Big Data Optimization con algoritmos metaheurísticos utilizando frameworks de computación distribuida

Carolina Salto, Gabriela Minetti, Hugo Alfonso, Carlos Bermúdez, Javier Vargas, Franco Morero
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
e-mail: {saltoc, minettig, alfonsoh, bermudezc}@ing.unlpam.edu.ar

Resumen

La comunidad científica ha encontrado en el uso de los recursos tecnológicos disponibles una aliada para abordar problemas de gran complejidad e identificados como irresolubles. Tales problemas han sido abordados con técnicas exactas o heurísticas para lograr su resolución, o al menos conseguir soluciones de alta calidad, cuando los mismos se clasifican como NP-duros. Inicialmente, los problemas se planteaban en entornos estáticos, pero en los últimos años se les trata de resolver reproduciendo las características dinámicas y de alta dimensionalidad que los alteran. La optimización de estos problemas, conocida como *Big Data Optimization*, se puede realizar diseñando algoritmos metaheurísticos secuenciales y distribuidos (*solvers*) bajo *frameworks* de programación de alto nivel como los que incorporan el paradigma MapReduce para el manejo de *Big Data*. Dichos *solvers*, en principio, serán diseñados y testeados con problemas académicos, con el objetivo de analizar el comportamiento en cuanto a eficiencia y escalabilidad. En consecuencia, nuestro objetivo central es adaptar estos *solvers* para abordar problemas de interés en contextos reales (científico, industrial, entre otros) donde estamos trabajando, y puntualmente en problemas de planificación y de diseño de redes de distribución de agua y de sensores en plantas industriales.

Palabras claves: Big Data, Optimización, Algoritmos metaheurísticos, Solvers

Contexto

Esta línea de investigación surge como una proyección del Proyecto "Técnicas inteligentes avanzadas y sistemas distribuidos aplicados a la resolución de problemas de decisión complejos" que se desarrolló hasta el 2020 en la Facultad de Ingeniería de la UNLPam. Este proyecto se lleva adelante en el Laboratorio de Investigación de Sistemas Inteligentes (LISI) de la Facultad de Ingeniería y es dirigido por la Dra. Salto.

En el LISI, desde su creación en 1998, nos hemos abocado al estudio de algoritmos cada vez más eficientes para la solución de problemas complejos, tanto de optimización como de diseño. En este dominio, el objetivo consiste en obtener algoritmos nuevos que den solución al problema y que necesiten un esfuerzo computacional más pequeño que los algoritmos existentes, así como caracterizar su comportamiento para las clases de problemas que demanda la comunidad científica e industrial en general. Cabe destacar que desde hace varios años, los integrantes de este laboratorio mantienen una importante vinculación con investigadores de las universidades argentinas, Universidad Nacional de San Luis y Universidad Nacional de Rio Cuarto, y de la Universidad de Málaga (España), con quienes se realizan trabajos conjuntos.

Introducción

La optimización de los recursos es una exigencia que deben hacer frente la mayoría de las organizaciones en la actualidad. Dicha optimización no se

