

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 05/01/2015 | Aceptado: 15/02/2015

Técnicas para el tratamiento de restricciones en el problema de conformación de equipos de proyectos de software

Techniques for handling constraints on the problem of software project team conformation

Katerine Escalera Fariñas^{1*}, Ana Lilian Infante Abreu¹, Margarita André Ampuero¹, Alejandro Rosete Suárez¹

^{1*} Facultad de Ingeniería Informática, Instituto Superior Politécnico José Antonio Echeverría. Calle 114 # 11901 e/ Ciclovía y Rotonda Marianao La Habana, Cuba. {kescalera, ainfante, mayi, rosete}@ceis.cujae.edu.cu

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

Resumen

El reto de conformar equipos de proyectos de software apropiados, resulta de gran importancia para las empresas de desarrollo de software, dado que el equipo juega un papel importante en el éxito o fracaso de un proyecto. Este trabajo hace referencia a un modelo formal para la conformación de equipos de proyectos de software. El modelo responde a un problema de optimización combinatorio multiobjetivo planteando cuatro funciones objetivos y doce tipo de restricciones. Para dar solución al problema se implementaron algoritmos metaheurísticos y para tratar las restricciones que incluye el modelo se implementó la estrategia de rechazo. Esta estrategia en el problema planteado presenta inconvenientes, ya que un porcentaje considerable de las soluciones generadas es no factible. Existen diferentes técnicas para el manejo de las restricciones, en este trabajo se presenta un análisis de las diferentes técnicas que pueden ser aplicadas al problema de conformación de equipos de proyectos de software.

Palabras clave: conformación de equipos de software, algoritmos metaheurísticos, técnicas para tratamiento de restricciones

Abstract

The challenge of forming appropriate teams of software projects is of great importance for software development companies, as the team plays an important role in the success or failure of a project. This paper refers to a formal model for forming software project teams. The model responds to a multiobjective combinatorial optimization problem putting four goals and twelve functions restrictions. To solve the problem metaheuristic algorithms were implemented and to address the constraints that includes model rejection strategy was implemented. This strategy has the problem posed problems because a significant percentage of the generated solutions are not feasible. There are different techniques for handling constraints, this paper analyzes different techniques that can be applied to the problem of forming teams of software projects is presented.

Keywords: *conformation teams of software, metaheuristics algorithms, techniques for handling constraints*

Introducción

La conformación de un equipo de proyecto de software responde a un problema multiobjetivo ya que implica tomar en cuenta diferentes factores, no solo aquellos que contribuyan a la asignación del personal adecuado a cada uno de los roles, las competencias y la carga de trabajo, sino aspectos vinculados a la conformación del equipo como un todo, como las relaciones entre los miembros.

La cantidad de combinaciones de asignaciones posibles, en dimensiones relativamente significativas de roles a cubrir y empleados disponibles, provoca que el proceso de conformación del equipo sea prácticamente imposible de abordar sin la ayuda de herramientas de soporte a la decisión que se basen en algoritmos de solución de modelos matemáticos, que representen el problema a resolver lo más objetivamente posible.

Este trabajo toma como base un modelo para la conformación de equipos de proyectos de software (André, 2009; André, Baldoquín et al., 2011) que propone optimizar cuatro funciones objetivo: maximizar las competencias de los trabajadores, minimizar las incompatibilidades entre los miembros del equipo, balancear la carga de trabajo y minimizar el costo del desarrollo a distancia (para aquellas organizaciones que incluyan este modelo de trabajo), e incluye doce tipos de restricciones.

Existen diferentes técnicas para el manejo de las restricciones entre las que se encuentran: estrategia de penalización, rechazo, reparación, entre otras. Para tratar las restricciones que incluye el modelo, solo está implementada la estrategia de rechazo. Sin embargo, esta estrategia en el problema planteado presenta inconvenientes ya que el porcentaje de soluciones no factibles generadas es elevado.

Este trabajo tiene como objetivo realizar un estudio de las diferentes estrategias para tratar las restricciones que pueden ser aplicadas al problema de conformación de equipos de proyectos de software.

