

Metaheurísticas secuenciales y distribuidas: Adaptación de parámetros y entornos de ejecución

Carolina Salto, Gabriela Minetti, Natalia Stark, Carlos Bermudez, Hugo Alfonso, German Dupuy
 Laboratorio de Investigación en Sistemas Inteligentes (LISI)
 Facultad de Ingeniería - Universidad Nacional de La Pampa
 Calle 110 Esq. 9 (6360) General Pico - La Pampa - Rep. Argentina
 Te. / Fax: (02302) 422780/422372, Int. 6302

Resumen

La contribución de esta línea de investigación es proponer mecanismos adaptativos que puedan guiar el cambio de ciertos parámetros de los algoritmos evolutivos, tanto secuenciales como paralelos, usando información del estado del proceso de búsqueda. Otro aspecto importante es el referido al entorno de ejecución de estos algoritmos, a tales efectos se analiza la incidencia del uso de plataformas heterogéneas, tanto en la calidad de los resultados finales como en los tiempos de ejecución.

CONTEXTO

Esta línea de investigación se desarrolla en el marco del proyecto de investigación "Resolviendo problemas complejos con técnicas metaheurísticas avanzadas", dirigido por la Dra. Carolina Salto, y llevado a cabo en el Laboratorio de Investigación de Sistemas Inteligentes (LISI), de la Facultad de Ingeniería de la Universidad Nacional de La Pampa. Este proyecto mantiene desde hace varios años una importante vinculación con investigadores de la Universidad Nacional de San Luis (Argentina) y de la Universidad de Málaga (España) con quienes se han realizado varias publicaciones conjuntas.

I. INTRODUCCIÓN

La eficiencia de un algoritmo evolutivo (AE) depende de muchos parámetros, tales como tamaño de la población, representación de los individuos, estrategias de selección, probabilidades de los operadores, número total de generaciones, entre otros. Determinar cuál es el valor más apropiado de un parámetro para un determinado problema es en sí mismo un problema complejo, debido a que existe una interacción no lineal entre los parámetros y no existe un modelo matemático de tal interacción.

Entre los parámetros críticos para el rendimiento de un AE podemos mencionar la probabilidad de mutación p_m . Valores grande de p_m transforman al AE en un algoritmo de búsqueda puramente aleatorio, pero por otro lado es necesario algún grado de mutación para prevenir la convergencia prematura del AE a soluciones subóptimas. En la literatura se han sugeridos muchos valores para p_m , los cuales se derivan de la experiencia o prueba y ensayo [4], [6], [8], [13]. Es difícil, sino imposible, hallar un seteo apropiado para el valor de p_m a fin de alcanzar un rendimiento óptimo, debido a que las probabilidades varían con el problema que se intenta resolver,

y también están influenciadas con las diferentes etapas del proceso genético.

Esta problemática de seteo de parámetros también está presente en los AEs distribuidos. En particular, los AEs distribuidos introducen otro operador responsable de la estructura del algoritmo y del esquema de intercomunicación entre las subpoblaciones o islas (política de migración), conocido como el operador de migración. Por lo tanto, se necesitan parámetros adicionales a los de un AE secuencial; esta situación aumenta la complejidad en el uso de la técnica. Como parámetros que definen la política de migración podemos encontrar:

- periodo de migración: el periodo de evolución aislada de las subpoblaciones
- tasa de migración: el número de individuos que se intercambian.
- elección / reemplazo de los migrantes: estrategias para la selección de los individuos que se envían en una operación de migración y para la sustitución de los individuos locales con los entrantes.
- topología: este parámetro define el vecindario de cada subpoblación.

Aunque muchos parámetros regulan la migración, la mayoría de ellos son estándares y ampliamente usados, como una topología de intercomunicación en anillo, enviar uno o pocos individuos, reemplazar el peor individuo de la población receptora, y usar comunicación asíncrona para obtener buenos valores de speedup.

Para determinar el valor óptimo de los parámetros del AE, tanto secuencial como distribuido, se pueden considerar al menos dos opciones: ajuste (*tuning*) de parámetros y control de parámetros [7]. La primera opción está relacionada con hacer una experimentación exhaustiva para determinar, de una manera empírica, el valor del parámetro que permite obtener los mejores resultados para el problema en cuestión. Esta tarea implica un tiempo de seteo extra, la cual es computacionalmente muy costosa y raramente reportada en la literatura. En esta línea, se han reportado varias alternativas, tal como diseño de experimentos [14], F-RACE [5] y REVAC [12]. La segunda opción consiste en hacer cambios en los valores de los parámetros durante la ejecución del AE. Esto requiere valores iniciales para los parámetros y estrategias de control, las cuales pueden ser determinísticas, adaptativas o

auto-adaptativas. Estas estrategias manejan información que influyen los valores de los parámetros durante la búsqueda y definen un conjunto de criterios para producir los cambios. La línea de investigación que se describe en este trabajo utiliza la técnica de control de parámetros, en particular se plantean estrategias adaptativas.

