Distance and Time Overcurrent Relay Coordination with an Autosolver Framework

Joe Perez, P.E.
SynchroGrid
Bryan, TX USA
jperez@synchrogrid.com

Abram Krislock, Ph.D.
SynchroGrid
Bryan, TX USA
abram.krislock@synchrogrid.com

Luke Hankins, P.E.
SynchroGrid
Bryan, TX USA
lukeh@synchrogrid.com

Lucille Lamb
SynchroGrid
Bryan, TX USA
lucille.lamb@synchrogrid.com

***Abstract*—Relay coordination is becoming much more difficult as the complexity of the grid increases, yet it is a vital part of a comprehensive protection strategy for modern power systems. With the recent introduction of PRC-027-1 [1] and its requirement for coordination to be reevaluated at regular intervals, innovation to reduce the time and resources required for this activity is essential. Achieving coordination and ensuring that time overcurrent and distance relays operate in a predictable manner can be quite burdensome, especially in highly coupled power systems with tight loops in the topology structure. In a previous paper, we presented a prototype of a coordination autotuner framework and demonstrated its use for the automatic generation of tuned pickup, time dial, and curve settings for directional time overcurrent relays on a mix of synthetic and real-world grids.**

Here, we present research results from a DOE grant that demonstrate improvements we have made to our autotuner framework. The updated framework moves us significantly closer to a general-purpose coordination autotuner capable of performing the mundane, iterative work required during coordination studies. We focus on several key areas, including support for heterogeneous protective element coordination, additional contingencies, coordination constraint relaxation, and customizable solution optimization. We show the use of each in the experiments and demonstrate how they allow us to better support common cases encountered in real-world coordination studies. Together, these new capabilities address many simplifying restrictions in our previous work. We are increasingly confident that autotuning-assisted coordination studies are a viable and important advancement quickly coming on the horizon in system protection.

***Index Terms*—Directional Time Overcurrent, Microprocessor Relays, Time Dials, Wide Area Coordination, Distance Elements**

I. Introduction

Wide area coordination is one of the very critical studies needed in system protection. Given the new compliance requirements from PRC-027-1 [1], electric companies are under great pressure to perform relay coordination studies quickly and correctly. Current methods require a great deal of manual effort to verify coordination on a large scale. They are long, arduous, and prone to human error. Software automation can provide a solution to these difficulties. Engineers can avoid copy and paste drudgery and focus their expertise on the interesting parts of grid protection.

In this paper, we show the use of an Auto Solver framework to solve coordination problems. The framework gets coordination problem data directly from the short circuit model to avoid human error. The solver formulates coordination problems as Mixed-Integer Linear Problems, as shown in [2]. In previous works [3]–[5], we presented frameworks which would solve overcurrent coordination problems automatically. We demonstrated the solver's ability to tackle real-world problems. We also made the solver heterogeneous, considering distance and overcurrent relay responses when tuning overcurrent relay settings for coordination.

Now we have further improved our solver. First, we made the solver truly heterogeneous, allowing it to tune the overcurrent and distance relay settings simultaneously in one coordination study. We also added custom optimization to the solver, so that it can find coordination solutions based on additional concerns, such as the resource cost of changing settings. Additionally, we improved constraint relaxation within the solver, adding the ability for the solver to treat constraint violations differently based on their severity. For instance, a violation under normal conditions can be considered worse than a violation under contingencies.

The remainder of this paper describes the solver framework and shows examples of these new improvements. In Section II, we outline the framework of the solver and describe the software tools comprising it.

Section III gives more details about our improvements, showing examples for each. Section III-A shows the solver's ability to incorporate heterogeneous relay elements in the coordination problem, including both overcurrent and distance relay settings and responses. In Section III-B, we consider the "costs" of changing relay settings, and show how the solver finds different solutions to coordination depending on the costs. Section III-C shows how the solver can relax constraints to solve an otherwise unsolvable coordination problem. We also discuss how changing the constraint relaxation "weights" can change the solution results.

Finally, we conclude in Section IV.

II. Framework Overview

A. Coordination Auto Solver Framework

We briefly describe our autotuning framework, which is depicted in Figure 1. *Coordination Project Inception* is the start of the project where the area of the grid to coordinate is chosen. The buses, corresponding relays, initial CTI to aim for,

