMC	54	47	63.2
Promedio = 60.4 % Máximo= 70.6 % Mínimo= 45.8 %			

Con el objetivo de identificar cuáles son las restricciones que más se incumplen, se realizaron varios experimentos teniendo en cuenta los escenarios diseñados anteriormente y todas las restricciones que propone el modelo. Las tres restricciones que se incumplen son: roles de Belbin (71% de restricciones violadas), niveles de competencias (28% de restricciones violadas) y cantidad máxima de roles (1% de restricciones violadas).

2.5 Selección de técnica para el tratamiento de restricciones

Teniendo en cuenta las técnicas más usadas de acuerdo a la bibliografía consultada, las particularidades del problema y la premisa de evaluar el desempeño de técnicas más sencillas antes de otras más complejas se decide implementar las siguientes técnicas:

- Técnica de penalización: la penalización estática propuesta en (Homaifar et al. 1994; Michalewicz, Dasgupta, Richie and Schoenauer 1995), donde para cada restricción se asume un nivel de violación y para cada nivel un coeficiente de penalidad.
- Técnica de preservación: según el criterio de los autores estudiados al implementarse esta técnica se logra encontrar una estructura que permite preservar la factibilidad de las soluciones. Para implementar esta técnica se modificará el operador de sustitución implementado con el objetivo de restringir el área de búsqueda a soluciones factibles.

3 Resultados y discusión

3.1 Técnica de penalización

Se implementa la técnica de penalización para las soluciones no factibles a través de un peso asociado a cada restricción, dándole mayor peso a las restricciones más violadas. La función objetivo penalizada es: $F_p(x) = f(x) \pm \sum_{i=1}^m C_i \delta_i$, donde $F_p(x)$ es la función objetivo penalizada, $f(x)$ es la función objetivo sin penalizar, δ_i toma valor 1 si la restricción i es violada y 0 en caso contrario y C_i es igual a la suma de los pesos asociados a cada restricción violada. Para determinar el valor de los pesos se tuvo en cuenta el por ciento de violación para cada restricción utilizando la técnica de rechazo (roles de Belbin 71%, niveles de competencias 28%, cantidad máxima de roles 1%).

3.2 Técnica de preservación

Se implementa la técnica de preservación con el objetivo de organizar el espacio de búsqueda de forma tal que se garanticen las restricciones asociadas a los niveles mínimos de competencias de los trabajadores y las relacionadas con los roles de Belbin. Para ello se procede a acotar el espacio de búsqueda utilizando una multilista auxiliar, la cual contiene por cada rol necesario para conformar el equipo de proyecto, las siete combinaciones posibles de roles de Belbin. Cada combinación de rol de Belbin es definido por un código binario de tres dígitos correspondientes a rol de acción, mental y social respectivamente, donde el valor 1 indica que el trabajador posee dicho rol y 0 en caso contrario. Por cada combinación de rol de Belbin se tiene la lista de los trabajadores que las poseen. De esta manera una vez que se tenga el trabajador a cambiar, en vez de seleccionar uno aleatoriamente del área de búsqueda inicial que contiene todos los trabajadores, se toma de la lista de trabajadores que poseen las competencias para desempeñar este rol en el equipo y que cumpla con las restricciones asociadas a los roles de Belbin, asegurando así una solución factible. En la

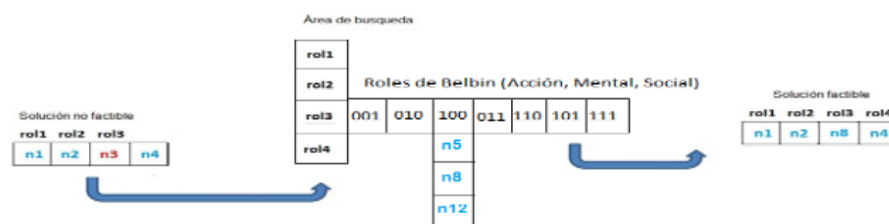


Figura 2: Flujo de la técnica de reparación propuesta

3.3 Análisis experimental

Para analizar el comportamiento de las técnicas propuestas se diseñaron cuatro escenarios de pruebas, teniendo en cuenta escenarios de organizaciones medianas, grandes y especialmente grandes: 250 trabajadores y 4 roles a cubrir, 500 trabajadores y 4 roles a cubrir, 500 trabajadores y 6 roles a cubrir, 1500 trabajadores y 6 roles a cubrir.