Cuando se estudia un algoritmo evolutivo paralelo, es importante tener en cuenta sobre qué plataforma de computación se ha implementado, ya que la arquitectura de hardware impacta notablemente en el tiempo requerido para realizar las operaciones, comunicaciones, sincronizaciones y compartimiento de datos. Hasta la última década, las propuestas tradicionales de metaheurísticas se enfocaban en supercomputadores y clusters de computadoras. Actualmente, las novedosas arquitecturas de computadoras paralelas emergentes tal como procesadores multicore, unidades de procesamiento gráfico (GPU), o entornos grid, proveen nuevas oportunidades para desarrollar técnicas de computación paralelas para mejorar la resolución de problemas y disminuir el tiempo de ejecución requerido.

Los ambientes de computación paralelos y distribuidos se han tornado populares en las últimas décadas como una forma de proveer el poder de cómputo necesario para resolver problemas complejos, representando una estrategia efectiva para la ejecución de AEs distribuidos. La mayoría de los resultados reportados sobre AEs distribuidos asumen que el ambiente de hardware subyacente tiene características idénticas (ambiente homogéneo) teniendo en cuenta no sólo componentes de hardware (procesadores, memoria, redes) sino también software (sistema operativo) [1]. Por otra parte, esta clase de homogeneidad en hardware es muy difícil de mantener en el transcurso del tiempo. Una de las principales razones son las roturas de las máquinas y la dificultad de construir un sistema igual teniendo en cuenta el mismo procesador, arquitectura interna y configuración. Por otra parte, el rápido desarrollo de la tecnología en el diseño de procesadores, redes y almacenamiento de datos sumado al constante decremento de la relación costo/rendimiento, posibilita el uso de nuevos recursos disponibles. Como consecuencia, la coexistencia de equipamientos nuevos y viejos en un ambiente de computación ha dado lugar a la emergencia de plataformas paralelas heterogéneas, las cuales son actualmente muy comunes en laboratorios, compañías, instituciones, campus, etc. Uno de los primeros trabajos que trata con ambientes de computación heterogéneos y AEs distribuidos se puede hallar en [2]. Trabajos más recientes relacionados con ambientes heterogéneos se pueden encontrar en [3], [11].

Las GPUs, originalmente diseñadas como dispositivos específicos para procesamiento gráfico, se han transformado en plataformas muy poderosas a costos accesibles para utilizar con propósitos de cómputo general [9]. Por esta razón, el estudio de implementaciones de AEs usando GPUs [10] ha crecido a pasos agigantados, ya que ayuda a reducir el tiempo de ejecución de estos algoritmos mediante la explotación del paralelismo masivo de tales dispositivos.

II. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

En esta sección se describen la línea de investigación y desarrollo, junto con las sublíneas que se desprenden.

La definición del valor apropiado para el periodo de migración en un AE distribuido requiere considerar varios valores (desde bajos a altos) siguiendo un estudio similar a un análisis factorial, realizar la experimentación previa correspondiente, para finalmente analizar cuál de esos periodos será usado en la resolución del problema. Esta tarea es computacionalmente costosa y el valor apropiado para el periodo de migración depende de las características del problema a resolver. En esta línea de investigación se propone una estrategia diferente: un AE distribuido proactivo que adaptativamente define el valor del periodo cuando la entropía (medida de diversidad genética) de cada subpoblación satisface una determinada condición; el objetivo es mantener la diversidad genética en valores aceptables. Por consiguiente, cuando la entropía cae debajo de cierto umbral (se detecta pérdida de diversidad), esas subpoblaciones proactivamente solicitan nuevo material genético al reducir el valor del periodo de migración de la subpoblación que le provee inmigrantes, es decir, se producen migraciones más frecuentes. De esta manera, si la topología de la estrategia de migración es un anillo, la isla i modifica el periodo de migración de la isla precedente $i-1$ solicitando migraciones más frecuentes. Análogamente, la propuesta también reacciona cuando detecta mucha diversidad, en este caso incrementa el periodo de migración de la isla $i-1$ para alargar el periodo de aislamiento entre esas islas. La continuación que se propone a esta línea de investigación es el uso de diferentes métricas para la toma de decisiones, como así también la incorporación de más parámetros de migración al mecanismo proactivo del AE distribuido.