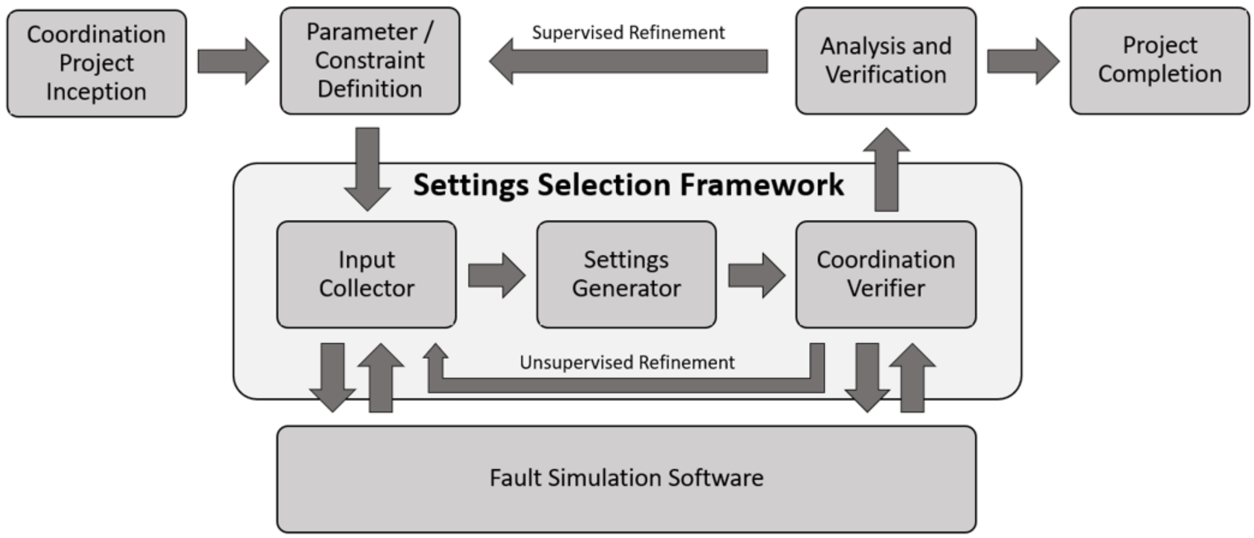


Fig. 1: Coordination Auto Solver Framework

and other parameters of the study are selected in *Parameter / Constraint Definition* and given to the *Settings Selection Framework* as inputs for the overall problem.

The *Input Collector* is an interface between the *Settings Selection Framework* and the fault simulation software. The *Input Collector* uses the algorithms and infrastructure of [6] to find all the relays, source/remote lines, backups, and other data needed for the coordination problem. The *Input Collector* tells the simulation software to run all the faults necessary to coordinate the relays in the study. The fault data, relay data, and backup relay data is given as input to the *Settings Generator*, which we simply call the "solver".

The solver finds the optimal settings for the relays which will coordinate them for the given circumstances. This solution can be given to the *Coordination Verifier*, which uses the fault simulation software to verify that coordination has been achieved. The output of the *Settings Selection Framework* is then given to a protection engineer for *Analysis and Verification*. The engineer can choose to refine the results by changing the problem definition and running the framework again. Once the engineer is satisfied, the project can be completed and the new relay settings accepted.

This paper focuses on improvements made to the *Settings Generator* and the *Input Collector*. These improvements add more features to the solver, allowing it to coordinate more general problems and deal with real-life situations compared to our previous work [3]–[5].

B. Software Setup

For the examples shown in this paper, we used the following software tools:

- ASPEN OneLiner 15.6 [7]
- IBM CPLEX 12.9 [8]
- C++20 with Microsoft Visual Studio 2019, 16.9.3 [9]
- Windows 10
- SARA 3.0.25 [6]

III. Improvements and Examples

A. Heterogeneous Element Coordination

In our previous works [3]–[5], we described our *Settings Generator* as an optimization problem solver, based heavily upon work in [2], [10]. The problem was setup to minimize the total response time to line-end faults while ensuring relay coordination. We used this solver model to coordinate response times in hypothetical and real-world grids, but only for overcurrent relays.

It is common for utilities to require **heterogeneous element coordination** as it is a more accurate representation of real-world relays. They also do so for the increased flexibility in resolving coordition issues. Indeed, they consider their distance elements as the primary operating elements and their overcurrent elements as the secondary elements. In cases where it is infeasible to coordinate overcurrent elements on their own, considering the fastest responding element instead of the overcurrent element only can help to achieve the desired coordination time interval.

Our model for distance response times is inspired by the following: Utilities usually choose their distance elements' reach setting based on their philosophy prior to attempting to coordinate those elements with their neighboring elements. When checking for and resolving coordination issues involving distance elements, it is common to consider the reach setting as fixed and only modify the delay setting. To emulate this process, we choose to model distance elements in our solver as having a fixed reach setting and a tuneable delay setting. As a result, we determine a distance element's response by first comparing the apparent impedance with the element's reach setting, and using either the element's delay setting or "No Operation" as the response depending on whether the apparent impedance is less or greater than the reach setting.