Los casos de prueba a realizar tienen en cuenta las tres funciones objetivo del modelo: Maximizar competencias, minimizar incompatibilidades y balancear carga de trabajo. Se tomaron en cuenta las restricciones de que un trabajador solo puede desempeñar un rol y las competencias mínimas establecidas para cada rol. Además, se tomaron en cuenta las restricciones asociadas con la sinergia del equipo: considerar la presencia de todas las categorías de roles de Belbin, elegir la presencia de al menos una persona con el rol cerebro y que exista un balance entre las categorías de roles de Belbin (la presencia de miembros del equipo con preferencia por los roles de acción deben ser mayor que los que tienen preferencia por los roles mentales y la preferencia por los roles mentales deben ser mayor que por los roles sociales).

Las condiciones relacionadas con el ambiente experimental fueron: la implementación de los algoritmos en Java bajo el ambiente de desarrollo Eclipse 3.2, compilados con el JDK 1.7.0 y las corridas fueron ejecutadas en un Intel Core 2 Duo con 2.66 GHz y 4 Gb de RAM. Se ejecutaron 20 corridas de cada uno de los algoritmos implementados para cada escenario de prueba.

Para evaluar el desempeño de las dos técnicas de tratamiento de restricciones implementadas se seleccionaron dos de los métodos de solución de problemas multiobjetivo: factores ponderados y multiobjetivo puro. Se emplean los algoritmos Escalador de Colinas Estocástico Multiobjetivo (ECEMO), Escalador de Colinas Estocástico Multiobjetivo con Reinicio (ECEMOR), Escalador de Colinas Estocástico Multiobjetivo de mayor Distancia (ECEMODist) y Recocido Simulado Multiobjetivo Multicaso (RSMMC) con el método multiobjetivo puro, y los algoritmos Escalador de Colinas (EC) y Recocido Simulado (RS) con el método factores ponderados, debido a que son los que mejor comportamiento tuvieron en la solución del problema en los experimentos realizados en (Infante 2012).

Los parámetros utilizados para la ejecución de los algoritmos se muestran en la **¡Error! No se encuentra el origen de la referencia.5**. Los valores establecidos para los parámetros son los mismos en todos los escenarios.

Tabla 4: Parámetros de los algoritmos

Algoritmos	Parámetros
RS	Temperatura inicial: 2 Temperatura final: 0 Esquema de Cauchy: $T_k = T_0 / (1 + k)$ donde $k = \#$ iteración y T_0 es la temperatura inicial Cantidad de iteraciones: 100000
EC, ECEMO, ECEMOR, ECEMODist, RSMMC	Cantidad de iteraciones: 100000

3.3.1 Medidas de rendimiento de los algoritmos

Todos los métodos de solución utilizados tienen en cuenta la medida Tiempo. Para comparar el rendimiento de los algoritmos metaheurísticos implementados con el método de solución de factores ponderados se tiene en cuenta la medida calidad de la solución, expresada en función de la evaluación ponderada de todos los objetivos (el valor ideal de la medida es que la evaluación sea 1). Para realizar las pruebas se utilizan como pesos en los objetivos: Maximizar competencias (0.36), Balancear carga de trabajo (0.31) y Minimizar incompatibilidades (0.33), teniendo en cuenta las encuestas aplicadas a expertos en (André, Baldoquín and Acuña 2011).