Materiales y métodos

El problema de conformación de equipos de proyectos implica 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.

Este trabajo toma como referencia el modelo propuesto en (André, 2009) que representa el problema de conformación de equipos de software. El modelo propone cuatro funciones objetivos: maximizar las competencias de los trabajadores, minimizar las incompatibilidades entre los trabajadores de un equipo de proyecto, balancear la carga del personal del equipo y minimizar el costo de trabajar a distancia. El último objetivo 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.

El modelo toma en cuenta un conjunto de restricciones que garantizan la asignación individual de personas a roles:

- 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.
- 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.

Existen otro conjunto de restricciones asociadas a 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 et al., 2008; Rodríguez, 2008; André, 2009; André, Baldoquín et al., 2011):

- Un conjunto de restricciones garantizan 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 deben sobrepasar la preferencia por desempeñar los roles sociales.
- 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 Cerebro.
- La persona que desarrolla el rol Jefe de Proyecto debe ser extrovertida y planificada (subtipo E_ _J) según el test de Myers-Briggs.

Metaheurísticas

Dado que el modelo de conformación de equipos de software corresponde a un problema combinatorio, se utilizaron algoritmos metaheurísticos para su solución.

Los algoritmos metaheurísticos son algoritmos aproximados de propósito general. Ofrecen en un tiempo de ejecución razonable una buena solución al problema planteado pero no garantizan obtener una solución óptima.

Entre los algoritmos implementados se encuentran Escalador de Colinas de Mejor Ascenso con Reinicio (Juels and Watenberg, 1994; Jones, 1995), Recocido Simulado (Kirkpatrick, Gelatt et al., 1983), Búsqueda Tabú (Glover, 1986), Híbridos de GRASP (Resende and Ribeiro, 2003) como: GRASP y Escalador de Colinas de Mejor Ascenso, GRASP y Recocido Simulado, GRASP con Tabú, Escalador de Colinas Estocástico Multiobjetivo (Díaz, 2001), Escalador de

¹Test creado por el Dr. Meredith Belbin, cuya metodología establece que en un equipo debe haber presencia de las tres categorías de roles (mentales, sociales 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

Colinas Estocástico Multiobjetivo con Reinicio (Díaz, 2001), Escalador de Colinas Estocástico Multiobjetivo por mayor distancia (Díaz, 2001), Búsqueda Tabú Multiobjetivo (Baykasoglu, Ozbaku et al., 2002), Recocido Simulado Multiobjetivo Multicaso (Haidine and Lehnert, 2008) y Recocido Simulado Multiobjetivo de Ulungu (Ulungu and Teghem, 1994).

Representación del problema

Representación del problema

Para el problema planteado, la representación de una solución se muestra en la Figura 1. Dado que el problema consiste en asignar trabajadores a los roles definidos en un equipo de proyecto y que se puede necesitar que se asignen 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.

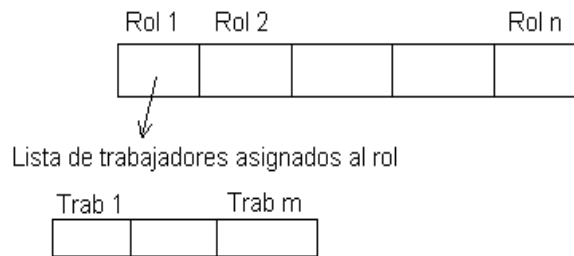


Figura 1. Representación de una solución.

El problema planteado tiene una particularidad, asociada al hecho de que se puede querer buscar una solución fijando uno o varios trabajadores en uno o más roles.

Los operadores utilizados son sustitución y permutación. El operador sustitución, escoge uno de los roles de la solución y sustituye uno de los trabajadores que juegan ese rol por otro del espacio de soluciones. El operador permutación escoge dos roles de la solución aleatoriamente, selecciona un trabajador de cada rol y los intercambia.

