

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 17/08/2016 | Aceptado: 10/10/2016

Experiencias usando algoritmos genéticos en la planificación de proyectos

Experiences by using genetic algorithms in project scheduling

Roberto García Vacacela^{1*}, Iliana Pérez Pupo², Nicolás Villavicencio¹, Pedro Y. Piñero², Susej Beovides²

¹Universidad Católica Santiago de Guayaquil, Vicerrectoría de Investigación y Postgrado, roberto.garcia@cu.ucsg.edu.ec, nicolas.villavicencio@cu.ucsg.edu.ec

² Universidad de las Ciencias Informáticas, Laboratorio de Investigaciones en Gestión de Proyectos, Carretera A San Antonio km 2 ½, CP 17830. iperez@uci.cu, ppp@uci.cu

* Autor para correspondencia: roberto.garcia@cu.ucsg.edu.ec

Resumen

Los procesos de planificación de proyectos se presentan como el problema de organizar un conjunto de actividades respetando sus relaciones de precedencia, asignando, además, recursos humanos y no humanos a las mismas y sin violar su disponibilidad en cada instante de tiempo. Actualmente aproximadamente el 61% de los proyectos de tecnologías de la información son renegociados o cancelados provocando impacto negativo tanto económicos como sociales. Muchos de estos fracasos se deben a deficientes procesos de planificación provocados por la poca alineación a estándares, la insuficiente experiencia y la falta de herramientas que ayuden a generar cronogramas de proyectos óptimos o cuasi óptimos en entornos de múltiples proyectos. En este contexto los procesos de planificación han sido tratados en la literatura científica como problemas de planificación de proyectos con recursos limitados; siendo este un problema de optimización combinatoria de la clase NP-Duro. El objetivo de este trabajo es presentar algunas experiencias para la resolución de este problema aplicando algoritmos genéticos. Se presenta la modelación matemática de este problema de optimización adaptado al escenario de aplicación de organizaciones orientadas a proyectos de desarrollo de software. Se discute el diseño de un algoritmo genético con diferentes variantes de operadores. Finalmente se muestran los resultados de la aplicación de las técnicas propuestas en la biblioteca PSLIB y en un escenario real de aplicación asociado a las bases de datos experimentales generadas a partir de la aplicación de la plataforma para la dirección integrada de proyectos GESPRO.

Palabras clave: planificación de proyectos; proyecto de software; algoritmos genéticos; metaheurísticas; gespro.

Abstract

Usually project scheduling is presented as problem to order project tasks keeping the tasks precedence relationship. Besides, in this problem, human and non-human resources should be assign to each tasks without violate the resources availability each time. Nowadays around 61% of software´s projects are challenged o cancelled with high negative impacts both economic and social too. Many of these fails have a commons reasons such as: project scheduling errors, low level of knowledge of standards and not enough tools to help specialist in project scheduling. Different authors present project scheduling problems, as optimization NP-hard problem, with limited resources. In order to solve projects scheduling in software´s projects environments. In this paper authors discuss mathematics formalization of multiple projects optimization problem. They present a genetic algorithm´s design and different crossover and mutation operators. Finally, the genetic algorithm designed, were applied in two project scheduling´s data sets: PSLIB data set and scheduling data set of Project management Laboratory generated form GESPRO system.

Keywords: *Project scheduling; software projects; metaheuristics; genetic algorithm; gespro.*

Introducción

La Gestión de Proyectos (GP) conocida también como Dirección o Administración de Proyectos (PM, siglas en inglés *Project Management*), busca planear, organizar, asegurar y coordinar recursos y personas para cumplir con los objetivos, entregables, y criterios de éxito en los proyectos. Sin embargo, actualmente los proyectos siguen presentando dificultades con la planificación, la definición del alcance, estimaciones erróneas e insuficiente gestión. Un estudio realizado en el 2014 por el *The Standish Group International, Incorporated* arrojó como resultado que históricamente las cifras de proyectos entregados satisfactoriamente, cerrados o fallidos y renegociados se ha movido por alrededor del 35%, 18% y 43% respectivamente (Standish Group International, 2015). Como se puede apreciar aproximadamente el 61% de los proyectos tienen dificultades durante su ejecución causando con frecuencia impactos negativos tanto desde el punto de vista económico como social.

Entre las causas fundamentales del fracaso de los proyectos (Villavicencio, 2016) se señalan: insuficiente formación de los recursos humanos y errores frecuentes en los procesos de planificación ya sea del alcance, el tiempo o la logística (Mossalam y Arafa 2014; Amoui, et. al, 2012; Delgado 2013; Piñero y Torres, 2013). También, se señala poca interacción con los clientes, dificultades en la gestión de los riesgos y en la gestión de los proveedores.

Como se puede apreciar muchos de los problemas están asociados a dificultades con la planificación de los proyectos, problema que adquiere un alcance mayor en las organizaciones orientada a proyectos donde es preciso planificar

proyectos que se ejecutan simultáneamente. Esta situación puede ser mitigada por acciones orientadas a mejorar la calidad de las planificaciones soportada en los procesos de construcción de los cronogramas de proyectos.

