International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Implementation of Optimization Algorithms for the Creation of a Real-Time Coordination System for Overcurrent Relays

Vijaykumar S. Kamble[1], Brajesh Mohan Gupta[2], Prabodh Khampariya[3], Amol A. Kalage[4]

Submitted: 15/09/2023 **Revised**: 18/11/2023 **Accepted**: 29/11/2023

Abstract: DE and ACO were chosen based on their documented superior performance across diverse literature, and their novelty in the security industry, where no prior implementation attempts had been made. Various approaches exist for coordinating overcurrent relays, each offering unique performance advantages that distinguish it from others. As the number of coordination pairings grew, so did GA exhibited the least favorable outcomes, proving to be the slowest with the greatest impact on execution time. This observation was derived from a comparative analysis of the outputs of the three algorithms. In contrast, ACO and DE not only demonstrated superior speed but also consistently yielded better results, showcasing resilience to an escalation in the number of coordination pairs.

Keywords: *ACO, DE, GA, Optimization, Overcurrent Relays*

1. Introduction

Protection is a crucial element extensively employed across the entire electrical power system. Due to its cost-effectiveness compared to other protection methods, in sub-transmission and distribution networks, overcurrent relays are often used. The primary objective in coordinating these overcurrent relays is to find settings that decrease time. These relays need to operate in response to faults within the protective zone. Simultaneously, it aims to establish a predetermined backup schedule for relays situated in zones adjacent to the protective area. Consequently, the highest fault current detected by a relay within its safe zone must surpass the fault currents detected in neighboring zones. It is acknowledged that the overcurrent protection principle may be beyond the scope for specific configurations of mesh systems, essentially pushing the limits of its protective capacity. Coordinating distribution system overcurrent relays is essential to fulfill fundamental requirements such as sensitivity, reliability, selectivity, and speed.

Through the implementation of real-time optimization methods in coordinating overcurrent relays, we achieved enhanced results. These included accelerated fault extinction, a diminished probability of erroneous tripping, and a decreased measure of fault extinction delay across seasons. The swift attainment of these outcomes was a direct result of our ability to expedite the optimization process.

[1]*Research Scholar, Department of Electrical Engineering, School of Engineering, SSSUTMS, Sehore, M.P.(India)*
[1]*Assistant Professor, Department of Electrical Engineering, AISSMS's IOIT, Pune, Maharashtra (India)*
[2, 3] *Professor, Department of Electrical Engineering, School of Engineering, SSSUTMS, Sehore, M.P.(India)*
[4]*Professor, Department of Electrical Engineering, SIT, Lonavala, Pune(India)*

2. Literature Survey

Over the past few decades, the predominant approach among protection specialists for coordinating Directional Overcurrent Relays (DOCRs) has been manual coordination. However, this manual coordination has been reframed as an optimization challenge because of its complicated and nonlinear nature of the underlying problem. To address this complexity, various optimization strategies have been proposed. One such strategy, introduced in 1988, involved coordinating DOCRs using linear programming within the framework of deterministic optimization theory (LP). This approach presented the challenge as a linear function, with dials generated based on provided pickup current values. Subsequent research has further explored the application of LP to this issue, appreciating its straightforward nature.

In the contemporary landscape, heuristic methods from the realm of artificial intelligence (AI) have experienced a remarkable surge in popularity for addressing coordination challenges. Noteworthy examples of these approaches encompass the particle swarm optimization (PSO) and genetic algorithm (GA). The GA has garnered frequent attention across diverse literature due to its robustness, simplicity and straightforward execution. This method is rooted in progressive principles, mirroring the natural gene selection, encompassing key processes such as selection, reproduction, and mutation, forming the foundational concepts of the algorithm.

Lately, hybrid approaches have surfaced to tackle coordination challenges, offering advantages such as a reduced search space, quicker execution times, and a decrease in the required number of iterations required to arrive at a solution. Among these, hybridizing the blending in particle swarm optimization (PSO) and genetic algorithm (GA) stand out as two recently devised methods seamlessly integrated with linear programming (LP).

