

Una Estrategia para la Selección de Individuos Recibidos en la Migración en un Algoritmo Genético Distribuido

Germán Dupuy¹ - Carolina Salto¹⁻²

¹Facultad de Ingeniería - Universidad Nacional de La Pampa – Argentina - ²CONICET

germandupuy24@gmail.com - saltoc@ing.unlpam.edu.ar

Abstract. La estrategia de migración juega un rol importante en el diseño de un algoritmo evolutivo distribuido. En este trabajo, se impone que los individuos recibidos en una subpoblación por los efectos de la migración formen parte de las parejas para la recombinación. De esta manera, el nuevo material genético que incorporan estos nuevos individuos se puede propagar a la próxima generación. El algoritmo evolutivo distribuido se evalúa con varias instancias del problema de planificación de software. Los resultados indican que en la mayoría de los casos, el algoritmo propuesto es capaz de obtener soluciones de calidad similar a un algoritmo evolutivo distribuido tradicional pero en una menor cantidad de evaluaciones.

1 Introducción

Los algoritmos genéticos (AGs) [1][2] mantienen una población de soluciones que evolucionan de acuerdo a la aplicación de operadores genéticos, siguiendo la idea de la supervivencia de los individuos más aptos. El grado de adaptación de un individuo se evalúa de acuerdo al problema a resolver, mediante una función de aptitud. Para problemas complejos, la población estará compuesta por individuos de tamaño considerable, por consiguiente su evaluación será costosa en términos computacionales, lo que supone una utilización considerable de recursos. Una forma sencilla de minimizar los requerimientos es recurrir a los AGs distribuidos (dAG), los cuales modifican el comportamiento típico de un algoritmo secuencial ya que en la mayoría de los casos utilizan una población estructurada [3]. El uso de técnicas de procesamiento paralelo y distribuido no sólo implica mejoras desde el punto de vista de la eficiencia sino también desde el punto de vista de la calidad de la búsqueda genética [4] [5].

En un dAG, la población se divide en un conjunto de subpoblaciones (islas), entre las que se realizan intercambios esporádicos (migraciones) entre ellas. En este tipo de algoritmos, además de tener en cuenta los parámetros típicos de un AE, se debe establecer una política de migración, definiendo la topología de comunicación entre las islas, la frecuencia de migración, la cantidad de migrantes, y la forma en la que se integran las soluciones migradas. Estos parámetros afectan directamente el rendimiento del dAG.

El foco del trabajo está en definir una estrategia para los individuos que se incorporan a una población debido al intercambio producto de la migración. La propuesta se centra en la obligatoriedad de elección de los nuevos individuos recibidos en una subpoblación

como padre para la recombinación. Esto marca un contraste con las reglas tradicionales de un dAG, donde el individuo recibido compite con el resto de los individuos de la subpoblación en el proceso de selección de los padres para participar en la recombinación. Esto puede ocasionar que estos individuos no se seleccionen como padre, con lo cual se estaría perdiendo el posible nuevo material genético que podrían aportar.

La nueva propuesta se compara con un dAG tradicional a fin de determinar su efectividad para mejorar la calidad de soluciones en una menor cantidad de evaluaciones. Para analizar el comportamiento de nuestra propuesta se considera el problema de planificación de proyectos de software (PPS) [5], el cual representa uno de los mayores desafíos de la ingeniería de software.

El resto del artículo está organizado como sigue. La sección 2 describe las principales características de los dAG. La sección 3 presenta el dAG propuesto y explica su funcionamiento. Luego, en la sección 4 se describe el problema considerado (PPS) para evaluar la propuesta y la parametrización utilizada. Finalmente en las secciones 5 y 6 se presentan los resultados y las conclusiones arribadas a partir de estos, respectivamente.