La confección de un cronograma de proyecto se presenta como el problema de organizar un conjunto de actividades, respetando sus relaciones de precedencia, y asignando a las mismas, recursos humanos y no humanos sin violar la disponibilidad en cada instante de tiempo de estos recursos. En este sentido las técnicas clásicas de planificación de proyectos PERT y CPM, no consideran explícitamente la asignación de los recursos a las tareas, sino que constituyen herramientas de ayuda para la graficación y el análisis de los cronogramas una vez construidos los mismos. En este contexto desde finales de los sesenta el problema de construcción de cronogramas de proyectos, suele abordarse como un problema de optimización (Villavicencio, 2016), conocido como problema de planificación de proyectos con recursos limitados siendo es un problema de optimización combinatoria NP-Duro (Demeulemeester y Herroelen, 2002). Para la resolución de este problema de optimización se concentran los esfuerzos en dos tipos de restricciones:

- Restricciones asociadas a la precedencia inicio-fin (PMBOK, 2013; PMI, 2014) entre las actividades.
- Restricciones asociadas a la asignación de recursos donde generalmente los mismos se encuentran en una cantidad fija y limitada en cada espacio de tiempo.

El objetivo del RCPSP (El Problema de Secuenciación de Proyectos con Recursos Limitados) es hallar el momento de inicio de cada actividad de tal forma que se minimice el tiempo de duración del proyecto (Brucker, Drexl, et. al., 1999). Pero la construcción de cronogramas se complejiza aún más en los proyectos de desarrollo de software, donde intervienen diversas variables como las competencias de los recursos humanos, la formación de equipos balanceados o el desarrollo simultáneo de múltiples proyectos con recursos limitados (Browninga y Yassineb, 2010). En estos escenarios las soluciones al problema RCPSP generalmente no son suficientes (Bruni, Beraldi, et. al., 2011).

Desde los años noventa Abdel-Hamid en (Abdel-Hamid, 1993), plantea el problema de la ausencia de resultados de investigación asociados al tratamiento de la planificación en entornos multiproyectos de desarrollo de software. Luego en (Gonsalves, Ito, et. al, 2008), se muestran algunos resultados para el entorno multiproyecto, pero en escenarios teóricos, en los que se consideran proyectos de pocas actividades y no se describen las características de los recursos humanos. Así mismo otros autores presentaron aplicaciones de la teoría de la planificación en proyecto de software (Kea y Liub, 2010; Gonsalves, Itoh, 2010; Santos y Tereso, 2011; Di Penta, et. al., 2011; Heimerl y KOLISCH, 2010) pero generalmente asociados a lograr la representación de las características de los recursos humanos en los modelos de planificación que no son para múltiples proyectos. Varios autores coinciden en que la planificación de proyectos de

desarrollo de software en entornos multiproyectos se mantiene como un campo abierto a la investigación aún en nuestros días (Beovides, 2015; Bolivar, 2013; Pinedo, 2015).

El objetivo de este trabajo es presentar problema de planificación de múltiples proyectos de desarrollo de software en escenarios con recursos limitados, como un problema de optimización. Y discutir algunas experiencias en la aplicación de metaheurísticas en la resolución de este complejo problema de optimización. Para la solución del problema de optimización se presentan varios autores han empleado metaheurísticas (Chen, Shi, et. al., 2010; Chen y Zhang, 2013). En este trabajo en particular se propone el uso de algoritmos genéticos. Se opta por esta metaheurística considerando las siguientes ventajas:

- Pueden ser paralelizables porque realizan la búsqueda a partir de una población de puntos, no de punto simple.
- Utilizan solo la información de la función objetivo, sin necesidad de conocimiento previo como derivadas u otro conocimiento auxiliar y permiten la resolución de problemas multiobjetivos de forma simultánea.

Los algoritmos genéticos son una de las metaheurísticas más empleadas en la solución de los problemas de planificación de proyectos en especial del problema RCPSP en (Kolisch y Hartmann, 2006). La función objetivo en el trabajo de Hartmann se centra en la duración del proyecto. Muchos trabajos posteriores toman como referencia este trabajo entre ellos los trabajos (Gonzalves, et. al., 2011; Chicano, Luna, et. al., 2011; Kea y Liub, 2010; Masmoudia, y Häitb, 2013), donde aplican algoritmos genéticos nuevamente al problema RCPSP.

El trabajo está organizado de la siguiente forma. En la sección 2 se presenta la metodología empleada en la investigación, se conceptualiza el problema de optimización de cronogramas de proyectos usando meta-heurísticas. En la sección 3 se presentan los resultados de la investigación y finalmente se presentan las conclusiones del trabajo.

Metodología computacional

Para la resolución del problema de optimización presentado se propone el uso de metaheurísticas en particular el uso de algoritmos genéticos y de algoritmos para la estimación de distribuciones bajo un enfoque univariado.

Se siguieron los siguientes pasos durante el desarrollo de la investigación:

1. Conceptualización del problema de optimización asociado.
2. Diseño de un algoritmo genético para la solución al problema de optimización.
3. Experimentación en bases de datos asociadas a la planificación de proyectos.
4. Análisis de resultados y discusión

Conceptualización del problema de optimización asociado.