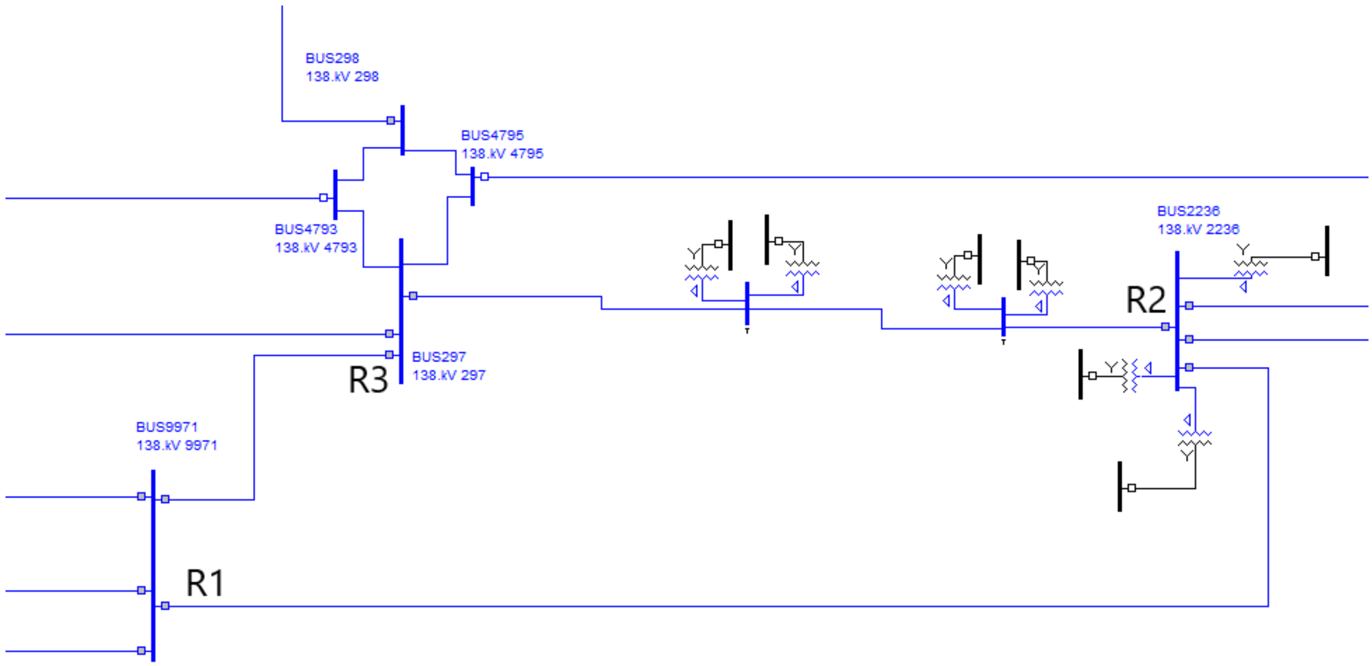


Fig. 2: Perturbation Experiment: 3 Bus System

To create a more accurate problem, we choose to include instantaneous overcurrent elements in our solver. Instantaneous overcurrent relays usually have a near-instantaneous delay and a pickup setting determined by the utility's philosophy. Therefore, we model those elements as invariants: Neither their pickup nor their delay are tuneable. Nevertheless, it is helpful to take the response of instantaneous overcurrent elements into account in a heterogeneous coordination study since they may be the means of resolving issues between an otherwise slow primary relay and a fast backup relay.

1) Perturbation Experiment: As a proof of concept, our first experiment starts with a well-coordinated area, shown in Figure 2. The loop is coordinated to 0.28 seconds. We decrease $R1$'s zone 2 delay to 0.017s, creating a violation under normal conditions: For a line-end fault on the line between buses 9971 and 2236, $R1$'s zone 2 responds in 0.017s and $R3$'s time overcurrent element responds in 0.139s. The original settings are shown in Table Ia. In this paper, we will always give pickup values in secondary amps, reach settings in secondary ohms, and relay response times in seconds. We show that our solver is able to resolve the issue by increasing $R1$'s zone 2 delay.

When running this experiment, we allowed the solver to choose curves from among the standard ANSI U curve family. The settings it suggested are shown in Table Ib; the values in bold correspond to settings that were modified. The solver did indeed increase $R1$'s zone 2 delay, which increases our confidence in its ability to solve real-world problems.

Because the solver aims to minimize the total response time to line-end faults, the $R1$ zone 2 delay was not the only setting that the solver chose to modify. It also modified all the time overcurrent curves and pickups, and the TMS setting for $R1$ and $R2$. This implies that changing these other settings allowed the solver to further optimize the solution.

Relay	Element	Curve	Pickup	TMS	Reach	Delay
R1	Z1				1.8500	0.0000
	Z2				3.4200	0.0170
	TOC	U1	1.0000	3.5600		
	IOC		16.0000			0.1340
R2	Z1				1.7700	0.0000
	Z2				3.3600	0.3330
	TOC	U1	0.6300	1.5800		
	IOC		20.0000			0.1340
R3	Z1				0.6600	0.0000
	Z2				1.8200	0.3830
	TOC	U1	1.2600	0.5000		
	IOC		32.3800			0.1340

(a) Original Settings

Relay	Element	Curve	Pickup	TMS	Reach	Delay
R1	Z1				1.8500	0.0000
	Z2				3.4200	**0.2850**
	TOC	**U4**	**3.2749**	**0.5145**		
	IOC		16.0000			0.1340
R2	Z1				1.7700	0.0000
	Z2				3.3600	**0.5434**
	TOC	**U4**	**2.0199**	**0.5085**		
	IOC		20.0000			0.1340
R3	Z1				0.6600	0.0000
	Z2				1.8200	**0.2850**
	TOC	**U4**	**8.6338**	0.5000		
	IOC		32.3800			0.1340

(b) Suggested Settings

TABLE I: Perturbation Experiment Original and Suggested Settings

2) Infeasibility Experiment: Certain coordination problems which have no solution, called "infeasible" problems, occur