2 Algoritmos Genéticos Distribuidos

La idea básica de los dAGs [6] se basa en particionar la población en varias subpoblaciones semi-aisladas, cada una de las cuales tiene asociada un AG independiente, explorando diferentes regiones prometedoras del espacio de búsqueda. Estas subpoblaciones pueden, ocasionalmente, interactuar con sus vecinas a través del intercambio de pocos individuos, simulando un proceso migratorio por medio del cual se puede inyectar nuevo material genético y evitar así una convergencia prematura. Esto asegura que se comparta material genético de tanto en tanto a través del sistema genético. La Figura 1 ilustra la distribución de las poblaciones y la topología de intercomunicación, en este caso anillo en donde la isla i envía individuos a la isla $i+1$ y recibe de la isla $i-1$ utilizando una comunicación asíncrona [7], de modo que al momento de la migración, la isla envía el individuo y procede a la próxima iteración sin esperar recibir otro de la isla vecina.

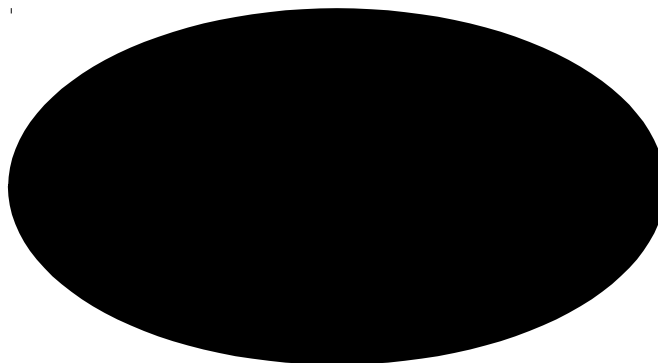


Figura 1. Modelo isla

La estructura elemental de un dAG i se muestra en el Algoritmo 1. Cada dAG i genera aleatoriamente la población inicial $P_i(t)$ de μ soluciones y las evalúa. Luego, la población entra en un ciclo evolutivo que consiste de la aplicación de operadores genéticos, para crear λ descendientes ($P'_i(t)$). Este ciclo incluye una fase adicional de intercambio de individuos con un conjunto de algoritmos vecinos, denotados como dAG j . Finalmente cada iteración termina seleccionando μ individuos para construir la nueva población $P_i(t+1)$, a partir de la población $(\mu + \lambda)$. La mejor solución es identificada como el mejor individuo que se encontró para maximizar la función de fitness.

La mayoría de los trabajos referentes a estudios sobre la política de migración en un dAG hacen referencia a cuál es el periodo de migración más adecuado para resolver determinado problema [8] [9], a cuál es la estrategia para seleccionar los individuos [9], a cuál es la topología más adecuada [10] o a establecer estrategias para la adaptación de algunos de estos parámetros [11][12]. Además existen estudios que indican cuál es la mejor estrategia a seguir para seleccionar el emigrante y cuál es la forma de ingresarlo en la población destino [9].

Algoritmo 1 - AG distribuido elemental (dAG i)

```

 $t = 0$ ; {current generation}
initialize ( $P_i(t)$ );
evaluate ( $P_i(t)$ );
while ( $t < max_{generations}$ ) do
     $P'_i(t) = evolve(P_i(t));$  {recombination and mutation}
    Evaluate ( $P'_i(t)$ );
     $P'_i(t) = send/receive\ individuals\ from\ dAG_j;$  {interaction with neighbours  $j$ }
     $P_i(t+1) = select\ new\ population\ from\ P_i(t) \cup P'_i(t);$ 
     $t = t+1$ ;
endwhile

```

3 Nuestra propuesta

Tradicionalmente, en un dAG se producen intercambios entre las islas con el objetivo de introducir alguna diversidad en las subpoblaciones, y de esta manera prevenir el quedar atrapados en óptimos locales. Los individuos que se aceptan en la población destino reemplazan a otros locales de acuerdo al criterio establecido en la política de migración, pasando a formar parte de la población actual. Estos nuevos individuos compiten con el resto en la etapa de reproducción, pudiendo ser o no elegidos.

