

# **ALGORITMOS DE REPARACIÓN PARA EL MANEJO DE RESTRICCIONES EN LA PLANIFICACIÓN DEL MANTENIMIENTO DE LOCACIONES PETROLERAS**

AdeS Jose Rasjido, AdeS Daniel Pandolfi y Mg Andrea Villagra  
Docentes Investigadores de la UNPA  
{jrasjido, dpandolfi, avillagra}@uaco.unpa.edu.ar

UNPA – UACO  
Universidad Nacional de la Patagonia Austral – Unidad Académica Caleta  
Olivia  
Departamento de Ciencias Exactas y Naturales  
Caleta Olivia, 2010

## **Resumen**

Los ecosistemas se ven afectados directa y negativamente por muchas actividades industriales riesgosas, como lo es el transporte y la explotación del petróleo. Por lo tanto es importante que las compañías petroleras lleven a cabo un mantenimiento correcto de los yacimientos petroleros. En este informe se explica las técnicas de reparación implementadas en la herramienta evolutiva PAE, para el manejo de las restricciones en la planificación del mantenimiento de las locaciones petroleras. Las comparaciones y los resultados obtenidos, muestran que el algoritmo de reparación propuesto requiere un menor esfuerzo computacional (evaluaciones) con la misma calidad de soluciones para los conjuntos de instancias utilizadas.

**Palabras Claves:** Metaheurísticas, Planificación, Manejo de Restricciones, Algoritmos de Reparación.

## 1. INTRODUCCIÓN

Las actividades petroleras de explotación y transporte del petróleo son muy importantes para el desarrollo económico de la sociedad industrial moderna, por ejemplo en la zona norte de la provincia de Santa Cruz. Sin embargo, estas actividades son generadoras de riesgos que se traducen en contaminaciones accidentales que afectan directamente al ecosistema. El impacto ambiental ocasionado por la industria petrolera comprende los efectos de todas y cada una de las fases involucradas en las etapas tanto de exploración, explotación, transformación, distribución y comercialización. En muchas de las regiones donde se ha explotado el petróleo, se han generado contingencias de diferentes magnitudes como consecuencia de las operaciones desarrolladas en: tanques de almacenamiento, oleoductos, equipos de perforación, buques petroleros, etc. Además de los impactos generados en el medio como consecuencia de la exploración y explotación del petróleo, existen también daños ocasionados a las personas que lo emplean o se encuentran en contacto con él dentro de su ámbito laboral.

En los últimos años, las Empresas dedicadas a la explotación, producción y transporte del recurso, 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 *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. Los Algoritmos Evolutivos (AEs) han sido aplicados exitosamente en la resolución de diversos tipos de problemas de planificación tales como *scheduling* o *routing* [Chang P. et al. 2005], [Jaskowski P. y Sobotka A. 2006]. Los AEs son metaheurísticas que comparten un concepto base común que es simular la evolución de los individuos que forman la población usando un conjunto de operadores predefinidos de selección y de búsqueda.

Algunos enfoques para AEs utilizan la multirecombinación [Eiben A.E. y Bäck Th. 1997] y enfoques con múltiples padres y múltiples *crossover* conocidos como MCMP (*Multiple Crossover Multiple Parent*) [Esquivel S. et al. 1997].

La mayoría de los problemas de optimización incluyen restricciones de cierta clase, que constituyen grandes desafíos a la resolución de problemas de optimización. Las restricciones son límites impuestos a las variables de decisión y en general las restricciones son una parte integral de la formulación de cualquier problema [Dhar, V., y Ranganathan, N. 1990]. En el dominio específico de problemas discretos, tales como *Knapsack Problem*, *Set Covering Problem*, *Vehicle Routing Problem*, y todos los tipos de *Scheduling Problems*, están sujetos a restricciones. El espacio de búsqueda de estos problemas se subdivide en dos subconjuntos, el espacio de soluciones factibles y el espacio de soluciones no factibles. Estos subespacios no necesariamente son convexos y no necesariamente están conectados. Para la resolución del

problema de optimización sujeto a restricciones los algoritmos de búsqueda, tal como lo son AEs, promueven soluciones que navegan en ambos subespacios.

El problema consistirá entonces en cómo tratar soluciones no factibles para explorar el subespacio de soluciones factibles [Michalewicz Z. 1997]. Es importante destacar que los diferentes trabajos realizados por el grupo de investigación apuntaron al desarrollo de tareas de investigación para el problema de planificación del mantenimiento de locaciones petroleras en aspectos de planificación estática, dinámica y manejo de restricciones [Villagra A. et al. 2008].

El manejo de restricciones en algoritmos evolutivos puede abordarse desde distintas técnicas tales como: Funciones de penalidad, Decodificadores, Algoritmos de reparación, y operadores de preservación de restricciones, y en los últimos años se han publicado numerosos avances en la optimización en el manejo de restricciones [Efrén Mezura – Montes Ed. 2009]. Las funciones de penalización han sido ampliamente discutidas en la literatura [Smith A. y Coit D. 1997]. Existen enfoques que penalizan soluciones no factibles como nuevos y promisorios enfoques que también penalizan las soluciones factibles. Las técnicas de penalización pueden subdividirse como estáticas, dinámicas, adaptativas y mortales.

Las técnicas basadas en algoritmos de reparación mapean una solución desde el espacio no factible al espacio factible. Los procesos de reparación que modifican genéticamente la solución no factible son conocidos como Lamarkianos y transforman una solución no factible en factible la que es evolucionada por el algoritmo. Un enfoque menos destructivo de la solución no factible, permite combinar el aprendizaje y la evolución, estrategia conocida como Baldwiniana. En este último enfoque las soluciones son reparadas solo para su evaluación. Estudios analíticos y empíricos indican que esta técnica reduce la velocidad de convergencia del algoritmo evolutivo y permite converger a óptimo globales [Whitley, D. et al. 1994].

En este informe se presenta en detalle las técnicas de reparación que se han trabajado durante el desarrollo del proyecto de investigación: Manejo de Restricciones con Algoritmos Evolutivos aplicados a la Planificación del Mantenimiento de Locaciones Petroleras.

El documento se organiza de la siguiente manera: en la sección 2 se desarrolla el marco teórico del informe, en el cual se detalla una descripción y la formulación del problema en las subsecciones 2.1 y 2.2 respectivamente, continua con una breve introducción a los Algoritmos Evolutivos en la subsección 2.3, luego en la subsección 2.4 se explica la herramienta evolutiva PAE, y finalmente en la subsección 2.5 se explican detalladamente las técnicas de reparación para el manejo de restricciones. En la sección 3 se muestran y se analizan los resultados obtenidos de los distintos experimentos realizados. Y en la sección 4 se muestran las conclusiones obtenidas a partir del análisis de los resultados.