Although Ant Colony Optimization (ACO) hasn't been applied in coordination studies, it has recently found utility in the examination of power generation scheduling, power flow economic dispatch and reactive power flow planning. ACO has demonstrated its efficacy as a potent tool in solving intricate problems across diverse domains. Notably, ACO offers a distinct advantage over Genetic Algorithms (GA) through the utilization of a pheromone matrix which plays a global memory role, contributing to more efficient and rapid convergence of solutions.

DE is a form of Evolutionary Algorithm (EA). However, in contrast to many other EAs, DE stands out for its simplicity and ease of implementation. In various studies comparing DE with other algorithms, its overall performance, including accuracy, convergence speed, and robustness, proves highly compelling. These attributes render DE particularly attractive for addressing real-world optimization problems, emphasizing the importance of obtaining an approximate solution within a reasonable computational timeframe.

3. Real-Time Coordination of Relay

The study of coordinating DOCRs predominantly centers around a fixed network structure within an Interconnected Mesh Power System (IMPS), framed as an issue of optimization. However, practical power systems operate in dynamic environments with ever-changing topologies due to line outages, transformer failures, and variations in generating unit status. In certain scenarios, alterations in network topology or element operation can lead to a lack of selectivity, causing the protective system to operate non-selectively, resulting in a loss of coordination.

This paper aims to address this challenge by proposing a solution to coordinate all protections seamlessly during every change in element operation or network topology. Achieving this goal necessitates the creation of an algorithm that operates in real time, works synergistically regarding optimization algorithm, ensuring continuous coordination effectiveness in dynamic operating conditions.

Figure 1 depicts the flow diagram for coordination of overcurrent relays in real-time.

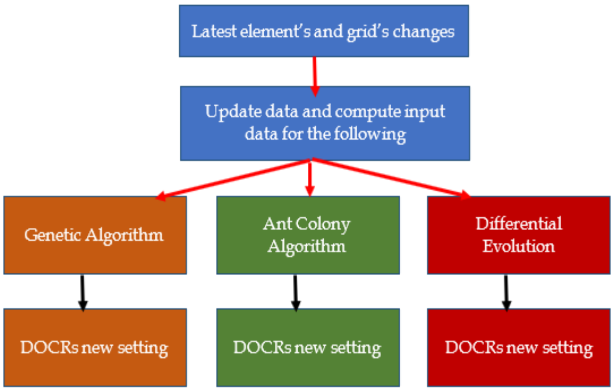

Fig 1 : Flow Diagram of Real Time Coordination.

4. Real Time Algorithm

An algorithm that operates in real time involves gathering data on the most recent changes in elements and the network. This data serves as input for subsequent relay coordination calculations. It is thought that the online update hardware mechanism has already been created; the required hardware merely has to be installed together with an appropriate real-time algorithm.

The following is a summary of the algorithm:

1. Initiate the process by updating the system's data in accordance with changes in elements and the network.
2. Construct or modify the Y_{bus} utilizing the inverse of the Inspection method and the Incident method based on the acquired data.
3. Automatically generate lists of "Relay Names" and "Coordination Pairs."
4. Execute a load flow analysis, employing methods such as Newton Raphson or an alternative technique.
5. Construct or modify the Zbus using Partial Inversion Motto and Block construction method.
6. Conclude the algorithm by running fault analysis, employing methods like Symmetrical Components method or Thevenin's method.

Upon completion of the aforementioned steps, the algorithm will have determined the pairs of coordination as well as the fault currents and maximum load for each relay, serving as input for the optimization strategies used on the initial topology of the network. This is represented in Figure 2.

