

MANEJO DE RESTRICCIONES CON FUNCIONES DE PENALIDAD PARA LA PLANIFICACIÓN DEL MANTENIMIENTO DE LOCACIONES PETROLERAS

Ing. J. Valdez, AdeS. D. Pandolfi y Mg. A. Villagra,
Docentes Investigadores de la UNPA,
{jvaldez,dpandolfi,avillagra}@uaco.unpa.edu.ar

2011

Universidad Nacional de la Patagonia Austral
Unidad Académica Caleta Olivia
Departamento de Ciencias Exactas y Naturales
Caleta Olivia

Resumen

La explotación y transporte de petróleo son actividades muy importantes para el desarrollo económico de la sociedad moderna industrial. Sin embargo, estas actividades son generadoras de riesgos que se traducen en contaminaciones que afectan el medio ambiente. Por esta razón, es importante que las empresas petroleras lleven a cabo un correcto mantenimiento de sus locaciones. En casos de planificación de mantenimiento de 100 o más locaciones, PAE es una herramienta que provee un plan de visitas para el mantenimiento, usando un algoritmo evolutivo para producir múltiples soluciones a este problema con restricciones. En este trabajo, se implementa una herramienta de optimización evolutiva aplicando técnicas de manejo de restricciones basadas en funciones de penalidad. Detalles de la implementación, resultados experimentales y beneficios son presentados en este informe.

Palabras Claves: Metaheurísticas, Planificación, Manejo de Restricciones, Funciones de Penalidad.

1. Introducción

Muchos problemas de optimización incluyen ciertos tipos de restricciones que constituyen grandes desafíos para resolverlos apropiadamente. Estas restricciones por lo general, son límites impuestos a variables de decisión. Además las restricciones son parte integral de la formulación de cualquier problema [Coello Coello C.,2002] y cualquier situación donde deba tomarse una decisión involucra restricciones. Lo que distingue a diversos tipos de problemas es la forma de estas restricciones (reglas, dependencia de datos, expresiones algebraicas, y otras formas).

Los Algoritmos Evolutivos (AEs) han sido aplicados exitosamente en la resolución de problemas de optimización en el campo de la ciencia y la ingeniería [Goldberg D.E., 1989]. Los AEs fueron originalmente diseñados para resolver problemas de optimización sin restricciones, pero los últimos años, los investigadores han incorporado técnicas de manejo de restricciones en estos algoritmos. Los grandes desafíos en problemas de optimización con restricciones surgen de los límites variados en las variables de decisión, los tipos de restricciones involucradas, la interferencia y la interrelación entre las restricciones y la función objetivo [Venkatraman S. et. al., 2005].

Hace pocos años atrás, se han propuesto varios métodos para el manejo de restricciones para los AEs para problemas de optimización. Estos métodos han sido agrupados por Michalewicz y otros investigadores en cuatro categorías [Michalewicz, Z. y Schoenauer, M. 1996], [Koziel, S. y Michalewicz, Z. 1999], [Coello Coello C. 2002]:

- *Métodos basados en preservar la factibilidad de las soluciones.* La idea de estos métodos esta basada en la especialización de los operadores que transforman padres factibles en hijos factibles. Estos métodos asumen únicamente restricciones lineales o comienzan desde un punto de comienzo factible o desde una población inicial factible.
- *Métodos basados en funciones de penalidad.* Varios algoritmos evolutivos incorporan métodos de manejo de restricciones basados en el concepto de funciones de penalidad exterior que penalizan soluciones no factibles. Estos métodos difieren en importantes detalles tales como, la forma en que se diseña la función de penalidad y como esta es aplicada a las soluciones no factibles.
- *Métodos que hacen una clara distinción entre soluciones factibles y soluciones no factibles.* Estos métodos enfatizan las diferencias entre soluciones factibles y no factibles dentro del espacio de búsqueda. Uno de estos métodos distingue entre individuo factibles e individuos no factibles $y : f(x) < f(y)$, es decir, que cualquier solución factible es mejor que cualquier otra no factible.
- *Otros métodos híbridos.* Estos métodos combinan técnicas de computación evolutiva con procedimientos determinísticos para problemas de optimización numérica.

Una forma muy común de incorporar el manejo de restricciones en AEs es a través de las funciones de penalidad, propuestas inicialmente en [Courant R. 1943],[Carroll C. W. 1961] y [Fiacco A. V. and McCormick G. P, 1968.]. Se pueden considerar dos tipos de funciones de penalidad: exterior e interior. El método más común utilizado en AEs es el enfoque de penalidad exterior. Y es el que se utilizará en este trabajo.

Los AEs con enfoques mulirecombinativos [Eiben A.E. y Bäck Th. 1997.] y enfoques con múltiples padres [Esquivel S., et. al. 1997] aplicados a la resolución de diversos problemas de planificación y routing se han convertido en estrategias exitosas. Particularmente, en problemas de planificación agregando una nueva variante al enfoque mutirecombinativo llamada MCMP-SRI (*Multiple Crossover, Multiple Parents - Stud and Random Immigrants*)[Pandolfi D. et. al. 2002].

En los últimos años, las empresas dedicadas a la explotación, producción y transporte del petróleo, han visto la necesidad de implementar medidas de prevención a fin de evitar y/o minimizar los daños ocasionados al medio ambiente, personas y bienes materiales. Una forma de atacar el problema de contaminación por petróleo es prevenir el incidente. Normalmente éstos se producen por fallas de equipos o del material y fallas humanas. Los primeros pueden ser subsanados mediante inspecciones periódicas y un mantenimiento adecuado; y los segundos, mediante la instrucción y el entrenamiento del personal en forma permanente. La minimización del tiempo entre cada período de inspección de las locaciones se obtiene con una mejora en la planificación del mantenimiento y esto puede ser abordado como un problema de planificación (scheduling). Se ha demostrado, que muchos problemas de scheduling pertenecen a la clase NP-hard [Brucker P. 2004.] reflejando así la relevancia industrial de este tipo de problemas.