Para evaluar el rendimiento del método de solución multiobjetivo puro se utilizan las métricas:

- Tasa de error (Veldhuizen and Lamont 2000): Indica el porcentaje de soluciones que no son miembros del frente de Pareto verdadero. El valor ideal de la métrica es 0.
- Distancia generacional (Veldhuizen and Lamont 2000): Indica qué tan lejos están los elementos del frente de Pareto actual respecto al frente de Pareto verdadero. El valor ideal es 0.
- Dispersión (Schott 1995): Mide la varianza de la distancia de cada miembro del conjunto de óptimos de Pareto con respecto a su vecino más cercano. Su valor ideal es 0.

Se tomó como “frente de Pareto verdadero” aquellas soluciones no dominadas de las obtenidas por todos los algoritmos ejecutados en cada escenario, respectivamente; debido a que no se conoce y no es factible calcular el frente de Pareto verdadero real.

Para analizar los resultados de las medidas antes mencionadas se utiliza el test no paramétrico prueba de Friedman (Derrac et al. 2011) y los post-hoc de Holm y Shaffer, con el objetivo de decidir cuál algoritmo y/o técnica es superior.

3.3.2 Análisis de los resultados obtenidos por técnica utilizando las medidas

En la Tabla 6 se muestran los resultados de comparar todas las técnicas entre sí (N técnicas vs N técnicas), utilizando el método de factores ponderados. Se muestran los *rankings* de Friedman para cada técnica y medida. En la última fila de la tabla aparece el valor del p-value. La técnica de mejor *ranking* se encuentra sombreada y marcada en negrita y cursiva.

Tabla 5: Resultados del test de Friedman para las medidas calidad de la solución y tiempo en algoritmos de trayectoria

Algoritmo/Ranking	Calidad de la solución	Tiempo
Preservación	1,9375	1,125
Penalización	2,125	1,875
Rechazo	1,9375	3
p-value	0,91	0,0008

De acuerdo al *ranking* en cuanto a la medida calidad de la solución, las técnicas de preservación y rechazo superan a la técnica de penalización. Aunque teniendo en cuenta el valor de *p-value* se puede decir que no existen diferencias significativas entre los resultados. En cuanto a la medida tiempo, la técnica que supera al resto es preservación, aunque la técnica de penalización ofrece mejores resultados que la de rechazo. Por otra parte, el valor de *p-value* muestra que existen diferencias significativas entre los resultados obtenidos ya que el *p-value* es menor que 0,05, por lo que se realiza un análisis *post-hoc* para caracterizar estas diferencias, tal como se muestra en la Tabla 7.

Tabla 6: Resultados del Test de Friedman para múltiples comparaciones teniendo en cuenta la medida tiempo por técnica

#	Hipótesis	Holm	Shaffer
1	Preservación vs Rechazo	0.000531	0.000531
2	Penalización vs Rechazo	0.048898	0.024449
3	Preservación vs Penalización	0.133614	0.133614

Teniendo en cuenta los resultados de la Tabla 7 y utilizando un nivel de significancia (α) = 0.05, los dos métodos seleccionados aceptan dos de las tres hipótesis. Estas hipótesis se muestran sombreadas en gris y marcadas en negrita. Las técnicas de preservación y penalización superan significativamente a la técnica de rechazo.

En la Tabla 8 se muestran los *rankings* de Friedman por cada técnica para cada medida, utilizando el método multiobjetivo puro. En la última fila de la tabla aparece el valor del *p-value*. La técnica de mejor *ranking* se encuentra sombreada de color gris y marcada en negrita.

