

ALGORITMOS EVOLUTIVOS PARALELOS DISTRIBUIDOS PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN

Natalia Stark, Carolina Salto

Laboratorio de Investigación en Sistemas Inteligentes (LISI)
Departamento de Informática - Facultad de Ingeniería
Universidad Nacional de La Pampa
Calle 110 Esq. 9 (6360) General Pico – La Pampa – Rep. Argentina
e-mail: { nstark, saltoc }@ing.unlpam.edu.ar
Te. / Fax: (02302) 422780/422372, Ext. 6302

RESUMEN

Los algoritmos evolutivos (AEs) se usan para buscar soluciones a problemas complejos. Una tendencia actual consiste en disponer de la mayor cantidad de recursos computacionales para alcanzar los resultados de forma más rápida por medio de un trabajo cooperativo.

Este trabajo presenta una versión distribuida de un algoritmo evolutivo. El algoritmo consta de un conjunto de islas o subpoblaciones cooperantes interconectadas con un proceso coordinador, representando una topología estrella. Todas las islas se conducen internamente de acuerdo a un plan reproductivo propio, en este caso un AE de estado estacionario, e intercambian un individuo a una cierta frecuencia mediante el proceso coordinador. Se compara experimentalmente el desempeño de esta propuesta con un modelo distribuido consistente en un anillo de islas evolutivas y con un AE panmítico a fin de identificar cuáles son sus aciertos y debilidades. Además se analiza el modelo propuesto frente a políticas de migración y reemplazo. Reportamos que el algoritmo propuesto presenta el mejor comportamiento de entre los evaluados al encontrar soluciones de mejor calidad y una disminución en el costo de búsqueda. Se validan los resultados sobre dos problemas de optimización como lo son Max-Unos y P-Picos.

PALABRAS CLAVES: Algoritmos Evolutivos Paralelos, Problemas de Optimización, políticas de migración.

Workshop de Agentes y Sistemas Inteligentes

I. INTRODUCCIÓN

La aplicación de Algoritmos Evolutivos (AEs) a problemas de optimización ha sido muy importante durante los últimos años [5]. Es posible encontrar este tipo de algoritmos para buscar soluciones a problemas complejos tales como tareas de optimización con restricciones, con ruido o con elevada epístasis y/o multimodalidad, para los cuales otras técnicas pueden insumir mucho tiempo y que, por lo general, proveen una única solución óptima. Normalmente la mayoría de los AEs utilizan una única población (panmícticos), aunque también existe una cierta tradición en el uso de poblaciones estructuradas, especialmente en el caso de implementaciones paralelas. La inclusión del paralelismo, distribución de tareas en varios procesadores, en el diseño de los AEs ha sido muy importante dando lugar a mecanismos de búsqueda y optimización mejorados: *algoritmos evolutivos paralelos (AEPs)*. Entre los AEPs más conocidos figuran los distribuidos y los celulares [6]. En el primer caso la población se divide en conjuntos de subpoblaciones o *islas* en las que se ejecutan AEs aislados. Entre estas islas se producen intercambios de individuos esporádicamente para introducir diversidad en cada una de ellas y guiar la búsqueda hacia las porciones más prometedoras del espacio. Este proceso es conocido como *migración*. La especificación de un AEP distribuido define el tamaño y el número de islas, la topología de conexión entre ellas, el porcentaje de migración (la fracción de la población que migra), la frecuencia de migraciones y la política para seleccionar emigrantes y reemplazar individuos existentes con inmigrantes, *políticas de migración y reemplazo*.

En el caso de los celulares se introduce el concepto de *vecindario*, de forma que un individuo sólo puede relacionarse con sus vecinos en el bucle reproductor que aplica los operadores de variación. La topología más popular es una rejilla toroidal en la que residen todos los individuos. En la Fig. 1 se muestran estos tres tipos de algoritmos: AE panmíctico con todos sus individuos en la misma población (a); estructurando la población se distingue entre AEs distribuidos (b) y AEs celulares (c).

Existen publicaciones con revisiones que pueden utilizarse para hacer un seguimiento de los AEPs en el tiempo, como por ejemplo [3, 4, 6, 8].

El objetivo de este trabajo es contrastar el comportamiento de dos modelos de AEPs distribuidos entre sí y con una versión secuencial a fin de identificar cuáles son sus aciertos y debilidades. El primer modelo distribuido es una propuesta propia, mientras que el segundo fue desarrollado en la Universidad de Málaga.

Con la propuesta propia de un modelo distribuido se pretenden conseguir beneficios en cuanto a la reducción del costo computacional para encontrar un valor óptimo. Además de mejorar la calidad del algoritmo al difundir eficientemente el mejor valor de aptitud encontrado en una determinada subpoblación al resto de las subpoblaciones que integran la población global.