refiere a ahorrar o a suprimir, sino que se define como la mejor forma de realizar una actividad. Por lo tanto, los administradores deben tomar medidas urgentes y con visión a futuro. Estas decisiones se relacionan tanto con la optimización de procesos y recursos como con el pronóstico de las tendencias de mercado. La primera decisión es quizás la más sencilla de interpretar, mientras que la segunda es la más compleja, ya que la creatividad debe anteponerse a la racionalidad e imaginar cuáles serán las tendencias futuras. Hoy en día, los procesos de toma de decisiones son cada vez más complejos y globalizados debido a: (i) las dimensiones de dichos problemas cuando se tratan de abordar en entornos reales (industriales, científicos, entre otros), (ii) el carácter combinatorio de muchos de ellos y (iii) la naturaleza del objetivo que intenta alcanzar. Todo ello teniendo en cuenta criterios vinculados a la eficiencia del sistema, sus costos de explotación y de distribución, además de los tiempos de recepción, de ejecución y de entrega de materiales, servicios y productos. La gran mayoría de los problemas de decisión complejos del mundo real, cuando son modelados como problemas de optimización numérica con restricciones, pertenecen a la clase NP-duros. En las últimas décadas se ha comenzado a considerar los cambios dinámicos que generalmente se presentan tanto en la función objetivo como en las restricciones que se deben considerar en tales problemas, y algunos equipos de investigación han comenzado a abordarlos ([1], [2], [3], [4], [5], [6]). Este tipo de problema es referenciado en la literatura como Problema de Optimización con Restricciones Dinámico (Dynamic Constrained Optimization Problem (DCOP)) ([2], [7], [8], [9]). Se considera DCOP a todo aquel problema de búsqueda en el cual la función objetivo y/o sus restricciones cambian a lo largo del tiempo; cambios que se materializan en el espacio de búsqueda de las soluciones y por ende en cuáles sean las soluciones óptimas a encontrar. Pero su abordaje en entornos dinámicos está siendo recientemente considerado y debe contemplar variantes de tales cambios que requieren distintos tipos de implementación. Esas variantes son: (i) tanto la función objetivo como las restricciones son dinámicas ([10], [11], [12]); (ii) sólo la función objetivo es dinámica pero las restricciones se mantienen estáticas ([13], [14], [15]) y (iii) la función objetivo es estática y las restricciones son dinámicas ([16], [17], [18]). En los tres casos, el espacio de búsqueda puede contar con áreas de soluciones que no son factibles de acuerdo a las restricciones, por lo tanto tales soluciones pueden requerir reparaciones para trasladarlas a zona factible. Cuando los problemas de optimización poseen un gran número de variables, los cálculos están drásticamente más allá de la capacidad de las computadoras de uso general. Ante estos problemas de optimización a gran escala (*Big Data Optimization*), a menudo se adoptan estrategias tácticas, como la hibridación de diferentes tipos de algoritmos, el uso de búsqueda local, enfoques para la adaptación de parámetros y la relajación de restricciones, todo contribuye a resolver el problema en tiempo polinomial. Por lo tanto, cómo resolver la optimización a gran escala plantea un serio desafío.

La computación distribuida ha cambiado enormemente el panorama de *Big Data*. A diferencia de la computación centralizada tradicional que escala el hardware para aumentar la capacidad de cómputo, la computación distribuida incrementa el hardware con la incorporación de computadoras más pequeñas con la potencia equivalente a una supercomputadora. En consecuencia, para resolver los problemas de *Big Data Optimization*, se descompone el problema original de tal manera que disminuya la complejidad computacional. Uno de los paradigmas de computación distribuida más usados para procesar *Big Data* es MapReduce (MR), propuesto por Google [19]. MR divide grandes volúmenes de datos en porciones más pequeñas (chunks), las cuales son procesadas en paralelo [20]. Hadoop es un *framework open source* y muy popular que implementa el paradigma MR [21], tanto en la industria como en el ámbito académico. Este *framework* provee una plataforma distribuida lista para usar, la cual es fácil de programar, escalable, confiable, tolerante a fallas, además planifica automáticamente las tareas paralelas [22]. La filosofía de Hadoop es enviar el código a los sitios donde están los datos almacenados en el sistema de archivos distribuido y procesar los

datos de forma local y simultánea. Posteriormente ha surgido un *framework open source* derivado de Hadoop, Apache Spark, que está atrayendo cada vez más interesados en la comunidad de *Big Data*. Como contiene resultados intermedios en la memoria en lugar de almacenarlos en el disco (como lo hace Hadoop), los trabajos MR de varias pasadas se pueden llevar a cabo minimizando el número de operaciones de entrada/salida en el sistema de archivos distribuido. Por lo tanto, logra un mayor rendimiento que Hadoop. Otro *framework open source* que implementa MR es MR-MPI [23], que permite un mayor control de la plataforma paralela permitiendo mejorar la performance del ancho de banda y reducir los costos de latencia. El paradigma MR, también, contribuye a construir nuevos modelos de optimización y de *machine learning*, en particular algoritmos metaheurísticos escalables, como *solvers* de problemas de optimización combinatoria, que despiertan un gran interés en la comunidad científica e industrial. En la literatura, muchos investigadores informan de algoritmos metaheurísticos programados en Hadoop ([24], [25], [26], [27], [28]) y Spark ([29], [30], [31]), y algunos bajo MR-MPI [32], según el conocimiento de los autores.