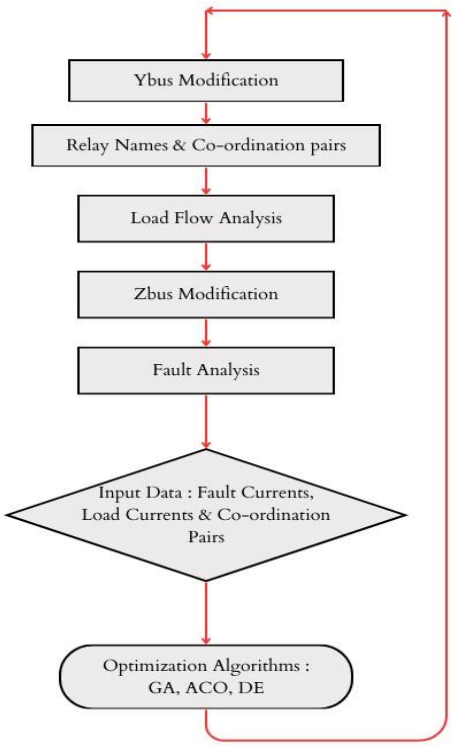

Fig 2: Real-Time Coordination Flow Diagram.

Yet, to guarantee that the relay configurations derived from the subsequent algorithm for coordinating align with one or more output without an element affecting synchronization, it is imperative to compute maximum load and currents at fault considering various n-1 contingency topologies. Prior to sending data to the optimization algorithms in charge of coordinating overcurrent relays, this algorithm incorporates sensitivity filtration. This crucial step guarantees the efficient coordination of all pairs in the system. Coordination pairs not aligning with the specified sensitivity analysis level will be excluded from the coordination process. This strategy prevents the algorithms for optimization from expending unnecessary effort on attempting to figure out how to configure these insensitive relay pairs, recognizing that such pairs lack suitable settings. Equation (1) gives the sensitivity analysis formula:

$$Sensitivity = \frac{I_{SC^{2\emptyset}{}_{Backup}}}{k \times I_{Load}} \qquad (1)$$

where k = 1.4, the smallest number between [1.4, 1.6]. Relays that fail to meet sensitivity criteria, even when considering the minimum k, will be excluded. Here, the term "sensitivity" refers to the relationship between the pickup current and short circuit current in the two-phases of a relay. A satisfactory sensitivity level requires the short circuit current in the two-phases to be at least 1.5 times the pickup current. This criterion is essential because currents situated close to the typical curve of relay's vertical asymptotic zone exhibit extended operation times. Using the pickup current as a point of reference, faults located near or within the vertical asymptotic area, or too close to the pickup current, may decrease sensitivity, leading to potential loss of sensitivity in certain instances.

Optimization Algorithms' Objective Function

It is critical to define the objective function that will be used to assess the effectiveness of relay settings, gauging their ability to meet specified needs. This type of function can comprise a combination of various criteria, directly influencing the optimization algorithms' (such as ACO, DE and GA) output quality. An indicator is required to convey whether a setting is unfavorable (outside satisfaction limits), satisfactory (within satisfaction limits), or optimal before its inclusion in the evaluation of objective functions. In the context of relay coordination, the time, specifically Coordination Time Interval (CTI), serves as indicator.

The initial step involves evaluating all settings for each relay within the population. The differentiating factor at this stage lies in the current flowing across a short circuit observed by both primary relay and backup relay. Despite this procedural difference using the manual approach, the underlying logic aligns with the Coordination Time Interval (CTI) principle. Consequently, equation (2) stipulates that the genuine calculation of CTI involves the subtraction of the backup time from the primary time.

$$CTI_{real} = t_{backup} - t_{primary} \qquad (2)$$

This represents the authentic CTI of the relays, commonly referred to as the limitations of the primary and backup relays. To calculate the indicator, subtract the pre-specified CTI from the actual CTI, displayed in equation (3).

$$CTI_{indicator} = CTI_{real} - CTI_{pre-specified} \quad (3)$$

As previously explained, the Coordination Time Interval (CTI) indicator serves to categorize settings as unfavorable (beyond satisfaction limits), satisfactory (within satisfaction limits), or optimal, allowing for appropriate reward or penalty assignment before integration into the objective function. Termed as CTI error, this indicator results from the subtraction of two CTIs. A zero error implies ideal settings, a rare occurrence, and most relays will not attain this perfection. A positive error suggests that the settings are good or acceptable, maintaining coordination, but with a time lag higher than the predefined threshold.

