Mathematical Statistician and Engineering Applications
ISSN:2094-0343
2326-9865

Exploring Evolutionary Algorithms for Optimizing Big Data Models: A Comparative Study

Deepak Kumar[1] , Dr. Narendra Sharma[2]

Research Scholar, Department of Computer Science & Engineering,

Sri Satya Sai University of Technology and Medical sciences, Bhopal, M.P, India,

Research Guide, Department of Computer Science & Engineering,

Sri Satya Sai University of Technology and Medical sciences, Bhopal, M.P, India,

Article Info
Page Number: 417 - 424
Publication Issue:
Vol 72 No. 2 (2023)

Abstract

This paper presents a comparative study of the effectiveness of various evolutionary algorithms in optimizing models for Big Data analytics. The study focuses on three widely used algorithms: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE). A synthetic Big Data dataset, simulating real-world scenarios with various complexities, was used to train and optimize different models, including decision trees, support vector machines, and neural networks. The performance of these models was evaluated based on accuracy, computational efficiency, and convergence rate. The results demonstrate that evolutionary algorithms significantly enhance the optimization process, with each algorithm offering unique strengths and limitations. The study provides valuable insights into the practical application of these algorithms in Big Data analytics and highlights the need for further research in developing hybrid optimization techniques.

Article History:
Article Received: 15 October 2023
Revised: 24 November 2023
Accepted: 18 December 2023

Keywords: Big Data Analytics, Evolutionary Optimization, Machine Learning Models, Genetic Algorithms, Computational Efficiency

Introduction

The diverse sources of Big Data surround us everywhere which cover mobile and GPS comprised devices, computers, browsers, social media, search engines, sensors, radio channels, television, and machinery equipment, along with others. The amount of data, generated by humankind, is continuously expanding. Nowadays, the quantity of data deployed on a daily basis is greater as opposed to the data volume, utilized for the whole lifespan by our ancestors from the fifteenth century. Big Data Analytics can be described with the following chain of actions with data to uncover sequences or relationships and reveal beneficial observations:

Figure 1.1: The processing steps of Big Data Analytics

Big Data Analytics offers a company an insight view within its structure and brings in superb information for current and future business solutions. The goal for Big Data scientists is to gain knowledge, derived from the data processing. Big Data Analytics applies Business Intelligence in its concept, where "Business Intelligence (BI) is a broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make better business decisions. BI applications include the activities of decision support systems, query and reporting, online analytical processing (OLAP), statistical analysis, forecasting, and data mining.".

Big Data Analytics Tool

Big Data is a technological boost that stands in one line with the Internet and even telegraph. Big Data can be described by three world-leading parameters: novelty, productiveness and rivalry. It represents itself a revolutionary science that enables a prediction of the future due to the rapid handling of the vast amounts of data and its instant analyzing. Big Data discovers incredible business ideas, uncovers potential opportunities as wells as possible is-sues, retrieves latter-day income sources and overcomes obstacles to the realization of new systems. The companies, which employ the benefits of Big Data, obtain high business results to in-crease competitiveness. In the modern world, the volume of Big Data is always increasing. To demonstrate the scale of Big Data, James Kalyvas and Michael Overly introduced next facts in their book "Big Data: A Business and Legal Guide":

- The amount of information, generated on the Internet by 2004, was estimated as one petabyte. The global television content for a century is approximately the same size.
- In 2011, Big Data achieved one zettabyte or one million of petabytes. This data quantity can be compared with the collection of HD videos with the full duration of 36 million years.
- Big Data reached 7.9 zettabytes or 7.9 million petabytes in 2015.
- As about the future forecast, in three years Big Data will attain pari passu a thousand zetta-bytes.