A partir de este concepto, proponemos una variante de un dAG tradicional, la cual se centra en obligar la elección de las soluciones inmigrantes para formar parte del pool de padres que participan de la etapa de recombinación. A este algoritmo lo denominaremos dAG_MasP (dAG con Migrantes como Padres). La razón detrás de forzar esta elección es que estos nuevos individuos pueden aportar nuevo material genético que se ha perdido en la población actual.

La propuesta trabaja de la siguiente manera. Si en la generación anterior se produjo la recepción de un individuo y éste es aceptado e incorporado en la población local, se presenta la situación graficada para $P_i(t)$ (este nuevo individuo está resaltado) en la Figura 2. En la etapa de selección de las parejas para la reproducción, se impone que la primera pareja estará formada por uno de los individuos recibidos y otro nativo (círculo negro) seleccionado según el operador de selección establecido. En caso de que

existiesen más individuos migrantes, este proceso de conformación de parejas se repite hasta considerar todos ellos. Una vez conformadas las parejas, el proceso evolutivo continúa en la forma tradicional (recombinación, mutación, selección para el reemplazo).

isla *i*

Figura 1. Procedimiento realizado por el dAG_MasP durante la etapa de recepción de individuos desde la isla vecina

4 Parametrización

En esta sección, se presenta la información necesaria para reproducir la experimentación realizada en este trabajo de investigación. Primero introduciremos el problema usado para validar el comportamiento de nuestra propuesta: problema de planificación de proyectos software. Finalmente, se presentarán los parámetros usados por nuestro algoritmo.

4.1 Problema de planificación de proyectos de software

El problema de planificación de proyectos de software (PPS) es tradicionalmente uno de los principales problemas a los que se deben enfrentar los administradores de proyectos. Se tiene un conjunto E de empleados y un conjunto T de tareas que requieren ciertas habilidades para ser realizadas. Además se cuenta con un grafo de precedencia de tareas (TPG) que indica las tareas a completar antes de iniciar otra. Cada empleado posee un conjunto de habilidades, recibe un salario mensual y tiene un grado máximo de dedicación al proyecto. Cada tarea está caracterizada por un conjunto de habilidades necesarias para poder ser llevadas a cabo y un esfuerzo expresado en persona-mes.

El objetivo del problema es, entonces, asignar empleados a tareas, minimizando tanto el costo como la duración del proyecto, asegurando una calidad mínima. Se debe considerar que cada tarea la realiza al menos una persona y que las habilidades de los empleados que realizan cierta tarea deben cubrir el conjunto de habilidades requeridas por esa tarea. Si alguna de estas restricciones no se cumplen, se considera que la solución al problema no es factible. Recientemente, Minku ha propuesto una mejora, en la que se resuelve el obstáculo del trabajo extra, mediante la normalización del grado de dedicación [13].

La calidad de una solución depende de tres factores: la duración y el costo del proyecto y la factibilidad de la solución. La función de aptitud para medir la calidad de una solución x es la siguiente:

$$f(x) = \begin{cases} \frac{1}{q} & \text{si la solución es factible} \\ \frac{1}{(p+q)} & \text{en caso contrario} \end{cases}$$

Donde $q = w_{\text{costo}} \times \text{costo} + w_{\text{tiempo}} \times \text{tiempo}$ y $p = w_{\text{pena}} + w_{\text{t_inc}} \times t_{\text{inc}} + w_{\text{h_nocub}} \times h_{\text{nocub}}$. Existen parámetros predefinidos como w_{costo} , w_{tiempo} , w_{pena} , $w_{\text{t_inc}}$, $w_{\text{h_nocub}}$. Mientras que t_{inc} es la cantidad de tareas que no tienen empleados asociados y h_{nocub} es la cantidad de habilidades requeridas no cubiertas para realizar todas las tareas de proyecto, tiempo es la duración del proyecto y costo es el producto entre el salario del empleado y el tiempo de dedicación al proyecto. Como los dos términos p y q aparecen en el denominador y el objetivo es minimizarlos, entonces $f(x)$ se debe maximizar.