Para resolver el problema de optimización asociado a la construcción de cronogramas de proyectos han surgido variantes del problema RCPSP (Brucker, Drexl, et. al., 1999). Una de estas variantes de este problema es la planificación de proyectos con múltiples modos de procesamiento (MRCPSP) en (Bruni, Beraldi, et. al., 2011), que considera que las actividades de un proyecto se pueden ejecutar con distintas cantidades de recursos. Este enfoque se basa en que la cantidad de recursos que se emplea en la ejecución de una tarea condiciona su duración, y establece la relación inversa, a mayor cantidad de recurso asignado, menor duración de la tarea. Otra variante es el problema de planificación de múltiples proyectos con recursos limitados (Pinedo, 2015), donde se considera que varios proyectos se ejecutan con los mismos recursos. En nuestro caso el problema a resolver es precisamente un problema RCMPSP (*Resource-Constrained Multi-Project Scheduling Problem*) que es el problema que se presentan en las organizaciones orientadas a proyectos.

Se introduce la siguiente notación para la formulación del problema de optimización.

En el sistema serán considerados conjunto $I = \{1, \dots, n\}$ de proyectos donde cada uno tiene una fecha de inicio $PStart_Date_i$ y una fecha de fin $PDue_Date_i$. Cada proyecto $i \in I$ está compuesto por un conjunto $J_i = \{j : j = 0, 1, \dots, N_{j_i}\}$ de tareas (nodos), donde $j = 0$ y $j = N_{j_i}$ son tareas ficticias (nodos inicio y fin), con tiempo de duración igual a cero y que no requieren recursos para su procesamiento, solo indican el inicio y fin del proyecto.

Para simplificar el problema suponemos que se cumplen las siguientes asunciones. Primera, la relación de precedencia entre las tareas son siempre del tipo *inicio-fin*¹ y solo entre tareas de un mismo proyecto. Segunda se considera cuatro competencias fundamentales $Competencias = \{análisis, diseño, implementación, prueba\}$. Además, se considera que cada recurso tiene una sola competencia principal.

Los miembros del equipo de desarrollo, se denotan como $r \in R_K$, $R = \bigcup_{k \in K} R_k$ indicando que cada recurso r tiene una competencia principal asociada k . Se conciben a las tareas como expresiones atómicas, lo que significa que cada tarea implica un tipo de competencia específica. Por tanto cada una de las $j \in J_i \setminus \{0, N_{j_i}\}$ tareas, está asociada, a una y solo una de las cuatro competencias identificadas. Para simplificar la notación identificaremos cada competencia con un valor numérico que representaremos con el término $K = \{1, 2, 3, 4\}$, $k \in K$ de forma tal que $k=1$

¹ Inicio-Fin: el inicio de una tarea depende que se haya concluido completamente la que la precede.

representa *análisis*, $k=2$, diseño así sucesivamente. Una tarea j que requiere la competencia k para poder ejecutarse, requiere algún recurso humano con esa competencia específica k .

Cada tarea $j \in J_{jk}$ de un proyecto i posee una fecha de inicio s_{ij} , y una fecha de fin f_{ij} . Cada tarea comienza en un momento t y el conjunto de todos los momentos en que ocurren acciones en el proyecto se denotará por T .

Se dice que cada tarea j es ejecutada de un modo m de M_j posibles. El modo está asociado al nivel de las competencias del recurso humano r que ejecuta dicha tarea. La duración de una tarea j , la denotaremos por d_{jr} , está determinada por la capacidad productiva del recurso que la ejecuta ($d_{jr^*} \leq d_{jr}$ si la capacidad productiva del recurso r^* es mayor que la del recurso r). Se considera que, si una tarea j es ejecutada por un recurso con alto nivel de competencias profesionales, se ejecuta en modo $m=1$, si es ejecutada por un recurso con un nivel medio de competencia profesional se ejecuta en modo $m=2$, y si la competencia del recurso es baja capacidad se ejecuta en modo $m=3$. Cada recurso r tiene un salario c_r por cada unidad de tiempo laborada.

Se define el problema de optimización asociado a la construcción de cronogramas de proyectos como un modelo línea en enteros donde la variable de decisión: $x_{ijrt} = \{0,1\}$ tal que $x_{ijrt} = 1$ si en el proyecto $i \in I$ la actividad $j \in J_{ik}$, es ejecutada por el recurso $r \in R_k$, en el momento t y $x_{ijrt} = 0$ en cualquier otro caso.

La función objetivo de este problema de optimización queda formulada como minimizar Z , donde t indica el tiempo en que se ejecuta cada tarea.

$$\min Z = \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T w_i t x_{ijrt} \quad (1)$$

Sujeta a las siguientes restricciones:

$$\sum_{r \in R_k} \sum_{t \in T} x_{ijrt} = 1, \quad i \in I, j \in J_{ik}, k \in K \quad (2)$$

$$\sum_{r \in R_k} \sum_{t \in T} x_{ijrt} = 1, \quad i \in I, j \in J_{ik}, k \in K$$

La (2) es una restricción básica que indica que cada tarea debe ser realizada como máximo una vez durante el desarrollo del proyecto.

$$\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} \leq \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt}, \quad \forall i \in I, j \in A_{ih}, h \in J_{ik}, k \in K \quad (3)$$

La
$$\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} \leq \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt} \quad , \forall i \in I, j \in A_{ih}, h \in J_{ik}, k \in K$$
 (3 representa la restricción garantiza que no se violen las relaciones de precedencia entre las tareas, donde A_{ih} es el conjunto de las actividades antecesoras de $h \in J_{ik}$.