## 2. MARCO TEÓRICO

### 2.1. DESCRIPCIÓN DEL PROBLEMA

Las empresas petroleras realizan visitas de mantenimiento preventivo a cada una de sus locaciones petroleras (pozos productores, inyectoras, baterías y colectores). Un yacimiento está formado por bloques y a su vez éste por baterías. Cada batería está formada por pozos de producción que son en promedio entre 15 y 20. Cada pozo tiene diferente nivel de producción que es conocido a priori y varía en el tiempo. La producción del pozo define la categoría y la cantidad de veces que debe visitarse al mes. Los pozos no pueden ser visitados más de una vez al día y dependiendo del tipo de pozo, existen ciertas tareas que se deben realizar. Cada tarea tiene asignado un determinado equipamiento necesario, una frecuencia de realización y un tiempo aproximado de su duración. Actualmente, el recorrido que realizan los encargados de las locaciones, se planifica en base a la experiencia de los mismos. La jornada laboral comienza a la mañana y se visitan las locaciones en dos turnos de tres horas. Luego de finalizado cada turno el responsable debe regresar a la base de operaciones, realizar determinadas actividades administrativas y luego comenzar con el siguiente turno. El tiempo demandado en cada locación dependerá del tipo de la misma. Existen contingencias aleatorias que hacen que el plan de mantenimiento de un turno no se cumpla según lo planificado, produciendo la necesidad de replanificar las visitas. La replanificación se presenta en [Villagra A. et al. 2007, (a)] y [Villagra A. et al. 2007, (b)]. Además puede ocurrir que en el momento de replanificación, por determinados motivos, se debe incorporar la visita a ciertas locaciones, provocando esto un conjunto de restricciones en la replanificación. Cuando ocurre esto, cada responsable redefine el nuevo itinerario utilizando su experiencia. Las restricciones con visitas obligatorias se presentaron en [Villagra A. et al. 2007(c)]. Finalmente existen locaciones que por determinadas causas deben visitarse dos o tres veces en distintos turnos, y de igual forma a lo anterior, los encargados realizan la planificación de las mismas en base a su experiencia.

### 2.2. FORMULACIÓN DEL PROBLEMA

El problema puede ser precisamente formulado como se definió en [Pinedo M. 1995]:

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

Esto denota un problema de planificación de máquina única con  $n$  trabajos sujetos a una secuencia dependiente de la configuración de los tiempos, donde los trabajos a ser planificados son el servicio de mantenimiento (o la intervención) en cada una de las locaciones petroleras. El objetivo es minimizar el *makespan* ( $C_{\max}$ ) sujetos a los tiempos dependientes de preparación de la secuencia. Este modelo conduce hacia una optimización del problema que es *NP-Hard* [Pinedo M. 1995]. El *makespan* puede ser calculado como:

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

Donde  $S_{jk}$  representa el costo (en tiempo) de trasladarse desde la locación petrolera  $j$  hasta la locación petrolera  $k$ ,  $t_k$  se refiere al tiempo de mantenimiento en la locación  $k$ , y  $n$  es el número total de locaciones petroleras en el yacimiento. Por otra parte, la formulación mencionada se puede extender como sigue:

**Definición 1:** Sean  $\mathbf{OW1} \subset \{1, \dots, N\}$  y  $\mathbf{OW2} \subset \{1, \dots, M\}$  dos subconjuntos de todas las locaciones petroleras en el yacimiento donde  $\mathbf{OW1} \cap \mathbf{OW2} = \emptyset$ .  $\mathbf{OW1}$  representa las locaciones petroleras que deberían ser visitadas dos veces y  $\mathbf{OW2}$  representa las locaciones petroleras que deberían ser visitadas 3 veces. Además, las locaciones petroleras en  $\mathbf{OW1}$  o en  $\mathbf{OW2}$  no pueden ser planificadas en el mismo turno. Una solución que no cumple con las condiciones mencionadas, se considera no factible.

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

### 2.3. ALGORITMOS EVOLUTIVOS

La motivación detrás de la simulación de la evolución proviene de al menos cuatro fuentes alternativas [Fogel D. 1997].

La optimización es la primera de ellas. La evolución, en sí misma, es un proceso de optimización y como prueba están los “órganos de extrema perfección” (Darwin 1859) como el ojo, formador de imágenes. Si bien la optimización no implica la perfección, la evolución puede descubrir soluciones funcionales altamente precisas para problemas planteados por el ambiente de un organismo. Más aún, los mecanismos que van siendo evolucionados son, desde una perspectiva ingenieril, muy elaborados: la función es la única cualidad expuesta ante la selección natural y la funcionalidad es lo que se optimiza a través de la selección iterativa y la mutación cambiante

Por esta razón resulta natural el tratar de describir la evolución en términos de un algoritmo que pueda usarse para resolver difíciles problemas ingenieriles de optimización. Los métodos clásicos del gradiente descendente, *hill climbing* determinístico o búsqueda aleatoria han resultado insatisfactorios al aplicárseles a problemas de optimización no lineales, especialmente, aquéllos con componentes estocásticos, caóticos o temporales. Pero estos son los problemas que la naturaleza parece haber resuelto bien. La evolución provee inspiración para computar soluciones que previamente aparecían como intratables.

La adaptación robusta es otra de las fuentes. El mundo real nunca es estático y los problemas de optimización temporal son algunos de los más desafiantes, pues requieren estrategias de

comportamiento variable acorde con la retroalimentación (feedback) acerca del éxito o el fracaso de la estrategia actualmente aplicada. Holland (1975), en el marco de los planes reproductivos (algoritmos genéticos) describió un procedimiento para evolucionar estrategias. Esto se hacía en la forma de *strings* codificados o basándose en reglas de comportamiento, llamadas sistemas clasificadores, explotando la recombinación de piezas exitosas de estrategias competitivas, a partir del conocimiento adquirido por individuos independientes. Como resultado se obtiene un procedimiento robusto que puede ajustar su performance al *feedback* del ambiente.

Las máquinas inteligentes son otra motivación. La inteligencia puede definirse como la capacidad de un sistema para adaptar su comportamiento a fin de cumplir ciertos objetivos en una categoría de ambientes. En consecuencia, el comportamiento inteligente requiere capacidad de predicción dado que la adaptación a circunstancias futuras requiere predecir esas circunstancias y tomar vías de acción apropiadas. La evolución ha creado criaturas de creciente inteligencia a lo largo del tiempo. En vez de intentar crear máquinas inteligentes mediante la replicación de aspectos de los seres humanos, ya sea imitando las reglas que ellos siguen o sus conexiones neuronales, un enfoque alternativo consiste en simular la evolución sobre una clase de algoritmos predictivos.

Por último, la biología, es otra fuente de motivación. En lugar de intentar usar la evolución como una herramienta para resolver un problema ingenieril, existe un deseo de capturar la esencia de la evolución en una simulación de computadora para ganar una nueva visión en la física de los procesos evolutivos naturales. De esta forma, existiría la posibilidad de estudiar sistemas biológicos alternativos que sean meras imágenes plausibles de lo que la vida puede ser. También aparece la pregunta de qué propiedades de tales sistemas imaginarios son comunes a la evolución de la vida en la tierra. Aunque todo modelo es incompleto e intentar evaluar lo que la vida podría ser en otras instancias pertenece al campo de la especulación pura, algunas simulaciones conocidas como vida artificial han generado algunos patrones que parecen corresponderse con fenómenos naturales.

Finalmente, la respuesta a la pregunta por qué simular la evolución descansa en la falta de buenas alternativas. Pues no podemos utilizar los métodos clásicos de optimización para encontrar mínimos globales de funciones, que están rodeados de mínimos locales. Los sistemas expertos que intentan imitar la inteligencia humana no son robustos ante los cambios en el dominio de aplicación y son incapaces de predecir correctamente circunstancias futuras para tomar decisiones acertadas. En contraste con ello, si explotamos exitosamente el uso de la aleatoriedad, es decir el uso de la incertidumbre, todos los caminos posibles están abiertos para la Computación Evolutiva. Nuestro desafío consiste entonces en no restringir la potencialidad de la Computación Evolutiva para descubrir nuevas soluciones a nuevos problemas de manera impredecible. Sin embargo, como siempre, los últimos avances en este campo vendrán de la cuidadosa abstracción e interpretación de los procesos que lo inspiran.