En esta misma línea, se está trabajando en la adaptación de otro parámetro de un AE secuencial como lo es la probabilidad de mutación, esta variación se realiza dependiendo de la diversidad genética presente en la población. El nuevo método adaptativo pretende mejorar el comportamiento del AE, evitando la convergencia prematura y la pérdida de diversidad. Una de las ventajas de este método es que no se introducen grandes cambios en la formulación original del algoritmo. El objetivo de nuestra estrategia adaptativa es incrementar el valor de p_m si la diversidad genética (medida por el valor de la entropía de la población) se pierde gradualmente, con la intención de mantener una población distribuida en el espacio de búsqueda. El valor de p_m se decrementa cuando se observa un incremento en la diversidad de la población. De esta manera, estos cambios en el valor de la probabilidad es una fuente adicional para mantener el balance entre la exploración y explotación del espacio de búsqueda. En este caso, la acción futura está orientada a perfeccionar el mecanismo auto-adaptativo considerando nuevos criterios e incorporando otros parámetros del AE como pueden ser las probabilidades de cruce. Por otra parte, para fusionar esta línea de investigación con la anterior, se podría considerar este AE auto-adaptativo en un entorno

distribuido y así avanzar en la propuesta de AEs distribuidos heterogéneos.

Acompañando la propuesta adaptativa también se puede considerar dentro de esta línea la plataforma de computación en la cual son implementadas las distintas propuestas. En este sentido hemos considerado la ejecución un algoritmo evolutivo distribuido en un conjunto de procesadores con rendimientos dispares, desde modernos a viejos. Las diferentes subpoblaciones son evolucionadas por procesadores con diferentes velocidades, en consecuencia en un determinado momento cada una se encuentra en diferentes estados de la evolución. Tan pronto una isla ejecutando en un procesador rápido llega al intercambio de soluciones, envía una solución a sus vecinos y continua con su procedimiento de optimización de una manera asíncrona. Las soluciones desde sus vecinos más lentos deberán tomar más tiempo para diseminarse a otras islas. La metodología consiste en calcular las diferencias relativas entre las velocidades de cada procesador. Con esta información, se puede establecer un ordenamiento que puede ayudar a tomar decisiones más informadas de los valores de los parámetros del algoritmo a fin de contar con un mecanismo que permita al AE distribuido tratar con diferencias en las características en las velocidades de los procesadores. El trabajo futuro está orientado a incluir más niveles de heterogeneidad, por tal motivo consideraremos unidades de procesamiento gráficos y procesadores multicore.

III. RESULTADOS OBTENIDOS/ESPERADOS

En esta sección se detallan los resultados obtenidos en las distintas líneas de investigación en desarrollo, presentadas en la sección anterior.

Los resultados del AE adaptativo que usa una estrategia de control para modificar la probabilidad de mutación, denominado APmAE, fue comparado con los producidos por las distintas variantes de AE utilizando probabilidades de mutación estáticas, con el fin de mostrar la robustez del algoritmo adaptativo como un método de búsqueda alternativo [16]. Como estamos interesados en la auto-adaptación de la probabilidad de mutación, en el caso del AE se presenta un conjunto de valores para la probabilidad de mutación pertenecientes a $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5\}$, con el fin de caracterizar el efecto de la probabilidad de mutación en la calidad de las soluciones obtenidas. Como resultado general obtuvimos resultados muy prometedores. En particular, el APmAE obtuvo mejor calidad de soluciones que las distintas variantes de AE con probabilidad de mutación fija, con velocidades de convergencia similares. Un punto importante a resaltar es que nuestra propuesta no adiciona tiempo de cómputo al aplicar la estrategia de control diseñada para adaptar los valores de la probabilidad de mutación. Finalmente, como el APmAE determina en forma dinámica la probabilidad de mutación durante el proceso de búsqueda, este método ayuda a aliviar el tiempo de seteo requerido, al menos en un 85 %, para determinar la probabilidad de mutación más adecuada.