Tamaño del espacio de solución

El tamaño del espacio de soluciones se calcula a través de las r -permutaciones (Johnsonbaugh, 2004) como $n!/(n-m)!$, donde n es la cantidad de trabajadores y m la cantidad de roles. Poner ejemplo para una instancia de 60 trabajadores y

6 roles, suponiendo que un trabajador solo puede desempeñar un rol y que el rol Jefe de Proyecto se asigna al inicio y permanece fijo durante la asignación, el espacio de soluciones es 6×108 .

Funciones objetivo

Siendo n la cantidad de trabajadores y m la cantidad de roles a cubrir para formar un equipo de proyecto, el modelo referido propone optimizar las siguientes funciones objetivo (André, 2009):

- Maximizar las competencias de los trabajadores

$$\max \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \quad (1)$$

Siendo $x_{ij} = 1$ si el empleado i es asignado al rol j y 0 en caso contrario, y c_{ij} la competencia neta del empleado i para desempeñar el rol j .

Para determinar la competencia neta, es necesario tener en cuenta los índices de competencia del trabajador i en las competencias necesarias para desempeñar el rol j , así como la experiencia del trabajador en el desempeño de dicho rol.

El índice de competencias tiene en cuenta los niveles de competencia que tiene el trabajador i en las competencias genéricas y técnicas necesarias para el desempeño del rol j y el peso que tiene cada competencia en el rol en función de su importancia.

La experiencia se determina a partir de la cantidad de proyectos en los que el trabajador i ha desempeñado el rol j , así como la evaluación recibida en su desempeño.

- Balancear la carga de trabajo de los trabajadores

$$\min \sum_{i=1}^n \left[\left(L_i + \sum_{j=1}^m b_j x_{ij} \right) - ME \right] \quad (2)$$

Donde

$$L_i = \sum_{j=1}^m g_{ij} \quad y \quad ME = \frac{\sum_{i=1}^n \sum_{j=1}^m (g_{ij} + b_j)}{n} \quad (3)$$

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.

La carga de trabajo de un empleado se calcula a partir de la cantidad de proyectos en los que está asignado y la complejidad del rol que está desempeñando en esos proyectos. Asimismo, el valor asignado a la carga de trabajo que implica asumir el rol j en un proyecto determinado está en dependencia de las características del proyecto.

- Minimizar las incompatibilidades entre los miembros del equipo

$$\min \sum_{h=1}^{n-1} \sum_{i>h}^n s_{hi} u_h u_i \quad (4)$$

Siendo s_{hi} la incompatibilidad existente entre los trabajadores h e i y $u_i=1$ si el empleado i es asignado al menos a un rol y 0 en caso contrario.

El cálculo de la incompatibilidad se realiza a partir de determinar las relaciones recíprocamente negativas entre dos miembros del equipo y en inicio se establece que su valor es 1 ó 0.

- Minimizar el costo de desarrollar software a distancia

$$\min \sum_{i=1}^n \sum_{j=1}^m l_{ij} x_{ij} \quad (5)$$

Donde l_{ij} es el costo del empleado i según la lejanía que tenga del proyecto y el rol j que va a desempeñar.

El cálculo del coeficiente que representa el costo de asignar el empleado i al rol j cuando el proyecto se desarrolla a distancia, depende del costo de desarrollar ese rol a distancia y el costo del empleado i de acuerdo a su localización respecto a la localización del proyecto.

Restricciones

El modelo propone 12 tipos de restricciones, 6 de las cuales corresponden con los patrones para la conformación de equipos propuestas en (André, 2009) .A continuación se detallan las restricciones:

$$\sum_{i=1}^n x_{ij} = A_j \quad \text{donde } A_j \in \mathbb{N} \quad (6)$$

Las restricciones de tipo 1(Ecuación 6) garantizan que todos los roles sean cubiertos exactamente en su demanda (A_j).

$$\sum_{j \in J_r} x_{ij} \leq 1 \quad \text{siendo } r = 1, \dots, R \quad (7)$$

Las restricciones de tipo 2 (Ecuación 7) garantizan que una persona solo pueda asumir un rol dentro de cada subconjunto de roles que se consideran incompatibles entre sí.

