DATA ALLOCATION IN DISTRIBUTED DATABASE SYSTEMS

Minor Project (IT897) Report Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY
in
INFORMATION TECHNOLOGY
by
NIKHIL VERMA
(222IT026)

DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

October, 2023

DECLARATION

I hereby *declare* that the Minor Project (IT897) entitled “DATA ALLOCATION IN DISTRIBUTED DATABASE SYSTEMS”, which is being submitted to National Institute of Technology Karnataka, Surathkal, in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Information Technology in the Department of Information Technology, is a *bonafide report of the work carried out by me.* The material contained in this report has not been submitted at any University or Institution for the award of any degree.

Place: NITK Surathkal
Date: 25/10/2023

Nikhil Verma (222IT026)
Department of Information Technology

CERTIFICATE

This is to certify that the Minor Project Work (IT897) Report entitled “DATA ALLOCATION IN DISTRIBUTED DATABASE SYSTEMS” submitted by Nikhil Verma (222IT026) as the record of the work carried out by him, is accepted as the Minor Project Work (IT897) Report submission in partial fulfilment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

Dr. Shrutilipi Bhattacharjee
Minor Project (IT897) Guide

Dept. of Information Technology
NITK Surathkal, Mangalore

ACKNOWLEDGEMENT

I have taken efforts in this project, however, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I would like to express my gratitude towards all members of National Institute of Technology Karnataka Surathkal, Information Technology Department for their kind cooperation and encouragement which helped me in completion of this project.

ABSTRACT

Effective data allocation strategies are crucial to ensure optimal performance, fault tolerance, and scalability in such distributed environments. The problem of allocating data to sites in distributed database system is a NP hard optimization. In this we have proposed a hybrid algorithm based on Particle Swarm Optimization, Genetic Algorithm and Variable Neighborhood search algorithm. Performance of our proposed approach in experimentally evaluated against state of the art methods using benchmarks reported in literature.

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Data allocation problem in distributed database system is a NP hard optimization problem i.e. these problem cannot be solved in polynomial time but these problem can be approximately solved using metaheuristic algorithms. The data allocation problem (DAP) is the task of assigning data fragment to sites of distributed database system in order to minimize transaction cost. Metaheuristic algorithm can be used to find approximate solution to the data allocation problem. Recently hybrid algorithm are being used to solve such optimization problem because this increases search efficiency. In this study we have proposed a hybrid method consisting of Particle Swarm Optimizatio (PSO), Genetic Alrorithm (GA) and Variable Neighborhood search (VNS).

Particle Swarm Optimization (PSO) is a population-based optimization algorithm inspired by the collective behavior of birds or fish. PSO aims to iteratively improve candidate solutions by adjusting their positions in the search space based on their fitness values. Like other metaheuristics, PSO efficiently explores extensive search spaces within acceptable computational time, requiring minimal problem-specific information. Although PSO doesn't guarantee finding the optimal solution, it strives to discover feasible and high-quality solutions for optimization problems.

Genetic Algorithm (GA) is also a powerful population-based optimization algorithm designed to improve candidate solutions iteratively by mimicking the process of natural selection and evolution. GAs are part of a family of metaheuristics, capable of exploring extensive search spaces efficiently within reasonable computational time. They operate without requiring in-depth problem-specific knowledge. Although GAs do not guarantee finding the optimal solution, they excel at discovering feasible and high-quality solutions for optimization problems.

Variable Neighborhood Search (VNS) is a versatile metaheuristic algorithm used for solving optimization problems. VNS iteratively explores different neighborhood structures to refine candidate solutions, aiming to find high-quality solutions within an acceptable computational timeframe. VNS falls within the category of metaheuristics, which are efficient tools for traversing vast search spaces without requiring exhaustive problem-specific information. While VNS does not guarantee optimal solutions, it discovers feasible and often high-quality solutions for optimization problems.