Las aplicaciones de la Computación Evolutiva son posibles en varias áreas [Beasley D. 1997][Bäck T. et al. 2000]. Una división gruesa, y no excluyente, abarcaría las siguientes áreas: Planificación, Diseño, Simulación e Identificación, Control y Clasificación.

Los Algoritmos Evolutivos (AEs) utilizan el proceso de aprendizaje colectivo de una población de individuos. Usualmente un individuo representa un punto de búsqueda en el espacio de posibles soluciones de un problema. Además los individuos pueden incorporar información adicional como por ejemplo, parámetros para la estrategia de los AEs.

Los descendientes de los individuos se generan por procesos aleatorios que intentan modelar la mutación y la recombinación. La mutación corresponde a una errónea auto-replicación de individuos, por lo general pequeñas modificaciones. La recombinación intercambia información entre dos o más individuos.

A cada individuo se le asigna una medida de su calidad o aptitud, evaluándolo en su ambiente. Como requerimiento mínimo es posible realizar una comparación de su calidad, lo que nos conduce a una decisión binaria: mejor o peor. Acorde con esta medida de adaptabilidad, el proceso de selección favorece a los mejores individuos a fin de que se reproduzcan más frecuentemente que aquellos relativamente peores.

Un Algoritmo Genético para un problema particular debe tener los siguientes cinco componentes:

- Una representación genética para una solución potencial del problema.
- Un modo para crear una población inicial de soluciones potenciales
- Una función de evaluación que juegue el rol del ambiente, evaluando las soluciones desde el punto de vista de su *fitness*.
- Operadores genéticos que alteren la composición de los individuos (soluciones).
- Valores para varios parámetros que usa el algoritmo evolutivo, como el tamaño de la población, cantidad máxima de generaciones, probabilidades de aplicación de los operadores genéticos (mutación y crossover).

## 2.4. PAE: Una herramienta evolutiva

Para resolver el problema de planificación de recorrido de las locaciones petroleras se utilizó un Algoritmo Evolutivo. El cromosoma representa el orden en que serán visitadas las locaciones (gen del cromosoma) dentro de la planificación.

Para el AE se implementó un proceso de múltiple recombinación y múltiples padres [Goldberg, D.E. 1989], [Eiben A.E. y Bäck Th. 1997], [Esquivel S. et al. 1997] donde se genera un *pool* de múltiples padres, conformado por un individuo semental y por individuos generados aleatoriamente (inmigrantes aleatorios), este proceso es llamado MCMP-SRI y es una variante de multirecombinación [Pandolfi D. et al. 2002]. Este método fue aplicado en diferentes problemas de planificación de máquina única para casos estáticos y casos dinámicos y los resultados obtenidos fueron satisfactorios. En los problemas estáticos, MCMP-SRI se aplicó para resolver problemas de *Earliness* y *Tardiness* [Pandolfi D. et al. 2001], *Weighted Tardiness* [De San Pedro et al. 2001], *Average Tardiness* [Pandolfi D. et al. 2003] y *Weighted Number of Tardy Jobs* [De San Pedro et al. 2003]. En los problemas dinámicos MCMP-SRI fue aplicado para resolver problemas de adaptabilidad para *Earliness* y *Tardiness* [Villagra A. et al- 2001], en problemas de dinámica parcial y total para *Weighted Tardiness* [Lasso M. et al. 2003] y *Average Tardiness* [De San Pedro M.E. et al. 2003].

El proceso para crear descendientes, en MCMP-SRI, es el siguiente: de la vieja población de individuos, se selecciona un individuo, el semental, a través de selección proporcional. Se genera un pool de apareamiento con  $n_2$  padres generados aleatoriamente. El semental se aparea con cada padre del pool de apareamiento y las parejas se someten a operaciones de recombinación, y generan  $2*n_2$  descendientes. El mejor de los  $2*n_2$  descendientes, se almacena en un pool de hijos temporal. Esta operación de recombinación se repite  $n_1$  veces, para diferentes puntos de corte cada vez, hasta que el pool de hijos se complete. Finalmente,

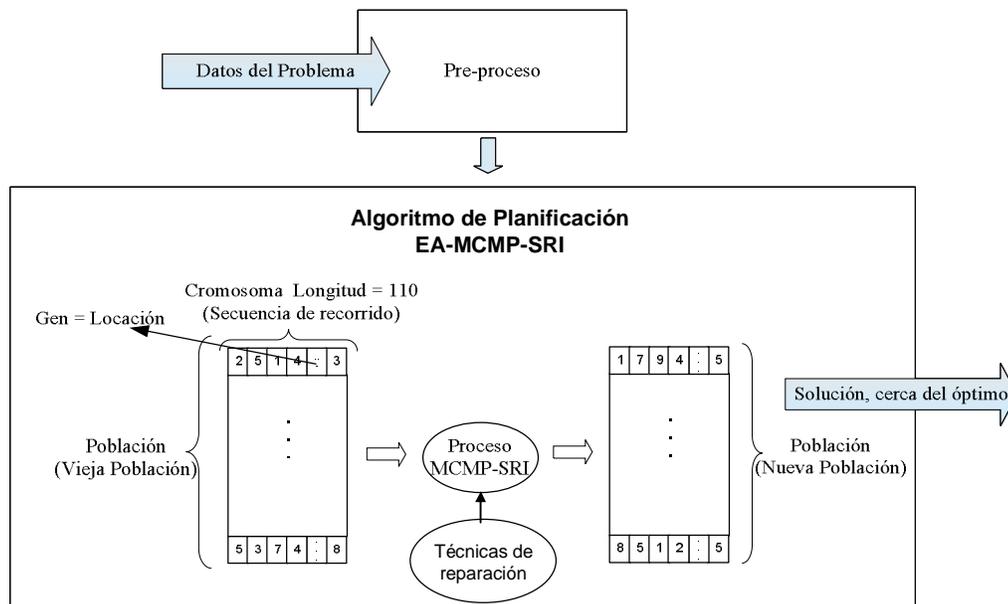
el mejor descendiente creado de  $n_2$  padres y  $n_1$  operaciones de recombinación, se inserta en la nueva población. El método de recombinación utilizado fue PMX (*Partial Mapped Crossover*): [Goldberg D. y Lingle R. 1987] que puede verse como una extensión del cruzamiento de dos puntos para representaciones basadas en permutaciones. La selección de individuos fue a través de 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  $i \in \{m+1, \dots, n\}$  decodifica en un número entero que pertenece a  $OW1$  o  $OW2$ . Cuando una solución se evalúa se aplica una función de penalidad o un algoritmo de reparación, según sea el enfoque de manejo de restricciones que se esté utilizando, a las soluciones no factibles.

PAE contempla tres enfoques o técnicas de reparación, cada una de las cuales se implementan de forma diferente. Estos enfoques serán tratados con más detalle en la sección 2.5. Básicamente el objetivo de estas técnicas es la de reparar soluciones no factibles, que cumplan con las restricciones propias del problema.

En la Figura 1 se muestra un esquema del proceso utilizado para la resolución del problema de planificación. Se sistematizaron los datos de entrada para poder utilizar la información en el algoritmo. Se procesa la información, y se produce la planificación a través del algoritmo genético anteriormente comentado, aplicando además las técnicas de reparación para soluciones no factibles.