Este documento se organiza como sigue. En la sección siguiente se exponen las características de los modelos distribuidos y el proceso de búsqueda de un algoritmo genético de estado estacionario. La sección III presenta los distintos algoritmos analizados. En la sección IV se describen los

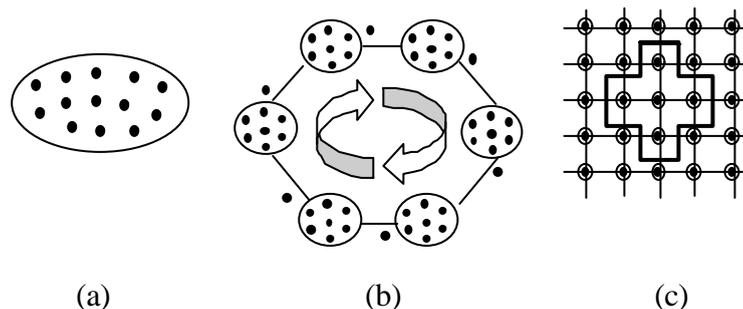


Fig. 1: AEs panmícticos (a) y estructurados (b) y (c).

problemas elegidos para realizar las pruebas. En la sección V se exponen los parámetros empleados para la ejecución de los algoritmos y las variables utilizadas para comparar su desempeño. Los resultados de las pruebas realizadas se presentan en la sección VI. Por último, en la sección VII se resumen algunas conclusiones y se sugieren futuras líneas de investigación.

II. CARACTERIZACIÓN DE LOS MODELOS DISTRIBUIDOS

En esta sección se presenta una descripción en pseudocódigo de un AG de estado estacionario (AGee) y una versión distribuida que puede realizarse a partir de él. (ver Fig. 2). Se muestra una versión simple, suponiendo el uso de selección, cruce, mutación y reemplazo.

A partir de esta versión simple es fácil obtener algoritmos más elaborados que incorporen operadores novedosos o dependientes del problema, así como variar el tipo, número y parámetros de los operadores.

El algoritmo que se emplea como caso de estudio es un AGee [1]. En el modelo básico de estado estacionario se genera en cada paso un nuevo individuo, *nuevo_indi*, aplicando selección, cruce, mutación y se lo inserta en la población junto con sus padres. Este nuevo individuo desplaza al peor individuo existente o a uno seleccionado en forma aleatoria. El reemplazo se puede realizar siempre o sólo si es mejor que el peor individuo existente.

En la versión distribuida síncrona (AGd) no se supone una topología de conexión predeterminada entre las subpoblaciones. Dicha topología podría referirse a cualquiera de las conocidas, por ejemplo anillo, estrella, árbol, etc.

El modelo distribuido presenta una fase de inicio elaborada al tener que generar las islas (y sus subpoblaciones iniciales) y controlar la migración.

```

[AGee] procedimiento ciclo_reproductor (ag)
inicio
    padre1= Seleccionar (ag.población)
    padre2 = Seleccionar (ag.población)
    Cruzar (ag.pc, padre1, padre2,nuevo_indi.crom)
    Mutar (ag.pm, nuevo_indi.crom)
    nuevo_indi.aptitud = ag.Evaluar (nuevo_indi.crom)
    Insertar_nuevo_indi (ag, nuevo_indi, [si mejor | siempre])
    Recolectar_estadísticas (ag)
fin procedimiento

[AGd]procedimiento AGee_distribuido
inicio
    Iniciar_sistema (ag)
    Generar_subpoblaciones (ag)
    Evaluar_subpoblaciones (ag)
    Recolectar_estadísticas_globales (ag)
para s = 1 hasta MAX_PASOS hacer
    para i = 1 hasta NUMERO_DE_ISLAS hacer
        Ciclo_reproductor (ag[i])
    fin para
    si Debe_migrar (ag, s) entonces
        Migrar (ag, [mejor | aleatorio]) // ¿quién migra?
        Recolectar_estadísticas_globales (ag)
    fin si
fin para
    Solución =mejor_indi_encontrado_durante_evolución
fin procedimiento

```

Fig. 2: Pseudocódigo de un AGee panmictico y su versión distribuida.

En el algoritmo, *solución* es aquel individuo con el mayor valor de aptitud encontrado durante todo el proceso de evolución

Aunque el criterio de terminación mostrado en la Fig. 2 consiste en completar un cierto número de pasos, MAX_PASOS, otro posible criterio es encontrar una solución (si se conoce de antemano).

En este trabajo se consideran dos modelos de AEPs distribuidos atendiendo a la forma en que las islas se interconectan. Ambos modelos aplican un mismo AGee. Uno de los modelos es una propuesta propia de AE con población global distribuida, en adelante denominada *modelo_estrella* (detalles del mismo se describen en la próxima sub-sección), y el otro es un modelo desarrollado por el grupo de investigación de Ingeniería de Software de la Universidad de Málaga [10] en adelante referenciado *modelo_anillo*. Para esta última versión se utilizó un paquete, denominado *arg*, que implementa un AE panmítico y su correspondiente distribuido. El AE contiene el código de un AGee que implementa selección por torneo, cruce clásico de un punto y mutación binaria. Utiliza una población inicial aleatoria formada por cromosomas binarios. El AE distribuido adiciona las modificaciones necesarias para establecer la conexión de las subpoblaciones de acuerdo a una topología de anillo unidireccional. De forma que cada subpoblación envía una copia del mejor individuo a la próxima isla en el anillo. Este individuo se incorpora en la subpoblación destino, reemplazando al peor individuo existente o a uno seleccionado al azar. El paquete de software fue desarrollado con una metodología orientada a objetos e implementado en JAVA.