Para evaluar los resultados obtenidos con el AE distribuido y proactivo, hemos realizado una comparación con varios algoritmos AEs distribuidos (no proactivos) que mantienen el parámetro de migración fijo con valores para la frecuencia de migración en $\{1, 64, 128, 256, 512\}$, es decir, desde un máximo de ensamble entre las islas hasta un proceso de búsqueda prácticamente en aislamiento. Los resultados indicaron que el algoritmo proactivo es capaz de encontrar soluciones al problema planteado de calidad similar al mejor algoritmo AE distribuido homogéneo [15]. El algoritmo proactivo ha funcionado especialmente bien sobre las instancias más difíciles del problema, donde claramente se impone al mejor homogéneo. Un punto interesante de nuestra propuesta es que presenta una mejor velocidad de convergencia, es decir, ha obtenido sus mejores soluciones en un número más bajo de generaciones que cualquiera de las versiones homogéneas.

En cuanto a la sublínea referida al entorno de ejecución, hemos dado un primer paso referente a la paralelización de un AE básico para ser ejecutado sobre una GPU. Los resultados iniciales han sido alentadores, ya que se ha logrado resolver un conjunto de instancias de grandes dimensiones de un problemas de optimización, mejorando los óptimos conocidos en tiempos de cómputo razonables. Por otra parte, hemos desarrollado una propuesta sobre un procedimiento metodológico, y el subsecuente diseño algorítmico, para tratar con ambientes paralelos heterogéneos. Resultados preliminares muestran que el nuevo mecanismo tiene un rendimiento similar a un conjunto de máquinas modernas ejecutando el mismo procedimiento de optimización, esto permite a los investigadores usar todos los recursos de procesamiento disponibles en los laboratorios.

IV. FORMACIÓN DE RECURSOS HUMANOS

La actividad de formación de recursos humanos está orientada a la formación de becarios de iniciación en la investigación. En este marco, un alumno de la carrera Ingeniería en Sistemas se encuentra desarrollando las actividades planificadas para una Beca de Estímulo a las Vocaciones Científicas, otorgada por el CIN para el periodo 2012-2013. Por otra parte, es interesante resaltar que en el LISI se trabaja con alumnos avanzados en la carrera Ingeniería en Sistemas en temas relacionados a la resolución de problemas de optimización usando técnicas inteligentes, con el objeto de guiarlos en el desarrollo de sus tesis de grado y, también, de formar futuros investigadores.

REFERENCES

- [1] E. Alba. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley, 2005.
- [2] E. Alba, A.J. Nebro, and J.M. Troya. Heterogeneous computing and parallel genetic algorithms. *Journal of Parallel and Distributed Computing*, 62:1362–1385, 2002.
- [3] J. Dominguez E. Alba. A methodology for comparing the execution time of metaheuristics running on different hardware. In Jin-Kao Hao and Martin Middendorf, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 7245 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2012.
- [4] Thomas Bäck. Self-adaptation in genetic algorithms. In *Proceedings of the First European Conference on Artificial Life*, pages 263–271. MIT Press, 1992.

- [5] M. Birattari, T. Stutzle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *GECCO 2002*, pages 11 – 19, 2002.
- [6] K. DeJong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975.
- [7] A. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions Evolutionary Computation*, 3(2):124–141, 1999.
- [8] J. J. Greffentette. Optimization of control parameters for genetic algorithms. *IEEE Transaction on System Man and Cybernetic*, 16(1):122–128, 1986.
- [9] T. Wong K. Fok and M. Wong. Evolutionary computing on consumer graphics hardware. *Intelligent Systems*, 22(2):69–78, 2007.
- [10] W. Langdon. Graphics processing units and genetic programming: an overview. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, pages 1–13, 2011.
- [11] S. Mostaghim, J. Branke, A. Lewis, and H. Schneck. Parallel multi-objective optimization using master-slave model on heterogeneous resources. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1981 –1987, 2008.
- [12] V.Nannen and A. Eiben. Relevance estimation and value calibration of evolutionary algorithm parameters. *Proceedings of the Joint International Conference for Artificial Intelligence (IJCAI)*, pages 975–980, 2006.
- [13] Gabriela Ochoa, Inman Harvey, and Hilary Buxton. On recombination and optimal mutation rates. In *in Proceedings of Genetic and Evolutionary Computation Conference*, pages 488–495. Morgan Kaufmann, 1999.
- [14] E. Ridge and D. Kudenko. Tuning an algorithm using design of experiments. In *Experimental Methods for the Analysis of Optimization Algorithms*, pages 265 – 286. Springer, 2010.
- [15] C. Salto, F. Luna, and E. Alba. Heterogeneity through proactivity: Enhancing distributed eas. *Seventh International Conference on Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 279–284, 2012.
- [16] N. Stark, G. Minetti, and C. Salto. A new strategy for adapting the mutation probability in genetic algorithms. *XVIII Congreso Argentino de Ciencias de la Computación (CACIC 2012)*, pages 51–59, 2012.