$$\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \leq |R_k| \quad , k \in K \quad (4)$$

La
$$\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \leq |R_k| \quad , k \in K$$
 (4 presenta la restricción que

garantiza que no se exceda el uso de los recursos disponibles en el instante t.

$$\sum_{r \in R_k} \sum_{t \in T} t x_{ijrt} \geq TPStart_Date_i \quad , i \in I, j = 1 \quad (5)$$

Finalmente la
$$\sum_{r \in R_k} \sum_{t \in T} t x_{ijrt} \geq TPStart_Date_i \quad , i \in I, j = 1$$
 (5 asegura la restricción que todas las tareas inician después de que el proyecto ha iniciado formalmente. En lo adelante denotamos por C al conjunto de las restricciones siendo $c_q \in C$ la q -ésima restricción.

Diseño de algoritmo genético para la resolución del problema.

Se propone en esta sección un algoritmo genético Algoritmo 1 para la resolución del problema de optimización planteado. En el caso particular se propone una estrategia que combina las restricciones del problema en la función objetivo. Básicamente se propone la penalización de las soluciones considerando el nivel de incumplimiento de las restricciones que ocurran. Veamos a continuación la descripción de los componentes fundamentales.

Algoritmo 1 Vista general de un algoritmo genético

```
t = 0;
InicializarPoblacion P (t); // inicializar aleatoriamente la población inicial
Evaluar P (t); // evaluar todos los individuos de la población
Mientras no se cumpla Condición de Parada hacer
    t = t + 1;
    P'(t) = SeleccionarPadres P(t); // seleccionar padres
    Recombinar P'(t); // recombinar los "genes" de los padres seleccionados con algún operador de cruzamiento
    Mutar P'(t); // alterar el material genético aleatoriamente, modificando los individuos de forma independiente
    Evaluar P'(t);
```

$P(t+1) = \text{Reemplazo } P(t), P'(t);$ // aplicar estrategia de reemplazo seleccionando los sobrevivientes
fin del ciclo

Forma de representación de la información: para los valores de los genes se empleó una representación binaria $x_{ijrt} \in \{0; 1\}$. De esta forma cada gen representa que en el proyecto $i \in I$ la actividad $j \in J_{ik}$, es ejecutada por el recurso $r \in R_k$ y comienza en el momento t .

Diseño de los cromosomas: los genes de cada cromosoma representan a los múltiples proyectos ordenados por su prioridad. Cada gen es un arreglo multidimensional de tres dimensiones, donde cada escaque identifica a una tarea, el momento en que se realiza y el recurso asociado a la misma. Es importante señalar que, entre las tareas siempre existe una holgura de espera² que denotaremos por he_{ij} indicando la diferencia entre los momentos en que se ejecutan la tarea i y la j siempre que la tarea i sea predecesora de tarea j .

Definición de la población: la población está formada por diferentes combinaciones de formas diferentes de organizar los proyectos en análisis. Se establecerá tamaño fijo de la población y poblaciones de 50 cromosomas. En la población inicial se siembran un conjunto de posibles soluciones de planificación de los proyectos generadas por expertos humanos.

Definición de la estrategia de reemplazo: basada en truncamiento donde solo lo mejores individuos pasan a la siguiente generación. Este método de reemplazo mantiene poblaciones del mismo tamaño.

Definición de la condición de parada: se considera condición de parada un número fijo de generaciones.

² Holgura de espera representa la diferencia de tiempo entre el fin de una actividad y el inicio de la siguiente.

Método de evaluación: evalúa los individuos considerando la función objetivo que se describe en la

$$\min Z_1 = Z + \frac{1}{2|C|} \sum_{q=1}^{|C|} v_{c_q} \bar{Z}, \quad i \in I, j \in A_{ih}, h, A_{ih} \in J_{ik}, k \in K, c_q \in C, |C| = 4$$

$$Z = \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T w_i t x_{ijrt}$$

$$v_{c_1} = \frac{\left| 1 - \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right|}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right)},$$

$$v_{c_2} = \frac{\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt}}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt} \right)}$$

$$v_{c_3} = \frac{\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k|}{\max_{i \in I} \left(\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k| \right)},$$

$$v_{c_4} = \begin{cases} 0 & \text{si } \sum_{r \in R_k} \sum_{t \in T} t x_{ijrt} \geq TPStart_Date_i, j = 1 \\ 1 & \text{eoc.} \end{cases}$$

(6 que

relaciona la evaluación de los cromosomas según la función objetivo definida con la penalización de los individuos que no cumplen con las restricciones.

$$\min Z_1 = Z + \frac{1}{2|C|} \sum_{q=1}^{|C|} v_{c_q} \bar{Z}, \quad i \in I, j \in A_{ih}, h, A_{ih} \in J_{ik}, k \in K, c_q \in C, |C| = 4$$

$$Z = \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T w_i t x_{ijrt}$$

$$v_{c_1} = \frac{\left| 1 - \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right|}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right)},$$

$$v_{c_2} = \frac{\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt}}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt} \right)}$$

$$v_{c_3} = \frac{\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k|}{\max_{i \in I} \left(\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k| \right)},$$

$$v_{c_4} = \begin{cases} 0 & \text{si } \sum_{r \in R_k} \sum_{t \in T} t x_{ijrt} \geq TPStart_Date_i, j = 1 \\ 1 & \text{eoc.} \end{cases}$$

En la

(6,

\bar{Z} representa la media de los valores de la expresión Z en la población. Z es empleada para evaluar a cada uno de los

cromosomas. Se presentan además las expresiones v_{c_q} que representan a cada una de las restricciones del problema. Nótese, que la función objetivo empleada relaciona en una única expresión los valores de la función objetivo Z con las cuatro restricciones descritas en el problema de optimización.