**Figura 1.** Proceso de PAE.



## 2.5. MANEJO DE RESTRICCIONES Y ALGORITMO DE REPARACIÓN

En general, las restricciones son una parte integral para la formulación de cualquier problema. Coello Coello [Coello Coello 2002] describe una taxonomía útil de técnicas de manejo de restricciones, entre las cuales 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* (o la aptitud) de las soluciones no factibles y tienen preferencia con las factibles en el proceso de selección. Las representaciones espaciales y los operadores están diseñados para representar sólo las soluciones factibles y los operadores son capaces de preservar la viabilidad de la descendencia generada. Los algoritmos de reparación modifican una solución no factible con el objetivo de lograr que estas logren cumplir ciertas restricciones. La separación de objetivos y restricciones consiste en usar estos valores como un criterio separado en el proceso de selección de un AE; esto es opuesto a las funciones de penalidad, en donde el valor de la función objetivo y la restricción son mezclados en un simple valor. Finalmente los métodos híbridos son una combinación de algoritmos diferentes y/o mecanismos.

A continuación se presenta una introducción acerca de las “funciones de penalidad” y luego se explica en detalle los “algoritmos de reparación” que se han propuesto e implementado para el manejo de restricciones de la planificación del mantenimiento de locaciones petroleras.

### 2.5.1 Funciones de Penalidad

Existen dos tipos básicos de funciones de penalidad: *funciones de penalidad exterior*, las cuales penalizan las soluciones no factibles, y las *funciones de penalidad interior*, las cuales penalizan soluciones factibles.

La idea principal de las funciones de penalidad interior es que una solución óptima requiere que una restricción sea activa (por ejemplo rigurosa), por lo que esta solución óptima se encuentra en el límite entre la factibilidad y la no factibilidad. Conociendo esto, una penalidad es aplicada a soluciones factibles cuando una restricción no es activa (tales soluciones son llamadas *soluciones interiores*). Para una restricción simple, este enfoque es sencillo. Para los casos más comunes de múltiples restricciones, la implementación de *funciones de penalidades interiores* es considerablemente más compleja.

Existen tres tipos de *funciones de penalidades exteriores*: (i) métodos de barreras, en los cuales las soluciones no factibles no son consideradas. (ii) funciones de penalidad parcial, en la cual una penalidad es aplicada cerca del límite de la factibilidad, y (iii) funciones de penalidad global que son aplicadas en todas partes de la región no factible.

Una buena comparación de seis estrategias de funciones de penalidad aplicadas a problemas de optimización continua, es dada por Michalewicz (1995). Estas estrategias incluyen enfoques estáticos y dinámicos, y se explican a continuación.

### **Funciones de Penalidad Estática**

Un método simple para penalizar soluciones no factibles es aplicar una penalidad constante a aquellas soluciones que violan la factibilidad en cualquier sentido. Es decir la función objetivo penalizada debería ser la función objetivo no penalizada, mas una penalización (para un problema de minimización). Una variación es penalizar de acuerdo a la distancia que falta para llegar a la factibilidad o dicho de otra forma el “**costo de completación**”. Es decir tener en cuenta una distancia métrica por cada restricción, y agregar una penalidad que se vuelve más severa a mayor distancia de la factibilidad. Las distancias métricas pueden ser continuas o discretas, y pueden ser lineales o no lineales.

### **Funciones de Penalidad Dinámica**

Una variación de las funciones de penalidad “basadas en la distancia”, que alivia muchas de estas dificultades, es la de incorporar un aspecto dinámico que (generalmente) incrementa la rigurosidad de las penalidades para una distancia dada a medida que avanza la búsqueda. Esta presenta la propiedad de permitir, con alta probabilidad, soluciones no factibles muy tempranamente en la búsqueda, pero continuamente incrementando la penalidad impuesta hasta que eventualmente se mueva la solución final hacia la región factible.

### **Funciones de Penalidad Adaptativa**

En este tipo de funciones la idea principal es que las funciones de penalidad permitan además un incremento o un decremento de la penalidad impuesta durante la evolución, por lo tanto, la función estará condicionada por un multiplicador que se actualizará adaptativamente, por ejemplo, a medida que la búsqueda avanza, el multiplicador de la función de penalidad es actualizado cada cierta cantidad de generaciones, basadas en si la mejor solución fue o no factible durante ese intervalo.

En base a la clasificación de penalidades mencionadas anteriormente, se implementaron en PAE tres funciones de penalidad. La primera es una penalidad estática, denominada **STT**, que consiste en aplicar una función de penalidad a las soluciones no factibles, sumando al *fitness* un valor de penalidad estático. La formulación es como sigue:

$$F(p) = f(p) + P(p) \quad (3)$$

donde  $F(p)$  es la función de *fitness* con la penalidad aplicada,  $f(p)$  es el *fitness* del individuo sin la penalización aplicada, y  $P(p)$  es la función de penalidad.

La segunda función de penalidad, denominada **DYN**, consiste en multiplicar el valor de penalidad por otro valor que se obtiene de la siguiente función monotónicamente incremental:

$$V(g) = \left(\frac{g}{G}\right)^2 \quad (4)$$

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

$$F(p) = f(p) + [P(p) \cdot V(g)] \quad (5)$$

La tercera función de penalidad, denominada **ADP**, se propuso un coeficiente adaptativo ( $\rho$ ) que es el responsable de modificar la función de penalidad. Este coeficiente  $\rho$  es actualizado teniendo en cuenta el grado de restricciones violadas. Para lograr este objetivo, se utiliza una ventana de generaciones que es analizada para determinar si el valor del coeficiente se debe incrementar o decrementar según corresponda. La función de *fitness* es calculada como sigue:

$$F(p) = f(p) + [P(p) \cdot \rho] \quad (6)$$

donde  $p \in (0,2)$  y es calculado analizando el promedio de restricciones violadas en la ventana de generaciones llamada  $AV_w$  y el promedio de violaciones en la generación actual llamado  $AV_c$ , como sigue a continuación:

$$V_g = \frac{AV_w - AV_c}{AV_w} \cdot 100 \quad (7)$$

teniendo en cuenta el grado de violación  $V_g$ ,  $\rho$  es actualizado como:

$$\rho = \begin{cases} \rho + k & \text{si } V_g < -0,25 \\ \rho - k & \text{si } V_g > 0,25 \\ \rho & \text{en otro caso} \end{cases} \quad (8)$$

donde  $k$  es establecido en 0,025.