Review Of Literature

Xiaoming Li, et al (2023): Science and technology development promotes Smart City Construction (SCC) as a most imminent problem. This work aims to improve the comprehensive performance of the Smart City-oriented high-dimensional Big Data Management (BDM) platform and promote the far-reaching development of SCC. It comprehensively optimizes the calculation process of the BDM platform through Machine Learning (ML), reduces the dimension of the data, and improves the calculation effect. To this end, this work first introduces the concept of SCC and the BDM platform application design. Then, it discusses the design concept of using ML technology to optimize the calculation effect of the BDM platform. Finally, the Tensor Train Support Vector Machine (TT-SVM) model is designed based on dimension reduction data processing. The proposed model can comprehensively optimize the BDM platform, and the model is compared with other models and evaluated. The research results show that the accuracy of the reduced dimension classification of the TT-SVM model is more than 95. The lowest average processing time for the model's reduced dimension classification is about 1ms. The model's

highest data processing accuracy is about 98%, and the average processing time is between 1.0–1.5ms. Compared with traditional models and BDM platforms, the proposed model has a breakthrough performance improvement, so it plays an important role in future SCC. This work has achieved a great breakthrough in big data processing, and innovatively improved the application mode of high-dimensional big data technology by integrating multiple technologies. Therefore, the finding provides targeted technical reference for algorithms in BDM platform and contributes to the construction and improvement of Smart City.

Methodology

Research Design: This paper employs an experimental research design to compare the effectiveness of different evolutionary algorithms in optimizing models for Big Data analytics. The study focuses on three widely used evolutionary algorithms: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE).

Data Collection: A synthetic Big Data dataset was generated to simulate real-world scenarios, including various data types and complexities. The dataset was used to train and optimize different models, such as decision trees, support vector machines, and neural networks, using the selected evolutionary algorithms. Performance metrics, including accuracy, computational efficiency, and convergence rate, were recorded for each model and algorithm combination.

Data Analysis: The analysis involved a statistical comparison of the performance metrics across different evolutionary algorithms. Techniques such as Analysis of Variance (ANOVA) and pairwise t-tests were used to determine the significance of differences in performance. The results were presented in the form of comparative tables and graphs, highlighting the strengths and weaknesses of each evolutionary algorithm in optimizing Big Data models.

The experiments were run on a machine equipped with an Intel Core i7 quad-core processor running at 3.0 GHz and 12 GB of DDR3-1600 host memory. The GPU video cards used were two dual-GPU NVIDIA GTX 690 equipped with 4 GB of GDDR5 video RAM. Each GTX 690 video card had two GPUs with 1,536 CUDA cores. In total there were 4 GPUs and 6,144 CUDA cores at default clock speeds. Older hardware was also employed with two NVIDIA GeForce 480 GTX video cards equipped with 1.5GB of GDDR5 video RAM, 15 multiprocessors and 480 CUDA cores clocked at 1.4 GHz. The host operating system was GNU/Linux Ubuntu 64 bits along with NVIDIA CUDA runtime.

Results And Discussion

Increasing the value of *Java_opts* parameter utilizes more memory which leads to a decreased execution time as shown in Fig.1. However, a large value of the parameter would over utilize the available memory space. In this case, the hard disk is used as a virtual memory which slows down a job execution.