Chapter 2

LITERATURE REVIEW

Nasser Lotfi [1] proposed a novel hybrid algorithm for solving data allocation problem using differential evolution and variable neighborhood search algorith. In his work he has first used differential evolution algorithm to optimize a set of randomly generated solution. Then on each solution in the population he has applied variable neighbor hood search algorithm. the best solution among the population is selected as solution to DAP.

Kaya et al. [2] in their review paper explained application of Artificial Bee Colony (ABC) algorithm to solve combinatorial optimization problems. In this research they have studied different modification to the ABC algorithm for solving large variety of problems.

Mahi et al. [3] proposed a new approach to solve data allocation problem in distributed database using Particle Swarm Optimization (PSO) algorithm. In their research they have evaluated their approach using benchmark method to test DAP problem of distributed database system.

Chikhout et al. [4] proposed solution to the problem of data placement in storage as a service federated cloud. First they have defined the data placement as multi objective optimization problem then they have used a type of Genetic algorithm. In this to generate good initial population they have used CPLEX to find solution using exact method. To test their proposed approach they have used cost model for federated cloud system.

Adi et al. [5] presented a solution to QAP and DAP using Ant Colony Optimization algorithm. They have also proposed cost model for data allocation problem in distributed database. This cost model is used as a benchmark for testing solution to data allocation problem.

Wang et al. [6] proposed a hybrid heuristic method to solve large-scale combinatorial optimization problem. They have used diversification technique to generate diverse initial population and improvement technique to enhance the quality of solution in the population.

2.1 Outcome of Literature Review

On reviewing above literature we learned that the data allocation problem in distributed database is a single objective optimization problem and problem of data placement in federated cloud is a multi objective optimization problem. We have also learned that hybrid algorithms are giving very good results for single objective optimization problem. In the literature review we also learned about method to test our proposed algorithm on benchmark set.

Chapter 3

METHODOLOGY

This study proposes a novel hybrid algorithm to data allocation problem in distributed database using particle swarm optimization, genetic algorithm and variable neighbourhood search.