## 2.5.2 Algoritmos de Reparación

Los algoritmos de reparación implementados para resolver nuestro problema están basados en los enfoques Lamarkiano, Baldwiniano y Annealing, y son enfoques que transforman soluciones no factibles en soluciones factibles, por medio de la aplicación de procesos específicos. Los mismos se describen a continuación:

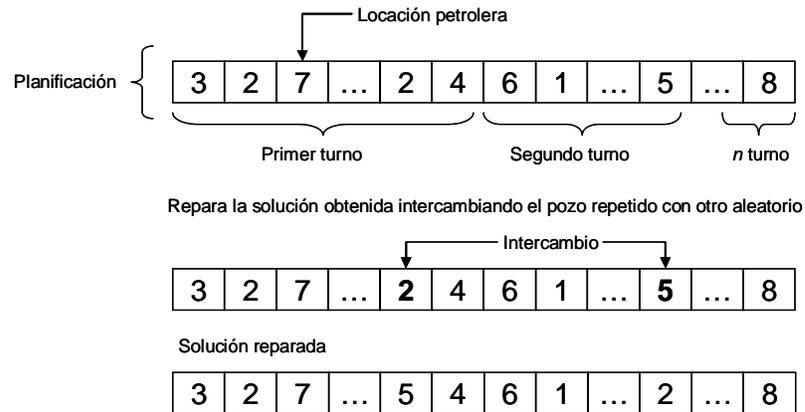
- **Enfoque Lamarkiano (LMCK):** este consiste en modificar genéticamente una solución no factible y transformarla en una solución factible. Con este enfoque cada individuo no factible de una nueva población es reparado. Este proceso se repite en todas las generaciones.
- **Enfoque Baldwiniano (BLDW):** Este es un enfoque menos destructivo para las soluciones no factibles, en donde se permite combinar el aprendizaje y la evolución. La idea es que la solución no factible solo sea reparada para su evaluación, es decir que el objetivo de reparar la solución es solo con el fin de obtener el *fitness* de esa solución reparada, pero en la población, el cromosoma del individuo o la solución no es modificada (permanece no factible). Estudios analíticos y empíricos indican que esta técnica reduce la velocidad de convergencia del algoritmo evolutivo y permite la convergencia hacia un óptimo global [Whitley D. et al. 1994].
- **Enfoque Annealing (ANNL):** Este enfoque está basado en el concepto de *Simulated Annealing* [Kirkpatrick S. et al. 1983]. Aquí las soluciones no factibles son aceptadas con cierta probabilidad. En las primeras etapas (es decir en las primeras generaciones) las soluciones no factibles tienen una alta probabilidad de ser aceptadas para formar parte o contribuir con la siguiente generación, pero esta probabilidad va disminuyendo conforme pasan las generaciones. En el caso de que una solución no factible no sea aceptada, entonces se procede a reparar dicha solución. Esta probabilidad de aceptación se puede calcular de la siguiente manera:

$$Pr ob_A = 1 - \left( \frac{g}{G} \right)^2 \quad (9)$$

donde  $g$  es la generación actual y  $G$  es el número total de generaciones.

En todos los enfoques mencionados, el proceso de reparar una solución consiste en intercambiar una locación petrolera, situada en una ubicación incorrecta dentro de la planificación, con otra locación petrolera elegida aleatoriamente y que sea distinta a la primera locación. Este proceso se repite, hasta que todas las locaciones cumplen con las restricciones del problema.

En síntesis, cuando no se cumple una restricción y se aplica el enfoque Lamarkiano, una solución no factible es reemplazada por la solución reparada, mientras que en el enfoque Baldwiniano la solución solo es reparada para su evaluación, pero no para su reemplazo. Por último el enfoque Annealing permite con una probabilidad de que una solución no factible forme parte o contribuya a la siguiente generación, siendo esta probabilidad condicionada por el paso del tiempo, es decir a mayor número de generaciones transcurridas, menor es la probabilidad que una solución no factible sea aceptada. La Figura 2 muestra un ejemplo de reparación de una solución no factible para cualquiera de los tres enfoques vistos.

**Figura 2.** Proceso de reparación de una solución no factible.

Básicamente el proceso de reparación consiste en los siguientes pasos:

- Verificar si una locación es visitada más de una vez en un mismo turno.
- Si se cumple que la locación es visitada dos o más veces en un mismo turno, entonces la segunda ocurrencia de la locación es intercambiada aleatoriamente con otra locación distinta que no pertenezca al mismo turno. Lo mismo sucede con el resto de las ocurrencias de la locación en cuestión, hasta que se cumpla la restricción de que una locación petrolera solo puede ser visitada en un turno a la vez.

### 3. RESULTADOS

Para probar la performance de los enfoques aplicados para este problema se realizaron tres experimentos, considerando un yacimiento petrolero con 110 locaciones petroleras y 2 conjuntos diferentes **OW1** y **OW2** respectivamente. Se definieron 6 instancias, descritas en la Tabla 1, en donde la columna “**Ins**” representa el nombre de la instancia y **|OW1|** y **|OW2|** representa la cantidad de locaciones petroleras que deberían ser visitadas dos veces y tres veces respectivamente.

**Tabla 1.** Instancias usadas en los experimentos determinados por OW1 y OW2.

Ins	OW1	OW2
I <sub>1</sub>	15	0
I <sub>2</sub>	17	0
I <sub>3</sub>	19	0
I <sub>4</sub>	8	8
I <sub>5</sub>	9	9
I <sub>6</sub>	10	10

Dicho de otra forma, la Tabla 1 se puede interpretar de la siguiente manera, para la primera instancia  $I_1$ , de las 110 locaciones petroleras, 15 de ellas volverán a ser visitadas en un instante dado, por lo que la solución que representa la planificación del recorrido estará compuesta por 125 locaciones. Esto quiere decir que a nivel de representación de la solución, cada locación repetida se contempla como una locación diferente, pero al momento de analizar las restricciones se contemplan como iguales, por ejemplo en el caso cuando se quiere determinar si estas dos locaciones (de nuevo, representadas en la solución como locaciones diferentes pero con la intención de que se analice como una locación que se vuelve a visitar) se visitan en un mismo turno. Para la segunda instancia  $I_2$ , de las 110 locaciones petroleras, 17 de ellas volverán a ser visitadas, por lo que la solución estará compuesta por 127 locaciones. Para la tercera instancia  $I_3$ , de las 110 locaciones petroleras, 19 de ellas volverán a ser visitadas, por lo que la solución estará compuesta por 129 locaciones. A partir de la cuarta instancia existen locaciones petroleras que se visitarán dos veces y locaciones que se visitarán 3 veces. Por lo tanto, para  $I_4$ , de las 110 locaciones petroleras, 8 de ellas volverán a visitarse una vez y otras 8 de ellas se visitarán dos veces, por lo que la solución estará compuesta por 134 locaciones. En este caso la cantidad de locaciones estará dada por la siguiente ecuación:

$$L = TL + OW1 + 2 * OW2 \quad (4)$$

donde  $L$  es la cantidad total de locaciones petroleras,  $TL$  es una constante que representa la cantidad de locaciones sin repetir (establecida en 110), y en donde  $OW1$  y  $OW2$  es la cantidad de locaciones que se repiten 2 y 3 veces respectivamente. A partir de esta ecuación la cantidad total de locaciones para las instancias  $I_5$  e  $I_6$  es de 137 y 140 respectivamente.

Es importante aclarar que todos los experimentos realizados se han hecho aplicando técnicas tanto de reparación como de aplicación de funciones de penalidades, para el manejo de restricciones. Como el propósito de este informe es la descripción del trabajo realizado con los algoritmos de reparación, no se explicará en detalle los experimentos realizados aplicando funciones de penalidad correspondientes a las instancias  $I_1$ ,  $I_2$ , y  $I_3$ , ver [Villagra A. et al. 2010], pero si se mostrarán los resultados con el fin de hacer una comparativa entre los distintos manejos de restricciones.