A. Modelo Propuesto

El modelo propuesto consiste en un esquema distribuido de islas de población única. En general este modelo puede funcionar como un algoritmo no distribuido tradicional (una única isla) o como un algoritmo (paralelo) distribuido. El algoritmo consta de islas comunicantes en una topología de estrella. Todas las islas se conducen internamente de acuerdo a un plan reproductivo propio, AGee.

La fase de comunicación envía / recibe un único individuo. Cada isla migra la mejor solución hallada o una estructura seleccionada en forma aleatoria a un proceso central, denominado *coordinador*. El coordinador se encarga de coleccionar y seleccionar la mejor estructura de entre las recibidas y de difundir una copia de esa mejor estructura a las distintas islas. En cada subpoblación se reemplaza la peor estructura, o una seleccionada al azar, por la que llega desde el coordinador. Este proceso se denomina *migración* y se realiza en forma determinada cada una cierta cantidad de pasos, *frecuencia de migración*.

A continuación se describen los pasos en los que se basa el modelo propuesto de AE distribuido de islas de población única.

Primer paso: se divide la cantidad de trabajo entre múltiples módulos, es decir, se fracciona a la población entre la cantidad de islas. En la Fig. 3 se presenta un ejemplo de una población inicial distribuida en n islas.

Segundo paso: en cada isla, se aplican los pasos de un AGee (ver Fig. 4): selección de individuos para el apareamiento, cruce, mutación y reemplazo del individuo generado por otro de la subpoblación.

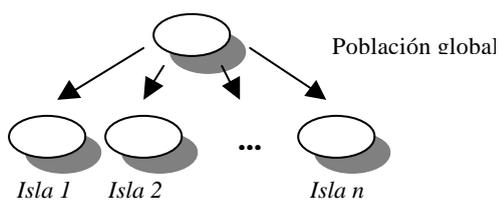


Fig. 3: Distribución de una población inicial en n islas

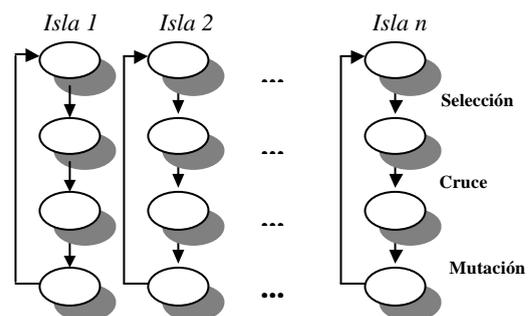


Fig. 4: Ejecución paralela.

Tercer paso: luego de un determinado número de pasos (frecuencia de migración) se interrumpe la ejecución del algoritmo. Cada isla envía al coordinador un individuo elegido de acuerdo a cierta política de migración. El coordinador colecciona los individuos enviados por cada isla y selecciona el mejor entre ellos (aquel con mayor aptitud). En la Fig. 5 se muestra el esquema de esta fase de comunicación.

Cuarto paso: una vez que el coordinador selecciona al individuo más apto de entre los recibidos envía una copia de este mejor individuo, individuo global, a todas las islas.

Las islas, luego de enviar un individuo al coordinador, esperan por la copia del individuo global. Cuando lo reciben, reemplazan el peor de su población o por uno elegido en forma aleatoria, según la política de reemplazo acordada. Vuelve a repetirse todo el proceso una determinada cantidad de pasos predefinidos para el algoritmo o hasta que alguna isla encuentre la solución, en caso de ser conocido el valor óptimo. La Fig. 6: Etapa de Sincronización, muestra este procedimiento y la etapa de sincronización entre las islas y el proceso central.

El modelo antes descrito ha sido implementado en JAVA. Para la fase de comunicación se empleó el protocolo RMI-IIOP.

III. DESCRIPCIÓN DE LOS ALGORITMOS UTILIZADOS

En esta sección se presentan los algoritmos empleados al resolver los problemas de optimización propuestos: un AE secuencial y distintos AEs distribuidos homogéneos, tanto a nivel de búsqueda como de ejecución. Por otra parte, se analiza el comportamiento de los algoritmos en distintos ambientes: distribuido en ambiente de máquina única y distribuido sobre 4 procesadores, para evaluar cuál de ellos presenta mejor desempeño. La paralelización a nivel físico se logró empleando una red LAN con cuatro computadoras personales homogéneas, entendiéndose como tales a aquellas que tienen una misma arquitectura y ejecutan el mismo sistema operativo. Concretamente las pruebas se han realizado sobre procesadores Pentium II con 64 MB de RAM que ejecutan Windows'98.