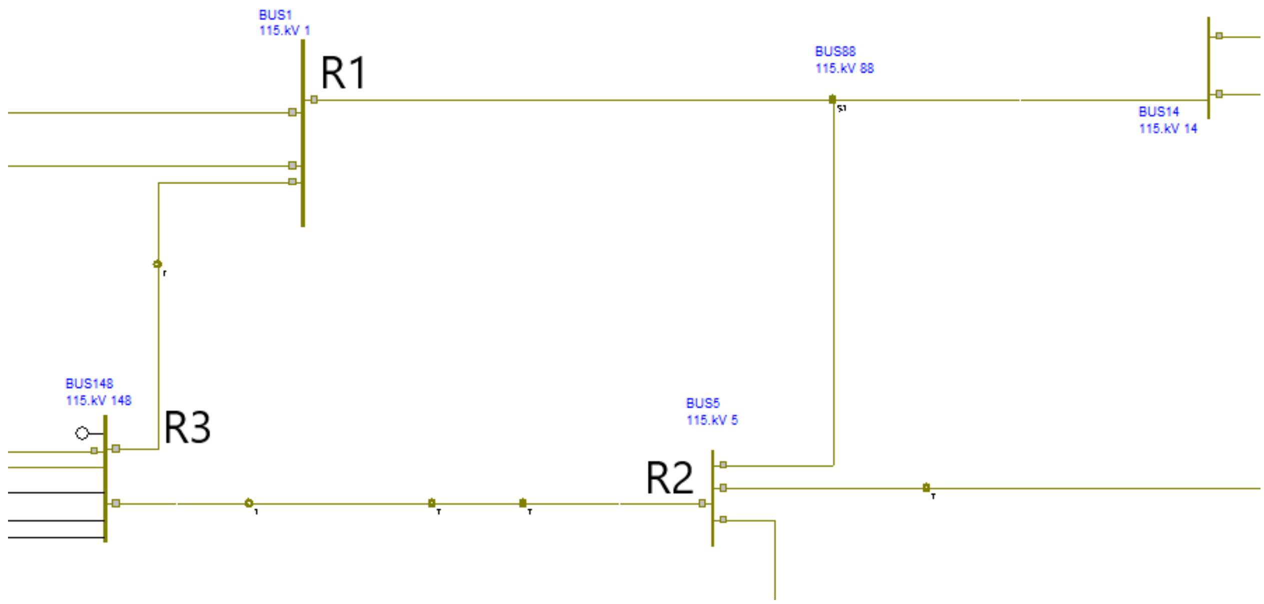


Fig. 3: Infeasibility Experiment: 3 Bus System

when the constraints making up the problem are contradictory, so that no combination of relay settings can satisfy the constraints. This can happen when the problem size becomes large, when some relays are considered "fixed" or for other reasons.

It is well known that coordinating overcurrent elements only is difficult, and this can lead to an infeasible problem as well. One way of solving this is by including distance relays, making it a heterogeneous coordination problem. For example, Figure 3 shows a set of terminals for which the solver cannot achieve coordination to 0.29 seconds under normal conditions with time overcurrent elements only. In fact, $R1$ and one of its backups (not $R3$) have CTI violations that can only be resolved by modifying that backup. By taking the distance elements into account, we expect to resolve those violations by making the distance elements at $R1$ respond faster than its time overcurrent element.

When running this experiment, we again allowed the solver to choose ANSI U curves. The solver found a solution to this problem when we took the distance elements into account; Table II shows the settings it suggested. We conclude from this experiment that using distance and overcurrent elements together can be a solution for areas for which it is difficult to achieve coordination with only one of the element types.

3) Solution Analysis: In Section III-A1 and Section III-A2, we simply presented the settings that the solver suggested without many comments. Here we perform a more detailed analysis of the settings suggested by the solver in Section III-A2, shown in Table II.

First, we look at the distance elements' settings:

- The solver suggested a delay of 0.58 seconds for $R2$'s zone 2, which is twice the desired CTI. Since it is when considering a relay as the backup in a coordination pair that it makes sense to increase the time in which it responds, this could mean that $R2$'s zone 2 element is overreaching $R3$'s zone 1 element. In that case, we would need to adjust the reach settings before using our solver to suggest settings. It would be interesting to see if our solver could suggest a better delay setting for this zone 2 element if it could also tune distance reach settings.
- The other delays are reasonable: The zone 1 delays were set to 0.0 seconds and the other zone 2 delays were made as small as possible while avoiding CTI violations.

Relay	Element	Curve	Pickup	TMS	Reach	Delay
R1	Z1				0.3100	0.0000
	Z2				2.5100	0.2900
	TOC	U4	8.1322	0.5000		
R2	Z1				0.8900	0.0000
	Z2				1.9700	0.5800
	TOC	U4	3.7427	0.5000		
R3	Z1				2.0800	0.0000
	Z2				3.9800	0.2900
	TOC	U4	3.4280	0.5000		

TABLE II: Infeasibility Experiment: Suggested Settings

Next, we look at the time overcurrent elements' settings:

- The curves are all set to $U4$ (extremely inverse) and the time dials to 0.5 (the smallest possible for an ANSI U curve). This is due to the solver's setting to minimize each relay's response time to a line-end fault, where possible. Table III shows the line-end response times for each time overcurrent element when using the settings suggested by the solver. Indeed, their response times are small. Having such fast-responding time overcurrent elements could cause issues with other relays downstream.
- For this experiment, we allowed the solver to change the pickups, TMS settings, and the curves. The pickup values

it suggested are high. However, in real-world scenarios the pickups are generally low for sensitivity purposes. It is possible to restrain the solver to limit the pickup.

Relay Element	Line-End Response Time (seconds)
R1 TOC	0.1342
R2 TOC	0.1231
R3 TOC	0.2364

TABLE III: Solution Analysis: Time Overcurrent Line-End Response Times

Relay	Instantaneous Pickup	Delay
Internal-1	7.7000	0.0000
	7.7000	0.0000
Internal-2	22.8000	0.1000
Internal-3	18.0000	0.1000
Internal-4	31.0000	0.0000
	31.0000	0.0000
Internal-5	N/A	N/A
External-1	15.8000	0.0000
External-2	13.0000	0.0000

TABLE IV: Instantaneous Settings for the Relays in the "No Cost vs Cost" Case Study

B. Customizable Solution Optimization

Coordination problems such as those illustrated above can have multiple solutions. The solver suggests the one that is theoretically optimal, which produces the lowest total response time to line-end faults. This solution may not always satisfy all of an engineer's requirements as there are other real-world considerations they must take into account besides minimizing total response time while avoiding coordination issues.