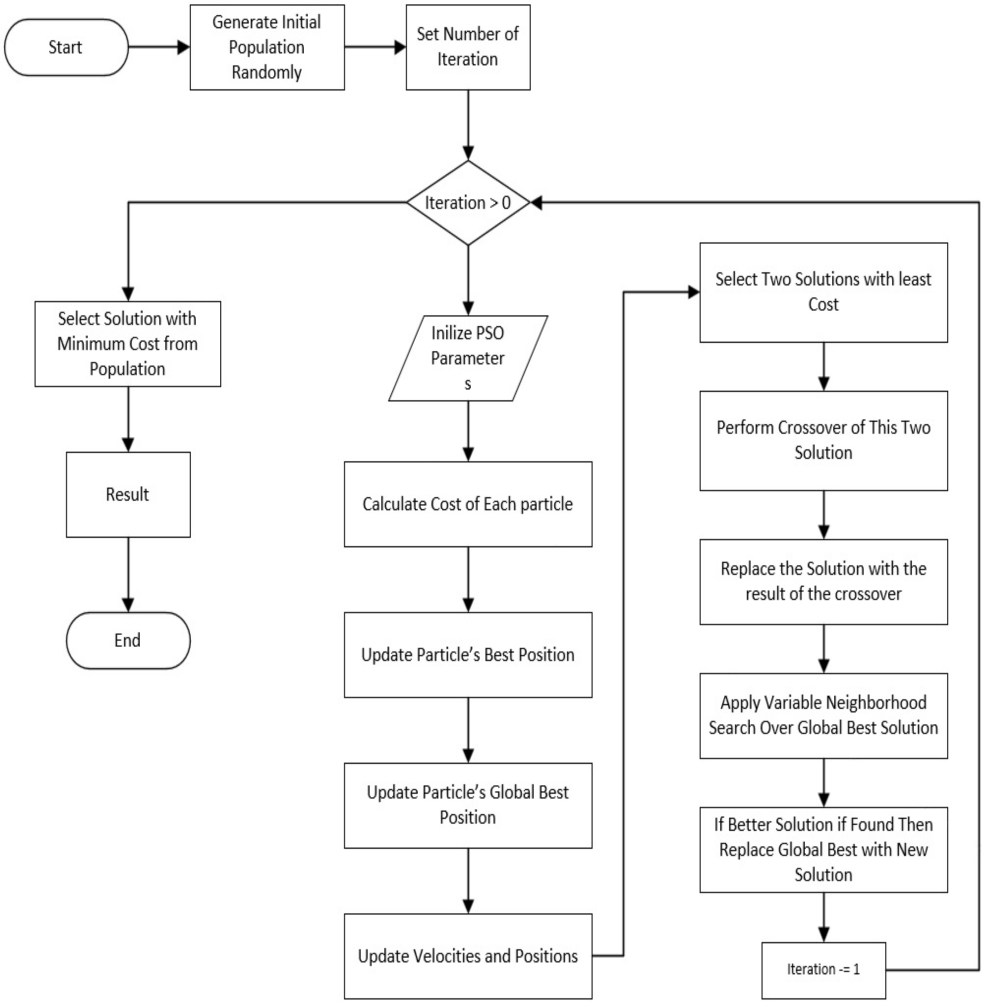

Start
Generate Initial Population Randomly
Set Number of Iteration
Iteration > 0
Select Solution with Minimum Cost from Population
Result
End
Inilize PSO Parameter s
Calculate Cost of Each particle
Update Particle's Best Position
Update Particle's Global Best Position
Update Velocities and Positions
Select Two Solutions with least Cost
Perform Crossover of This Two Solution
Replace the Solution with the result of the crossover
Apply Variable Neighborhood Search Over Global Best Solution
If Better Solution if Found Then Replace Global Best with New Solution
Iteration -= 1

Figure 3.0.1: Flow Chart of The Proposed Solution to DAP

3.1 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a computational optimization algorithm inspired by the social behavior of birds and fish. PSO is a heuristic optimization technique that can be used to find approximate solutions to optimization and search problems.

The fundamental idea behind PSO is to model the behavior of a swarm of particles in a multidimensional search space, where each particle represents a potential solution to the optimization problem. The particles move through the search space, and their movements are influenced by their own best-known position and the best-known position of the entire swarm. The PSO algorithm uses the concepts of exploration and exploitation. Exploration is achieved through the randomness in particle movement, allowing the search space to be thoroughly explored. Exploitation occurs by guiding particles toward the best-known positions found so far. The main steps of the PSO algorithm are:

- Initialization: In this step population of particle is randomly initialized by assigning random velocities to each particle.

- Objective Function Evaluation: We calculate objective function value evaluated for each particle in the population.

- Updating Particle's Best Position: Update the particle's best-known position (local best) if its current position has a better fitness value compared to its previous best.

- Updating Global Best Position: In this step we Update the global best position (best solution found by the entire swarm) based on the fitness values of all particles.

- Updating Velocities and Positions: First we Update the velocities of each particle based on its previous velocity, its best-known position, and the global best position. Then the position of each particle is updated based on its current velocity.

- Stopping Criterion: We check a stopping criterion (e.g., reaching a specified number of iterations or achieving a desired fitness threshold). If the stopping criterion is met, then we terminate the algorithm; otherwise, we go back to step 2.

3.2 Genetic Algorithm

A Genetic Algorithm (GA) is a search heuristic based on the principles of natural selection and genetics. It's often used to find approximate solutions to optimization and search problems. GAs are inspired by the process of natural selection, where the fittest individuals are more likely to survive and pass their genetic information to the next generation. The main components and steps of a typical Genetic Algorithm:

- Initialization: In this step we create initial population of potential solutions (chromosomes) to the problem at hand. Each solution is represented by a set of parameters called a chromosome.

- Objective Function Evaluation: We calculate objective function value evaluated for each particle in the population.