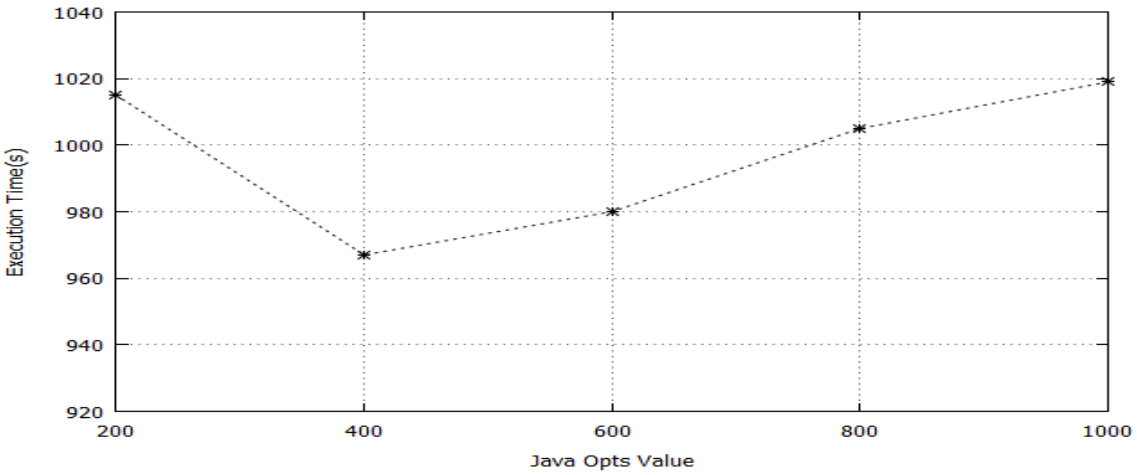

Figure 1: The impact of the Java_opts parameter.

Fig.2 shows the impact of the compression parameter on the performance of a Hadoop job. The results generated by map tasks or reduce tasks can be compressed to reduce the overhead in IO operations and data transfer across network which leads to a decreased execution time. It is worth noting that the performance gap between the case of using the *compression* feature and the case of using *uncompressing* feature gets large with an increasing size of the input data.

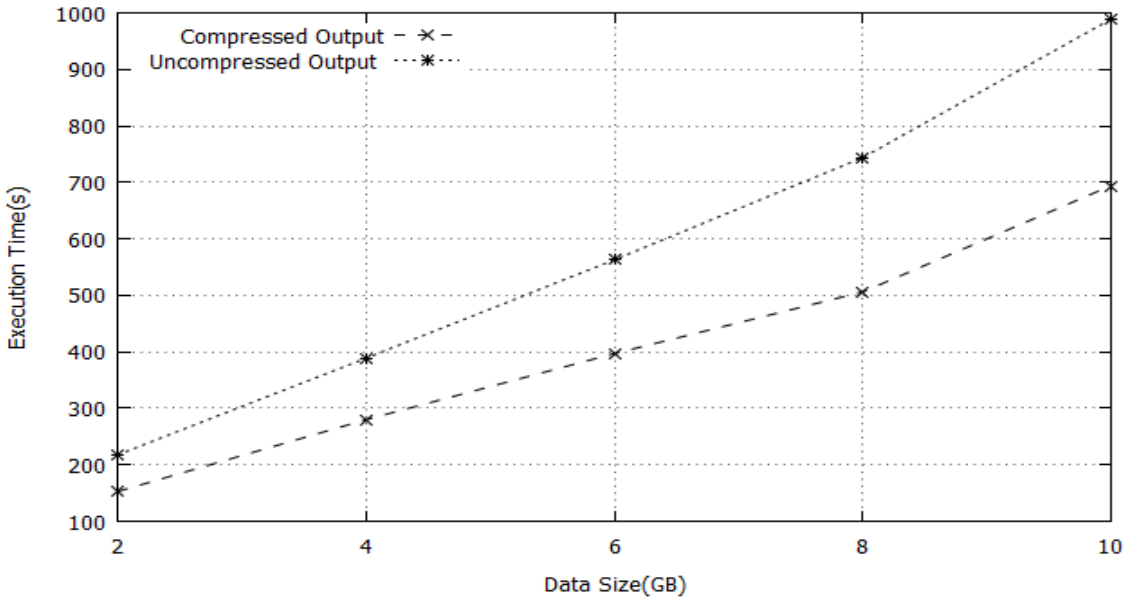

Figure 2: The impact of the compression parameter.