En el caso que compete hemos optado por la penalización por la complejidad del problema en cuestión y de encontrar soluciones completamente correctas y que además garanticen un uso cuasi óptimo u óptimo de los recursos. Es interesante resaltar que, en la propuesta las restricciones están incorporadas de forma tal que penalizan de forma dinámica cambiando los valores en función de la media de los valores de la evaluación de la población respecto a la expresión Z.

Definición del operador de selección: se aplica el método de la ruleta aportando potencia y robustez a la búsqueda, selecciona las mejores combinaciones en la organización de tareas y recursos disponibles.

Definición del operador de cruzamiento: permite el cambio de código genético entre los individuos introduciendo nuevas combinaciones de genes. Se presentan varias variantes de este operador.

Variante operador de cruzamiento basado en el cruce en un punto, horizontal: se selecciona un punto de cruce entre los cromosomas y se intercambian los genes. O sea, el punto de cruce representa un corte horizontal separando proyectos diferentes de la misma población. Luego, se intercambian la planificación de proyectos completos.

$$\min Z_1 = Z + \frac{1}{2|C|} \sum_{q=1}^{|C|} v_{c_q} \bar{Z}, \quad i \in I, j \in A_{ih}, h, A_{ih} \in J_{ik}, k \in K, c_q \in C, |C| = 4$$

$$Z = \frac{1}{n} \sum_{i=1}^n \sum_{t=1}^T w_i t x_{ijrt}$$

$$v_{c_1} = \frac{\left| 1 - \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right|}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} x_{ijrt} \right)}, \quad v_{c_2} = \frac{\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt}}{\max_{i \in I} \left(\sum_{r \in R_k} \sum_{t \in T} (t - d_{ijr}) x_{ijrt} - \sum_{r \in R_k} \sum_{t \in T} t x_{ihrt} \right)} \quad (6)$$

$$v_{c_3} = \frac{\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k|}{\max_{i \in I} \left(\sum_{i \in I} \sum_{j \in J_{ik}} \sum_{r \in R_k} \sum_{t \in T} x_{ijrt} - |R_k| \right)}, \quad v_{c_4} = \begin{cases} 0 & \text{si } \sum_{r \in R_k} \sum_{t \in T} t x_{ijrt} \geq TPStart_Date_i, j = 1 \\ 1 & \text{eoc.} \end{cases}$$

Variante operador de cruzamiento basado en dos puntos vertical: centrado en intercambio de recursos: esta variante corresponde con una variante tradicional donde se seleccionan dos puntos de cruce entre los cromosomas y se intercambian los recursos asociados a las tareas comprendidas entre estos dos momentos.

Variante operador de cruzamiento heurístico basado en intercambio de recursos: en este caso se seleccionan los padres a cruzar y se seleccionan todos los hitos de ejecución que son comunes a dos o más proyectos. Se forman pares con los hitos de ejecución análogos, o sea, que son los mismos en los dos cromosomas. Se construye un nuevo cromosoma hijo que tiene los mejores recursos posibles ubicados en los hitos comunes.

Definición del operador de mutación: provoca la posible aparición de nuevas características en el individuo al mutar uno de sus genes. La probabilidad de que este proceso ocurra es baja asegurando de esta forma, que no se convierta en sobreactivo y destruya las ventajas del cruzamiento. Se proponen varias variantes de este operador.

Variante mutación basada en la reasignación de recursos: en la misma los cromosomas seleccionados son sometidos a un proceso interno que comienza con la selección aleatoria de un punto en el tiempo de vida de los proyectos. En este punto se toman las tareas dos a dos y se intercambian sus recursos. La reasignación de recursos puede provocar que aumente o disminuya el tiempo de ejecución de las mismas.

Variante mutación basada en la eliminación de holguras: en esta implementación los cromosomas seleccionados son sometidos a un proceso que elimina las holguras de espera de las tareas siempre que no se sobrecarguen los recursos asignados a las mismas en más de un 5%.

Resultados y discusión

Para la validación de investigación se emplearon dos bases de datos. Se emplea la base de datos PSPLib publicada en (PSPLib, 2016) contiene conjuntos de datos asociados al problema RCPSP. Aunque este no es exactamente el problema que se abordó en la investigación se empleó esta base de datos para poder comparar los resultados del algoritmo propuesto con otras bases de datos. El problema RCPSP puede ser resuelto también con el algoritmo genético propuesto con la salvedad de que los cromosomas están compuestos por un único proyecto, que es un caso particular del problema que resuelve nuestra propuesta.