One such consideration could be the cost of changing the relay's settings in the field. Suppose that a utility decides to coordinate their grid while minimizing the cost to deploy the new settings to the field. If there are multiple options for settings that resolve their coordination issues, they will choose the one that involves the lowest deployment cost.

To capture this scenario, we add a "cost of change" term to the objective function formulated in [2]. First, we assign each relay a "cost" which represents the cost of changing its settings. Then, we make our "cost of change" term such that if a relay's settings are modified, the cost of changing that relay will be added to the objective value to be minimized along with the response time. We refer to w as the overall weight constant for the "cost of change" term in the objective function. The overall weight constant for the response time term is $1 - w$. To test our new objective function, we found a complex substation connected to another utility's grid via two external transmission lines. We assume that it costs more, in time and in resources, to have relays belonging to the other utility modified. This substation is shown in Figure 4.

We ran three coordination studies on the overcurrent relays in this area under normal conditions. All three studies coordinated seven relays (two of which are external relays). We used a shortest valid time interval of 0.33 seconds. All studies tuned the curve, the TOC pickup, and TMS settings of internal relays, and the TOC pickup and TMS of the two external relays. The potential curves were to be chosen from among the ANSI U curves. Most relays also had an instantaneous element, which the solver did not tune. The instantaneous settings are shown in Table IV. We assigned internal relays a cost of change of 1 and external relays a cost of change of 10.

In the first study, we disregarded the cost of changing the relays by setting $w = 0$. This corresponds to the state of our solver before we started our present work: The total response time of the relays to line-end faults is minimized, regardless of how many relays must be changed. The results of this study are given in Table Va. We see that every relay was modified in order to minimize the total response time.

In the second study, we only considered the cost of changing relays by setting $w = 1$. The results are shown in Table Vb. Notice that three relays were not modified, including one of the external ones. The consequence is that the relay trip times, which refer to the time in which each relay responds to a line-end fault, are greater than the ones in Table Va. We expected this given that the tuner no longer attempts to minimize the relay trip times. We use this case only with the goal of comparing it to the other studies, since the CTI values it offers are excessive.

In the third study, we set $w = 0.2$ to represent the case where we want to minimize the relay response times while also minimizing the total cost of modifying the relays if possible. The results are shown in Table Vc. Compared to the results of the second study, we only have one external relay which was not modified, but that should still represent a lower deployment cost than the solution found in the first study.

This experiment shows that we can alter our auto-solver's objective function to better represent the various real-life considerations that an engineer may have when coordinating an area of the power grid. This means they can have greater flexibility and control over the final settings in coordination studies. Additionally, it represents one of the first solutions to an important yet constraining assumption in our previous work: the assumption that every relay in the problem can be modified.

C. Contingencies and Constraint Relaxation

In Section III-A2, we considered an infeasible problem where overcurrent relays alone would not coordinate. We ran another problem including distance relays, which did coordinate. Even without running another problem, an engineer may ignore or "suppress" a violation based on their expertise and knowledge of the grid. The engineer may already know that a particular CTI issue is addressed when using distance relays.

Another option to solve infeasible problems is for the protection engineer to allow certain violations in order to make the problem feasible. We call this option "Constraint Relaxation." We demonstrate this technique in our solver by reducing the required CTI, for instance from 0.33 seconds to 0.25 seconds. Then we add "cost" terms like we did in Section III-B, but instead of "cost of change" terms, we add

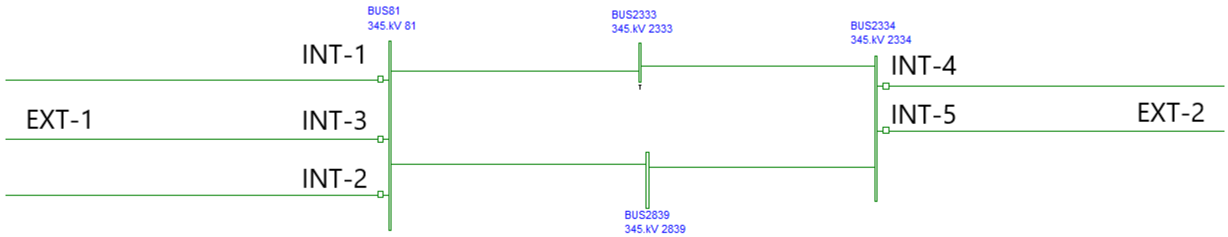


Fig. 4: A Complex Substation With Two External Transmission Lines

Relay	Original Curve	New Curve	Original Pickup	New Pickup	Original TMS	New TMS	Relay Trip Time (sec)
Internal-1	U4	U4	0.5500	0.5500	10.5000	0.5000	0.0325
	U4	U4	0.5500	0.5500	12.0000	0.5000	0.0325
Internal-2	U1	U4	1.8000	1.0000	3.2000	0.5000	0.0208
Internal-3	U3	U4	1.5000	1.0000	3.3000	0.5000	0.0250
Internal-4	U1	U4	0.9000	0.9000	3.3000	0.5000	0.0147
	U1	U4	0.9000	0.9000	3.3000	0.5000	0.0147
Internal-5	U2	U4	0.5000	0.5000	3.6000	0.5000	0.0216
External-1	U3	U3	0.5000	2.8254	6.4000	0.5000	0.1391
External-2	U3	U3	0.5000	3.5496	3.5000	0.5000	0.2772

(a) “No cost” results with $w = 0$. All relays have changed pickup and/or TMS values in order to minimize trip times.