Tabla 7: Resultados del test de Friedman por técnica para las diferentes medidas en los algoritmos multiobjetivo

Algoritmo/Ranking	Tasa de error	Distancia Generacional	Dispersión	Tiempo
Preservación	2.3125	2.125	2.3125	1.3125
Penalización	1.4375	1.5	1.625	2.0625
Rechazo	2.25	2.375	2.625	2.625
p-value	0.022	0.038	0.144	0.0009

De acuerdo al *ranking*, la técnica que supera al resto por ser la primera, en cuanto a las medidas de calidad, es penalización. En cuanto a la medida tiempo la mejor técnica es preservación. En cuanto a las medidas tasa de error, distancia generacional y tiempo el *p-value* es menor que 0.05 por lo que se puede decir que existen diferencias significativas entre los resultados y se procede a utilizar los *post-hoc* para caracterizar estas diferencias, tal como se muestran en las tablas 9, 10 y 11.

Tabla 8 Resultados del Test de Friedman para múltiples comparaciones teniendo en cuenta la medida tasa de error

#	Hipótesis	Holm	Shaffer
1	Preservación vs Penalización	0.039985	0.039985
2	Penalización vs Rechazo	0.043113	0.039985
3	Preservación vs Rechazo	0.859684	0.859684

En cuanto a la medida tasa de error, se aceptan dos de las tres hipótesis, las que se muestran sombreadas en gris y marcadas en negrita en la Tabla 9. No se detectan diferencias significativas entre las técnicas de preservación y rechazo, y la técnica de penalización supera significativamente a las técnicas de preservación y rechazo.

Tabla 9 Resultados del Test de Friedman para múltiples comparaciones teniendo en cuenta la medida distancia generacional

#	Hipótesis	Holm	Shaffer
1	Penalización vs Rechazo	0.039985	0.039985
2	Preservación vs Penalización	0.1542	0.0771
3	Preservación vs Rechazo	0.4795	0.4795

Tabla 10 Resultados del Test de Friedman por técnica para múltiples comparaciones teniendo en cuenta la medida tiempo

#	Hipótesis	Holm	Shaffer
1	Preservación vs Rechazo	0.000616	0.000616
2	Preservación vs Penalización	0.03779	0.033895
3	Penalización vs Rechazo	0.111612	0.111612

En cuanto a la medida distancia generacional, se acepta una de las tres hipótesis, las que se muestran sombreadas en gris y marcadas en negrita en la Tabla 10. La técnica de penalización supera significativamente a la técnica de rechazo. En cuanto a la medida tiempo, se aceptan dos de las tres hipótesis, las que se muestran sombreadas en gris y marcadas en negrita en la Tabla 11. La técnica de preservación supera significativamente a las técnicas de rechazo y penalización. Una vez analizados todos los resultados obtenidos para cada una de las técnicas, en la **¡Error! No se encuentra el origen de la referencia.2** se muestra un resumen de las técnicas que por medida quedaron primero en el *ranking*, así como las diferencias significativas encontradas con los *post-hoc*.

Tabla 11 Resumen de los resultados obtenidos por técnica

Tipo	Medida	Primero del <i>ranking</i>	Diferencias significativas
Trayectoria	Evaluación	Preservación y Rechazo	No existen diferencias
	Tiempo	Preservación	Preservación sobre Rechazo Penalización sobre Rechazo
Multiobjetivo	Tasa de error	Penalización	Penalización sobre Rechazo y Preservación.
	Distancia generacional	Penalización	Penalización sobre Rechazo
	Dispersión	Penalización	No existen diferencias

	Tiempo	Preservación	Preservación sobre Rechazo y Penalización
--	--------	--------------	---

Conclusiones

Una vez culminado el trabajo se concluye que:

- De acuerdo a la bibliografía consultada las técnicas más utilizadas para el tratamiento de las restricciones en problemas de optimización son penalización, reparación y preservación.
- En el problema de formación de equipos de proyectos de software se obtiene un por ciento significativo (60%) de soluciones no factibles generadas utilizando la técnica de rechazo y se identifican las restricciones asociadas a los niveles mínimos de competencias, la cantidad máxima de roles y el balance de los roles de Belbin como las más violadas.
- Basado en los estudios realizados para determinar de las restricciones que propone el modelo de formación de equipos las que más se incumplen y las diferentes técnicas para el tratamiento de restricciones se incorporaron a la solución del modelo las técnicas de penalización y preservación.
- De acuerdo a los resultados obtenidos aplicando el test de Friedman para obtener el comportamiento de las técnicas implementadas, se pudo comprobar que:
 - ✓ La técnica de preservación es la que mejor se comporta en cuanto a la medida tiempo utilizando ambos métodos de solución.
 - ✓ La técnica de preservación es la que mejor se comporta utilizando el método factores ponderados en cuanto a la medida calidad de la solución.
 - ✓ La técnica penalización es la que mejor se comporta utilizando el método multiobjetivo puro en cuanto a las medidas de calidad de la solución (tasa de error, distancia generacional y dispersión).
 - ✓ A pesar de que al aplicar las pruebas de Friedman teniendo en cuenta la medida tiempo no se logra una diferencia significativa entre las técnicas implementadas para todos los algoritmos, si se logra una disminución visible de los tiempos con las técnicas propuestas.

Para futuros trabajos se recomienda:

- Aplicar otras técnicas para el tratamiento de restricciones, como la técnica de reparación.
- Aplicar diferentes variantes de la técnica de penalización como la adaptativa y la dinámica en el problema de formación de equipos de proyecto de software.
- Probar las nuevas técnicas implementadas utilizando algoritmos poblacionales.

- Recopilar datos reales para la aplicación del modelo y realizar pruebas conformando equipos más cercanos a la realidad cubana como son: un trabajador con varios roles y un rol asignado a varios trabajadores.

Referencias

- ALIAY, M. M., M. GOLALIKHANIB AND J. ZHUANGBY A computational study on different penalty approaches for solving constrained global optimization problems with the electromagnetism-like method. *Optimization*, 2014, 63(3), 403-419.
- ANDRÉ, M., M. G. BALDOQUÍN AND S. T. ACUÑA Formal model for assigning human resources to teams in software projects. *Information and Software Technology*, 2011, 53(3), 16.
- ÁNGELES, M. D. L., M. GÓMEZ AND S. T. ACUÑA Procedimiento para Determinar las Necesidades de Competencias en Organizaciones Desarrolladoras de Software. *Revista Colombiana de Computación*, 2005, 7(2), 28.
- BARRETO, A. S. Apoio à Decisão Gerencial na Alocação de Recursos Humanos em Projetos de Software. Universidade Federal do Rio de Janeiro, 2003.
- CENTENO, A., D. REY AND A. MARTÍNEZ Algoritmo de optimización metaheurístico aplicado al cálculo del flujo de carga óptimo en redes eléctricas de potencia 2012.
- COELLO, C. A. Theoretical and numerical constraint-handling techniques used with evolutionary algorithm: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 2002, 191(11), 1245-1287.
- COELLO, C. A. AND N. CRUZ Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 2004, 36(5), 607-634.
- COELLO, C. A. C. AND N. C. CORTÉS Hybridizing a genetic algorithm with an artificial immune system for global optimization. *Engineering Optimization*, 2004, 36(5), 607-634.
- COELLO, E. M.-M. A. C. A. 2003. A simple multimembered evolution strategy to solve constrained optimization problems. . In *Proceedings of the Evolutionary Computation Group (EVOCINV) CINVESTAV-IPN Computer Science Section Electrical Engineering Department Mexico, D.F., 07300, MEXICO*.2003.
- DECARVALHO, L. R. Planejamento da alocação de recursos humanos em Ambientes de desenvolvimento de software orientados à Organização. Tese para a obtenção do grau de mestre em ciências em engenharia de sistemas e computação Universidade Federal do Rio de Janeiro, 2003.
- DERRAC, J., S. GARCÍA, D. MOLINA AND F. HERRERA A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 2011, 1, 3-18.
- ESCALERA, K., A. L. INFANTE, M. ANDRÉ AND A. ROSETE Técnicas para el tratamiento de restricciones en el problema de conformación de equipos de proyectos de software. *Revista Cubana de Ciencias Informáticas*, 2015, 9(No. Especial CCIA 17), 110-125.
- GUPTA, R. K., A. K. BHUNIA AND D. ROY A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliability of components. *Journal of Computational and Applied Mathematics*, 2009a, 232, 275-284.
- GUPTA, R. K., A. K. BHUNIA AND D. ROY A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliability of components. *Journal of Computational and Applied Mathematics*, 2009b, 232(2), 275-284.
- HOMAI FAR, A., C. X. QI AND S. H. LAI Constrained optimization via genetic algorithms. *Simulation*, 1994, 62(4), 242-253.