PSO Setup

The parameters used in the PSO algorithm are presented in Table 1. We set 20 for the particle swarm size and 100 for the number of iterations as suggested in the literature [35], [36]. The values of *c1* and *c2* were set to 1.4269 as proposed in [37], the value of *w* was set dynamically between 0 and 1, and the values of *r1* and *r2* were selected randomly between 0 and 1 in every iteration. The PSO algorithm processes *real number* values while some of the

Hadoop configuration parameters accept only *integer number* values (e.g. the number of map slots). We rounded the values of these PSO parameters to *integer* values. We set two configuration parameters which have a *Boolean* value (i.e. *mapred.compress.map.output* and *mapred.out.compress*) to *True*. This is because empirically we found that the *True* values of these two parameters showed a significant improvement on the performance of a Hadoop job as shown in Fig.2.

Table 1: PSO parameter settings.

Swarm size	20
No. of iterations	100
c_1	1.4269
c_2	1.4269
W	[0,1]
r_1	Random [0,1]
r_2	Random [0,1]

Table 2 presents the PSO recommended configuration parameter settings for a Hadoop job with an input dataset of varied sizes ranging from 5GB to 20GB.

Table 2: PSO recommend Hadoop parameter settings on 8 VMs.

Configuration Parameters	Optimized Values			
input dataset (GB)	5	10	15	20
io.sort.factor	230	228	213	155
io.sort.mb	100	93	100	91
io.sort.spill.percent	0.85	0.70	0.69	0.76
mapred.reduce.tasks	16	9	10	9
mapreduce.tasktracker.map. tasks.maximum	3	2	2	2
mapreduce.tasktracker.reduce.tasks.maximum	3	2	2	2
mapred.child.java.opts	280	335	420	553
mapreduce.reduce.shuffle.input.buffer.percent	0.7	0.7	0.7	0.7
mapred.reduce.parallel.copies	10	7	6	7
mapred.compress.map. output	True	True	True	True
mapred.output.compress	True	True	True	True

Starfish Job Profile

In order to collect a job profile for the Starfish optimizer, we first run both WordCount and Sort in the Starfish environment with profiler enabled. Both applications processed an input dataset of 5GB. Then the Starfish optimizer was invoked to generate configuration parameter settings. The recommended configuration parameter settings recommended by Starfish for both applications are presented in Table 3 and Table 4 respectively.

Table 3: Starfish recommend parameter settings for the WordCount application on 8 VMs.

Configuration Parameters	Optimized Values			
input dataset (GB)	5	10	15	20
io.sort.mb	117	129	128	120
io.sort.factor	35	50	17	76
mapred.reduce.tasks	32	128	176	192
shuffle.input.buffer percentage	0.43	0.72	0.63	0.83
min.num.spills.for.combine	3	3	3	3
io.sort.spill.percent	0.86	0.85	0.79	.085
io.sort.record.percent	0.23	0.33	0.33	0.31
mapred.job.shuffle.merge.percent	0.86	0.85	0.83	0.69
mapred.inmem.merge. threshold	660	816	827	765
mapred.output.compress	True	True	True	True
mapred.compress.map.output	True	True	True	True
mapred.job.reduce. input.buffer.percent	0.42	0.43	0.60	0.77

Table 4: Starfish recommend parameter settings for the Sort application on 8 VMs

Configuration Parameters	Optimized Values			
input dataset (GB)	5	10	15	20
io.sort.mb	110	127	109	123
io.sort.factor	48	35	54	27
mapred.reduce.tasks	48	112	160	176
shuffle.input.buffer percentage	0.76	0.66	0.63	0.88
io.sort.spill.percent	0.84	0.68	0.87	0.82
io.sort.record.percent	0.21	0.15	0.23	0.11
mapred.job.shuffle.merge.percent	0.77	0.88	0.89	0.76
mapred.inmem.merge. threshold	393	787	783	972
mapred.output.compress	True	True	True	True
mapred.compress.map.output	True	True	True	True
mapred.job.reduce. input.buffer.percent	0.65	0.63	0.52	0.79