Equations (4) and (5) articulate the boundaries specified for the relay settings.

$$dial_{min} \leq dial \leq dial_{max} \qquad (4)$$

$$I_{pickup_{min}} \leq I_{pickup} \leq I_{pickup_{max}} \qquad (5)$$

Here, 'dial' represents the relay dial configuration. found between the max. $dial_{max}$ and min. $dial_{min}$ ranges and I_{pickup} is the pickup current of relay detected within its max. $I_{pickupmax}$ and min. $I_{pickupmin}$ ranges.

The primary function encompasses the total count of violations, the combined primary time and backup time, and CTI equals the total of errors in the number of pairings that are coordinated. All three algorithms— Differential Evolution Algorithm, Ant Colony Algorithm, and Genetic Algorithm, share a common primary function in their implementation. It is represented in equation (6).

It is advisable to apply a slight penalty to encourage convergence towards zero error, aligning with the pre-defined CTI. In cases where the error has a negative value, indicating a coordination loss, a more substantial penalty is recommended to prevent significant mal coordination.

$$fitness = \left(\frac{NV}{NCP}\right) + \left(\frac{\sum_{a=1}^{NCP} t_{principal_a}}{NCP}\right) \times \alpha + \left(\frac{\sum_{b=1}^{NCP} t_{backup_b}}{NCP}\right) \times \beta + \left(\sum_{L=1}^{NCP} E_{CTI_L}\right) \times \delta \qquad (6)$$

Where, α, β and δ are variables that affect the effect of any sub-objective function in any other system. The number of violations of coordination limitations is denoted by NV, NCP stands for the quantity of pairs that are coordinated, In this context, '$t_{principala}$' represents the primary operation time of relay a, and '$t_{backupb}$' represents the backup operation time of relay b and $E_{CTI_L}$is the L^{th} coordination pair's CTI error.

5. Protection Coordination

Using a Genetic Algorithm

Genetic algorithms (GA) are natural selection-based adaptive heuristic search methods applied to genes that belong to the area of evolutionary computing.

Every living organism is composed of cells, and within each cell resides an identical set of chromosomes, analogous to individuals. These chromosomes, comprised of genes or DNA blocks, encode specific proteins or traits (such as eye color), each with its designated place on the chromosome. The chromosome stores potential solution information in the form of genes with binary, real, integer, or floating-point elements.

Reproduction involves a recombination process, commonly known as crossover, where offspring or new chromosomes are created through the merging of genes from specific parent chromosomes. Typically, chromosomes with favorable fitness values are selected as parents. However, it's important to note that relying solely on the best chromosomes for reproduction may result in reaching a premature solution or getting bound in a local optimum.

The algorithm commences with numerous solution sets, represented by chromosomes, collectively constituting a population. The solutions obtained from a specific population are selected to generate a new and refreshed population. with the anticipation that it will exhibit improvement over the previous one. The selection of solutions for forming new ones is guided by their fitness; the more apt a solution, the greater its likelihood of contributing to the reproduction process.

The entire procedure is reiterated until a specified condition is fulfilled, such as attaining the maximum number of repetitions or improving the best solution. This criterion is referred to as the stopping criteria.

In cases where the number of chromosomes is insufficient, the algorithm has limited opportunities for crossover, exploring only a fraction of the search space. Conversely, an excess of chromosomes results in a broader exploration of feasible solutions, but this comes at the cost of significantly increased execution time.

Ant Colony Algorithm

Ant agents refer to a group of artificial ants tasked with constructing solutions for an optimization problem. They communicate information about solution quality using a communication method similar to that seen in real ant colonies.

In the AS graph, the matrix delineates the search space, comprising discrete combinations (states) of control variables (stages). The Pheromone matrix stores data on chemical pheromones deposited by ants, providing information on the intensity of pheromones corresponding to each discrete setting. This matrix reflects the appealingness of potential routes with relation to the solution, with higher intensity indicating a greater likelihood of being an ant agent opted to be a part of the solution.