Una solución al problema se puede representar por medio de una cadena binaria que representa el grado de dedicación de cada empleado a cada tarea. Se considera que el grado de dedicación máxima de cada empleado es igual a 1 (ningún empleado trabaja horas extras). Como la dedicación de cada empleado es un valor real, el intervalo de dedicación se discretiza en ocho valores uniformemente distribuidos, tomando para ello tres bits. En consecuencia, la dimensión de la cadena binaria es igual a $|E| \times |T| \times 3$. Este tipo de representación permite el uso de operadores genéticos tradicionales.

4.2 Parametrización y metodología

Con el objetivo de evaluar nuestra propuesta dAG_MasP, se ha comparado con un dAG tradicional (dAG_trad), en el cual no se impone la condición de obligatoriedad en la selección de los individuos recibos en la etapa de reproducción.

Los parámetros comunes a ambos algoritmos son los que se describen a continuación. Ambos algoritmos utilizan una topología de anillo conformado por 4 islas, con un tamaño de subpoblación de 16 soluciones tentativas. La cantidad máxima de generaciones está determinada en 500. Los operadores genéticos usados son: selección por torneo binario, operador de cruce de un punto y mutación *bit_flip*. La probabilidad de cruce es de 0.9 mientras que la de mutación es de 0.05. Para construir la población de la nueva generación se usa selección proporcional considerando los $(\mu + \lambda)$ individuos. Estos valores fueron tomados de [14]. Se realizaron varias pruebas preliminares para calibrar el dAG en relación a la frecuencia de migración y la cantidad de individuos a migrar. Para esta experimentación se utilizó el dAG_trad y se consideraron frecuencias de 10, 25 y 50 generaciones y se migraron 1, 3 y 5 individuos cada vez, resultando 9 posibles combinaciones. Para cada una de ellas se realizaron 30 ejecuciones independientes para determinar la mejor combinación. Los resultados indicaron que la mejor combinación es intercambiar 3 individuos cada 10 generaciones, para el problema en cuestión, con significancia estadística.

Los algoritmos fueron implementados dentro de MALLBA [15], una librería de software desarrollada en C++ que permite el rápido prototipado de algoritmos paralelos e

híbridos. El sistema de computación utilizado para la experimentación consiste de una máquina AMD Phenom X3 a 2.66 GHz y 2 GB de RAM, ejecutando la versión 2.6.27.7 del kernel de Linux.

Debido al carácter no determinista del algoritmo, cada variante fue ejecutada 30 veces en forma independiente sobre 298 instancias de complejidad creciente, obtenidas a través del generador de instancias disponible públicamente en <http://tracer.lcc.uma.es/problems/psp>. Cada instancia representa un escenario distinto de la vida real, las cuales están organizadas en 6 grandes grupos, cuyas diferencias están dadas en: el número de empleados, el número de tareas, el grafo de precedencia de tareas y las habilidades necesarias para ejecutar cada tarea y las que posee cada empleado.

5 Resultados y discusión

Esta sección muestra los resultados obtenidos en la experimentación, a fin de analizar el impacto producido al forzar la elección de los nuevos individuos ingresados en el paso de migración. Comencemos con el análisis de la calidad de las soluciones obtenidas por dAG_MasP y dAG_trad al resolver el PPS.

La Tabla muestra el porcentaje de veces que cada algoritmo encuentra en promedio las mejores soluciones para cada uno de los grupos de instancias. En dicha tabla se resalta el porcentaje más alto para cada grupo de instancia. La última columna (ANOVA|K-W) representa los valores p calculados al realizar la prueba ANOVA o Kruskal-Wallis, según corresponda, sobre la calidad de las soluciones, con el objetivo de determinar la significancia estadística. En este trabajo se considera un nivel de significancia del 0.01. En el caso de existir diferencias estadísticas significativas entre los algoritmos, éstas se muestran con el símbolo “(+)”, mientras que en caso contrario se usa el símbolo “(-)”. En la mayoría de los casos, dAG_MasP encuentra, en promedio, mejores soluciones que dAG_trad. Cabe destacar que para el B6, el más amplio de los casos de prueba, dAG_MasP encuentra el mejor promedio, salvo para N con valor 12. En general se observa que la variante algorítmica dAG_MasP presenta un mejor desempeño en las instancias que se tiene en cuenta la restricción asociada a las habilidades, esto se puede apreciar claramente en B3 y B4, así como en el B6, exceptuando al último grupo (N=12), en donde el 56% de las instancias obtienen mejores valores de fitness con dAG_trad. Por razones de espacio no se incluye información sobre la cantidad de soluciones factibles obtenidas por ambos algoritmos, pero vale indicar que el porcentaje es similar en ambos algoritmos. Si bien las diferencias estadísticas entre la calidad de los resultados obtenidos por dAG_MasP y dAG_trad no son significativas (como se puede apreciar en la última columna de la Tabla 1), el mayor porcentaje de mejores soluciones obtenidas por dAG_MasP marca una superioridad de nuestra propuesta, sugiriendo que las nuevas soluciones recibidas por migración aportan material genético de interés en la población actual.