$$\sum_{j=1}^m x_{ij} \leq L \quad L \in \mathbb{N} \quad (8)$$

Las restricciones de tipo 3 (Ecuación 8) hacen que el número máximo de roles que pueda asumir un empleado en el proyecto se restrinja a la cantidad fijada por el usuario (L).

$$(1 - u_i) \sum_{j=1}^m x_{ij} = 0, (1 - u_i) + \sum_{j=1}^m x_{ij} > 0 \quad (9)$$

Las restricciones de tipo 4 (Ecuación 9) garantizan que las variables u_i tomen valor 1 si el empleado 1 es asignado al menos a un rol y 0 en caso contrario.

$$u_i(k_{hi} - v_{ih}) \leq 0, \forall h \in Z_j \quad (10)$$

Donde Z_j es el conjunto de competencias necesarias para desempeñar el rol j.

Las restricciones de tipo 5 (Ecuación 10) imponen el cumplimiento de condiciones mínimas en cuanto a las competencias necesarias en una persona para que se le asigne un rol dado.

$$L_i + \sum_{j=1}^m b_j x_{ij} \leq U \quad (11)$$

Donde U es el valor máximo de carga que puede asumir cada empleado y b_j es la carga de trabajo que implica asumir el rol j en un proyecto determinado.

Las restricciones de tipo 6 (Ecuación 11) establecen que la carga de trabajo total asignada a cada empleado no debe sobrepasar un valor máximo.

Las restricciones 7-11(Ecuaciones 12-16) que se muestran a continuación tienen en cuenta los roles de Belbin.

$$\sum_{k=1}^3 \left(\sum_{i=0}^n d_{ik} u_i \right) > 0 \quad \sum_{k=4}^6 \left(\sum_{i=1}^n d_{ik} u_i \right) > 0 \quad \sum_{k=7}^9 \left(\sum_{i=1}^n d_{ik} u_i \right) > 0 \quad (12)$$

La restricción 7 (Ecuación 12) impone que en el equipo de trabajo seleccionado deben estar representadas las tres categorías de roles de Belbin (roles de acción, roles mentales y roles sociales). Se asume una matriz D, donde las columnas representan los roles de Belbin, las tres primeras columnas están asociadas a los roles de acción, las columnas del 4 al 6 a los roles mentales y las tres últimas a los roles sociales.

$$\sum_{k=1}^3 \left(\sum_{i=1}^n d_{ik} u_i \right) > \sum_{k=4}^6 \left(\sum_{i=1}^n d_{ik} u_i \right) \quad (13)$$

La restricción 8 (Ecuación 13) impone que en el equipo de trabajo debe existir preferencia por desempeñar roles de acción sobre los roles mentales.

$$\sum_{k=4}^6 \left(\sum_{i=1}^n d_{ik} u_i \right) > \sum_{k=7}^9 \left(\sum_{i=1}^n d_{ik} u_i \right) \quad (14)$$

La restricción 9 (Ecuación 14) impone que las preferencias por desempeñar roles mentales debe sobrepasar la preferencia por los roles sociales.

$$x_{i1} \leq d_{i1} + d_{i7} \quad (15)$$

Las restricciones tipo 10 (Ecuación 15) imponen que si la persona i desarrolla el rol Jefe de Proyecto, debe tener como preferido al menos uno de los roles de Belbin: Impulsor o Coordinador. Para elaborar la restricción se asumen que en una matriz D , la preferencia o no por desempeñar los roles Impulsor y Coordinador se registran en las columnas 1 y 7 respectivamente.

$$\sum_{i=1}^n d_{i4} u_i \geq 1 \quad (16)$$

Las restricciones tipo 11 (Ecuación 16) imponen que en el equipo al menos una persona tenga como preferido el rol Cerebro. Se asumen que en la columna 4 de la matriz D se registra la preferencia o no por el rol Cerebro.

$$x_{i1} \leq \frac{(b_{i1} + b_{i4})}{2} \quad (17)$$

Las restricciones de tipo 12 (Ecuación 17) imponen que si la persona i desarrolla el rol Jefe de Proyecto, debe tener como subtipo E_ _J según el test de Myers-Briggs. Asumiendo una matriz B , la dimensión E/I se registra en la columna 1 y la J/P en la columna 4.