The algorithm begins with multiple solution sets (states), collectively forming the search space for AS graphs. This AS graph remains constant during the entire search procedure, maintaining consistency and not undergoing changes from one iteration to the next.

The whole process is reiterated until a specified condition is satisfied. This point is referred to as the criterion for stopping.

The AS-graph's total number of states is represented by its size. When there are too few states, the algorithm has less chances to find the best answer, examining only a portion of the search area. Conversely, an excess of states enhances the chance of encountering the optimum result but significantly slows down the overall procedure.

Differential Evolution Algorithm

Like a conventional evolutionary algorithm (EA), the Differential Evolution algorithm uses procedural operations. But unlike typical EAs, it disturbs members of the current generation population with scaled variations from randomly selected, unique population members. This eliminates the need for a separate probability distribution to generate offspring. This feature reduces the number of mathematical operations, resulting in shorter execution times with respect to other algorithms.

The trial solutions are called genomes or parameter vectors in the Differential Evolution community. Each vector of parameters encapsulates a wealth of information pertaining to potential solutions. Crossover is a mechanism akin to reproduction, generating offspring or new individuals by means of gene recombination from specific parameter vectors known as target or parents' vectors. The individuals generated through this process are referred to as trial vectors, emerge from this recombination process.

The procedure starts with several sets of solutions, each of which is a population made up of all the parameter vectors. Selected solutions from an existing population are utilized to form a new population. Through the selection process, the algorithm ensures that the population either preserves or enhances its performance in minimizing the objective function, thus preventing deterioration.

F, Cr, and NP are the three main control parameters in DE. They're regarded as constant values.

Algorithm Comparison and Real-Time Coordination: GA, DE & ACO

In this part, we assess the performance of each algorithm using the IEEE 14 bus test system, highlighting both their positive and negative aspects. The evaluation criteria include the execution time, a crucial factor determining the algorithm's quality, resilience, and convergence capabilities. The search area is systematically narrowed by subjecting the test system to repeated iterations and analyzing outcomes, aiming for more precise results and faster algorithm convergence.

The trio of algorithms—Differential Evolution, Genetic Algorithm & Ant Colony Optimization are equipped with settings for continuous dial and k. Adjusting the step size impacts the widening of the AS graph and influences the time required for the task, with a smaller time denoting faster execution. The decision to use dial and k for real-time coordination was based on information provided in Table 1. Careful consideration ensured that these parameters were neither excessively large nor too small, optimizing result accuracy.

The algorithm execution is set to terminate after surpassing one thousand iterations, simplifying comparisons across approaches by removing individual halting conditions. This iteration limit was chosen as, beyond 1,000 iterations, the progress in algorithms becomes less apparent compared to thresholds like 3,000 or 5,000 iterations. The simulation involves a total of 500 agents for GA, DE, and ACO scenarios, prioritizing speed among participating agents.

Parameter Settings

A comprehensive evaluation and comparison of three techniques were conducted using the IEEE Fourteen bus test system, as illustrated in Figure 3. The voltage on buses connected to the high voltage side of transformers was set at 34.5kV, while buses linked to the LV side maintained a voltage of 22 kV.

A comprehensive evaluation and comparison of three techniques were conducted using the IEEE Fourteen bus test system, as illustrated in Figure 3. The voltage on buses connected to the high voltage side of transformers was set at 34.5kV, while buses linked to the LV side maintained a voltage of 22 kV.

Table 1: Parameter settings of GA, ACO & DE.