Nuestra herramienta denominada PAE [Villagra A. et.al. 2008] (Planificador basado en un Algoritmo Evolutivo) es una herramienta basada en un enfoque evolutivo, tiene por objetivo la planificación de las visitas de mantenimiento de tal forma que: (a) Minimice el tiempo total de visitas, es decir, encuentre la planificación que en menor tiempo recorra las locaciones incluyendo el tiempo de intervención en cada una de las mismas. (b) Replanifique las visitas a partir de desviaciones en la planificación original. Frente a la ocurrencia de eventos que condicionan operativamente la ejecución de un plan de mantenimiento, proveer de planificaciones alternativas sin disminuir significativamente la calidad de las mismas. (c) Obtener soluciones que cumplimenten las restricciones impuestas al problema.

Particularmente en este trabajo se muestran los resultados obtenidos de la aplicación de enfoques de funciones de penalidad al problema con restricciones del mantenimiento de locaciones petroleras.

2. Marco Teórico

2.1. Dominio y descripción del problema

Las compañías petroleras llevan a cabo visitas de mantenimiento o de prevención para cada uno de sus locaciones petroleras (pozos productores, inyectoras, baterías y colectores). Un yacimiento petrolero esta formado por baterías, cada batería contiene cerca de 20 pozos petroleros. Cada pozo posee diferentes niveles de producción conocido a priori y estos varían en el tiempo. Esta producción define la categoría y el número de veces que sera visitada en el mes. Los pozos de petróleo no se pueden visitar más de una vez en el mismo turno y en función de su tipo es que hay algunas tareas que se llevarán a cabo. Cada tarea tiene su equipo necesario, una frecuencia de realización y un tiempo aproximado de duración. Actualmente la ruta es llevada a cabo por el equipo encargado de las visitas de mantenimiento programado es en base a la experiencia del

operario. Un día de trabajo comienza por la mañana y las locaciones petroleras son atendidas en dos turnos de tres horas. Después de un turno el equipo debe regresar a la base para llevar a cabo algunas tareas administrativas y luego continuar con el siguiente turno. El tiempo exigido en cada pozo de petróleo dependerá de su tipo. De vez en cuando, hay locaciones de petróleo que, por diversas causas se deben visitar más de una vez (dos o tres veces) en diferentes turnos, y estas son las limitaciones impuestas a este problema. En la Figura 1 se muestra una distribución de locaciones petroleras del yacimiento explotado en la zona norte de la provincia de Santa Cruz, Argentina.

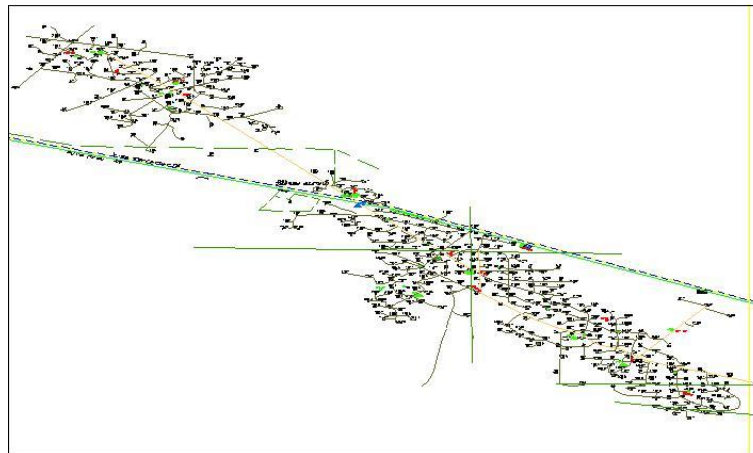


Figura 1: Mapa de ruta y distribuciones.

2.2. Algoritmos Evolutivos

En la naturaleza, los individuos deben adaptarse a su ambiente a fin de sobrevivir a un proceso llamado evolución, en el cual aquellas características que hacen que un individuo sea más apto para competir, se preservan cuando el individuo se reproduce y aquellas características que lo hacen más débil son eliminadas. Estas características son controladas por unidades llamadas “genes” y estos son los que forman conjuntos llamados “cromosomas”. Luego de subsiguientes generaciones no solamente sobreviven los individuos más aptos, sino también los genes aptos que son transmitidos a sus descendientes durante el proceso de recombinación sexual llamado “crossover”. Las primeras analogías entre el mecanismo de selección natural y un proceso de aprendizaje (u optimización) llevó al desarrollo de los llamados ‘Algoritmos Evolutivos’ (AEs) [Bäck T. 1996], en los que el principal objetivo es simular el proceso evolutivo en una computadora. Existen tres principales paradigmas dentro de los AEs, cuyas motivaciones y orígenes son independientes entre sí: las Estrategias de Evolutivas [Schwefel H. 1981], Programación Evolutiva [Fogel L. 1966] y Algoritmos Genéticos [Holland J., 1975].

En general, necesitamos los siguientes componentes básicos para implementar un AE para resolver un problema:

- una representación de las posibles soluciones al problema,
- una forma de crear la población inicial de posibles soluciones,

- una función de evaluación que hace el papel del medio ambiente, las soluciones de clasificación en términos de su "fitness",
- un procedimiento de selección de los padres que se reproducirán,
- los operadores utilizados para la evolución, que alteran la composición de los hijos (normalmente de cruce y mutación),
- los valores de varios parámetros que utiliza el algoritmo evolutivo (tamaño de la población, probabilidades de la aplicación de operadores, etc.)

Los Algoritmos Evolutivos han sido aplicados exitosamente en un amplio rango de aplicaciones [Richardson et al 1989, Goldberg D., 1989., Mitchell M., 1996], sin embargo, un aspecto que normalmente no se considera cuando se los usa para optimización, es que estos algoritmos son procedimientos de optimización sin restricciones y por esta razón es necesario encontrar mecanismos de incorporación de restricciones (que normalmente existen en cualquier aplicación del mundo real) en la función de aptitud.

2.3. Formulación del problema

El problema puede ser definido como se declara en [Pinedo, 1995]:

$$1|S_{jk}|C_{max} \quad (1)$$

Denota un problema de scheduling de máquina única con n tareas sujetas a tiempos de preparación dependientes de la secuencia. Donde las tareas a planificar son el servicio de mantenimiento (o intervención) en cada una de las locaciones petroleras.