- Selection: In this step we select individuals (chromosomes) from the current population based on their fitness. The fitter individuals have a higher chance of being selected, mimicking the natural selection process.

- Crossover (Recombination): In this step pair selected individuals to create offspring (children) through a crossover operation. Crossover combines the genetic information of the parent chromosomes to generate new potential solutions.

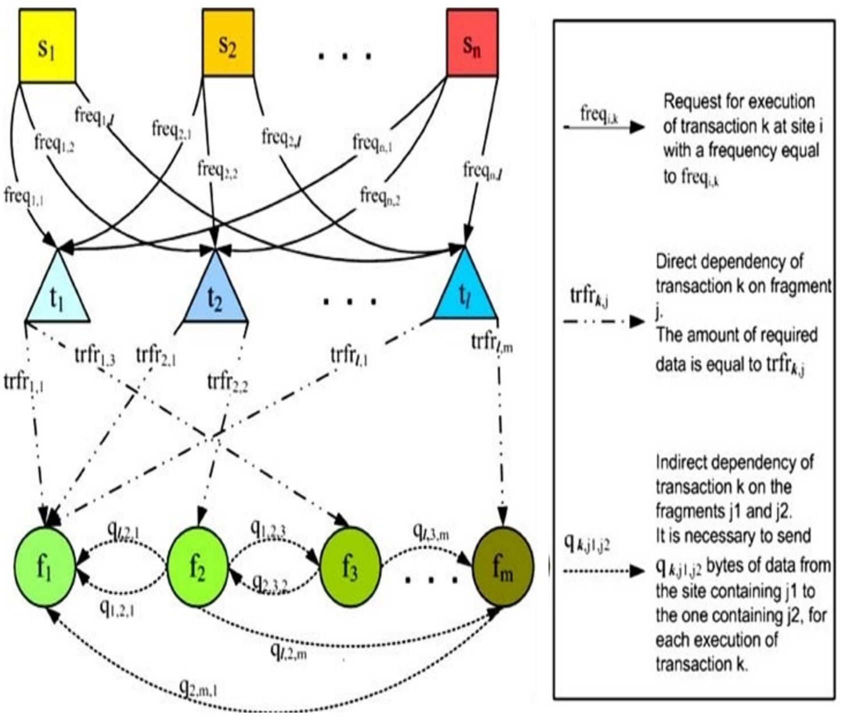
- Mutation: In this step we apply a mutation operation with a certain probability to some of the offspring. Mutation introduces small random changes in the offspring's genetic information, promoting genetic diversity in the population.

- Replacement: We replace the old generation with the new population of offspring and mutated individuals.

- Stopping Criterion: We check a stopping criterion (e.g., reaching a specified number of iterations or achieving a desired fitness threshold). If the stopping criterion is met, then we terminate the algorithm and return the best solution; otherwise, we go back to step 2.

3.3 Data Allocation Problem

The purpose of the DAP is to determine a placement of fragments at the best different sites so as to minimize the total transaction cost when a query is taken from one site to another. Firstly,a dataset is required to solve the DAP. The dataset contains n sites, m fragments and l transactions. The dataset containing sites, fragments and transactions is constructed according to the benchmark method [1]. The dependency of sites on fragments to execute transactions in shown in figure 3.3.1.



S1
S2
. . .
Sn
freq1,l
freq1,2
freq2,1
freq2,l
freqn,1
freq2,2
freqn,2
freqn,l
freq1,1
t1
t2
. . .
tl
trfr1,3
trfr2,1
trfr1,1
trfr2,2
trfrl,1
trfrl,m
ql,2,1
q1,2,3
ql,3,m
q2,3,2
q1,2,1
ql,2,m
q2,m,1
f1
f2
f3
. . .
fm
freqi,k
Request for execution of transaction k at site i with a frequency equal to freqi,k
trfrk,j
Direct dependency of transaction k on fragment j.
The amount of required data is equal to trfrk,j
qk,j1,j2
Indirect dependency of transaction k on the fragments j1 and j2.
It is necessary to send qk,j1,j2 bytes of data from the site containing j1 to the one containing j2, for each execution of transaction k.