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). Los algoritmos de búsqueda generan soluciones de ambos espacios, 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 y por tanto son más difíciles de garantizar en el momento de conformar la solución.

Técnicas para el tratamiento de restricciones

Mucho de los problemas de optimización están sujetos a restricciones. Estas pueden ser de cualquier tipo: lineal o no lineal, de igualdad o desigualdad.

Existen diferentes técnicas para el manejo de las restricciones que actúan principalmente en la representación de la solución o de la función objetivo. Estas técnicas se pueden clasificar como: estrategias de rechazo, estrategias de penalización, estrategias de reparación, entre otras (Talbi, 1965). A continuación se explica en qué consisten cada una de estas técnicas.

Estrategia de rechazo: consiste en aceptar durante la búsqueda solo soluciones factibles, descartando las soluciones no factibles que son generadas. Este tipo de estrategia generalmente es usada cuando la porción de soluciones no factibles del espacio de búsqueda es pequeño (Talbi, 1965).

En algunos problemas las regiones factibles del espacio de soluciones puede ser discontinuo, por lo que un camino entre dos soluciones factible solo puede existir si se aceptan soluciones no factibles (Talbi, 1965).

Estrategia de penalización: esta estrategia considera en la búsqueda soluciones no factibles pero estas son penalizadas. Existen diferentes alternativas para definir las sanciones, entre las que se encuentran (Talbi, 1965):

Restricciones violadas: consiste en contar el número de restricciones que se incumplen y agregar a la función objetivo una función de penalización.

Dadas M restricciones, la función de penalización $f_p(x)$ de $f(x)$ se puede definir como se muestra en la Ecuación 18:

$$f_p(x) = f(x) + \sum_{i=1}^m w_i \alpha_i \quad (18)$$

donde $\alpha_i = 1$ si la restricción i es violada y $\alpha_i = 0$ en caso contrario, w_i es el coeficiente asociado con cada restricción i .

Estrategia de reparación: consiste aplicar un operador de reparación a cada solución no factible que se genere con objeto de transformarla en otra factible (Talbi, 1965).

Modificación de operadores: Está diseñada de forma que asegure la factibilidad de las soluciones generadas (Talbi, 1965).

Estado del arte de utilización de técnicas para el tratamiento de restricciones

En (Coello and Cruz, 2004) se propone un algoritmo genético para la optimización global donde se manejan las restricciones de todo tipo (lineal, no lineal, de igualdad y desigualdad). En este trabajo utilizan la estrategia de penalización, donde la función de penalización es la siguiente:

$$\Phi(\vec{x}) = f(\vec{x}) \pm \left[\sum_{i=1}^n r_i * G_i + \sum_{j=1}^p c_j * L_j \right]$$

Donde $\Phi(\vec{x})$ es la nueva función objetivo a optimizar, G_i y L_j son las funciones de las restricciones $g_i(\vec{x})$ y $h_j(\vec{x})$ respectivamente, r_i y c_j son constantes positivas normalmente llamadas factores de penalización.

En (Villagra, Lasso et al., 2010) aplican al problema de planificación del mantenimiento de locaciones petroleras funciones de penalidad usando un algoritmo multirecombinatorio donde estas funciones serán: Estáticas (se consideren enfoques en los cuales los factores de penalidad no dependen del número de generación actual y por esta razón se mantienen constantes durante todo el proceso evolutivo), Dinámicas (se considera cualquier función de penalidad en la cual está involucrado el número de generación actual en el cálculo de la función de penalidad correspondiente), Recocido (están basadas en la idea de Recocido Simulado), Adaptativa (usan una función de penalidad que se retroalimenta del proceso de búsqueda) y Mortal (rechazo de individuos no-factibles).

En (Mezura and Coello) para el manejo de restricciones en algoritmos evolutivos utilizan un conjunto de técnicas donde las restricciones son manejadas como objetivos (un problema mono-objetivo con restricciones se convierte en un problema multiobjetivo sin restricciones).