El objetivo es minimizar el makespan (C_{max}) sujetos a los tiempos que dependen de la preparación de la secuencia. Este modelo conduce a un problema de optimización que es NP-Hard [Pinedo, 1995]. El makespan puede calcularse como:

$$\sum_{k=1}^n (S_{jk} + t_k) \quad (2)$$

donde S_{jk} representa el costo (en tiempo) de ir del pozo j al pozo k , t_k el respectivo tiempo de mantenimiento en la locación k , y n el número total de locaciones petroleras en el yacimiento.

Sean $OW1 \subset \{1, \dots, N\}$ y $OW2 \subset \{1, \dots, M\}$ dos subconjuntos de todas las locaciones petroleras en el yacimiento petrolero donde $OW1 \cap OW2 = \emptyset$. $OW1$ representa las locaciones de petróleo que deben ser visitadas dos veces y $OW2$ representa las locaciones de petróleo que deben ser visitadas tres veces. Además, las locaciones petroleras en $OW1$ o en $OW2$ no se pueden programar en el mismo turno. Una solución que no cumpla las condiciones anteriores se considera una solución no factible.

Como se explica en otras secciones, la introducción de las restricciones antes mencionadas afectarán el diseño del AE, ya que es necesario considerar estas restricciones para resolver este problema.

2.4. Preparación de datos

Para resolver el problema fue necesario preparar los datos de entrada, ya que originalmente las distancias entre las locaciones petroleras no estaban procesadas. Se realizó el cálculo de las distancias entre las locaciones petroleras basados en el plano de caminos y distribución del yacimiento. Matemáticamente, es sabido que la distancia entre dos puntos que se encuentran en cualquier lugar del sistema de coordenadas, está determinada por la relación denominada distancia euclídea. No obstante, en este problema sólo se puede calcular la distancia entre dos puntos, teniendo en cuenta el mínimo de caminos que existe para llegar a ellos. Por esta razón se utilizó el plano de las ubicaciones de las locaciones y se escalaron las distancias entre las mismas.

2.5. Diseño de PAE

Para instanciar el problema bajo fue necesario, desarrollar una adecuada codificación de las visitas a las locaciones petroleras que representa una solución al problema. Una planificación de visitas fue codificado en un cromosoma como una secuencia de pozos petroleros representados por números naturales. Por lo tanto, un cromosoma será una permutación $p = (p_1, p_2, \dots, p_n)$, donde n es la cantidad de locaciones petroleras que se visitarán. Cada elemento de p_i representa la locación i -ésima que será visitada de acuerdo a la secuencia de visitas.

Algoritmo 1 EA-MCMP-SRI

```
1:  $t = 0$  {generación actual}
2: inicializar Stud( $t$ )
3: evalúa Stud( $t$ )
4: mientras not max_evaluaciones hacer
5:   pool_apareamiento = Genera_Inmigrantes_Aleatorios  $\cup$  Selecciona (Stud( $t$ ))
6:   mientras not max_padres hacer
7:     mientras max_recombinaciones hacer
8:       evoluciona (pool_apareamiento){recombinación y mutación}
9:     fin mientras
10:  fin mientras
11:  evalúa (pool_apareamiento)
12:  Stud( $t+1$ ) = selec. nueva población del pool_apareamiento
13:   $t = t + 1$ 
14: fin mientras
```

En el Algoritmo 1 se presenta un esquema general de EA-MCMP-SRI, que se utiliza para resolver nuestro problema y se explica a continuación. El algoritmo crea una población inicial de soluciones al azar para la planificación, y estas soluciones son evaluadas. Después de eso,

la población se somete a un proceso iterativo multirecombinativo. El proceso para la creación de hijos se realiza de la siguiente manera. Los padres se aparean, entonces las parejas sufren un cruce y $2 \times n_2$ (n_2 máximo de padres) descendientes se crean. Lo mejor de esta descendencia $2 \times n_2$ se almacena en un pool de hijos temporal. La operación de cruce se repite n_1 veces (recombinaciones máximo) para cada punto de corte diferente, hasta que el pool de hijos se ha completado. Los hijos pueden o no someterse a mutación. Por último, los mejores hijos creados a partir de n_2 padres y n_1 cruce se insertan en la población. El operador de recombinación usado en este algoritmo fue PMX (*Partial Mapped Crossover*), propuesto por Goldberg and Lingle [D. Goldberg and R. Lingle. 1987]. Este puede ser visualizado como una extensión de cruce de dos cortes para cadenas binarias para la representación de permutación. Para la mutación se uso, la llamada *Swapping Mutation* (SM), que selecciona dos posiciones al azar e intercambia estos genes. El operador de selección usado para seleccionar un individuo fue la Selección Proporcional.

El cromosoma (solución) da el orden de la secuencia a seguir para visitar cada uno de las locaciones petroleras. Además, se tiene en cuenta que existen locaciones que deben ser visitadas más de una vez (de acuerdo con OW1 y OW2) lo que implica que muchas soluciones visitadas en el espacio de búsqueda serán no factibles. Por otra parte, para la versión con restricciones de nuestro problema, la longitud del cromosoma será $n = m + |OW1| + |OW2|$, es decir, $p = (p_1, \dots, p_m, p_{m+1}, \dots, p_n)$ donde m es la número total de locaciones petroleras y $|OW1| + |OW2|$ es el número de las locaciones petroleras que deben ser visitadas dos o tres veces. Por lo tanto, un $p_i \in m + 1, \dots, n$ decodifica en un número entero que pertenece a OW1 o OW2. Cuando una solución se evalúa se aplica una función de penalidad a las soluciones no factibles.

2.6. Manejo de Restricciones

En general, las restricciones son una parte integral para la formulación de cualquier problema. Una taxonomía útil de técnicas de manejo de restricciones incluye:

- (1) Funciones de penalidad,
- (2) representaciones espaciales y operadores,
- (3) algoritmos de reparación,
- (4) separación de objetivos y restricciones y
- (5) métodos híbridos.

Las funciones de penalidad disminuyen el fitness de las soluciones no factibles y tienen preferencia con las factibles en el proceso de selección. Una representación Especial y los operadores están diseñados para representar sólo las soluciones factibles y los operadores son capaces de preservar la viabilidad de la descendencia generada.