Figure 3.3.1: Transaction-fragment and site-transaction dependencies

Here S are sites, t are transaction and f are data fragments

Chapter 4

EXPERIMENTS, RESULTS

4.1 Implementation Details

For implementation and evaluation of the proposed model we have used PYTHON language. As various parameter are generated randomly within the constraint we have used pseudo random number generator. The various parameters for the implementation of the proposed solution are:

Table 4.1.1: Parameter values used in proposed hybrid solution

Parameter description	**Value**
Fragment size	10
Transmission costs between two sites	[0 - 1]
Number of transactions	20
Probability of transaction requested at a site	0.7
Probability of fragment accessed by transaction	0.4
Probability of a transaction needing data trans- mission between two sites	0.025
Population size	200
Number of generations	200

4.2 Results and Comparison

We have evaluated our proposed hybrid solution for instance size of 5,10,15 and 20 and have compared our results with the state of art methods to solve DAP. Our obtained cost and comparisons are shown in following table. The proposed method is also faster than the DEVNS algorithm for solving DAP. In experiment we found that the proposed method is four times faster than DEVNS method.

Table 4.2.1: Cost comparison of methods for different DAP instance sizes (*cost value is column* $\times 10^6$)

Size	**SA**	**RTS**	**ACO**	**PSO-DAP**	**DEVNS**	**Proposed approach**
5	0.04	0.04	0.04	0.02	0.03	0.010
10	0.31	0.31	0.31	0.05	0.06	0.032
15	0.98	0.98	0.98	0.41	0.52	0.47
20	2.62	2.61	2.61	0.77	1.47	1.23

Chapter 5

CONCLUSIONS & FUTURE WORK

5.1 Conclusion

This paper proposes an novel hybrid method to problem of allocation data fragments to sites in distributed database systems. This method works based on a strategy to combine particle swarm optimization to improve the quality of randomly generated solutions then these solution are further improved by genetic algorithm. On the best solution we apply variable neighborhood search algorithm to further find more optimized solution in the neighborhood of current best solution.

5.2 Future Scope

In this project we have implemented a hybrid method by combining best features of three algorithms. This solution can be used to allocate data fragment in distributed database. Also this method can be modified to find optimal data placement strategy for cloud storage service by modifying the cost calculation method.

REFERENCES

[1] Lotfi, Nasser. ”Data allocation in distributed database systems: a novel hybrid method based on differential evolution and variable neighborhood search.” SN Applied Sciences 1, no. 12 (2019): 1724.

[2] Kaya, Ebubekir, Beyza Gorkemli, Bahriye Akay, and Dervis Karaboga. ”A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems.” Engineering Applications of Artificial Intelligence 115 (2022): 105311.

[3] Mahi, Mostafa, Omer Kaan Baykan, and Halife Kodaz. ”A new approach based on particle swarm optimization algorithm for solving data allocation problem.” Applied Soft Computing 62 (2018): 571-578.

[4] Chikhaoui, Amina, Laurent Lemarchand, Kamel Boukhalfa, and Jalil Boukhobza. ”Multi-objective optimization of data placement in a storage-as-a-service federated cloud.” ACM Transactions on Storage (TOS) 17, no. 3 (2021): 1-32.

[5] Karimi Adl, Rosa, and Seyed Mohammad Taghi Rouhani Rankoohi. ”A new ant colony optimization based algorithm for data allocation problem in distributed databases.” Knowledge and Information Systems 20 (2009): 349-373.

[6] Wang, Haibo, and Bahram Alidaee. ”A new hybrid-heuristic for large-scale combinatorial optimization: A case of quadratic assignment problem.” Computers and Industrial Engineering 179 (2023): 109220.