Los parámetros requeridos para el AG se explican con detalle en la sección 2.4. La Tabla 2 muestra los valores de los parámetros de entrada que se establecieron para el EA-MCMP-SRI, en donde el parámetro “Individuos” es el tamaño de la población, “Generaciones” es la cantidad de generaciones que evolucionará la población, la “Probabilidad de recombinación” es la probabilidad de aplicar el mecanismo PMX, la “Probabilidad de mutación” es la probabilidad que tiene cada individuo de que sus alelos muten, “Operaciones de recombinación” es la cantidad de veces que se realizará el cruzamiento para llenar el pool de hijos requerido por el mecanismo MCMP-SRI, “Cantidad de padres para el crossover” en la cantidad de individuos, generados aleatoriamente, requeridos para completar el pool de apareamiento, y finalmente “Cantidad de ejecuciones por instancia” es la cantidad de corridas independientes para el AG. Los valores para estos parámetros no fueron elegidos al azar, sino que fueron seleccionados en base a estudios experimentales anteriores.

**Tabla 2.** Parámetros del Algoritmo Genético.

Parámetro	Valor
Individuos	15
Generaciones	3000
Probabilidad de recombinación	0,65
Probabilidad de mutación	0,05
Operaciones de recombinación	16 ( $n_1$ )
Cantidad de padres para el crossover	18 ( $n_2$ )
Cantidad de ejecuciones por instancia	30

En el primer experimento se compararon los resultados obtenidos con las tres funciones de penalidad usando las instancias  $I_1$ ,  $I_2$  y  $I_3$ . El segundo experimento básicamente consiste en comparar los resultados obtenidos después de ejecutar los diferentes algoritmos de reparación, utilizando los conjuntos de instancias mencionados anteriormente. Y finalmente el tercer experimento consiste en comparar las mejores técnicas del experimento 1 contra las mejores del experimento 2, pero en base a las instancias  $I_4$ ,  $I_5$  y  $I_6$ .

La Tabla 3 muestra los resultados obtenidos para el primer experimento, donde se muestra la siguiente información para cada una de las funciones de penalidad implementadas (STT, DYN y ADP). La columna “**Ins**” es el nombre de la instancia, la columna “**Mediana**” representa la mediana de kilómetros recorridos, la columna “**Promedio**” representa el promedio de kilómetros recorridos y finalmente, la columna “**Evals**” representa el número de evaluaciones en miles hechas por cada enfoque.

**Tabla 3.** Resultados obtenidos por STT, DYN y ADP para las instancias  $I_1$ ,  $I_2$  e  $I_3$ .

Ins	STT			DYN			ADP		
	Mediana	Promedio	Evals	Mediana	Promedio	Evals	Mediana	Promedio	Evals
-	444,94	447,45	<b>8432</b>	<b>443,93</b>	<b>438,46</b>	9863	444,73	441,99	9043
$I_2$	446,84	453,30	9091	<b>444,81</b>	450,34	9186	445,66	<b>448,39</b>	<b>8001</b>
$I_3$	457,99	459,01	9145	<b>446,60</b>	<b>449,83</b>	9473	447,23	453,74	<b>8919</b>

En la Tabla 3 se puede observar que para todas las instancias, DYN obtiene los valores mínimos para la mediana de kilómetros recorridos y para dos de las tres instancias ( $I_1$  e  $I_3$ ) obtuvo los valores mínimos para el promedio de kilómetros recorridos. En cuanto al número de evaluaciones, para las instancias  $I_2$  e  $I_3$ , ADP obtuvo los valores mínimos.

La Tabla 4 muestra los resultados obtenidos para el segundo experimento, donde se muestra la siguiente información para cada enfoque (LMCK, BLDW, ANNL).

**Tabla 4.** Resultados obtenidos por LMCK, BLDW y ANNL para las instancias  $I_1$ ,  $I_2$  e  $I_3$ .

Ins	LMCK			BLDW			ANNL		
	Mediana	Promedio	Evals	Mediana	Promedio	Evals	Mediana	Promedio	Evals
$I_1$	448,24	454,23	<b>8955</b>	448,73	455,03	9093	<b>444,55</b>	<b>453,51</b>	9088
$I_2$	453,44	468,18	<b>8587</b>	460,20	476,70	9414	<b>447,50</b>	<b>455,02</b>	8692
$I_3$	468,98	468,14	<b>8702</b>	479,32	477,87	9335	<b>447,92</b>	<b>464,96</b>	8980

En la Tabla 4 se puede observar que los valores mínimos para la mediana y promedio de kilómetros recorridos fueron obtenidos por ANNL. Sin embargo, el mínimo número de evaluaciones fue obtenido por el enfoque LCMK para todas las instancias.

La Tabla 5 muestra los resultados obtenidos del tercer experimento. En este experimento se realizaron corridas utilizando los enfoques en los que se obtuvieron mejores resultados para los experimentos 1 y 2, con el fin de comparar los distintos enfoques para el manejo de restricciones. Del experimento 1, las técnicas que mejores resultados brindaron fueron las funciones de penalización Adaptativa (ADP) y Dinámica (DYN). Por lo que estas dos fueron comparadas con la mejor del experimento 2, la técnica basada en ANNL. En la Tabla 5 se puede observar que para todas las variables de las dos instancias ( $I_4$  e  $I_6$ ), ANNL obtuvo los valores mínimos.

**Tabla 5.** Resultados obtenidos por ADP, DYN, y ANNL para las instancias  $I_4$ ,  $I_5$  e  $I_6$

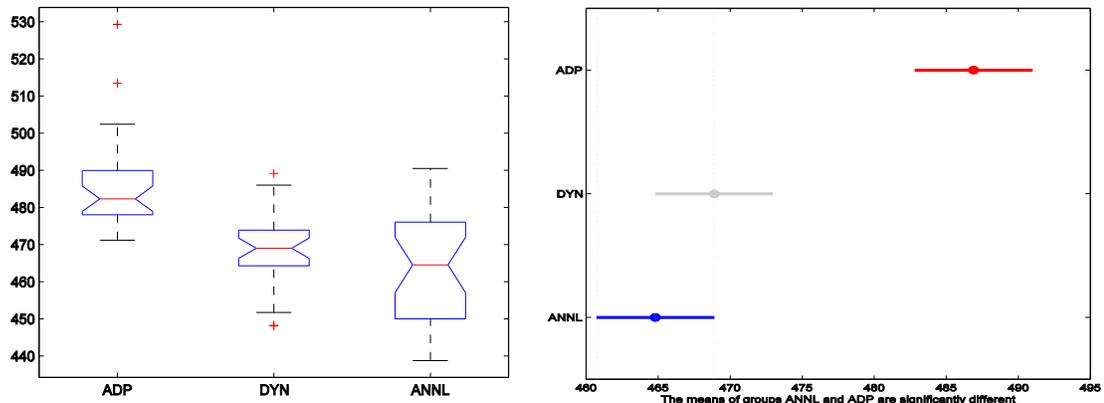
Ins	ADP			DYN			ANNL		
	Mediana	Promedio	Evals	Mediana	Promedio	Evals	Mediana	Promedio	Evals
-	482,29	486,89	9131	469,01	468,60	10165	464,42	464,88	6127
$I_4$	498,30	502,64	9139	478,62	480,47	9809	479,08	481,01	7077
$I_5$	514,36	514,51	9167	490,69	494,60	10057	490,23	492,61	7008

Para realizar un análisis más profundo de los resultados obtenidos, se realizó un análisis ANOVA. En la Figura 3(a) para la  $I_4$ , se puede observar que ANNL obtiene el mínimo valor entre las medianas de kilómetros recorridos. Sin embargo la mediana obtenida por DYN es muy similar y esto puede ser observado en la Figura 3(b), donde las diferencias entre ANNL y ADP son estadísticamente significativas, pero no existen diferencias estadísticamente significativas entre DYN y ANNL. Finalmente para la variable “Evals” se puede observar en la Figura 3(c) un menor esfuerzo computacional (evaluaciones) requerido por ANNL que el esfuerzo requerido por las otras dos técnicas.