- HU, X. AND R. EBERHART. Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. In *Proceeding of the sixth world multiconference of systemics, cybernetics and informatics*. 2002, vol. 5, p. 203-206.
- INFANTE, A. Algoritmos de trayectoria multiobjetivo aplicados al problema de asignación de recursos humanos a equipos de proyecto de software. Tesis de Maestría Instituto Superior Politécnico José Antonio Echeverría, 2012.
- ISHIBUCHI, H., S. KAIGE AND K. NARUKAWA. Comparison between lamarckian and baldwinian repair on multiobjective 0/1 Knapsack Problems. In *Evolutionary Multi-Criterion Optimization*. 2005, p. 370-385.
- JADAAN, O. S., L. RAJAMANI AND C. R. RAO. Adaptive penalty function for solving constrained evolutionary optimization. *Journal of Theoretical and Applied Information Technology*., 2009a.
- JADAAN, O. S., L. RAJAMANI AND C. R. RAO. Adaptive penalty function for solving constrained evolutionary optimization. *Journal of Theoretical and Applied Information Technology*., 2009b, 5(3).
- KIRKPATRICK, S., C. D. GELATT AND M. P. VECCHI Optimization by Simulated Annealing. *Science*, 13 mayo 1983 1983, 220, 10.
- KOWALCZYK, R. Constrained genetic operators preserving feasibility of solutions in genetic algorithms In *Genetic Algorithms in Engineering Systems: Innovations and Applications*. 1997, p. 191 - 196
- LIEPINS, G. E. AND M. D. VOSE Representational Issues in Genetic Optimization. *Journal of Experimental and Theoretical Computer Science*, 1990, 2(2), 101-115.
- LUKASIEWYCZ, M., M. GLAß, C. HAUBELT AND J. TEICH. A Feasibility-Preserving Local Search Operator for Constrained Discrete Optimization Problems. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*. 2008a, p. 1968-1975.
- LUKASIEWYCZ, M., M. GLAß AND J. TEICH. A Feasibility-preserving Crossover and Mutation Operator for Constrained Combinatorial Problems. In *Proceedings of Conference on Parallel Problem Solving from Nature*. 2008b, p. 919-928.
- MEZURA, E. AND C. A. COELLO. Use of Multiobjective Optimization Concepts to Handle Constraints in Genetic Algorithms. In L.J.A.R.G. AJITH ABRAHAM ed. *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*. London, , 2003, p. 229--254.
- MICHALEWICZ, Z. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. *Evolutionary Programming*, 1995, 4, 135-155.
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. Edition ed., 2013.
- MICHALEWICZ, Z., D. DASGUPTA, R. L. RICHIE AND M. SCHOENAUER. A co-evolutionary algorithm for numerical optimization with nonlinear constraints. In D.B. FOGEL. *Proceeding of the Second IEEE International Conference on Evolutionary Computation*. Piscataway, New Jersey. IEEE Press., 1995, p. 647-651.
- MICHALEWICZ, Z., D. DASGUPTA, R. L. RICHIE AND M. SCHOENAUER Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering Journal*, 1996, 30(4), 851-870.
- MICHALEWICZ, Z. AND C. Z. JANIKOW *Handling Constraints in Genetic Algorithms*. Edition ed., 1991.
- MICHALEWICZ, Z. AND S. KOZIEL A decoder-based evolutionary algorithm for constrained parameter optimization problems. *Parallel Problem Solving from Nature-PPSN V*, 1998.
- MICHALEWICZ, Z. AND G. NAZHIYATH. Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *IEEE International Conference*. 1995, vol. 2, p. 647-651.
- MICHALEWICZ, Z. AND M. SCHOENAUER Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 1996, 4(1), 1-32.
- NAKANO, R. Conventional Genetic Algorithm for Job Shop Problems. . In R.K.B.A.L.B. BOOKER. *Proceedings of the Fourth International Conference on Genetic Algorithms*. San Mateo, California. : Morgan Kaufmann Publishers., 1991, p. 474-479.