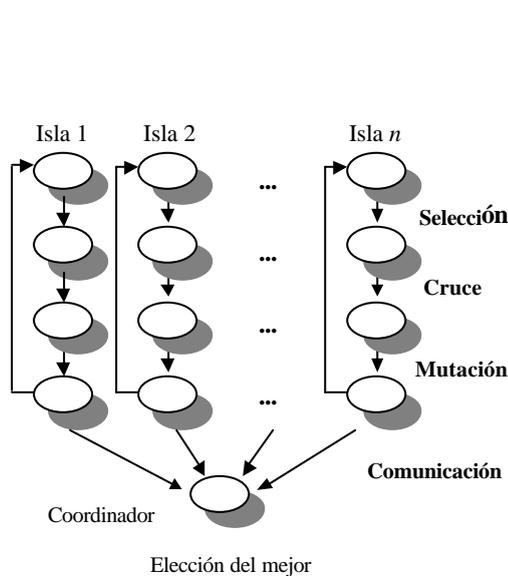


Fig. 5: Etapa de Comunicación.

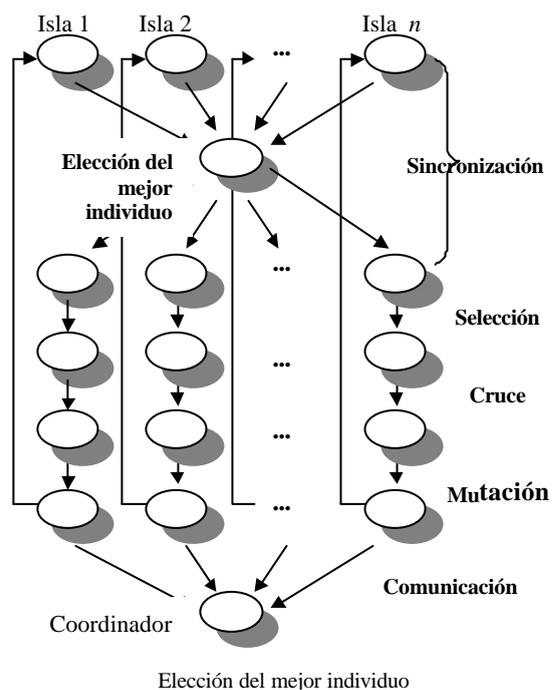


Fig. 6: Etapa de Sincronización.

Los algoritmos empleados fueron: **AE**: AE Panmítico de estado estacionario, **AEd1**: AE distribuido con topología de interconexión en forma de estrella sobre un procesador (según modelo_estrella), **AEd1p**: AE distribuido con topología de interconexión en forma de estrella sobre 4 procesadores, **AEd2**: AE distribuido con topología de interconexión en forma de anillo unidireccional sobre un procesador (según modelo_anillo), y **AEd2p**: AE distribuido con topología de interconexión en forma de anillo unidireccional sobre 4 procesadores.

El caso base AE utiliza una población de una determinada cantidad de individuos y genera en cada paso una cadena nueva aplicando los operadores de selección, cruce y mutación y la inserta en la población junto con sus padres desplazando siempre a la peor cadena existente.

Para los algoritmos AEd1 y AEd2 se emplearon 4 subpoblaciones distribuidas sobre la misma máquina. Para los algoritmos AEd1p y AEd2p se usa el mismo modelo descrito para los AE distribuidos pero trabajando en 4 máquinas diferentes. En todo caso, para los algoritmos distribuidos el tamaño de la población se fijó en un cuarto del total de la población que utiliza el algoritmo panmítico para mantener el tamaño de esta siempre fijo.

Con el objeto de observar el comportamiento de los AE distribuidos frente a las políticas de selección de emigrantes y reemplazo, se desarrollaron nuevos experimentos. Existen dos alternativas para seleccionar los individuos que emigran a una isla [7]: elegir uno al azar (A) o seleccionar el individuo con mejor aptitud en la isla (M); y dos opciones para reemplazar a los individuos existentes en la isla con los inmigrantes: elegir un individuo al azar (A) o reemplazar al peor (P). Teniendo en cuenta estas alternativas surgen cuatro combinaciones posibles de selección de emigrantes y reemplazo:

AA: migrar un individuo seleccionado al azar en cada isla y reemplazar un individuo existente en la subpoblación destino, elegido en forma aleatoria, con el inmigrante.

AP: migrar un individuo seleccionado al azar en cada isla y reemplazar el peor individuo de la subpoblación por el inmigrante.

MA: migrar el mejor individuo de cada isla y efectuar el reemplazo de un individuo elegido en forma aleatoria por el inmigrante.

MP: migrar el mejor individuo de cada isla y reemplazar el peor de la isla destino por el inmigrante.

IV. PROBLEMAS ESTUDIADOS

Con el fin de analizar el comportamiento del proceso de búsqueda de los algoritmos propuestos se han elegido dos problemas de prueba.

El primero, *Max-Unos (Onemax)* es un problema clásico de maximización que consiste en hallar una cadena binaria que tenga todos los bits en uno (ecuación 1). Formalmente, este problema puede definirse como encontrar un vector

$$\bar{x} = \{x_1, x_2, \dots, x_n\}, \text{ con } x_i \in \{0,1\}, \text{ que maximice } F(\bar{x}) = \sum_{i=1}^n x_i \quad (1)$$

Aparentemente es un problema simple. Sin embargo, aunque es fácil mejorar el valor de aptitud en las primeras generaciones, se convierte en un problema complicado de optimización, pues es difícil poner en uno el último bit que está en cero. Esto hace que en muchos casos el algoritmo se estanque, por lo cual resulta esencial mantener la diversidad.