2.7. Funciones de penalidad

La forma más común de incorporar restricciones en un algoritmo evolutivo ha sido a través de funciones de penalidad [Schwefel P. 1995]. Teniendo en cuenta las conocidas dificultades asociadas con su utilización se han propuesto diversas formas de automatizar la definición de factores buenos de penalidad, lo cual continúa siendo una debilidad importante cuando se usan funciones de penalidad. Las funciones de penalidad fueron originalmente propuestas por Courant [Courant R. 1943] y luego fueron expandidas por Carrol [Carroll C. W. 1961] y Fiacco & McCormick [Fiacco A. V. and McCormick G. P, 1968.]. La idea de este método es transformar un problema de optimización con restricciones en uno sin restricciones agregando (o sustrayendo) un cierto valor a la función objetivo basado en la cantidad de violación la restricción presentada en una solución. Se pueden considerar dos tipos de funciones de penalidad: exterior e interior. En el caso de penalidad exterior, se comienza con una solución no-factible y desde ahí se mueve hacia una solución factible. En el caso de penalidad interior se inicia la búsqueda en la zona factible y se trata de evitar salir de la zona (hacia la región no-factible). A diferencia de la exterior, los puntos factibles para iniciar la búsqueda puede ser el “cuello de botella” de este enfoque. Idealmente, la penalidad debe mantenerse lo más baja posible, justo sobre el borde inferior donde las soluciones no-factibles son óptimas (a esto se lo denomina, regla de penalidad mínima [Davis L. 1987]. Esto se debe a que si la penalidad es muy alta o muy baja entonces podría convertirse en un problema muy dificultoso para un AE [Davis L. 1987]. Si la penalidad es muy alta y el óptimo se encuentra en el borde de una región factible, el AE puede ser empujado fuera de la región factible muy rápidamente y no podrá regresar. Una penalidad grande no favorece la exploración de una región no-factible. Por otro lado, si la penalidad es muy baja, se empleará mucho tiempo de exploración en regiones no-factibles pues la penalidad es insignificante con respecto a la función objetivo [Smith A. and Coit D. 1997]. Estos temas son de mucha importancia en AEs pues muchos de los problemas en los cuales se usan tienen los óptimos en los bordes de la región factible.

Las más importantes tipos de funciones de penalidades son:

Penalidades estáticas: en el que los factores de penalidad no dependerán del número actual de generación, y por lo tanto, se mantienen constantes durante todo el proceso evolutivo;

Penalidades dinámicas: se refiere a cualquier función de penalización en el que está involucrado el número de generación actual en el cálculo de los factores sanción correspondiente;

Penalidades annealing: basada en la idea de recocido simulado [S. Kirkpatrick, et. al. 1983], los coeficientes de penalización se cambian una vez en muchas generaciones, la pena mayor en el tiempo (es decir, la temperatura disminuye con el tiempo) para que los individuos no factibles son fuertemente penalizados en las últimas generaciones;

Penalidades adaptativas: también son penalidades dinámicas, se consideró por separado para una mayor claridad;

Penalidades mortales: se elimina la solución que no cumple con la restricción.

2.8. Funciones de penalidad consideradas

Para manejar restricciones en un AEs con funciones de penalidad, la función fitness $f(p)$ es usualmente transformada en $F(p) = f(p) + \mathcal{P}(p)$ (para una minimización del problema) donde $\mathcal{P}(p)$ es llamada la función de penalidad. En este trabajo se definieron dos funciones de penalidad:

1. Una función de penalidad que calcula el costo de la reparación de la solución no factible con respecto a los valores de fitness. Debe tenerse en cuenta que la solución no factible sólo es reparada con el propósito de evaluarla para determinar el valor de penalidad.

$$\mathcal{P}_1(p) = | f(p) - f(p') |$$

donde p es la solución no factible, $f_{(p)}$ representa el fitness de esta solución y $f_{(p')}$ la función fitness de p' (la solución reparada).

2. Consideremos una solución no factible $p = (p_1, \dots, p_n)$. Habrá componentes que no satisfagan las restricciones del problema. La siguiente función de penalidad $\mathcal{P}_2(p)$ considerará cada una de esas componentes de la siguiente manera:

$$\mathcal{P}_2(p) = 2 \times \sum_{h \in H} s_{hk}$$

donde H es el conjunto de locaciones petroleras en la solución p que no cumplen con la restricción del problema, h es una locación petrolera en particular, y s_{hk} representa el costo (en kilómetros) de pasar desde la locación petrolera h a la base de operaciones k .

De acuerdo con las funciones de penalidad básicas definidas \mathcal{P}_1 y \mathcal{P}_2 , se presentan los siguientes mecanismos de aplicación de funciones de penalidad basados en enfoques estático, dinámico, adaptativo, mortal y annealing que se detallan a continuación:

A. Penalidades Estáticas

Dos funciones de penalidad son definidas para este tipo:

- S1:** Consiste en aplicar la primera función de penalidad a la solución no factible. Es decir, se añade a la función de fitness el valor de penalización obtenido:

$$F(p) = f(p) + \mathcal{P}_1(p)$$

- S2:** Consiste en aplicar la segunda función de penalidad a la solución no factible. Es decir, se añade a la función de fitness el valor de penalización obtenido:

$$F(p) = f(p) + \mathcal{P}_2(p)$$

B. Penalidades Dinámicas

Para las penalidades dinámicas, la primera y la segunda función de penalidad (\mathcal{P}_1 and \mathcal{P}_2) son multiplicadas por el valor retornado por la siguiente función monótona creciente:

$$V(g) = \left(\frac{g}{G}\right)^2$$

donde g es la generación actual, G es el número total de generaciones y $0 \leq V(g) \leq 1$.

D1: $F(p) = f(p) + [\mathcal{P}_1(p) \times V(g)].$

D2: $F(p) = f(p) + [\mathcal{P}_2(p) \times V(g)].$

C. Penalidades annealing

Para las penalidades *annealing*, las dos funciones de penalidad (\mathcal{P}_1 and \mathcal{P}_2) se aplican basados en los principales conceptos que intervienen en el enfoque de *Simulated Annealing*.