En la Figura 4(a) para la  $I_5$  se puede observar que hay una pequeña diferencia entre las medianas de DYN y ANNL, con una mediana obtenida por DYN ligeramente mas pequeña que la obtenida por ANNL. Esto es confirmado en la Figura 4(b) en donde las diferencias estadísticas son detectadas entre ADP y las otras dos técnicas, ANNL y DYN. Finalmente para la variable “Evals” se puede observar en la Figura 4(c) que también para estas instancias se requiere un menor esfuerzo computacional por parte de ANNL con respecto a las otras técnicas.

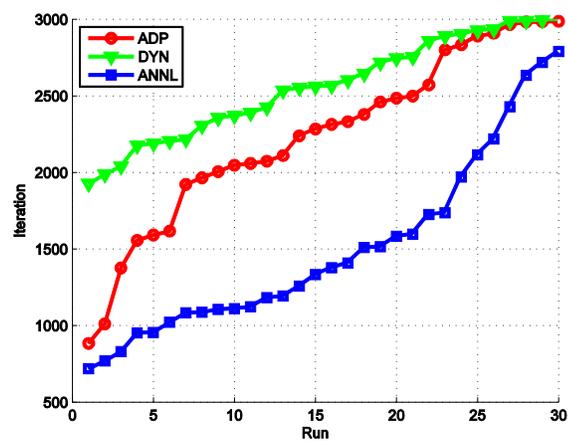
En la Figura 5 (a) para la  $I_6$  se puede observar que ANNL obtiene el mínimo valor para la mediana de kilómetros recorridos. Sin embargo la mediana obtenida por DYN es muy similar y con valores más compactos y esto puede ser observado en la Figura 3(b), donde las diferencias entre ANNL y ADP son estadísticamente significativas, pero no existen diferencias estadísticamente significativas entre DYN y ANNL. Finalmente para la variable “Evals” se puede observar en la Figura 5(c) un menor esfuerzo computacional (evaluaciones) requerido por ANNL que el esfuerzo requerido por las otras dos técnicas.

**Figura 3.** (a) Box-plot, (b) diferencias estadísticas, y (c) esfuerzo computacional para  $I_4$  con los enfoques ADP, DYN y ANNL respectivamente.



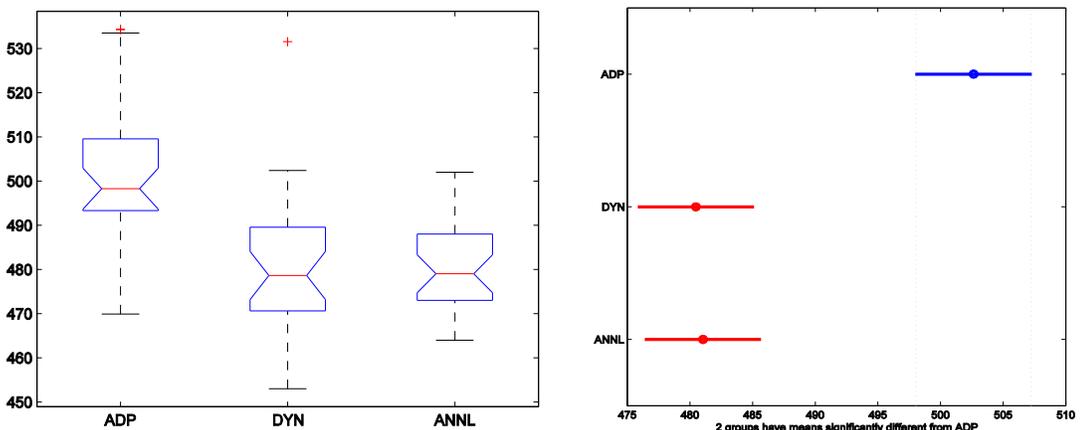
(a)

(b)



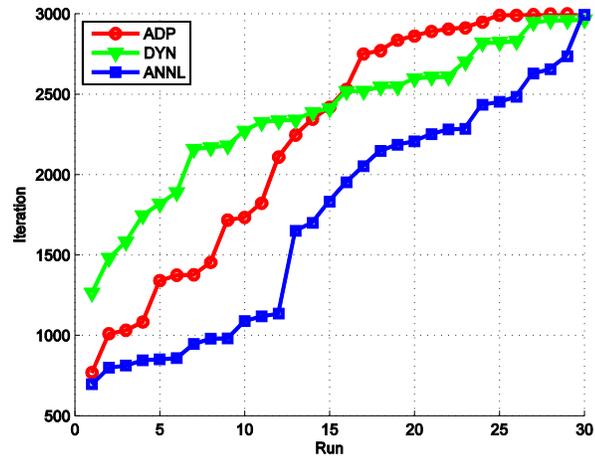
(c)

**Figura 4.** (a) Box-plot, (b) diferencias estadísticas, y (c) esfuerzo computacional para  $I_5$  con los enfoques ADP, DYN y ANNL respectivamente.



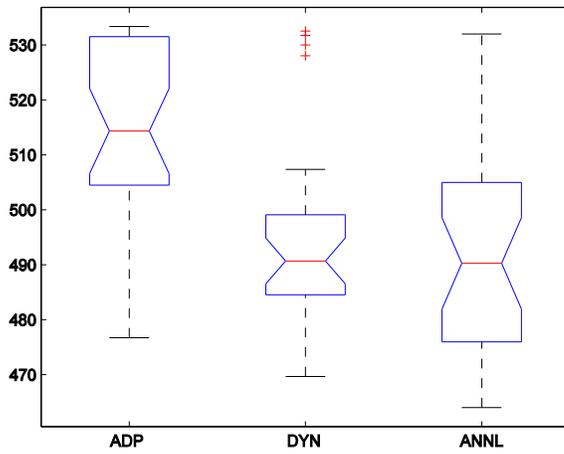
(a)

(b)

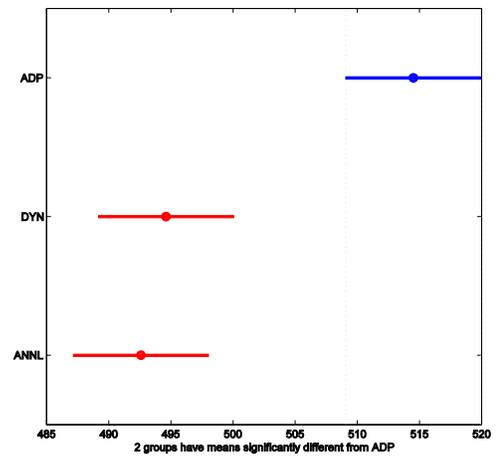


(c)

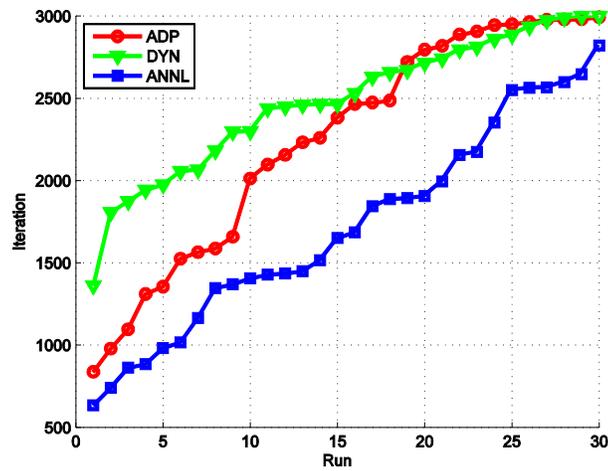
Figura 5. (a) Box-plot, (b) diferencias estadísticas, y (c) esfuerzo computacional para  $I_6$  con los enfoques ADP, DYN y ANNL respectivamente.