El Segundo, *P-Picos (Ppeaks)* es un generador de problemas multimodales [2]. Un generador de problemas es una tarea fácilmente parametrizable con un grado de dificultad afinable, de manera que permite generar instancias con una complejidad tan grande como se desee. Cada vez que se ejecuta el algoritmo se resuelve una instancia distinta, lo que permite obtener conclusiones sobre la clase del problema como tal y no sobre una instancia concreta, por muy difícil que esta pudiera resultar.

La idea de P-Picos consiste en generar P cadenas aleatorias de N bits. Estas cadenas representan la localización de P cimas o picos en el espacio de búsqueda. Para evaluar una cadena de bits arbitraria, primero se localiza el pico más cercano (en el espacio de *HammingD*). El valor de aptitud de una cadena es el número de bits que la cadena tiene en común con el pico más cercano, dividido por N (ecuación 2).

$$F(\vec{x}) = \left\{ \frac{1}{N} \max_{i=1}^P (N - \text{Ham} \min \text{gD}(\vec{x}, \text{Pico}_i)) \right\} \quad (2)$$

Usando un mayor (o menor) número de cimas se obtienen problemas más (o menos) epistáticos, es decir, problemas en los que existe una fuerte interacción entre genes en un cromosoma. En estos casos, la epístasis mide el grado en el cual la contribución del valor de un gen depende de los valores de otros genes.

V. EXPERIMENTACIÓN

Esta sección se organiza en dos subsecciones. En la primera se exponen los parámetros utilizados para la ejecución de los algoritmos y en la segunda se describen las variables empleadas para comparar el desempeño de los mismos.

A. Parametrización

Con el fin de hacer una comparación justa entre los algoritmos se ha utilizado la misma parametrización con los dos problemas; los detalles se resumen en la Tabla I. Se analizó el costo de resolver un problema midiendo el número de evaluaciones de la función objetivo.

Para la experimentación se han realizado, para cada algoritmo y para cada problema, series de 30 ejecuciones independientes.

La condición de parada para todos los algoritmos consistió en encontrar una solución o, en su defecto, llegar a realizar 200.000 pasos. Para cada experimento se consideró una población de 512 individuos que, para el caso de los algoritmos distribuidos, fue dividida en 4 islas o subpoblaciones de 128 individuos. Sobre cada una de las 4 islas se aplicó un AGee que emplea selección por torneo binario para hacer la elección entre el conjunto de padres, durante el proceso evolutivo aplica el cruce simple de un punto (SPX) con una probabilidad de 0.85% y a cada hijo generado le aplica mutación por inversión binaria con una probabilidad de 0.19%. Los valores de los parámetros fueron seleccionados como la mejor combinación de probabilidades luego de varias pruebas iniciales. Se substituyó al individuo considerado en cada instante, independientemente del valor de aptitud de su descendiente. La frecuencia de migración para los AEs distribuidos se calcula multiplicando el tamaño de la población de la isla por 32, lo que indica que la migración se produce con una frecuencia de 4096 pasos, como fue sugerido en [10].

B. Variables de desempeño

Las variables de desempeño utilizadas para evaluar el comportamiento de los algoritmos son:

- *#Evaluaciones Promedio*: número promedio del total de evaluaciones del AGee.

Tabla I - Parametrización utilizada en los algoritmos.

<i>Tamaño de la población</i>	512 individuos
<i>Selección de padres</i>	Torneo Binario
<i>Cruce</i>	SPX, pc = 0.85%
<i>Mutación</i>	Binaria, pm=0.19%
<i>Frecuencia Migración</i>	4096 pasos
<i>Reemplazo</i>	Siempre
<i>Máximo de Pasos</i>	200.000

- *#Evaluaciones Promedio Mejor Aptitud*: número promedio de evaluaciones necesarias para terminar aquellos experimentos que alcanzan la solución óptimas (esfuerzo o costo de calcular una solución). En el caso de los algoritmos distribuidos, el total de evaluaciones se calculó como la sumatoria de las evaluaciones realizadas en cada una de las islas
- *Mejor Aptitud Promedio*: promedio de las mejores soluciones halladas de cada ejecución.
- *Aptitud Promedio Poblacional*: media de los valores de aptitud promedio de los individuos de cada población final.
- *Peor Aptitud Promedio*: promedio de los peores valores hallados de cada ejecución.
- *Tiempo [s]*: tiempo promedio de cada ejecución, medido en segundos.
- *Evaluaciones/segundo*: promedio de evaluaciones por unidad de tiempo para cada ejecución.
- *# de islas promedio*: promedio de islas que encontraron el valor óptimo, para el caso de los AE distribuidos.
- *% éxitos*: porcentaje del total de experimentos (ejecuciones) en los cuales la solución óptima ha sido alcanzada (tasa de éxitos).