Tabla 1. Porcentaje de veces que dAG_MasP y dAG_trad encuentran mejores soluciones

Instancias	dAG MasP	dAG trad	ANOVA K-W
B1	50	50	-
B2	33	67	-
B3	100	0	-
B4	65	35	-
B5	33	67	-

B6	N=8	62	38	-
	N=9	52	48	-
	N=10	54	46	-
	N=11	62	38	-
	N=12	44	56	-

Otra métrica a evaluar es el esfuerzo numérico de las dos opciones algorítmicas, medido en la cantidad de evaluaciones que cada algoritmo necesita realizar en promedio para obtener las mejores soluciones. La figura 3 muestra las evaluaciones promedio de cada algoritmo para cada grupo de instancias. En 8 de los diez distintos grupos de instancias, dAG_MasP obtiene en promedio las mejores soluciones en una leve menor cantidad de evaluaciones. Esto marca la superioridad de la propuesta dAG_MasP al obtener buenas soluciones con un menor esfuerzo numérico que un dAG tradicional. Sumando a esto, vale destacar que los tiempos totales de ejecución de dAG_MasP y dAG_trad no presentan diferencias, por lo que el proceso de selección de los nuevos individuos ingresados a la población para formar parte del conjunto de padres no hace incurrir al algoritmo en un mayor tiempo de cómputo (por razones de espacio no se muestran las tablas correspondientes).

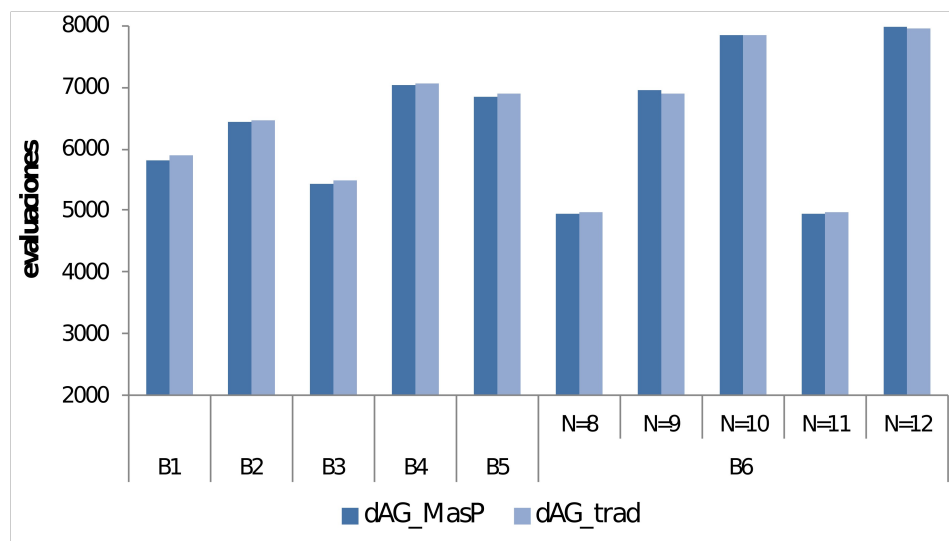


Figura 1. Cantidad de evaluaciones promedio de cada algoritmo y grupo de instancias