Conclusion

In conclusion, this study has demonstrated the potential of evolutionary algorithms in optimizing models for Big Data analytics. The comparative analysis of GA, PSO, and DE across various Big Data models reveals that these algorithms can significantly improve model performance in terms of accuracy, computational efficiency, and convergence rate. However, the study also underscores that no single algorithm outperforms the others across all metrics, suggesting that the choice of algorithm should be context-dependent, based on specific model

requirements and data characteristics. The findings indicate the promise of hybrid approaches that combine the strengths of different evolutionary algorithms to further enhance optimization outcomes. Future research should explore these hybrid models and investigate their application in more complex and dynamic Big Data environments.

References

[1] C. Li, A. Qouneh, and T. Li. iswitch: coordinating and optimizing renewable energy powered server clusters. In *Proc. the Int'l Symposium on Computer Architecture (ISCA)*, 2012.

[2] C. Li, R. Zhou, and T. Li. Enabling distributed generation powered sustainable high-performance data center. In *Proc. of the IEEE Int'l Symposium on High-Performance Computer Architecture (HPCA)*, 2013.

[3] M. Li, J. Tan, Y. Wang, L. Zhang, and V. Salapura. Sparkbench: A comprehensive benchmarking suite for in memory data analytic platform spark. In *Proc. of ACM Int'l Conference on Computing Frontiers (CF)*, 2015.

[4] M. Li, L. Zeng, S. Meng, J. Tan, L. Zhang, N. Fuller, and A. R. Butt. mronline: Mapreduce online performance tuning. In *Proc. of ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, 2014.

[5] S. Li, S. Hu, s. Wang, L. Su, T. Abdelzaher, I. Gupta, and R. Pace. Woha: Deadline-aware map-reduce workflow scheduling framework over hadoop clusters. In *Proc. of IEEE ICDCS*, 2014.

[6] X. Li, Y. Wang, Y. Jiao, C. Xu, and W. Yu. Coomr: Cross-task coordination for efficient data management in mapreduce programs. In *Proc. Int'l Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2013.

[7] Z. Li, Y. Cheng, C. Liu, and C. Zhao. Minimum standard deviation difference-based thresholding. In *Proc. Int'l Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2010.

[8] Lin, M.and Wierman, A. L., and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proc. IEEE Int'l Conference on Computer Communications (INFOCOM)*, 2011.

[9] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. *ACM SIGMETRICS*, 40(1), 2012.

[10] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew. Geographical load balancing with renewables. In *ACM SIGMETRICS*, 2011.

[11] Y. Lu, T. F. Abdelzaher, and A. Saxena. Automated and agile server parameter tuning by coordinated learning and control. *IEEE Trans. on Parallel and Distributed Systems*, 25(4), 2014.

[12] MapR. The executive's guide to big data. http://www.mapr.com/resources/white-papers.

[13] N. Mi, G. Casale, L. Cherkasova, and E. Smirni. Burstiness in multi-tier applications: Symptoms, causes, and new models. In *Proc. ACM/IFIP/USENIX Int'l Middleware Conference*, 2008.

[14] J. Moses, R. Iyer, R. Illikkal, S. Srinivasan, and K. Aisopos. Shared resource monitoring and throughput optimization in cloud-computing datacenters. In *Proc. of the IEEE Int'l Symposium on Parallel and Distributed Processing (IPDPS)*, 2011.

[15] NREL. Measurement and instrumentation data center. http://www.nrel.gov/midc/.

[16] P. Padala, K.-Y. Hou, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, and A. Merchant. Automated control of multiple virtualized resources. In *Proc. of the EuroSys Conference (EuroSys)*, 2009.

[17] V. Patil and V. Chaudhary. Rack aware scheduling in hpc data centers: An energy conservation strategy. In *Proc. of IEEE Int'l Parallel and Distributed Processing Symposium (IPDPS)*, 2011.