Relay	Original Curve	New Curve	Original Pickup	New Pickup	Original TMS	New TMS	Relay Trip Time (sec)
Internal-1	U4	U4	0.5500	0.5500	10.5000	0.5000	0.0325
	U4	U4	0.5500	0.5500	12.0000	12.0000	0.7803
Internal-2	U1	U5	1.8000	1.0000	3.2000	0.5000	0.0418
Internal-3	U3	U5	1.5000	2.5657	3.3000	0.5000	0.0609
Internal-4	U1	U1	0.9000	0.9000	3.3000	0.5000	0.0827
	U1	U1	0.9000	0.9000	3.3000	3.3000	0.5458
Internal-5	U2	U5	0.5000	4.4243	3.6000	0.5000	0.1395
External-1	U3	U3	0.5000	3.5859	6.4000	0.5000	0.1989
External-2	U3	U3	0.5000	0.5000	3.5000	3.5000	0.3656

(b) “Cost only” results with $w = 1$. Three relays have kept their original values in order to minimize the cost.

Relay	Original Curve	New Curve	Original Pickup	New Pickup	Original TMS	New TMS	Relay Trip Time (sec)
Internal-1	U4	U4	0.5500	0.5500	10.5000	0.5000	0.0325
	U4	U4	0.5500	1.0349	12.0000	0.5000	0.0855
Internal-2	U1	U4	1.8000	1.0000	3.2000	0.5000	0.0208
Internal-3	U3	U4	1.5000	1.0000	3.3000	0.5000	0.0250
Internal-4	U1	U4	0.9000	0.9000	3.3000	0.5000	0.0147
	U1	U4	0.9000	0.9000	3.3000	0.5000	0.0147
Internal-5	U2	U4	0.5000	0.5000	3.6000	0.5000	0.0216
External-1	U3	U3	0.5000	2.8254	6.4000	0.5000	0.1391
External-2	**U3**	**U3**	**0.5000**	**0.5000**	**3.5000**	**3.5000**	**0.3656**

(c) Mixed results with $w = 0.2$. All relays have changed except for one external relay **which is shown in bold**.

TABLE V: “No Cost vs Cost” Case Study Results

“cost of violation” terms. Thus, each relay pair which violates the original CTI of 0.33 adds a “cost” which is proportional to the size of the violation.

We use different costs or “weights” for different types of CTI violations. Adjusting these weights tells the solver to consider some violations as more costly than others. In our experiment, we use weights based on whether or not the violation involved a contingency. Since we know that the grid should coordinate under normal conditions whenever possible, we use a large weight for those types of violations. We use a smaller weight for violations under N-1 contingencies. With these weights, the solver will tend toward solutions which violate N-1 constraints, only violating normal conditions constraints if no other solution can be found.

1) Constraint Relaxation Experiment: In the problem shown in Figure 5, the solver was unable to find settings that would allow the overcurrent relays to be coordinated to 0.33 seconds under N-1 contingencies. We use the constraint relaxation feature to find a solution. For this experimental situation, we choose to relax the required CTI to 0.17 seconds even though it is below the permissible limit. We keep the preferred CTI at 0.33 seconds. If a constraint must be violated, the resulting CTI must be at least 0.17 seconds, and the solver will prefer violating a N-1 constraint over a N-0 constraint. We allow the solver to choose from among the ANSI U curves if necessary. The contingencies considered in this experiment