Un valor de probabilidad de aceptación $Prob_A$ se calcula de la siguiente manera:

$$Prob_A = 1 - \left(\frac{g}{G}\right)^2$$

donde g es la generación actual y G el número total de generaciones. Empleando esta probabilidad de aceptación ($Prob_A$), las soluciones no factibles son aceptadas con una alta probabilidad en las primeras etapas. A medida que la población evoluciona esta probabilidad disminuye. Por lo tanto, las soluciones no factibles se penalizan con más frecuencia al final de la ejecución del algoritmo. Las dos funciones de penalidad de basadas en *annealing* son:

A1: Consiste en aplicar el enfoque de penalidad $S1$ a la solución no factible con una probabilidad $Prob_A$ y la función fitness extendida se define como:

$$F(p) = \begin{cases} f(p) + \mathcal{P}_1(p) & \text{if } r \leq Prob_A \\ f(p) & \text{en otro caso} \end{cases}$$

donde $r \in (0 \cdot 1)$.

A2: Consiste en aplicar el enfoque de penalidad $S2$ a la solución no factible con una probabilidad $Prob_A$ y la función fitness extendida se define como:

$$F(p) = \begin{cases} f(p) + \mathcal{P}_2(p) & \text{if } r \leq Prob_A \\ f(p) & \text{en otro caso} \end{cases}$$

donde $r \in (0 \cdot 1)$.

D. Penalidades mortales

Una solución no factible pasa por un procedimiento de búsqueda local (*Hill Climbing*)(Stadler, 1996). Si la solución obtenida es todavía no factible esta solución es rechazada y una solución factible (alguno de sus padres) es aceptada.

E. Penalidades adaptativas

Para el enfoque de adaptativo se propone un coeficiente de adaptación (ρ) que modifica la función de penalización. ρ se actualiza teniendo en cuenta el grado de violación de las restricciones. Para lograr este objetivo se propone una ventana de generaciones (fijado en 10 generaciones) se analiza y el valor de penalidad impuesta es incrementado o disminuido según corresponda.

$\rho \in (0, 2)$ y se calcula analizando las violaciones promedio en la ventana de generaciones llamada AV_w y las violaciones promedio generación actual llamada AV_c como se indica a continuación:

$$V_g = \frac{AV_w - AV_c}{AV_w} \times 100$$

Teniendo en cuenta el grado de violación V_g , ρ es actualizado de la siguiente forma:

$$\rho = \begin{cases} \rho + k & \text{if } V_g < -0,25 \\ \rho - k & \text{if } V_g > 0,25 \\ \rho & \text{en otro caso} \end{cases}$$

donde k se establece en 0.025.

AD1: $F(p) = f(p) + [\mathcal{P}_1(p) \times \rho]$.

AD2: $F(p) = f(p) + [\mathcal{P}_2(p) \times \rho]$.

3. Resultados

Para probar la performance del algoritmo evolutivo, EA-MCMP-SRI, utilizando los diferentes enfoques de penalidad se realizaron dos experimentos. Además se definieron instancias que pertenecen a los conjuntos *OW1* y *OW2*. La Tabla 1 resume las instancias definidas, donde la primer columna (Ins) es el nombre de la instancia, la segunda columna (OW1) muestra la cantidad de locaciones petroleras que deben ser visitadas dos veces y la última columna (OW2) representa la cantidad de locaciones petroleras que deben visitarse tres veces.

Para todos los experimentos llevados a cabo y para cada uno de los enfoques de penalidad aplicados se realizaron 30 corridas independientes. La velocidad se definió en 12 segundos cada 100 metros y el tiempo de mantenimiento en una locación se estableció en 300 segundos. La parametrización utilizada del algoritmo evolutivo fue la siguiente: se estableció la población a 15 individuos. La población inicial se generó aleatoriamente. El operador de recombinación PMX, (Partial Mapped Crossover) fue aplicado con una probabilidad de 0,65, mientras que el operador de mutación, SW (Swap Mutation), fue aplicado con una probabilidad de 0,05. El número n_1 de recombinaciones y n_2 de padres fueron fijados en 16 y 18, respectivamente. El número máximo de evaluaciones por generación se ha calculado como $n_1 \times (n_2 - 1) \times T_{\text{tamaño poblacion}}$.

Tabla 1: Instancias usadas en los dos experimentos para OW1 and OW2.

Ins	OW1	OW2
I_1	15	0
I_2	16	0
I_3	17	0
I_4	18	0
I_5	19	0
I_6	20	0
I_7	8	8
I_8	9	9
I_9	10	10

Tabla 2: Parámetros del algoritmo evolutivo

Tamaño población	15
Tamaño cromosoma (poblaciones)	110
Criterio parada (generación)	500
Recombinación	PMX
Mutación	SW
Probabilidad recombinación	0,65
Probabilidad mutación	0,05
Nro. de recombinación (n1)	16
Nro. de padres (n2)	18

Los parámetros (tamaño de la población, criterio de parada, probabilidades, etc.) no fueron elegidos al azar, sino seleccionados de estudios experimentales anteriores. La Tabla 2 resume los parámetros utilizados.

Antes de realizar los experimentos fue necesario preparar los datos debido a que la información original sobre las distancias entre las locaciones petroleras no se encontraba automatizada. Se llevo a cabo el cálculo de distancias entre los pozos de petróleo basado en el mapa de rutas y la distribución del yacimiento petrolero. Se sabe que la distancia entre dos puntos que están en cualquier lugar del sistema de coordenadas cartesianas, se determina por la relación denominada distancia euclidiana. Sin embargo, en este problema la distancia entre dos puntos se puede calcular, teniendo en cuenta las rutas existentes de conexión a las locaciones petroleras. Por esta razón, se utilizó la hoja de ruta del yacimiento petrolero y las distancias fueron escaladas del plano.