En la actualidad se tiende a abordar el estudio de problemas de *Big Data Optimization* con el uso de *solvers*, basados en metaheurísticas, secuenciales y distribuidos bajo *frameworks* de programación de alto nivel como los que incorporan el paradigma MapReduce. Dentro de los problemas de interés actual, que en el LISI venimos abordando, se encuentran los de diseño óptimo de redes de agua ([33], [34], [35]) y de sensores en plantas industriales ([36], [37]), como también la planificación de tareas [38], entre otros. Generalmente, los *solvers* se prueban con problemas académicos, con el objetivo de analizar su desempeño. Esto permite estudiar las características, parámetros, comportamiento en cuanto a calidad de soluciones, entre otras métricas. También se aprovechan estas pruebas para evaluar la escalabilidad de los algoritmos propuestos. Los problemas usualmente considerados para estos testeos son: Knapsack ([39], [40]), OneMax [41], MaxCut [42], entre otros.

Línea de investigación y desarrollo

Los problemas de optimización dinámicos y, también, los que incluyan un gran número de variables (alta dimensionalidad) forman parte de lo que se conoce como *Big Data Optimization*. Estos se pueden resolver diseñando algoritmos metaheurísticos secuenciales y distribuidos (denominados *solvers*) bajo *frameworks* de programación de alto nivel como los que incorporan el paradigma MapReduce para el manejo de *Big Data*. Dichos *solvers*, en principio, serán diseñados y testeados con problemas académicos, con el objetivo de estudiar las características, parámetros, comportamiento en cuanto a calidad de soluciones y la escalabilidad de los algoritmos propuestos. Posteriormente, estos *solvers* serán adaptados para dar solución a problemas de diseño de redes de distribución de agua y de sensores en plantas industriales, además de problemas de planificación. Para una mejor comprensión del problema en su totalidad, y de su grado de complejidad, se formulan las siguientes preguntas:

1: ¿Cuál sería el diseño de *solvers* eficientes para resolver los problemas de *Big Data Optimization* planteados?

2: ¿Cuál de los *frameworks* disponibles actualmente permite a los *solvers* desarrollados alcanzar su mejor rendimiento, en cuanto a eficacia y eficiencia, para escalar a casos de estudio de alta complejidad y dimensionalidad?

3: ¿Cómo escalan estos *solvers* al considerar el incremento de los recursos computacionales para ejecutarlos?

4: ¿Cómo varían los tiempos de respuesta de los *solvers* al utilizar computación distribuida?

5: ¿Cuál es la eficiencia de los *solvers* desarrollados en comparación con los propuestos en la literatura para abordar los problemas de *Big Data Optimization* tratados?

El objetivo general de esta línea de investigación es evaluar la eficiencia de los *solvers* diseñados para resolver los problemas de *Big Data Optimization* académicos y los relacionados con el diseño óptimo de redes y de planificación, haciendo uso de *frameworks* de programación de alto nivel como los que incorporan el paradigma MapReduce.

 RedUNCI - UNdeC. ISBN: 978-987-24611-3-3

Para alcanzar el objetivo general propuesto se establecen los siguientes objetivos específicos a seguir durante los próximos cuatro años:

1: Diseñar e implementar los *solvers* para resolver los problemas de *Big Data Optimizatio*n planteados usando *frameworks* para el tratamiento de *Big Data*.

2: Realizar la experimentación para poder determinar el *framework* más adecuado y así mejorar el rendimiento, en cuanto a eficacia y eficiencia, de estos *solvers*.

3: Analizar la escalabilidad de los *solvers* en la ejecución al incrementar la disponibilidad de recursos computacionales (cantidad de nodos, capacidad de procesamiento, memoria, entre otros).

4: Estudiar los tiempos de respuesta de los *solvers* al utilizar computación distribuida.

5: Comparar los *solvers* desarrollados con los propuestos en la literatura para abordar los problemas de *Big Data Optimization* planteados desde los puntos de vista de eficiencia y eficacia.

Resultados esperados

Con este proyecto se espera que los *solvers* propuestos para la resolución de problemas de *Big Data Optimization* proporcionen soluciones eficientes de alta calidad a los casos de gran complejidad y alta dimensionalidad. También se espera aportar diseños de *solvers* innovadores y eficientes en el campo del diseño de redes de distribución de agua y de sensores en plantas industriales y de planificación.

Formación de recursos humanos

Cada año se incorporan al proyecto alumnos avanzados en la carrera Ingeniería en Sistemas y becarios de investigación. La tarea que se les asigna está relacionada a la solución de problemas de optimización usando técnicas inteligentes, con el objeto de guiarlos en el desarrollo de sus tesinas de grado y, también, de formar futuros investigadores científicos. Por otra parte, los docentes-investigadores que integran el proyecto realizan diversos cursos de posgrado relacionados con la temática del proyecto, con el objetivo de sumar los créditos necesarios para cursar carreras de posgrado.