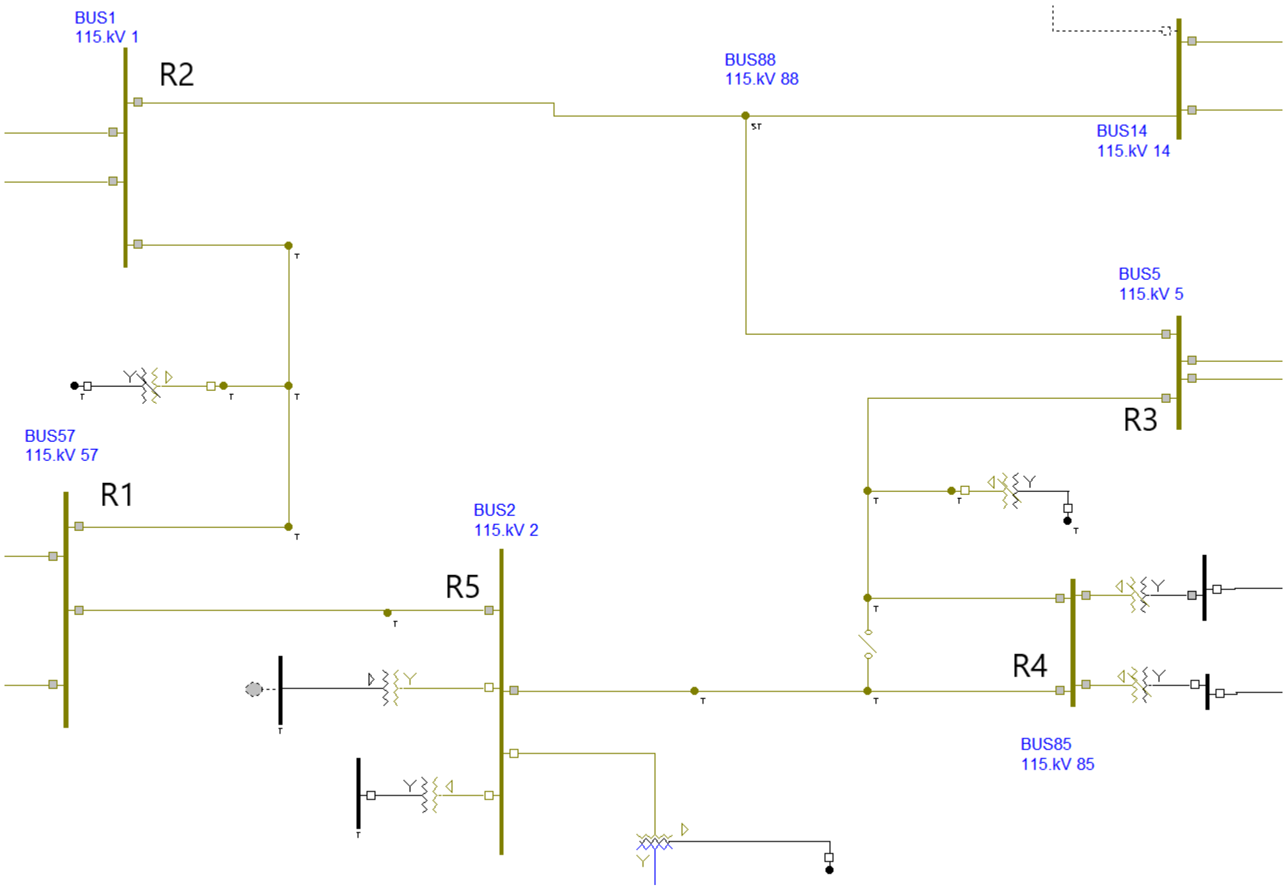


Fig. 5: Constraint Relaxation: 5 Bus System

are remote and transformer contingencies.

Given these inputs, the solver is able to find a solution. In Table VI, we show for each relay the number of CTI constraints which involve it as the primary relay, the number of those constraints which were violated, and the lowest CTI value associated with the constraints. We see that the lowest CTI values are still at least 0.17 seconds, as we directed the solver.

Relay	Total CTI Constraints	Violated Constraints	Lowest CTI
R1	48	7	0.1733
R2	144	4	0.2548
R3	144	48	0.1700
R4	60	36	0.1700
R5	60	6	0.1700

TABLE VI: Constraint Relaxation: Constraints and Violated Constraints

The results show us that using constraint relaxation can facilitate:

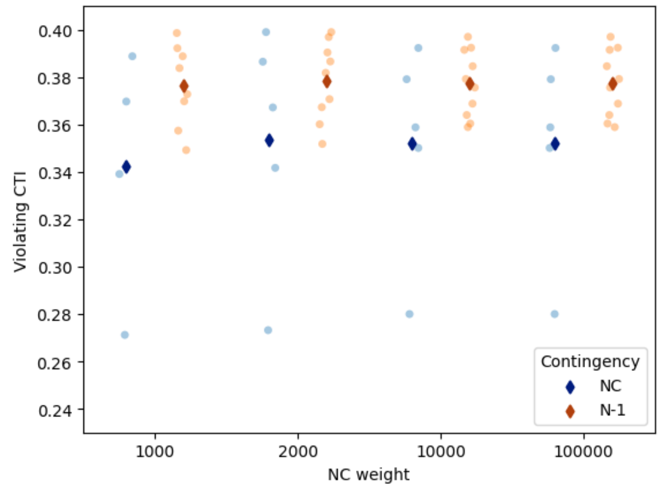
- **Identifying a difficult area**: Every CTI constraint involving $R3$ and one of its backups is violated, and all the violations involving $R4$ also involve $R3$ as the backup relay. This could be an indication that $R3$ and its problematic backup form an area that is difficult to coordinate.
- **Suggesting a potential solution**: Every relay except $R2$ has a CTI violation under normal conditions for a close-in fault. If we were to consider the response of distance or instantaneous overcurrent elements, these violations may be resolved.

Constraint relaxation is a helpful tool, either as an intermediate step to investigate what may be causing infeasibility, or as a final step to select relay settings.

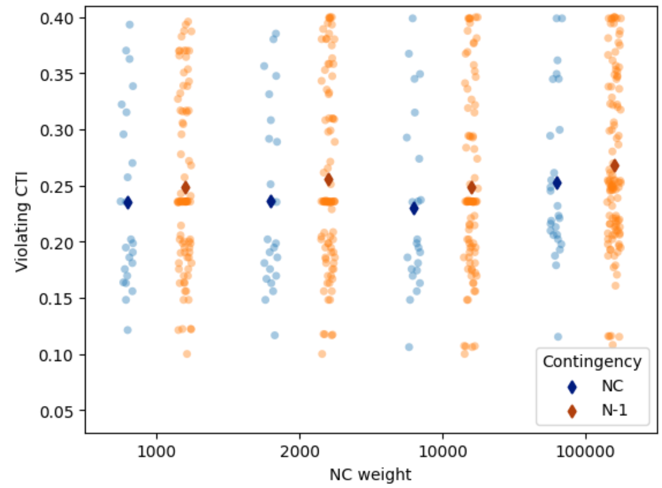
2) The weight of normal conditions: We investigated the influence of the value chosen to be the weight of violating N-0 constraints, W_{nc}. In Figure 6, we plot the CTI values obtained from the solver based on W_{nc} after running the solver multiple times on a 3-terminal system and a 12-terminal system containing only overcurrent relays, and with the weight of violating N-1 constraints remaining unchanged.