En (Centeno, Rey et al., 2012) se implementa una versión híbrida entre el esquema local y global del AEP (Algoritmo de Enjambre de Partículas). Entre los basamentos del algoritmo se encuentra que entre dos soluciones no factibles selecciona aquella que satisfaga el mayor número de restricciones.

En (Villagra, Lasso et al., 2010) se consideran dos tipos de funciones de penalidad, exterior e interior. La penalidad exterior comienza con una solución no factible y desde esta se mueve a una solución factible, mientras que la penalidad interior inicia la búsqueda en la zona factible y trata de evitar salir de esta zona. En este trabajo utilizan la función de penalidad exterior, ya que es el método más común utilizado en Algoritmos Evolutivos.

Discusión

Técnicas para el tratamiento de restricciones en el problema de conformación de equipos

Una vez estudiadas las diferentes técnicas para el tratamiento de restricciones y las particularidades del problema de conformación de equipos, se hace un análisis de las técnicas que pueden ser utilizadas en este problema para tratar las restricciones del mismo.

- La estrategia de rechazo solo considera soluciones factibles, las soluciones que incumplen alguna de las restricciones son consideradas soluciones no factibles, y por tanto no se toman en cuenta durante la búsqueda de una solución. En pruebas realizadas teniendo en cuenta esta técnica aproximadamente el 67% de las soluciones generadas son no factibles, por lo que gran parte del tiempo de búsqueda se emplea en construir soluciones que no serán aceptadas debido a que son soluciones no factibles.
- Con la estrategia de penalización se aceptan soluciones no factibles pero estas son penalizadas. Entre las funciones de penalización que pueden utilizarse para este problema se encuentran: $fp(x) = f(x) * (1 - P)$ donde $P = \text{cantidad de restricciones incumple} / \text{total de restricciones}$, $p(x) = f(x) + \sum_{i=1}^m w_i \alpha_i$ donde $\alpha_i = 1$ si la restricción i es violada y $\alpha_i = 0$ en caso contrario, w_i es el coeficiente asociado con cada restricción i . A pesar de que pueden utilizarse varias funciones de penalización esta técnica tiene como inconveniente que no garantiza obtener una solución factible, pudiéndose obtener como resultado final de la búsqueda una soluciones que incumpla con algunas de las restricciones.

- Teniendo en cuenta la complejidad del problema y la representación utilizada, no se ha encontrado un operador que garantice la factibilidad de las soluciones generadas.
- Uno de los operadores utilizados en la solución del problema es el operador de sustitución. Este operador escoge uno de los roles de la solución y sustituye uno de los trabajadores que juegan ese rol por otro del espacio de soluciones. En este caso se pudiera tener organizado el espacio de soluciones en función del cumplimiento de cada una de las restricciones individuales, lo que garantizaría asignar en un rol a las personas que cumplan con las restricciones. Sin embargo, las restricciones de equipo solo se pueden chequear una vez conformado el equipo y por tanto son más difíciles de garantizar en el momento de conformar la solución. Aunque esta variante no garantiza obtener siempre soluciones factibles, al menos disminuye el número de restricciones a chequear.
- Una alternativa viable es aplicar la estrategia de modificación de operadores explicada anteriormente y la estrategia de penalización para las restricciones de equipo.
- Otro elemento a considerar es partir de una solución que cumpla con todas las restricciones, es decir una solución factible y a partir de esta emplear cualquiera de las estrategias analizadas.

Teniendo en cuenta esto, para tratar las restricciones en el problema de conformación de equipos pueden utilizarse las estrategias de rechazo, penalización, puede modificarse el operar utilizado y la combinación de estas.

Conclusiones

A partir del estudio realizado se arriban a las siguientes conclusiones:

- Para el manejo de las restricciones existen diferentes técnicas: estrategias de rechazo, penalización, reparación, tratamiento de las restricciones como objetivos, entre otras.
- Entre las técnicas para el manejo de restricciones más utilizadas se encuentran las estrategias de penalización.
- Con la técnica de rechazo gran parte del tiempo se emplea en generar soluciones no factibles y puede no estarse explorando espacios de soluciones factibles.
- Con la técnica de penalización, puede obtenerse como resultado de la búsqueda una solución no factible.