VI. RESULTADOS EXPERIMENTALES

En esta sección se presentan y analizan los resultados obtenidos por los distintos algoritmos al resolver los dos problemas de optimización. Se organiza en dos subsecciones correspondientes a los distintos problemas de prueba. La primer subsección hace referencia al problema Max-Unos y la segunda al problema P-Picos. En cada una de ellas se describe: a) el estudio comparativo del AE que se ejecuta en forma panmítica contra los AEs distribuidos y de estos entre sí y b) el desempeño de los AEs distribuidos basados en el modelo_estrella frente a políticas de selección de emigrantes y reemplazo.

La política de migración para los algoritmos distribuidos consistió en seleccionar el mejor individuo para migrar, mientras que la política de reemplazo se centró en sustituir el peor individuo de la población destino por el inmigrante.

En las tablas donde se muestran los resultados, las columnas sombreadas se corresponden con los resultados de los algoritmos que distribuyen el trabajo sobre 4 procesadores

A. *Max-Unos*

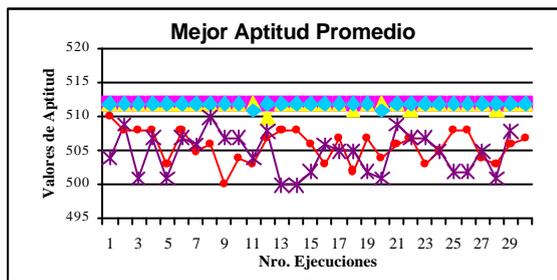
Para este problema se optó por una longitud de cromosoma de 512 genes. El valor óptimo esperado es 512, cada uno de los bits de la cadena binaria serán iguales a uno, por lo que se decidió detener el algoritmo cuando se encuentra dicho valor o cuando se alcanza el máximo de pasos.

ANÁLISIS ALGORITMOS PANMÍTICO Y DISTRIBUIDOS.

En la TABLA II se muestran los promedios de los resultados alcanzados por los distintos algoritmos. Analizando los valores mejor aptitud promedio puede concluirse que la mejor calidad de las soluciones halladas corresponde al algoritmo AEd1, debido a que el valor óptimo, 512, se alcanza en cada una de las ejecuciones (% éxitos = 100%). Sin embargo, el porcentaje de error del mejor valor encontrado en AEd1p y AEd2 con respecto al óptimo, calculado como $(\text{abs}(\text{valor_óptimo} - \text{mejor_valor}) / \text{valor_óptimo}) * 100$, es de tan sólo 0,033 y 0,013 respectivamente. Puede decirse que, con estos algoritmos se obtiene un alto porcentaje de valores óptimos en contrapartida del AE y del AEd2 que no han alcanzado el valor óptimo en las ejecuciones realizadas (% éxitos = 0%). Comparando los valores de mejor aptitud promedio, aptitud promedio poblacional y peor aptitud promedio para cada algoritmo puede decirse que AE, AEd1 y AEd1p muestran mayor concentración de los individuos de la población alrededor del mejor valor, respecto a los algoritmos AEd2 y AEd2p. Estas observaciones se pueden corroborar en Fig. 8 y Fig. 9 donde se muestran valores de mejor aptitud promedio y aptitud promedio poblacional alcanzados por los algoritmos en cada una de las ejecuciones.

TABLA II - RESULTADOS ALCANZADOS PARA MAX-UNOS EMPLEANDO LOS DISTINTOS ALGORITMOS.

PROBLEMA MAX-UNOS	AE	AEd1	AEd1p	AEd2	AEd2p
# Evaluaciones Promedio	200512,00	430206,52	587043,20	593052,40	800516,00
# Evaluaciones Promedio Mejor Aptitud	200512,00	103793,94	145533,77	130332,97	200129,00
Mejor Aptitud Promedio	505,05	512,00	511,83	511,93	505,73
Aptitud Promedio Poblacional	504,95	511,80	511,18	509,91	503,03
Peor Aptitud Promedio	504,10	511,53	510,93	507,07	499,97
Tiempo [s]	131,10	296,47	98,57	374,33	127,98
Evaluaciones / segundos	1525,04	1395,63	5919,67	1573,00	6256,82
# Islas solución promedio	-	3,30	0,93	1,07	0,00
% Éxitos	0,00	100,00	86,67	93,33	0,00



—■— AEd1 —▲— AEd1p —◆— AEd2 —●— AEd2p —*— AE

Fig. 8 : Mejor Aptitud Promedio alcanzada por los algoritmos.

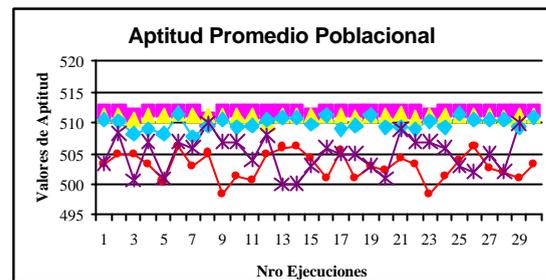


Fig. 9: Aptitud Promedio Poblacional alcanzada por los algoritmos

Por otra parte, en la TABLA II se puede apreciar que el algoritmo que se ejecuta en forma panmíctica, AE, requiere mayor cantidad de evaluaciones promedio para alcanzar la mejor solución (costo de búsqueda) que aquellos que distribuyen el trabajo; siendo superior el desempeño de los algoritmos que distribuyen el trabajo sobre un procesador, AEd1 y AEd2.