Se empleó en la experimentación la base de datos de “Asignación de Recursos y Tiempo” (GESPRO_Scheduling) del repositorio para investigaciones de proyectos terminados del Laboratorio de Investigaciones en Gestión de Proyectos que contiene 9315 tareas agrupadas en 88 proyectos y con 4 tipos de recursos. En la misma se pueden identificar el

problema RCMPSP que es precisamente el que queremos resolver. Se comparan los resultados de los algoritmos para diferentes corridas del algoritmo propuesto variando los operadores de cruzamiento y mutación.

Experimentación empleando PSPLIB

Se realiza la prueba con las instancias J30 (30 tareas y 41 recursos), J60 (60 tareas y contiene 49 recursos) y J120 (120 tareas y contiene 48 recursos) públicas en PSPLib. En estas bases de datos se emplean cuatro tipos de recursos, se puede hacer una analogía entre estos y las competencias esenciales en el desarrollo de proyecto de software como explicamos con anterioridad.

Se muestra a continuación los resultados de corrida del algoritmo propuesto tomando como operador de cruzamiento: el operador de cruzamiento basado en dos puntos vertical y como operador de mutación: la mutación basada en la reasignación de recursos. Se escogen para esta prueba estos dos operadores por la aplicabilidad a las bases de datos de PSPLIB.

Para la instancia J30 1-1 el algoritmo converge a 45 unidades de tiempo. Un valor muy cercano al 43 propuesto en PSPLib como mejor solución para este caso. Además, las mejores soluciones se encuentran para poblaciones de 40 o más individuos, después de las 25 generaciones.

Para la instancia J60 1-1 el algoritmo obtiene como mejor valor 77 unidades de tiempo, que coincide con la mejor solución propuesta por PSPLib. Las mejores soluciones se obtienen con poblaciones iniciales de 80 o más individuos, a partir de las 25 generaciones.

Se muestran a continuación los resultados de la corrida del algoritmo propuesto tomando en consideración que cada cromosoma de la población está formado por un solo proyecto.

Tabla 1 Resultados con las bases de datos J30 1-1, J60 1-1 y J120 1-1

Generaciones/ tamaño población	Pruebas J30					Pruebas j60					Pruebas J120						
	20	40	60	80	100	20	40	60	80	100	20	40	60	80	100	120	140
25	55	50	53	52	46	83	83	94	92	78	146	141	144	143	136	134	126
50	55	47	51	49	45	90	83	82	83	77	143	138	143	141	139	136	118
75	54	50	47	53	47	89	86	83	80	78	149	140	145	138	131	126	109
100	53	53	45	49	45	89	87	90	78	77	140	144	141	137	134	115	114

Para la instancia J120 1-1, el algoritmo obtiene como mejor valor 109 unidades de tiempo. Un valor cercano al 105 propuesto en PSPLib como mejor solución para este caso. Las mejores soluciones se obtienen con poblaciones iniciales de 140 individuos, y con 75 generaciones o más.

Experimentación empleando base de datos GESPRO_Scheduling

Se realiza la prueba considerando 5 proyectos simultáneos con 502 tareas con 100 tareas como promedio cada uno y con 21 recursos agrupados en cuatro tipos fundamentales orientados a las competencias en el desarrollo de software. Se conforman un conjunto de diseños diferentes del algoritmo genético propuesto, obtenidos a partir de la combinación de los operadores como muestra la Tabla 2.

Tabla 2. Diferentes diseños del algoritmo genético propuesto empleados en la prueba.

Diseño de Algoritmo Genético	Operador Cruzamiento	Operador Mutación
1Hor.Res	Cruce en un punto, horizontal	Reasignación de recursos
2Hor.Hol		Eliminación de holguras
3Ver.Res	Cruce en dos puntos vertical	Reasignación de recursos
4Ver.Hol		Eliminación de holguras
5Heu.Res	Cruce heurístico basado en intercambio de recursos	Reasignación de recursos
6Heu.Hol		Eliminación de holguras

Empleando cada diseño se desarrollaron 8 corridas todas ellas con 200 generaciones y poblaciones de tamaño 80 individuos, y se guardaron los mejores tres resultados de cada corrida. Se compararon los resultados de la media del tiempo de duración del proyecto en días y el máximo número de días que dura la ejecución conjunta de los 5 proyectos. Se realizó un análisis descriptivo empleando el comando *summary*, ver Tabla 3.

Para la comparación de los resultados de las corridas se aplicó el test de *Wilcoxon* sobre la plataforma de análisis de datos (Paquete R, 2016) para la comparación de los resultados de diferentes algoritmos, la Tabla 4 muestra los resultados obtenidos.

Tabla 3 Análisis descriptivo, resultados desde R comando *summary*.

Algoritmos y variables	Min	Median	Mean	Max
X1Hor.Res.MediaT	157.0	161.0	161.6	197.0
X1Hor.Res.MaxT	189.0	199.5	200.2	215.0
X2Hor.Hol.MediaT	157.0	159.0	160.6	196.0
X2Hor.Hol.MaxT	190.0	195.5	196.0	204.0
X3Ver.Res.MediaT	144.0	149.0	150.6	189.0
X3Ver.Res.MaxT	175.0	187.0	187.2	201.0
X4Ver.Hol.MediaT	134.0	136.0	137.0	154.0
X4Ver.Hol.MaxT	147.0	155.0	155.6	167.0
X5Heu.Res.MediaT	135.0	137.0	138.2	159.0
X5Heu.Res.MaxT	155.0	161.0	160.8	169.0
X6Heu.Hol.MediaT	104.0	107.0	107.3	121.0
X6Heu.Hol.MaxT	115.0	124.0	124.0	133.0

Finalmente se forman los siguientes grupos de algoritmos ordenados en función de la calidad de los resultados: $Grupo1=\{X6Heu.Hol\}$, $Grupo2=\{X4Ver.Hol\}$, $Grupo3=\{X5Heu.Res\}$, $Grupo4=\{X3Ver.Res\}$ y $Grupo5=\{X2Hor.Hol, X1Hor.Res\}$. O sea, el algoritmo con mejores resultados fue $X6Heu.Hol$.