- Los operadores utilizados en la solución del problema no garantizan obtener una solución factible.
- Para tratar las restricciones en el problema de conformación de equipos pueden utilizarse las estrategias de rechazo, penalización, puede modificarse el operar utilizado y la combinación de estas.

Referencias

- CHAPMAN, Pete and et.al. *CRISP-DM 1.0*. Washington D. C., EEUU: SPSS, 2000. págs. 1-76.
- CORTEZ, Paulo; SILVA, Alice. *Using Data Minig to predict secondary school student performance*. Guimaraes, Portugal : s.n., 2006.
- FAYYAD, Usama; PIATESKY, Gregory; PADHRAIC, Smyth. *Knowledge Discovery in Databases*. 1996. págs. 37-54.
- FERNÁNDEZ, Santiago. *Regresión Logística*. Madrid : Universidad Autónoma de Madrid, 2011.
- FERNÁNDEZ, Pita. Investigación: Determinación del tamaño muestral. Unidad de Epidemiología Clínica y Bioestadística. A Coruña. Cad Aten Primaria 1996. págs. 1-6.
- HAN, Jiawei. *Introduction to Data Mining*. San Francisco : Morgan Kaufmann, 2006. págs. 1-20.
- HERNÁNDEZ, José; RAMÍREZ, José; FERRI, César. *Introducción a la Minería de Datos* Madrid: Pearson, 2004. págs. 3-39.
- HUEBNER, Richard. *A survey of educational data-mining research*. Norwich : Norwich University, 2013. pág. 13.
- IBM Corporation. Rational Unified Process. *Concepts: Performance Testing*. [En línea] 2003. [Citado el: 8 de 11 de 2013.] http://students.mimuw.edu.pl/~zbyszek/posi/ibm/RUP_Eval/process/workflow/test/co_perfo.htm.
- KUMAR, Brijesh y SAURABH, Pal. *Mining Educational Data to analyze student's performance*. Rajasthan, India : IJACSA, 2011. págs. 63-69.

- MACLENNAN, Jamie. *Data Mining with Microsoft SQL Server 2008*. Indianapolis, EEUU, Wiley Publishing Inc. 2008. págs. 39-53.
- MERCERON, Agathe y KALINA, Yacef. *Educational Data Mining: a Case Study*. Sydney : University of Sydney, 2005. págs. 1-8.
- MICROSOFT CORPORATION. Algoritmo de Árboles de Decisión. [En línea] Microsoft Developer Network, 2012. [Citado el: 6 de 11 de 2013.] <https://msdn.microsoft.com/es-ec/library/ms175312.aspx>.
- MICROSOFT CORPORATION. Algoritmo Regresión Logística de Microsoft. [En línea] Microsoft Developer Network, 2012. [Citado el: 6 de 11 de 2013.] <http://msdn.microsoft.com/es-es/library/ms174806.aspx>.
- MICROSOFT CORPORATION. Algoritmos de minería de datos. [En línea] Microsoft Developer Network. [Citado el: 6 de 11 de 2013.] <https://msdn.microsoft.com/es-es/library/ms175595.aspx>.
- MORENO, María; QUINTALES; Luis y GARCÍA; Francisco. *Aplicación de técnicas de minería de datos en la construcción y validación de modelos predictivos y asociativos a partir de especificaciones de requisitos de software*. Salamanca : Universidad de Salamanca, 2001. págs. 1-14.
- VALLEJOS, Sofía. *Minería de Datos*. Corrientes, Argentina, Universidad Nacional de Noreste, 2006, págs. 11-16.
- ULUNGU, E. L. and J. TEGHEM. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 1994, 3 22
- VILLAGRA, A., M. LASSO, et al. Manejo de Restricciones y Mejora en los Procesos de Búsqueda en Algoritmos Evolutivos aplicados a Problemas Dinámicos y Scheduling. En: XII Workshop de Investigadores en Ciencias de la Computación. Red de Universidades con Carreras en Informática (RedUNCI), 2010, p. 116-120