Parameters	GA	ACO	DE
CTI	0.3	0.3	0.3
k	[1.4:1.6]	[1.4:1.6]	[1.4:1.6]
dial	[0.5:2.0]	[0.5:2.0]	[0.5:2.0]
k step	continuous	0.01	continuous
dial step	continuous	0.05	continuous
R	--	5	--
Q	--	100	--
F	--	--	0.8
r	--	--	0.5
Cr	--	--	0.5
individual-ants	500	500	500
iterations	1000	1000	1000

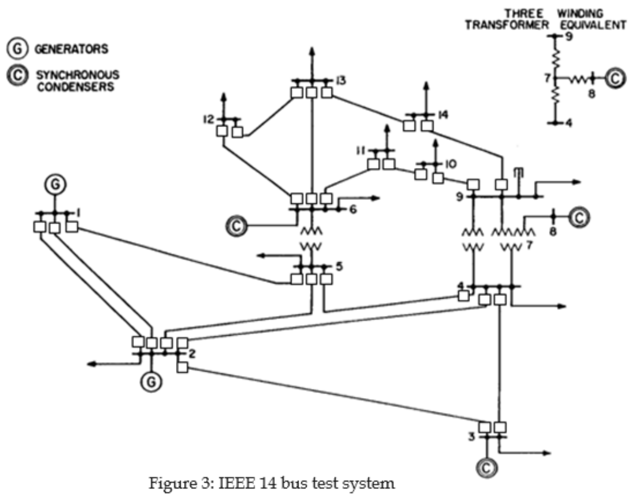

Figure 3: IEEE 14 bus test system

A comprehensive evaluation and comparison of three techniques were conducted using the IEEE Fourteen bus test system, as illustrated in Figure 3. The voltage on buses connected to the high voltage side of transformers was set at 34.5kV, while buses linked to the LV side maintained a voltage of 22 kV.

Table 1 presents data supporting the belief that all relays exhibit a highly inverted time characteristic curve. Unlike traditional naming conventions involving numerical designations, the real-time algorithm automatically assigns relay names as strings of numbers. Each relay's three-digit name signifies the proximity of the buses it connects: the first digit denotes the bus in close proximity, the second indicates the more distant bus, and the 3rd signifies the number of parallel lines that connect them.

For instance, relays connecting buses 1 and 2 near bus 1 are denoted as [1 2 1] and [1 2 2]. Relays near bus 2 for the same connection are represented as [2 1 1] and [2 1 2]. Impedance testing on lines connecting buses 1 and 2 revealed identical values, leading to shared grounding and identical maximum load currents of 815A for relays [1 2 2], [1 2 1], and [2 1 1]. Despite detecting the same maximum load currents, all three relays can withstand load currents up to 1,849 A at full capacity.

The figures presented aimed to minimize strain on participants, necessitating the computation of leakage current by opening the circuit to its far end. This operation served to obtain the maximum overcurrent detected by the relay while limiting the risk of malfunction in the remote end relay. Additionally, it was emphasized that changes in network topology or component operation would require recalculating load flow and fault analysis using a real-time method.

Specific data values include X_d values of 0.01 for generator one and 0.3 for generator two. The data from the first 14 bus loads represent the maximum load, accounting for 70% of the minimum load.

6. Result & Discussion

The simulation of the 14 bus test system is repeated, employing ACO, DE andGA, with parameters as outlined in the preceding section, specifically at minimum load conditions. Following the application of a sensitivity filter, total of 48 (forty-eight) relay coordination pairs found.

Each of the three algorithms underwent ten simulations, and the algorithm's convergence was determined by averaging the best fitness across iterations in the ten simulations. Figure 4 demonstrates this for comparison.

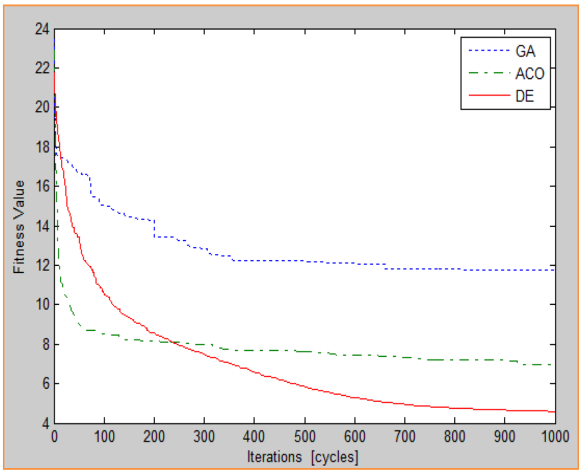

Fig 4 : Convergence analysis of average fitness for the 14 -bus test systems of GA, ACO & DE in 10 simulations at minimal load.