A pesar de que las soluciones encontradas son buenas, es importante resaltar en algunos de los casos que eran salida de los algoritmos como mejores soluciones, ya que no se cumplían completamente todas las restricciones del problema. Se recomienda que el proceso de planificación sea un proceso asistido por los algoritmos y las herramientas computacionales, pero verificado y ajustado por expertos humanos especialistas en gestión de proyectos. Se considera que esta temática es un campo abierto a la investigación donde se pueden presentar disímiles situaciones por las cuales no hay suficientes soluciones en la bibliografía.

Tabla 4 Resultados de la comparación de algoritmos con R. (*Wilcoxon signed rank test with continuity correction*)

Comparando	Variable Media de duración de los cinco proyectos	Variable máximo tiempo de duración de proyecto
X6Heu.Hol vs X4Ver.Hol	data: X6Heu.Hol.MediaT and X4Ver.Hol.MediaT V = 0, p-value = 1.797e-05 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)	data: X6Heu.HolMaxT and X4Ver.HolMaxT V = 0, p-value = 1.906e-05 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas.)
X5Heu.Res vs X4Ver.Hol	data: X5Heu.Res.MediaT and X4Ver.Hol.MediaT V = 184.5, p-value = 0.01634 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)	data: X4Ver.HolMaxT and X5Heu.ResMaxT V = 27.5, p-value = 0.0008016 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)
X5Heu.Res vs X3Ver.Res	data: X5Heu.Res.MediaT and X3Ver.Res.MediaT V = 0, p-value = 1.875e-05 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)	data: X5Heu.ResMaxT and X3Ver.ResMaxT V = 0, p-value = 1.895e-05 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)
X2Hor.Hol vs X3Ver.Res	data: X2Hor.Hol.MediaT and X3Ver.Res.MediaT V = 276, p-value = 2.753e-05 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)	data: X2Hor.HolMaxT and X3Ver.ResMaxT V = 235, p-value = 0.0004509 alternative hypothesis: true location shift is not equal to 0. (Se encuentran diferencias significativas)
X2Hor.Hol vs X1Hor.Res	data: X2Hor.Hol.MediaT and X1Hor.Res.MediaT V = 70.5, p-value = 0.06808 alternative hypothesis: true location shift is not equal to 0. (No se encuentran diferencias)	data: X2Hor.HolMaxT and X1Hor.ResMaxT V = 59.5, p-value = 0.01754 alternative hypothesis: true location shift is not equal to 0. (No se encuentran diferencias)

	significativas)	significativas)
--	-----------------	-----------------

Conclusiones

El problema de planificación de proyectos de desarrollo de software es un caso particular de los problemas en general de planificación de proyectos. Pero por lo general en las organizaciones orientadas a proyectos el problema que se presenta es el problema de la planificación de múltiples proyectos que se ejecutan simultáneamente.

Los algoritmos genéticos son técnicas factibles a ser aplicadas en la resolución de problemas de planificación de proyectos por su flexibilidad, solo utilizan información de la función objetivo, sin necesidad de conocimiento previo como derivadas u otro conocimiento auxiliar. Además, permiten la resolución de problemas multiobjetivos de forma simultánea.

En general la aplicación de los algoritmos genéticos en el problema de planificación de múltiples proyectos que se ejecutan simultáneamente arrojó buenos resultados, pero se recomienda seguir profundizando en la solución a este problema por su complejidad (NH-duro) y su aplicabilidad en la gestión de organizaciones orientada a proyectos.

En la propuesta se adopta un método basado en penalizaciones porque es complejo obtener soluciones completamente correctas por la complejidad del problema de optimización.

A pesar de que las soluciones encontradas son buenas, es importante resaltar en algunos de los casos que eran salida de los algoritmos como mejores soluciones no se cumplían completamente todas las restricciones del problema. Se recomienda que el proceso de planificación sea un proceso asistido por los algoritmos y las herramientas computacionales, pero verificado y ajustado por expertos humanos especialistas en gestión de proyectos.

Referencias

ABDEL-HAMID, T. K. A multiproject perspective of single-project dynamics. *Journal of Systems and Software*, 1993, 22(3): p. 151–165.

AMOUI, M.; DERAKHSHANMANESH, M., et al. Achieving dynamic adaptation via management and interpretation of runtime, models. *The Journal of Systems and Software*, 2012, 85, p. 2720–2737.

BEOVIDES, S. Algoritmo genético para optimizar la planificación del cronograma de un proyecto. Laboratorio de Gestión de Proyectos, Universidad de las Ciencias Informáticas. La Habana, 2015.