El costo de búsqueda es menor para los algoritmos distribuidos desarrollados según el modelo_estrella ya que se observa una menor cantidad en el número de evaluaciones promedio que realizan para encontrar el mejor valor en comparación con el modelo_anillo.

Los tiempos de ejecución de los algoritmos AEd1 y AEd2 son altos debido a que las ejecuciones de estos se realizaron en ambiente de máquina única; pero esto no se refleja en la cantidad de evaluaciones por segundo. AEd1p, presenta claramente una disminución del costo computacional, expresado en segundos, ya que el tiempo que tarda el algoritmo en hallar la mejor solución es menor respecto de los otros algoritmos.

La diferencia entre la cantidad de islas que encuentran la solución en los dos modelos distribuidos demuestra que la propagación del individuo más apto en el modelo_estrella es más eficiente que la propagación del mejor individuo en el modelo_anillo. Esto se debe a la forma de interconexión ya que en el modelo_estrella el coordinador siempre propaga, en forma directa, el mejor valor recibido de las islas.

ANÁLISIS POLÍTICAS DE MIGRACIÓN Y REEMPLAZO

Finalmente, con el objeto de observar el comportamiento del AEP frente a las políticas de selección de emigrantes y reemplazo, se decidió realizar otros experimentos empleando el modelo_estrella.

En la Tabla III se resumen los resultados obtenidos. Observando los valores promedios para todas las combinaciones se puede decir que se obtiene mejor calidad al seleccionar para migrar el mejor individuo de cada isla independientemente de la forma en que se realice el reemplazo en la isla destino, esto puede apreciarse analizando los valores de mejor aptitud promedio y aptitud promedio poblacional. Por otra parte, cuando se migró el mejor individuo se obtuvo mejor calidad al

Tabla III: RESULTADOS OBTENIDOS AL COMBINAR POLÍTICAS DE MIGRACIÓN Y REEMPLAZO PARA MAX-UNOS.

PROBLEMA MAX-UNOS	AEd1-AA	AEd1p- AA	AEd1- MA	AEd1p- MA	AEd1-AP	AEd1p- AP	AEd1- MP	AEd1p- MP
# Evaluaciones Promedio	390727,71	590004,23	392834,93	569005,00	393389,93	614030,63	430206,53	587043,20
# Evaluaciones Promedio Mejor Aptitud	104704,17	146114,00	98141,66	141106,67	97535,76	152115,93	103793,93	145533,77
Mejor Aptitud Promedio	511,97	511,87	512,00	511,90	512,00	511,77	512,00	511,83
Aptitud Promedio Poblacional	511,73	510,64	511,86	511,25	511,45	510,61	511,80	511,18
Peor Aptitud Promedio	511,27	509,47	511,30	511,00	510,59	509,67	511,53	510,93
Tiempo [s]	296,37	99,50	287,31	98,73	284,38	101,13	296,47	98,57
Evaluaciones / segundo	1422,87	5903,23	1388,17	5937,67	1375,00	6032,90	1395,63	5919,67
# Islas solución promedio	3,20	0,90	2,80	1,00	3,03	0,80	3,30	0,93
% Éxitos	96,67	86,67	100	90,00	100	76,67	100	86,67

reemplazar un individuo seleccionado en forma aleatoria en la isla destino; y menor costo computacional y de búsqueda al reemplazar al peor. Independientemente de las combinaciones entre las políticas de migración y reemplazo, puede apreciarse una ligera pérdida de calidad en las soluciones y menor tiempo de cómputo al emplear procesamiento en paralelo.

B. P-Picos

Para resolver el problema P-Picos se han realizado experimentos empleando una instancia de $P=100$ cimas de longitud de $N = 100$ bits cada una [2]. El máximo valor de aptitud que se puede conseguir es de 1.0. Al igual que para el problema Max-Unos, se decidió detener el algoritmo cuando se encuentra el mencionado valor óptimo o cuando se alcanza el máximo de pasos.

ANÁLISIS ALGORITMOS PANMÍTICO Y DISTRIBUIDOS.

En la Tabla IV se muestran los promedios de los resultados alcanzados para cada una de las 30 ejecuciones realizadas por los algoritmos. Si bien todas las versiones del algoritmo encuentran el valor óptimo, analizando los valores de aptitud promedio poblacional se observa mejor calidad en las soluciones halladas para los AE distribuidos desarrollados según el modelo_estrella.

El número de evaluaciones promedios para hallar el óptimo disminuye considerablemente al utilizar un algoritmo distribuido en lugar de uno que trabaje en forma panmítica. En particular la menor cantidad de evaluaciones fue reportada con AEd1, ya que requirió solo el 32% del número de evaluaciones promedio que empleó AE.

Observando los valores de aptitud promedio poblacional se desprende que los algoritmos AE, AEd1 y AEd1p son los que mantienen una mayor concentración de individuos en la zona próxima al valor óptimo, esto a su vez, puede ser corroborado observando la distribución del valor de aptitud de la media poblacional (ver Fig. 10) y de los peores individuos (ver Fig. 12) en cada una de las ejecuciones.