Table 2 displays the average number of violations of coordination constraints, the convergence of fitness averaged across iterations, and the time averages for Ant Colony Optimization, Differential Evolution & Genetic Algorithm.

Table 2: Comparison of averaged GA, ACO, and DE findings at minimum load.

Parameters	GA	ACO	DE
NV	1.4	0	0
Fitness	11.748	6.968	4.612
Time (sec)	5825	438	94

Based on these findings, it is evident that both Ant Colony Optimization & Differential Evolution exhibit lesser violations of coordination constraints, faster performance and superior convergence, compared to Genetic Algorithm. Notably, in both DE & ACO, all coordination pairs are successfully coordinated across the 10 simulations. In contrast, some of the 10 GA simulations do not achieve coordination for all pairs. DE particularly stands out for its exceptional performance in these aspects.

Table 3 displays the mean operation time, relay settings, Coordination Time Interval (CTI), and sensitivity for the ten for Ant Colony Optimization, Differential Evolution & Genetic Algorithm simulations.

Table 3: Mean relay settings, operation time, sensitivity, and CTI of GA, ACO and DE for 14 bus test system at minimum load.

Algorithm	Dial	K	Backup time	Primary time	CTI	Sensitivity
GA	1.1336	1.4903	2.7546	0.8511	1.9266	**4.03250**
ACO	1.1165	1.4434	2.1331	0.7974	1.3357	**4.17205**
DE	**0.7863**	**1.4324**	**1.4938**	**0.5297**	**0.9641**	**4.18241**

7. Conclusion

The analysis reveals that as the number of coordinating pairs increased, Genetic Algorithm (GA) exhibited the poorest outcomes, with the slowest execution time and a considerable impact on the total execution time. This conclusion stems from a comparative assessment of outcomes across the three algorithms. Conversely, both Ant Colony Optimization & Differential Evolution demonstrated significantly faster execution times, yielded superior results, and were relatively unaffected by an increase in the total number of coordinating pairs. In this particular study, the invested execution time is a crucial factor, alongside resultant quality, durability, and convergence capability.

Differential Evolution stands out with the best overall performance among the algorithms investigated in this paper when compared to others. The application of real-time coordination validated the theory's correctness. The relay operating time is reduced, system sensitivity is heightened, and preparations are made for potential emergencies.

References

[1] Mohamed M. Mansour, Said F. Mekhamer, and Nehad El-Sherif El-Kharbawe, "A modified particle swarm optimizer for the coordination of directional overcurrent relays", *IEEE Transactions on Power Delivery*, vol. 22, No.3, pp. 1400-1410, July 2007.

[2] Reza Mohammadi, Hossein Askarian Abyaneh, Farzad Razavi, Majid Al- Dabbagh, Seyed H. H. Sadeghi, "Optimal relays coordination efficient method in interconnected power systems", *Journal of Electrical Engineering*, vol. 61, No. 2, pp. 75-83, 2010.

[3] Kamble, V. S., Khampariya, P., & Kalage, A. A.(2023). “A Survey on the Development of Real-Time Overcurrent Relay Coordination Using an Optimization Algorithm.” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11(3s), 104–114

[4] Manohar Singh, B. K. Panigrahi, "Optimal overcurrent relay coordination in distribution system", *IEEE*, 2011.

[5] Kamble, V. S., Khampariya, P., & Kalage, A. A.(2023). “Application of optimization algorithms in the development of a real-time coordination system for overcurrent relays.” *The Scientific Temper (2023)* Vol. 14 (1): 165-171

[6] Dharmendra Kumar Singh and S. Gupta, "Optimal coordination of directional overcurrent relays: A genetic algorithm approach", *IEEE Students' Conference on Electrical, Electronics and Computer Science*, 2012.