En el **primer experimento** se utilizaron las seis primeras instancias de la Tabla 1 con el objetivo elegir la función de penalidad (\mathcal{P}_1 y \mathcal{P}_2) a utilizar en los distintos enfoques (estático, dinámico, adaptativo y annealing descriptos en la sección 2.7) se muestran los resultados obte-

Tabla 3: Resultados obtenidos con Penalidades Estáticas, Dinámicas, Annealing y Adaptativas con las instancias $I_1, I_2, I_3, I_4, I_5, I_6$

Ins	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{P}	Mediana	Promedio	Evals
I_1	S1	444,21	451,66	9073	D1	443,97	444,19	10227	A1	444,90	447,74	10311	AD1	445,00	449,25	9637
	S2	444,94	447,45	8432	D2	443,93	438,46	9863	A2	444,61	448,73	9923	AD2	444,73	441,99	9043
I_2	S1	446,03	464,29	10157	D1	445,16	449,98	10305	A1	445,32	452,64	10952	AD1	445,10	449,48	8132
	S2	445,70	452,85	8507	D2	443,91	446,85	8949	A2	446,79	450,05	9815	AD2	446,63	451,49	9382
I_3	S1	454,31	465,15	9123	D1	446,34	450,97	10499	A1	446,01	456,45	10752	AD1	446,88	452,81	9050
	S2	446,84	453,30	9091	D2	444,81	450,34	9186	A2	445,96	458,23	10628	AD2	445,66	448,39	8001
I_4	S1	466,72	471,13	8189	D1	447,08	464,54	10571	A1	461,70	458,84	10742	AD1	461,15	460,12	9079
	S2	458,80	463,37	9278	D2	445,44	454,29	9186	A2	446,74	453,12	10791	AD2	461,31	461,07	9496
I_5	S1	468,22	463,97	8934	D1	463,93	460,30	10356	A1	463,87	462,85	10984	AD1	460,78	460,52	9522
	S2	457,99	459,01	9145	D2	446,60	449,83	9473	A2	471,58	467,91	11146	AD2	447,23	453,74	8919
I_6	S1	473,95	471,72	7961	D1	467,11	463,43	10689	A1	462,55	460,09	10097	AD1	469,78	471,09	8993
	S2	469,20	466,72	8969	D2	446,99	456,44	10050	A2	463,15	464,42	11241	AD2	468,13	467,42	8785

nidos en la Tabla 3. Donde la primer columna (Ins) es el nombre de la instancia, las restantes columnas dividen a la información en cuatro grupos y en cada grupo se muestra la siguiente información: (\mathcal{P}) es la penalidad utilizada, (Mediana) representa la mediana de kilómetros recorridos, (Promedio) representa el promedio de kilómetros recorridos, y (Evals) representa el número de miles de evaluaciones efectuadas por cada enfoque. Recordemos que S1 y S2 corresponde al enfoque de penalidad estática; D1 y D2 al enfoque de penalidad dinámica; A1 y A2 a penalidad *annealing* y finalmente, AD1 y AD2 a la penalidad adaptativa.

El primer grupo (lado izquierdo de la Tabla 3) muestra los resultados utilizando el enfoque de penalidad estática y puede observarse que para casi todas las instancias la mínima mediana en kilómetros recorridos es obtenida por S2 (Instancias I_2, I_3, I_4, I_5, I_6) y el mínimo valor para el promedio de kilómetros recorrido es obtenido también por este enfoque para todas las instancias. En cuanto al número de evaluaciones, para las instancias I_4, I_5, I_6 , S1 logro el valor mínimo, mientras que para I_1, I_2 e I_3 el valor mínimo fue obtenido por S2.

El segundo grupo (centro izquierdo de la Tabla 3) pertenece al enfoque dinámico. En este caso puede observarse que D2 obtiene el mínimo valor para todas las instancias y para todas las variables analizadas.

El tercer grupo (centro derecho de la Tabla 3) muestra el enfoque *annealing*. Los mejores resultados para la mediana de kilómetros recorridos es obtenido por A1 en tres de seis instancias (I_2, I_5, I_6), para el promedio de kilómetros recorridos en cuatro de seis instancias (I_1, I_3, I_5, I_6) A1 obtiene mejores resultados, y para el número de evaluaciones en tres de seis instancias (I_4, I_5, I_6).

En el último grupo muestra los resultados obtenidos por el enfoque adaptativo. Se puede observar que para los menores valores para la mediana y promedio de kilómetros obtenidos es obtenido por AD2 en cuatro de las seis instancias (I_1, I_3, I_5, I_6). En cuanto al número de evaluaciones para las instancias pares los menores valores son obtenidos por AD2 y para las instancias impares los menores valores son obtenidos por AD1.

En general usando la segunda función de penalidad (\mathcal{P}_2) con todos los enfoques de penalidad se obtienen los menores valores y es por esta razón que en las siguientes experimentos se lo utiliza (\mathcal{P}_2) como función de penalidad.

Con el fin de encontrar el mejor enfoque de penalidad para este problema y con este conjunto de instancias se incorpora el enfoque mortal denominado en la Tabla 4 como (DTH) y se lo

Tabla 4: Resultados obtenidos con los enfoques de Penalidad Estática, Dinámica, Annealing, Adaptativa y Mortal con las instancias $I_1, I_2, I_3, I_4, I_5, I_6$

I_{ns}	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{P}	Mediana	Promedio	Evals
I_1	S2	444,94	447,45	8432	A2	444,61	447,85	9923	AD2	444,73	441,99	9043
I_1	D2	443,93	438,46	9863	DTH	444,55	453,51	9088	—	—	—	—
I_2	S2	445,70	452,85	8507	A2	446,79	450,05	9815	AD2	443,25	439,62	8238
I_2	D2	443,91	446,85	8949	DTH	446,14	455,03	8835	—	—	—	—
I_3	S2	446,84	453,30	9091	A2	445,96	458,23	10628	AD2	445,66	448,39	8001
I_3	D2	444,81	450,34	9186	DTH	447,50	455,02	8692	—	—	—	—
I_4	S2	458,80	463,37	9278	A2	446,74	453,12	10791	AD2	445,17	447,73	7823
I_4	D2	445,44	454,29	9186	DTH	446,83	461,66	9355	—	—	—	—
I_5	S2	457,99	459,01	9145	A2	471,58	467,91	11146	AD2	447,23	453,74	8919
I_5	D2	446,60	449,83	9473	DTH	447,92	464,96	8980	—	—	—	—
I_6	S2	469,20	466,72	8969	A2	463,15	464,42	11241	AD2	455,49	447,20	9318
I_6	D2	446,99	456,44	10050	DTH	461,82	463,69	8771	—	—	—	—

compara con otros enfoques.