- NEIL, D. Proyectos Informáticos: Fracasos y Lecciones Aprendidas. Revista de Derecho y Tecnologías de la Información, junio, 2006 2006, (4), 11.
- NGO-THE, A. AND G. RUHE A Systematic Approach for Solving the Wicked Problem of Software Release Planning. Soft Computing, 2008, 12(1), 95-108.
- ORVOSH, D. AND L. DAVIS. Using a genetic algorithm to optimize problems with feasibility constraints. In *Proceedings of the first IEEE Conference.*, 1994, vol. 2, p. 548-553.
- PADHYE, N., P. MITTAL AND K. DEB Feasibility Preserving Constraint-Handling Strategies for Real Parameter Evolutionary Optimization 2015.
- PAQUET, U. AND A. P. ENGELBRECHT A New Particle Swarm Optimiser for Linearly Constrained Optimisation. Evolutionary Computation, 2003, 1, 227-233.
- RASJIDO, J., D. PANDOLFI AND A. VILLAGRA Algoritmos de reparación para el manejo de restricciones en la planificación del mantenimiento de locaciones petroleras. Informe Científico Técnico UNPA, 2011, Vol. 3, Nº. 2, 41-62.
- SCHOTT, J. R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Maestría, 1995.
- STANDISHGROUP. Standish Group 2015 Chaos Report. In., 2015.
- TALBI, E. *METAHEURISTICS FROM DESIGN TO IMPLEMENTATION*. Edition ed.: John Wiley & Sons, 1965.
- VALDEZ, J., D. PANDOLFI AND A. VILLAGRA Manejo de restricciones con función de penalidad para la planificación del mantenimiento de locaciones petroleras 2011.
- VELDHUIZEN, D. A. V. AND G. B. LAMONT Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation, 2000, 8(2), 23.
- VILLAGRA, A., M. LASSO, D. PANDOLFI, E. D. S. PEDRO, et al. 2010a. Manejo de Restricciones y Mejora en los Procesos de Búsqueda en Algoritmos Evolutivos aplicados a Problemas Dinámicos y Scheduling. In *Proceedings of the XII Workshop de Investigadores en Ciencias de la Computación* 2010a.
- VILLAGRA, A., M. LASSO, D. PANDOLFI, E. D. S. PEDRO, et al. Manejo de Restricciones usando penalidades y Procesos de Búsqueda Mejorados en Algoritmos Evolutivos aplicados a Problemas Dinámicos de Scheduling 2010b.
- VOSE, G. E. L. A. M. D. Representational Issues in Genetic Optimization. Journal of Experimental and Theoretical Computer Science, 1990, 2(2):4 30.
- WHITLEY, D., S. GORDON AND K. MATHIAS Lamarckian evolution, the Baldwin effect and function optimization. Parallel Problem Solving from Nature - PPSN III., 1994.
- ZITZLER, E. AND L. THIELE Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation, 1999, 3(4), 257-271.