REFERENCES

[1] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[2] T. T. Nguyen and X. Yao, "Continuous dynamic constrained optimization—the challenges," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 769–786, 2012.

[3] T. T. Nguyen and X. Yao, "Evolutionary optimization on continuous dynamic constrained problems - an analysis," in *Evolutionary Computation for Dynamic Optimization Problems*, S. Yang and X. Yao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 193–217.

[4] M. Mavrovouniotis and S. Yang, "Ant colony optimization for dynamic combinatorial optimization problems," pp. 121–142, 2018.

[5] M. Ameca-Alducin, E. Mezura-Montes, and N. Cruz-Ramirez, "Dynamic differential evolution with combined variants and a repair method to solve dynamic constrained optimization problems: An empirical study," *Soft Comput.*, vol. 22, no. 2, p. 541–570, 2018.

[6] J. K. Kordestani and M. R. Meybodi, *Application of Sub-Population Scheduling Algorithm in Multi-Population Evolutionary Dynamic Optimization*. John Wiley Sons, Ltd, 2020, ch. 7, pp. 169–211.

[7] T. T. Nguyen and Xin Yao, "Benchmarking and solving dynamic constrained problems," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 690–697.

[8] H. K. Singh, A. Isaacs, T. T. Nguyen, T. Ray, and Xin Yao, "Performance of infeasibility driven evolutionary algorithm (idea) on constrained dynamic single objective optimization problems," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 3127–3134.

[9] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2678–2688, 2017.

[10] M. Andrews and A. Tuson, "Dynamic optimization: An investigation of practitioners requirements," *Proceedings of the The 24th Annual Workshop of the UK Planning and Scheduling Special Interest Group*, 2005.

[11] Y. Wang and M. Wineberg, "Estimation of evolvability genetic algorithm and dynamic environments," *Genetic Programming and Evolvable Machines*, vol. 7, no. 4, p. 355–382, Dec. 2006. [Online]. Available: https://doi.org/10.1007/s10710-006-9015-5

[12] D. M. Prata, E. L. Lima, and J. C. Pinto, "Simultaneous data reconciliation and parameter estimation in bulk polypropylene polymerizations in real time," *Macromolecular Symposia*, vol. 243, no. 1, pp. 91–103, 2006.

[13] L. Araujo and J. J. Merelo, "A genetic algorithm for dynamic modelling and prediction of activity in document streams," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 1896–1903. [Online]. Available: https://doi.org/10.1145/1276958.1277340

[14] P. Tawdross, S. K. Lakshmanan, and A. Konig, "Intrinsic evolution of predictable behavior evolvable hardware in dynamic environment," in *2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*, 2006, pp. 60–60.

[15] M. Rocha, J. Neves, A. Veloso, E. Ferreira, and I. Rocha, *Evolutionary Algorithms for Static and Dynamic Optimization of Fed-batch Fermentation Processes*, 01 2005, pp. 288–291.

[16] K. Deb, U. N, and K. Sindhya, "Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling," 05 2007, pp. 803–817.

[17] P. Ioannou, A. Chassiakos, H. Jula, and R. Unglaub, "Dynamic optimization of cargo movement by trucks in metropolitan areas with adjacent ports," 2002. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/15509

[18] K. Mertens, T. Holvoet, and Y. Berbers, "The dyncoaa algorithm for dynamic constraint optimization problems," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1421–1423. [Online]. Available: https://doi.org/10.1145/1160633.1160898

[19] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *OSDI*, 2004.

[20] A. Cano, C. García-Martínez, and S. Ventura, "Extremely high-dimensional optimization with mapreduce: Scaling functions and algorithm," *Information Sciences*, vol. 415-416, pp. 110–127, 11 2017.

[21] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[22] I. Hashem, N. Anuar, A. Gani, I. Yaqoob, F. Xia, and S. Khan, "Mapreduce: Review and open challenges," *Scientometrics*, vol. 109, 04 2016.

[23] S. J. Plimpton and K. D. Devine, "Mapreduce in mpi for large-scale graph algorithms," *Parallel Computing*, vol. 37, no. 9, pp. 610–632, 2011, emerging Programming Paradigms for Large-Scale Scientific Computing.