As W_{nc} increases, we observe that the average N-0 CTI value increases. This is especially visible in Figure 6b. The average N-1 CTI value also increases. Remarkably, in both figures, there is a group of violations that does not draw nearer to the average nor to the top. These must be violations that cannot be easily resolved and require special attention. They may be due to a specific type of fault which is particularly complicated to protect given the system's topology, impedances, etc.

According to the plots, the exact penalty chosen for violating N-0 CTI constraints does not matter so much as long



(a) 3 Relay Case, N-0 and N-1



(b) 12 Relay Case, N-0 and N-1

Fig. 6: Distribution of CTIs with Increased Penalties for N-0 Violations.

is it is significantly larger than the penalty for violating N-1 constraints.

IV. Conclusion

In this paper, we demonstrated that autotuning-assisted coordination studies are a viable advancement for system protection. Our coordination problem solver can handle a variety of real-world situations. We showed that our solver is now heterogenous – able to coordinate overcurrent and distance relays together in a single problem. We showed that including distance relays can change a coordination problem from infeasible to solvable.

We also demonstrated that our solver can be customizable to additional real-world concerns. The solver can be setup to consider the cost of deploying new settings to the field. Also, the solver can be set to discern between different types of coordination violations. This can then be used to allow certain types of unlikely violations in order to make a problem feasible.

We see many opportunities to make our solver practical for real-world studies. We plan to give the solver more control over which settings are fixed or variable to support different relay setting philosophies. This will ensure that the proper relay settings are tuned for coordination, and that we avoid unrealistic solutions where the response times are too short. We also plan to include every relay's downstream relays in the solver. This will further provide more realistic solutions when distance delays are tuned.

Our other major goal for the solver is to have it scale. We plan to do larger studies, which requires that the solver be fast. It also requires that the solver provides more detail about how to work with infeasible problems. In this way, we hope that we can help electric grids be better coordinated, and thus better protected, than ever before.

References

[1] "PRC-027-1: Coordination of Protection Systems for Performance During Faults." (2015, November). Retrieved February 22, 2018, from North American Electric Reliability Corporation.

[2] Y. Damchi, M. Dolatabadi, H. Mashhadi, and J. Sadeh, "MILP approach for optimal coordination of directional overcurrent relays in interconnected power systems." Electric Power Systems Research, 2018. 158. 267-274.

[3] N. Thomas, L. Hankins and J. Perez, "Auto-tuned Solution to Wide Area Coordination Issues of Distance and Directional Time Overcurrent Relay Settings." 2021 74th Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 2021.

[4] N. Thomas, L. Hankins and J. Perez, "Simplifying Wide Area Coordination of Directional Time Overcurrent Relays Using Automatic Settings Selection." 2019 72nd Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 2019.

[5] N. Thomas, J. Perez, L. Hankins, and H. Tribur. "An innovative and automated solution for NERC PRC-027-1 compliance." 2018 71st Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2018.

[6] M.J. Boecker, G.T. Corpuz, J. Perez, and L. Hankins. "Novel approach to relay setting development." 2017 70th Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2017. https://synchrogrid.com/sara-software/.

[7] ASPEN Oneliner. https://www.aspeninc.com/web/software/oneliner.

[8] "IBM ILOG CPLEX 12.7 User's Manual". IBM ILOG CPLEX Division, Incline Village, NV, 2017.

[9] "Microsoft Visual Studio". https://visualstudio.microsoft.com/.

[10] A.J. Urdaneta, R. Nadira, and L.G. Perez Jimenez. "Optimal coordination of directional overcurrent relays in interconnected power systems." IEEE Transactions on Power Delivery 3.3 (1988): 903-911.

V. Author Biographies

Joe Perez has 20 years of experience in the protection industry. He received his B.S. degree in Electrical Engineering from Texas A&M University in 2003. He is founder and CEO of SynchroGrid, which celebrated its 10-year anniversary in 2022 and has offices in College Station, Texas, and Denver, Colorado. He is also leading a new era in automation of relay setting development through new cutting-edge software SARA to streamline and automate system protection processes. Joe has authored numerous relay application notes and technical papers for WPRC, Texas A&M and Georgia Tech relay conferences. He is a registered professional engineer in Texas and Louisiana and is also a member of PSRC, IEEE, and PES.

Abram Krislock graduated from Texas A&M University with a Ph.D. in Physics in 2011. He has experience in both scientific computing research and professional software development. Abram spent 10 years working on Monte Carlo simulation and data analysis tools, making predictions for High Energy Physics experiments, leading to 11 publications in physics journals. He has 6 years of experience making web and desktop applications for both business and scientific purposes.

Luke Hankins graduated from Texas A&M University with a B.S. degree in Electrical Engineering in 2015. He is a registered professional engineer in the state of Texas and works for SynchroGrid in relay setting automation. He writes software applications in various programming languages, focusing on simplifying relay setting development. Luke also performs relay settings verification and mis-operation analysis.

Lucille Lamb graduated from Texas A&M University with a B.S degree in Applied Mathematics with a Computational Science Emphasis and a minor in Computer Science in 2020. Her unique background in mathematics and computer science enables her to develop effective and flexible software for the power industry. She specializes in providing automation to Wide Area Coordination studies and in building a solid framework for modeling the power grid.