Tabla IV - Resultados alcanzados para AE, AEd1 y AEd1p en la resolución de P-Picos.

PROBLEMA P-PICOS	AE	AEd1	AEd1p	AEd2	AEd2p
# Evaluaciones Promedio	87749,76	115617,20	255524,62	92006,00	244253,66
# Evaluaciones Promedio Mejor Aptitud	87749,76	28805,63	62517,00	55882,20	63635,80
Mejor Aptitud Promedio	1,000	1,000	1,000	1,000	1,000
Aptitud Promedio Poblacional	0,990	0,992	0,992	0,963	0,985
Peor Aptitud Promedio	0,990	0,988	0,990	0,930	0,972
Tiempo [s]	122,800	218,300	95,759	129,180	107,492
Evaluaciones / segundos	711,000	603,167	2643,414	712,695	2845,852
# Islas solución promedio	-	1,200	1,103	1,200	1,000
% Éxitos	100,00	100,00	100,00	100,00	100,00

VII. CONCLUSIONES Y TRABAJO FUTURO

En este documento se presentó un modelo de AEP distribuido consistente de un conjunto de islas que intercambian soluciones mediante un proceso coordinador formando una topología estrella. Se ha comparado su desempeño con un modelo distribuido, que consta de un anillo de islas evolutivas, y con un AE panmítico a fin de identificar cuáles son sus aciertos y debilidades. Además se ha analizado el modelo propuesto frente a políticas de migración y reemplazo. La ejecución de los AEPs se ha realizado atendiendo distintos ambientes de trabajo, distribuido en ambiente de máquina única y distribuido sobre 4 procesadores, para evaluar cuál de ellos presenta mejor desempeño. A partir del análisis de los resultados alcanzados se infiere que la incorporación de la distribución a un AE proporciona una mejora en cuanto a la calidad de las soluciones halladas, reduciendo el número de evaluaciones necesarias para encontrar valores óptimo o de muy buena calidad.

La interconexión de las islas en forma de estrella permite hallar mejor calidad en las soluciones, disminuyendo el costo computacional y el costo de búsqueda debido a que posibilita una rápida difusión de una buena solución encontrada en una isla a las restantes en comparación con una interconexión en forma de anillo.

Los mejores resultados se obtienen al seleccionar para migrar el mejor individuo independientemente de la política de reemplazo en la isla destino, mientras que los resultados de menor calidad corresponden a la combinación: migrar al azar reemplazar al peor.

El paralelismo produce una leve pérdida de calidad de las soluciones en comparación de la distribución en ambiente de máquina única, pero en esta los tiempos de ejecución son mayores.

Como próximos pasos se propone analizar la influencia de la migración en los AEPs y la de variar la topología de interconexión entre las islas durante la evolución a fin de tomar las ventajas de cada una. Además, analizar el comportamiento del modelo propuesto sobre problemas de mayor complejidad es un punto importante que abordaremos en el futuro..

RECONOCIMIENTOS

Este trabajo ha sido realizado con el apoyo de la Universidad Nacional de La Pampa. Se reconoce especialmente la cooperación de los integrantes del grupo del proyecto LISI por aportar nuevas ideas y críticas constructivas.

REFERENCIAS

- [1] Alba, E. Tesis Doctoral: *Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos*. Universidad de Málaga. España. 1999.
- [2] Alba, E.; Dorronsoro, B. *Auto-Adaptación en Algoritmos Evolutivos Celulares. Un Nuevo Enfoque Algorítmico*. Segundo Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. 2003.
- [3] Alba E., Tomassini. M. *Parallelism and Evolutionary Algorithms*. En IEEE Transactions on Evolutionary Computation, vol 6, pág. 443-462, Octubre 2002.
- [4] Alba E.; Troya J. M. *A Survey of Parallel Distributed Genetic Algorithms*. En Complexity vol.4, pág.31-52,1999.
- [5] Bäck, T., Fogel, D., Michalewicz, Z. *Handbook of Evolutionary Computation*. Oxford University Press. 1997.
- [6] Cantú-Paz, E., *A Survey of Parallel Genetic Algorithms*. En *Calculaterurs Paralleles, Réseaux et Systèmes Répartis*, 10(2), pág. 141-171, 1998.
- [7] Cantú-Paz, E., *Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms*. *Journal of Heuristics*. 7(4), pág. 311-334. 2001.
- [8] Chipperfield, A., Fleming, P. *Parallel Genetic Algorithms*, en *Handbook of Parallel and Distributed Computing*, A. Y. Zomaya, Ed. New York, NY: McGraw-Hill, pág. 1118-1143, 1996.
- [9] De Jong, K., Potter, M., Spears, W. *Using Problem Generators to Explore the Effects of Epistasis*. En *Proceedings of the Seventh International Conference on Genetic Algorithms*, pág 338-345. 1997.
- [10] GISUM: Grupo de Ingeniería de Software de la Universidad de Málaga, España. <http://polaris.lcc.uma.es>.