(a)



(b)



(c)

## 4. CONCLUSIONES

PAE es una aplicación construida con el objetivo de proveer una efectiva herramienta que facilite la planificación de las visitas de mantenimiento de locaciones petroleras, sujeta a restricciones. Los algoritmos evolutivos son metaheurísticas que usan modelos computacionales del proceso evolutivo. Para la planificación con restricciones de locaciones petroleras se usó una variante de un enfoque multirecombinativo llamado EA-MCMP-SRI, implementando diferentes técnicas de penalidad y técnicas de reparación para el manejo de restricciones.

De los experimentos llevados a cabo se puede remarcar que:

- En general la técnica de reparación Annealing (ANNL) y la técnica de penalización dinámica (DYN) obtienen valores similares para las variables de calidad analizadas (mediana y promedio de kilómetros recorridos).
- En referencia al esfuerzo computacional (evaluaciones) se puede decir que un 95% de certeza de que el enfoque Annealing requiere un menor número de evaluaciones con la misma calidad de soluciones que los otros enfoques.

Trabajos futuros incluirán el diseño, implementación y estudio de técnicas de manejo de restricciones más avanzadas, teniendo en cuenta los resultados observados en este trabajo. Además, la formulación de diferentes tipos de restricciones y planificaciones basados en múltiples equipos de mantenimiento. Se pretende en el futuro generar planificaciones (soluciones) que sean ejecutadas por varios equipos, proporcionando de esta manera una división del trabajo en el que múltiples equipos trabajan en paralelo para completar una planificación y así minimizar los tiempos de las planificaciones y maximizar la cantidad de visitas a aquellas locaciones que son críticas.

## 5. AGRADECIMIENTOS

Se agradece la cooperación del equipo de proyecto del LabTEM y la Universidad Nacional de la Patagonia Austral, de los cuales se recibe un apoyo continuo.

## 6. REFERENCIAS

Bäck T., Fogel D. B. and Michalewicz z., Eds., 2000. Evolutionary Computation 2, Advanced Algorithms and Operators. Institute of Physics Publishing, Bristol and Philadelphia, ISBN 0-7503-0665-3.

Beasley D. 1997. Why Evolutionary Computation? Possible applications of evolutionary computation, IOP Publishing Ltd and Oxford University Press pp A 1.2.

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

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*, 191(11-12):1245–1287.

De San Pedro M.E., Pandolfi D., Villagra A., Vilanova G., Gallard R. 2001; Stud and immigrants in multirecombined evolutionary algorithm to face weighted tardiness scheduling problems; CACIC'01 VII Congreso Argentino de Ciencias de la Computación, El Calafate, Argentina, octubre, pp. 1251-1258.

De San Pedro M.E., Villagra A., Lasso M., Pandolfi D., Diaz Vivar M., Gallard R. 2003; Solutions for the Weighted Number of Tardy Jobs in Single Machine Environments via Evolutionary Algorithms, CSITeA03 International Conference on Computer Science, Software Engineering Information Technology, E-Business and Applications, Rio de Janeiro, Brazil, pp. 438-443.

De San Pedro M.E., Lasso M., Villagra A., Pandolfi D., Gallard R., 2003; Solutions to the Dynamic Average Tardiness Problem in the Single machine Environments; CACIC'03 IX Congreso Argentino de Ciencias de la Computación, La Plata, Argentina, pp. 1251-1258.

Efrén Mezura – Montes Ed. 2009. *Constraint – Handling in Evolutionary Optimization*. Springer, ISBN 978-3-642-00619-7.

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.

Fogel D. 1997. *Why Evolutionary Computation? Introduction*, IOP Publishing Ltd and Oxford University Press pp A 1.1.

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

Goldberg, D. y Lingle R. 1987, Alleles, loci and the traveling salesman problem. *Proc. of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, pp. 154-159, Hillsdale, NJ.

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.

Lasso M., Pandolfi D., De San Pedro M.E., Villagra A., Gallard R., 2003; Heuristics to Solve Dynamic W-T problems in Single Machine Environments; CSITeA03 International Conference on Computer Science, Software Engineering Information Technology, E-Business and Applications, Rio de Janeiro, Brazil, pp. 432-437.

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

Pandolfi D., De San Pedro M., Villagra A., Vilanova G., Gallard R. 2002. Studs Mating Immigrants in Evolutionary Algorithm to Solve the Earliness-Tardiness Scheduling Problem . Cybernetics and Systems of Taylor and Francis Journal, (U.K.), pp 391-400.

Pandolfi D., Vilanova G., De San Pedro M.E, Villagra A.; Gallard R. 2001; Solving the single-machine common due date problem via studs and immigrants in evolutionary algorithms; *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Vol. III Emergent Computing and Virtual Engineering, pp. 409-413, Orlando, Florida.

Pandolfi D., Lasso M., De San Pedro M.E., Villagra A., Gallard R. 2003; Evolutionary Algorithms to solve average tardiness problems in the single machine environments; CSITeA03 International Conference on Computer Science, Software Engineering Information Technology, E-Business and Applications, Rio de Janeiro, Brazil, pp. 444-449.

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.

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

Smith A. and Coit D. 1997. Handbook of Evolutionary Computation, chapter Constraint Handling Techniques-Penalty Functions, page C 5.2. Oxford University Press and Institute of Physics.

Valdez J., Pandolfi D. y Villagra A, 2011. Manejo de Restricciones con Funciones de Penalidad para la Planificación del Mantenimiento de Locaciones Petroleras. Informe Científico Técnico. Universidad Nacional de la Patagonia Austral, UNPA.

Villagra A., Pandolfi D., Vilanova G., De San Pedro M.E; Gallard R. 2001; Adaptability of multirecombined evolutionary algorithms in the single-machine common due date problem; Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, Vol. III Emergent Computing and Virtual Engineering, pp. 401-404, Orlando, Florida.

Villagra A., Montenegro C., De San Pedro E., Lasso M. y Pandolfi D. (a), 2007. Restricciones en la Planificación del Mantenimiento de Locaciones Petroleras. CACIC, Corrientes, Argentina.

Villagra A., De San Pedro E., Lasso M., Montenegro C., y Pandolfi D. (b), 2007. Evolutionary Algorithm for the Oil Fields Preventive Maintenance Scheduling. WMSCI, Orlando.

Villagra A., Montenegro C., De San Pedro E., Lasso M., Pandolfi D. (c), 2007. Planificación con Restricciones del Mantenimiento de Locaciones Petroleras. RPIC (XII Reunión de Trabajo en Procesamiento de la Información y Control), Rio Gallegos.

Villagra A., Pandolfi D., Rasjido J., Montenegro C., Serón N. y Leguizamón G. 2010. Repairs Algorithms and Penalty Functions to Handling Constraints in an Evolutionary Scheduling. CACIC 2010.

Whitley, D., Gordon, S., and Mathias, K. 1994. Lamarckian evolution, the Baldwin effect and function optimization, Parallel Problem Solving from Nature - PPSN III, pp. (1994).