[7] A. Motie Girjandi, M. Pourfallah, "Optimal coordination of overcurrent relays by mixed genetic and particle swarm optimization algorithm and comparison of both", *International Conference on Signal, Image Processing and Applications*, vol. 21, 2011 IACSIT Press, Singapore.

[8] J. A. Sueiro, E. Diaz-Dorado, E. Míguez, and J. Cidrás, “Coordination of directional overcurrent relay using evolutionary algorithm and linear programming,” Int. J. Electr. Power Energy Syst., vol. 42, pp. 299–305, 2012.

[9] D. Birla, R. P. Maheshwari, and H. O. Gupta, “A new nonlinear directional overcurrent relay coordination technique, and banes and boons of near-end faults-based approach,” IEEE Trans. Power Deliv., vol. 21, pp. 1176–1182, 2006.

[10] K. A. Saleh, H. H. Zeineldin, A. Al-Hinai, and E. F. El-Saadany, “Optimal Coordination of Directional Overcurrent Relays Using a New Time–Current–Voltage Characteristic,” IEEE Trans. Power Deliv., vol. 30, pp. 537–544, 2015.

[11] P. P. Bedekar and S. R. Bhide, “Optimum Coordination of Directional Overcurrent Relays Using the Hybrid GA-NLP Approach,” IEEE Trans. Power Deliv., vol. 26, no. 1, pp. 109–119, Jan. 2011.

[12] Y. Damchi, J. Sadeh, and H. Mashhadi, “Applying Hybrid Interval Linear Programming and Genetic Algorithm to Coordinate Distance and Directional Over- current Relays,” Electr. Power Components Syst., vol. 44, pp. 1–12, 2016.

[13] M. Singh, B. Panigrahi, A. R. Abhyankar, and S. Das, “Optimal coordination of directional over-current relays using informative differential evolution algorithm,” J. Comput. Sci., vol. 5, pp. 269–276, 2014.

[14] M. Y. Shih and A. Conde, “Implementation of Directional Overcurrent Relay Coordination Approaches in Electrical Networks,” Electr. Power Components Syst., vol. 43, no. 19, pp. 2131–2145, 2015.

[15] A. Srivastava, J. Tripathi, S. R. Mohanty, and B. Panda, “Optimal Overcurrent Relay Coordination with Distributed Generation Using Hybrid Particle Swarm Optimization– Gravitational Search Algorithm,” Electr. Power Components Syst., vol. 44, pp. 1–12, 2016.

[16] M. Singh, B. K. Panigrahi, and A. R. Abhyankar, “Optimal coordination of directional over-current relays using Teaching Learning-Based Optimization (TLBO) algorithm,” Int. J. Electr. Power Energy Syst., vol. 50, no. Complete, pp. 33–41, 2013.

[17] A. A. Kalage and N. D. Ghawghawe, "Optimum Coordination of Directional Overcurrent Relays Using Modified Adaptive Teaching Learning Based Optimization Algorithm," Intell. Ind. Syst., vol. 2, no. 1, pp. 55–71, Mar. 2016.

[18] F. A. Albasri, A. R. Alroomi, and J. H. Talaq, "Optimal Coordination of Directional Overcurrent Relays Using Biogeography-Based Optimization Algorithms," IEEE Trans. Power Deliv., vol. 30, no. 4, pp. 1810–1820, Aug. 2015.

[19] V. N. Rajput, F. Adelnia, and K. S. Pandya, "Optimal coordination of directional overcurrent relays using improved mathematical formulation," IET Gener. Transm. Distrib., vol. 12, no. 9, pp. 2086–2094, 2018.

[20] M. Singh, B. Panigrahi, and A. R. Abhyankar, "Optimal coordination of electromechanical- based overcurrent relays using artificial bee colony algorithm," Int. J. Bio- Inspired Comput., vol. 5, pp. 267–280, 2013.

[21] T. Amraee, "Coordination of Directional Overcurrent Relays Using Seeker Algorithm," IEEE Trans. Power Deliv., vol. 27, no. 3, pp. 1415–1422, Jul. 2012.