En la Tabla 3 la primer columna (Ins) es el nombre de la instancia, las restantes columnas muestran la penalidad utilizada (columnas \mathcal{P}), la mediana de kilómetros recorridos (columnas Mediana), el promedio de kilómetros recorridos (columnas Promedio), y el número de miles de evaluaciones efectuadas por cada enfoque (columnas Evals). Se puede observar que el menor valor para la mediana de kilómetros recorridos es obtenida por D2 en cuatro de las seis instancias (I_1, I_3, I_5, I_6), mientras que AD2 obtiene los valores mínimos los dos instancias restantes (I_2, I_4). En cuanto al promedio de kilómetros recorridos AD2 obtiene los menores valores para cuatro de las seis instancias (I_2, I_3, I_4, I_6) mientras que AD2 obtiene los valores mínimos en las otras dos instancias (I_1, I_5). Finalmente, en cuanto al número de evaluaciones AD2 obtiene los valores mínimos en cuatro de las seis instancias (I_2, I_3, I_4, I_5).

Para realizar un análisis más detallado en cuanto al promedio de kilómetros recorridos se aplicaron test de Krustal-Wallis, ya que los resultados no poseían una distribución normal. Para las instancias I_3, I_4, I_6 se observaron diferencias estadísticamente significativas entre los enfoques D2, AD2 y S2. Con la instancia I_3 se observaron diferencias estadísticamente significativas entre D2 y AD2. En este caso se puede afirmar que D2 es mejor que AD2. En cuanto a la instancia I_4 se observaron diferencias estadísticamente significativas entre D2 y AD2 con respecto a S2. Es decir que los primeros dos enfoques (D2 y AD2) tienen una mejor performance que S2. Finalmente para la instancia I_6 se observó diferencia estadísticamente significativa entre AD2 y S2. Podemos asegurar con un 95 % de confianza que el mejor comportamiento para esta instancia fue obtenido por AD2.

Para las instancias restantes (I_1, I_2, I_5) no se encontraron diferencias estadísticamente significativas y por esta razón se muestra la distribución de los resultados en un diagrama de box-plot (Figura 2).

En la Figura 2(a) puede observarse que para la instancia I_1 los valores de las medianas son similares en todos los enfoques y todos los valores obtenidos por cada enfoque están cercanos de su correspondiente mediana. Esto se refleja en lo compacto de las gráficas.

En cuanto a la instancia I_2 los enfoques D2 y AD2 tienen valores de medianas similares y en ambos enfoques existen valores por debajo de la mediana, en el cuartil inferior (Figura 2 (b)).

Finalmente, para la instancia I_5 los resultados obtenidos por D2 y AD2 son bastante similares y mejores que los otros enfoques (Figura 2 (c)).

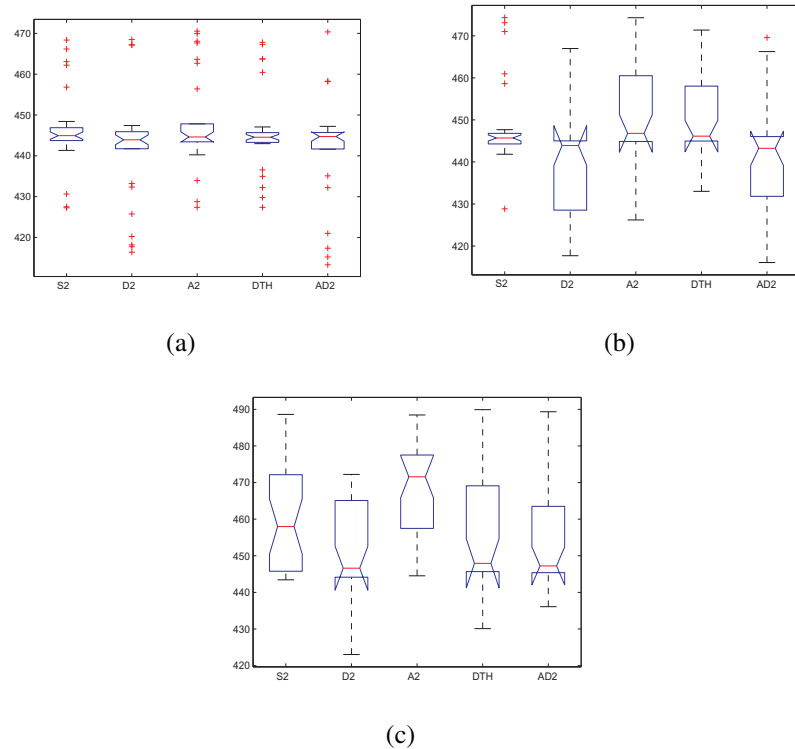


Figura 2: Box-plots para S2, D2, A2, DTH, y AD2 considerando: (a) Instancia I_1 , (b) Instancia I_2 , y (c) Instancia I_5 .

En el **segundo experimento** [Villagra et al 2010] se tomaron los enfoques que obtuvieron mejores resultados en la mayoría de las variables analizadas y se utilizaron las tres últimas instancias de la Tabla 1, que contienen locaciones petroleras que deben visitarse dos y tres veces. La Tabla 5 muestra para los tres enfoques y con las tres instancias seleccionadas, la siguiente información: la columna Ins es el nombre de la instancia, \mathcal{P} es el enfoque de penalidad utilizada, Mediana representa la mediana de kilómetros recorridos, Promedio representa el promedio de kilómetros recorridos, y Evals representa el número de miles de evaluaciones efectuadas por cada enfoque. Se puede observar que para todas las instancias D2 obtiene los valores mínimos para la mediana de kilómetros recorridos y para el promedio de kilómetros recorridos obtiene también los menores valores en dos de las tres instancias (I_7 , I_9). En cuanto al número de evaluaciones AD2 obtiene los valores mínimos para dos de las tres instancias (I_8 , I_9).

4. Conclusiones

PAE es una aplicación que se construyó con el objetivo de proveer una herramienta efectiva que facilite la planificación de visitas de mantenimiento a las locaciones petroleras sujetas a un conjunto de restricciones.

Los Algoritmos Evolutivos son metaheurísticas que usan modelos computacionales del proce-

Tabla 5: Resultados obtenidos por S2, D2, AD2 con las instancias I_7 , I_8 y I_9 .