6 Conclusiones

Una nueva estrategia para tratar los individuos que ingresan a una subpoblación por migración se propone en este trabajo para ser utilizada en un algoritmo evolutivo distribuido. El algoritmo resultante, dAG_MasP, considera como padre para la recombinación aquellos nuevos individuos ingresados por una operación de migración. Este algoritmo ha sido evaluado en un amplio conjunto de instancias del problema de planificación de proyectos de software y comparado con un algoritmo evolutivo

distribuido tradicional. Los resultados indican que además de encontrar soluciones de buena calidad, dAG_MasP las encuentran en una menor cantidad de evaluaciones comparado con un dAG tradicional.

Como trabajo futuro se propone el desarrollo de un operador genético más específico del problema. También se pondrá atención en el mecanismo de manejo de restricciones, tal como está planteado se usa un mecanismo de penalización pero se podrían considerar mecanismos más avanzados. Además del estudio experimental para proponer otros pesos para la función de aptitud.

7 Reconocimientos

Los autores agradecen el apoyo de la UNLPam y ANPCYT (proyecto 09F-049) y proyecto PICTO-2011 0278.

8 Bibliografía

- [1] M. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin: Springer, 1996.
- [2] T. Bäck, D. Fogel y A. Michalewicz, *Handbook of Evolutionary Computation*, New York: Oxford University Press, 1997.
- [3] M. Tomassini, *Spatially Structured Evolutionary Algorithms*, Springer, 2005.
- [4] E. Alba y M. Tomassini, «Parallelism and evolutionary algorithms,» *IEEE Transactions on Evolutionary Computation*, vol. 6, n° 5, pp. 443-462, 2002.
- [5] E. Alba y J. Chicano, «Software project management with GAs,» *Information Sciences*, vol. 177, pp. 2380-2401, 2007.
- [6] R. Tanese, «Distributed genetic algorithms,» de *Proceedings of the Third International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1989.
- [7] E. Alba y J. Troya, «Analyzing synchronous and asynchronous parallel distributed genetic algorithms,» *Future Generations of Computer Systems*, vol. 17, n° 4, pp. 451-465, 2001.
- [8] K. Skolicki y K. De Jong, «The influence of migration sizes and intervals on island models,» de *Proceedings of the Conference of Genetic and Evolutionary Computation*, 2005.
- [9] E. Cantú-Paz, «Migration Policies, Selection Pressure, and Parallel Evolutionary Algorithms,» *Journal of Heuristics*, vol. 7, pp. 311-334, 2001.
- [10] E. Alba y G. Luque, «Theoretical models of selection pressure for dEAs: topology influence,» de *Proceedings of the IEEE International Conference on Evolutionary Computation*, 2005.
- [11] Bororska, P. and Iazarova, M, «Migration Policies for Island Genetic Models on Multicomputer Platform,» de *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Dortmund, 2007.
- [12] Osorio, K.; Luque, G.; Alba, E., «Distributed evolutionary algorithms with adaptive migration period,» de *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, Córdoba (España), 2011.
- [13] L. Minku, D. Sudholt y X. Yao, «Evolutionary algorithms for the project scheduling problem: runtime analysis and improved design,» de *Proceedings of the International Conference on Genetic and Evolutionary Computation*, 2012.
- [14] G. Dupuy, N. Stark y C. Salto, «Algoritmo evolutivo para el problema de planificación en proyecto de desarrollo de software,» de *Actas del XIX Congreso Argentino de Ciencias de la*

Computación, 2013.

- [15] A. E, «A Library of Skeleton for Combinatorial Optimisation,» de *LNCS*, Berlin, 2002.
- [16] P. Spiessens y B. Manderick, «A massively parallel genetic algorithm,» de *Proceedings of the Fourth International Conference on Genetic Algorithms*, San Francisco, CA, USA, 1991.
- [17] E. Alba y B. Dorronsoro, *Cellular Genetic Algorithms*, Springer, 2008.