BOLIVAR, E. Planificación de múltiples proyectos de desarrollo de software utilizando métodos metaheurístico. Memoria para optar por el título de: Máster en Ciencias Matemáticas. Facultad de Matemática y Computación. Universidad de La Habana, 2013.

BRUCKER, P.; DREXL, A., et. al, Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 1999, 112(1): p. 3–41.

BRUNI, M. E.; BERALDI, P., et. al. A heuristic approach for resource constrained project scheduling with uncertain activity durations. *Computers & Operations Research*, 2011, 38(9): p. 1305–1318. Disponible en: <http://dx.doi.org/10.1016/j.cor.2010.12.004>

CHEN, W.; SHI, Y., et. al. An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences*, 2010 180(6): p. 1031–1039. Disponible en: <http://dx.doi.org/10.1016/j.ins.2009.11.044>

CHEN, W. N.; ZHANG, J. Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler. *IEEE Transactions on Software Engineering*, doi: 10.1109/TSE.2012.17, 2013, 39(1): p. 1-17.

CHICANO, F.; LUNA, F., et. al. Using multi-objective metaheuristics to solve the software project scheduling problem. *GECCO '11 Proceedings of the 13th annual conference on Genetic and evolutionary computation*, New York, USA, ISBN: 978-1-4503-0557-0 doi>10.1145/2001576.2001833, 2011, p. 1915-1922.

DELGADO, R. La Dirección Integrada de Proyecto como Centro del Sistema de Control en el Ministerio del Poder Popular para la Comunicación y la Información. La Habana, Cuba: Futuro, 2013.

DEMEULEMEESTER, E. L.; HERROELEN, W. S. *Project scheduling, a Research Hand- book*. New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers, 2002, 1st ed., p. 12-26.

DI PENTA, M.; HARMAN, M.; ANTONIOL, G. The use of search-based optimization techniques to schedule and staff software projects: an approach and an empirical study. *Software: Practice and Experience*, 2011, 41(5): p. 495–519.

GONSALVES, T.; ITO, A., et. al. Swarm intelligence in the optimization of software development project schedule, 2008, p. 587–592.

GONSALVES, T.; ITOH, K. Multi-objective optimization for software development projects. *Lecture Notes in Engineering and Computer Science*, 2010, vol. 2180, p. 1–6.

GONZALVES, J. F.; RESENDE, M. G.; MENDES, J. J. A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. *Journal of Heuristics*, 2011, 17(5): p. 467–486.

HEIMERL, C.; KOLISCH, R. Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectrum*, 10.1007/s00291-009-0169-4, 2010, vol. 32, p. 343–368.

KEA, H.; LIUB, B. Fuzzy project scheduling problem and its hybrid intelligent algorithm. *Applied Mathematical Modelling*, 2010, 34(2): p. 301–308. Disponible en: <http://dx.doi.org/10.1016/j.apm.2009.04.011>.

KOLISCH, R.; HARTMANN, S. Experimental investigation of heuristics for resource - constrained project scheduling: An update. *European Journal of Operational Research*, 2006, 174(1): p. 23–37.

MASMOUDIA, M.; HAÏTB, A. Project scheduling under uncertainty using fuzzy modeling and solving techniques. *Engineering Applications of Artificial Intelligence*, 2013, 26(1): p. 135–149.

MOSSALAM, A.; ARAFA, M. The role of project manager in benefits realization management as a project constraint/driver. *Housing and Building National Research Center, HBRC Journal*, 2014, p. 56-67.

PINEDO, M. L. *Scheduling, Theory, Algorithms, and Systems*. New York, Fifth Edition, Springer, ISBN: 978-3-319-26578-0 (Print) 978-3-319-26580-3 (Online). DOI 10.1007/978-3-319-26580-3, 2015, p. 459 - 505. Disponible en <http://www.stern.nyu.edu/mpinedo>.

PIÑERO, P.; TORRES, S. Paquete para la Dirección Integrada de Proyectos y la ayuda a la toma de decisiones. II Taller internacional las TIC en las Organizaciones, XV Convención y Feria Internacional, Informática 2013, ISBN 978-959-7213-02-4.

PROJECT MANAGEMENT INSTITUTE. *A Guide to the Project Management Body of Knowledge Fifth. Edition*. Estados Unidos de América: Project Management Institute. 2013.

PROJECT MANAGEMENT INSTITUTE. *PMI's Pulse of the Profession: The High Cost of Low Performance*. New York: Project Management Institute. 2014.

PSPLib, Scheduling projects library. Online 2016, Disponible en: <http://www.om-db.wi.tum.de/psplib>

R. Paquete estadístico de R. Online 2016, Disponible en: <http://cran.r-project.org/>

SANTOS, M.; TERESO, A. On the multi-mode, multi-skill resource constrained project scheduling problem – a software application. Springer Berlin / Heidelberg. In Soft Computing in Industrial Applications, 2011, vol.96, p. 239–248.

STANDISH GROUP INTERNATIONAL. The CHAOS Report. New York: The Standish Group International, Inc., 2015.

VILLAVICENCIO, N. Modelo integrado para la mejora de la productividad en organizaciones orientadas a proyectos de tecnologías de la información. Tesis para optar al grado de: Máster en Diseño, Gestión y Dirección de Proyectos. Fundación Universitaria Iberoamericana – FUNIBER, 2016.