[24] A. Cano, C. García-Martínez, and S. Ventura, "Extremely high-dimensional optimization with mapreduce: Scaling functions and algorithm," *Information Sciences*, vol. 415-416, pp. 110–127, 2017.

[25] F. Ferrucci, P. Salza, and F. Sarro, "Using hadoop mapreduce for parallel genetic algorithms: A comparison of the global, grid and island models," *Evolutionary Computation*, vol. 26, no. 4, pp. 535–567, 2018.

[26] L. Di Geronimo, F. Ferrucci, A. Murolo, and F. Sarro, "A parallel genetic algorithm based on hadoop mapreduce for the automatic generation of junit test suites," in *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, 2012, pp. 785–793.

[27] A. Verma, X. Llorà, D. E. Goldberg, and R. H. Campbell, "Scaling genetic algorithms using mapreduce," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 13–18.

[28] A. Verma, X. Llorà, S. Venkataraman, D. E. Goldberg, and R. H. Campbell, "Scaling ecga model building via data-intensive computing," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[29] C. Hu, G. Ren, C. Liu, M. Li, and W. Jie, "A spark-based genetic algorithm for sensor placement in large scale drinking water distribution systems," *Cluster Computing*, vol. 20, 06 2017.

[30] C. Paduraru, M.-C. Melemciuc, and A. Stefanescu, "A distributed implementation using apache spark of a genetic algorithm applied to test data generation," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1857–1863. [Online]. Available: https://doi.org/10.1145/3067695.3084219

[31] R. Qi, Z.-J. Wang, and S.-Y. Li, "A parallel genetic algorithm based on spark for pairwise test suite generation," *Journal of Computer Science and Technology*, vol. 31, pp. 417–427, 03 2016.

[32] C. Salto, G. Minetti, E. Alba, and G. Luque, *Developing Genetic Algorithms Using Different MapReduce Frameworks: MPI vs. Hadoop: 18th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2018, Granada, Spain, October 23–26, 2018, Proceedings*, 01 2018, pp. 262–272.

[33] C. Bermudez, G. Minetti, and C. Salto, "SA to optimize the multi-period water distribution network design," in *XXIX Congreso Argentino de Ciencias de la Computación (CACIC 2018)*, 2018, pp. 12–21.

[34] C. Bermudez, C. Salto, and G. Minetti, "Designing a multi-period water distribution network with a hybrid simulated annealing," in *XLVIII JAIIO: XX Simposio Argentino de Inteligencia Artificial (ASAI 2019)*, 2019, pp. 39–52.

[35] H. Alfonso, C. Bermudez, G. Minetti, and C. Salto, "A real case of multi-period water distribution network design solved by a hybrid SA," in *XXVI Congreso Argentino de Ciencias de la Computación – CACIC 2020*, no. 1-12, 2020.

[36] J. Hernandez, C. Salto, G. Minetti, M. Carnero, and M. C. Sanchez, "Hybrid simulated annealing for optimal cost instrumentation in chemical plants," *Chemical Engineering Transactions*, vol. 74, pp. 709–714, May 2019. [Online]. Available: https://www.cetjournal.it/index.php/cet/article/view/CET1974119

[37] J. Hernandez, C. Salto, G. Minetti, M. Carnero, C. Bermudez, and M. Sanchez, "Optimal instrumentation: Adjustment and hybridization of a simulated annealing based technique," in *XXV Congreso Argentino de Ciencias de la Computación (CACIC 2019)*, Oct. 2019, pp. –.

[38] F. Morero, C. Salto, and C. Bermudez, "Parallelism and hybridization in differential evolution to solve the flexible job shop scheduling problem," *Journal of Computer Science & Technology*, vol. 20, no. 1, pp. 33–42, 2020.

[39] K. Klamroth and M. M. Wiecek, "Time-dependent capital budgeting with multiple criteria," in *Research and Practice in Multiple Criteria Decision Making*, Y. Y. Haimes and R. E. Steuer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 421–432.

[40] H. Kellerer, U. Pferschy, and D. Pisinger, *Introduction to NP-Completeness of Knapsack Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 483–493.

[41] J. Schaffer and L. Eshelman, "On crossover as an evolutionarily viable strategy," in *ICGA*, 1991.

[42] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, p. 1115–1145, Nov. 1995. [Online]. Available: https://doi.org/10.1145/227683.227684