Ins	\mathcal{P}	Mediana	Promedio	Evals	\mathcal{A}	Mediana	Promedio	Evals	\mathcal{A}	Mediana	Promedio	Evals
I_7	S2	444.94	447.45	8432	D2	443.93	438.46	9863	AD2	444.73	441.99	9043
I_8	S2	446.84	453.30	9091	D2	444.81	450.34	9186	AD2	445.66	448.39	8001
I_9	S2	457.99	459.01	9145	D2	446.60	449.83	9473	AD2	447.23	453.74	8919

so evolutivo. Para este problema con restricciones del mantenimiento de locaciones petroleras se utilizó un enfoque multirecombinativo llamado EA-MCMP-SRI que implementa diferentes enfoques de penalidad para manejar las restricciones. De los experimentos llevados a cabo, se puede concluir que:

- En cuanto a la función de penalidad \mathcal{P}_2 obtuvo mejores resultados que \mathcal{P}_1 , en todos los enfoques utilizados y para las variables analizadas.
- EA-MCMP-SRI obtiene, en general, mejores resultados con los enfoques dinámico (D2) y Adaptativo (AD2).
- En cuanto al esfuerzo computacional (evaluaciones) de los enfoques, el enfoque Adaptativo (AD2) consume menos recursos que los otros enfoques.

Trabajos futuros incluirán el diseño, implementación y estudio de técnicas avanzadas para el manejo de restricciones. Además se incluirá la formulación de diferentes tipos de restricciones y *schedules* considerando multiples equipos de mantenimiento paralelizando las planificaciones.

5. Agradecimientos

Agradecemos la colaboración del LabTEm y a la Universidad Nacional de la Patagonia Austral, de la cual se recibe apoyo continuo.

6. Referencias

Bäck T., editor. Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.

Brucker P. 2004., Scheduling Algorithms, 3rd ed. Springer-Verlag New York.

Carroll C. W. 1961., The created response surface technique for optimizing nonlinear retrained systems. Operations Research, 9.169-184.

Chang P., Hsieh J., and Wang Y. 2005. Knowledge Incorporation in Evolutionary Computation, chapter F. Divina and E. Marchiori. Springer-Verlag.

Coello Coello C. 2002, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art", *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, No. 11-12. (4 January 2002), pp. 1245-128.

Courant R. 1943, *Variational Methods for the Solution of Problems of Equilibrium and Vibrations*. *Bulletin of the American Mathematical Society*, 49:1-23.

Davis L. 1987, *Genetic Algorithms and Simulated Annealing*. Pitman, London.

Dhar, V., and Ranganathan, N. 1990. *Integer Programming versus Expert Systems: An Experimental Comparison*. *Communications of the ACM* 33:323-336.

Eiben A.E. y Bäck Th. 1997. *An Empirical Investigation of Multi-Parent Recombination Operators in Evolution Strategies*; *Evolutionary Computation*, 5(3):347-365.

Esquivel S., Leiva A., Gallard R. 1997. "Multiple Crossover per Couple in Genetic Algorithms". *Proc. Fourth IEEE Conference on Evolutionary Computation (ICEC'97)*, pp 103-106, Indianapolis, USA.

Fiacco A. V. and McCormick G. P, 1968. *Extensions of SUMT for nonlinear programming: equality constraints and extrapolation*. *Management Science*, 12(11):816-828.

Fogel L., *Artificial Intelligence through Simulated Evolution*. John Wesley, New York, 1996.

Goldberg D. and Lingle R., 1987. *Alleles, loci and the traveling salesman problem*. In *International Conference on Genetic Algorithms*, pages 154-159.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley, Reading, MA.

Holland J. 1975, *Adaptation in Natural and Artificial Systems*. Ann Harbor: University of Michigan Press.

Jaskowski P. y Sobotka A. 2006. *Multicriteria construction project scheduling method using evolutionary algorithm*. *Operational Research an International Journal*, 6(3).

Kirkpatrick S., Gelatt J. and Vecchi M. 1983. *Optimization by simulated annealing*. *Science*, 220:671-6801.

Koziel S. and Michalewicz Z. 1999. *Evolutionary algorithms, homomorphous mapping, and constrained parameter optimization*. 7(1): 19-44, *Evolutionary Computation* .

Michalewicz Z. 1997. *Genetic Algorithms + Data Structures = Evolutions Programs*; Springer-Verlag, third, revised edition.

Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*.

Mitchell M., 1996. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, Massachusetts.

Pandolfi, D., Pedro, M. D. S., Villagra, A., Vilanova, G., and Gallard, R. 2002. Studying immigrants in evolutionary algorithm to solve the earliness-tardiness scheduling problem. *Cybernetics and Systems of Taylor and Francis*, 391-400.

Pinedo M., 1995. *Scheduling: Theory, Algorithms and System*. First edition Prentice Hall.

Richardson J.T, Palmer M., Liepins G., and Hilliard M. 1989. Some Guidelines for Genetic Algorithms with Penalty Functions, In J. David Schaffer, editor, *Proceedings of the Third International Conference on genetic Algorithms*, pages 191-197; George Mason University. Morgan Kaufmann Publishers.

Schwefel H. 1981. *Numerical Optimization of Computer Models*. John Wiley and Sons, Great Britain.

Schwefel P. 1995, *Evolution and Optimum Seeking*. John Wiley and Sons, New York.

Smith A. and Coit D. 1997, *Constraint Handling Techniques Penalty Functions*. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C 5.2. Oxford University Press and Institute of Physics Publishing.

Venkatraman, S. and Yen, G. 2005. A generic framework for constrained optimization using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 9(4):424-435.

Villagra A., de San Pedro E., Lasso M., Pandolfi D. 2008; "Algoritmo multirecombinativo para la planificación dinámica del mantenimiento de locaciones petroleras"; *Revista Internacional INFORMACIÓN TECNOLÓGICA*"; La Serena, Chile. Vol. 19 Número 4 pp. 63-70.

Villagra A., Pandolfi D., Rasjido J., Montenegro C., Serón N. y Leguizamón G. 2010, *Repair Algorithms and Penalty Functions to Handling Constraints in an Evolutionary Scheduling*; Congreso Argentino de Ciencias de la Computación 2010.