Doctoral Thesis in Electrical Engineering

Accelerated ADMM Variants for Distributed Optimization

Algorithms for Dynamic and Large Networks

JEANNIE HE

KTH ROYAL INSTITUTE OF TECHNOLOGY

Accelerated ADMM Variants for Distributed Optimization

Algorithms for Dynamic and Large Networks

JEANNIE HE

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology,
is submitted for public defence for the Degree of Doctor of Philosophy on Friday the 12th of June 2026,
at 1:30 p.m. in Q2, Malvinas väg 10, Stockholm.

Doctoral Thesis in Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden 2026

TRITA-EECS-AVL-2026:67
ISBN 978-91-8106-660-9

Printed by: Universitetsservice US-AB, Sweden 2026

Abstract

This doctoral thesis presents a comprehensive summary of research efforts with the aim of advancing the state-of-the-art in decentralized optimization. As modern distributed systems grow in scale and complexity, traditional optimization methods face significant bottlenecks. This work, presented as a compilation of five papers, specifically targets the Alternating Direction Method of Multipliers (ADMM). The central objective is to re-engineer ADMM to overcome the dual challenges of convergence latency and communication inefficiencies in peer-to-peer networks.

The first part of the thesis focuses on decentralization, convergence speed, and computational cost. While centralized ADMM algorithms struggle with scalability and bottlenecks, existing decentralized schemes rely on either excessive computations and messaging or completely sequential operations, causing a prolonged time required to reach convergence. To tackle these problems, we introduce two fast-converging decentralized ADMM algorithms. Our theoretical analysis confirms that our algorithms retain the classical convergence properties of centralized ADMM while maintaining a low per-node complexity of $O(1)$. Numerical simulations further demonstrate that our algorithms converge significantly faster than state-of-the-art decentralized implementations, providing clear conditions under which they outperform traditional benchmarks.

The second part of the thesis addresses the straggler problem, which arises from the conventional ADMM requirement for global synchronization, where faster nodes are forced to remain idle until the slowest nodes complete their local updates, impeding the progress of the entire network. Here, we introduce three algorithms to allow the system to remain productive even under single-point-of-failure scenarios or extreme hardware variance. The first algorithm achieves straggler-resilience by allowing the nodes to proceed to the next iteration even when one or more nodes have not provided an update for one or more iterations. The second algorithm is a decentralized version of the first algorithm. The second algorithm enforces fast convergence as well as robustness against stragglers and single points of failure through decentralized, asynchronous, and concurrent operations. The third algorithm extends the second algorithm by enforcing robustness against uncertainties with the help of a time-tracking scheme. Through theoretical analyses, we establish the convergence properties of our algorithms and show that our decentralized algorithms achieve a computational complexity of $O(1)$ for each worker node, whereas our centralized algorithm achieves a computational complexity of $O(N)$ for the central node and $O(1)$ for each of the remaining nodes. Through numerical simulations with various settings, we show that our algorithms have converged significantly faster than several state-of-the-art ADMM algorithms with well-established convergence properties.

The final part of the thesis extends these optimizations to highly volatile systems characterized by message dropouts and dynamic topologies. Here, we introduce two algorithms to adapt ADMM to dynamic systems, where new nodes may be added amidst the process at the same time as the system may encounter issues with stragglers and message dropouts. The algorithms achieve fast convergence and flexibility by allowing nodes to choose a step size that is best suited for their own system and by allowing the nodes to move on to the next iteration even when not all nodes have made an update for the current iteration. More importantly, these algorithms incorporate a contribution tracking mechanism to ensure consistency despite message loss. Furthermore, these algorithms enforce robustness against uncertainties by removing the need to predefine the waiting time or the minimum number of updates before moving to the next iteration. Here, an approximation mechanism is also introduced to give stragglers more time to compute their variables while the faster nodes move on to the next iteration.

To summarize, this thesis provides accelerated algorithms for fast convergence in distributed optimization, with solutions tailored for both large-scale and dynamic networks.

Keywords
ADMM, optimization, decentralization, asynchronous operations, dynamic networks, communication, algorithms, linear regression, distributed learning

Sammanfattning

Denna doktorsavhandling presenterar en omfattande sammanfattning av forskningsinsatser som syftar till att främja den senaste tekniken inom decentraliserad optimering. I takt med att moderna distribuerade system växer i skala och komplexitet möter traditionella optimeringsmetoder betydande flaskhalsar. Detta arbete, som presenteras som en sammanställning av fem artiklar, riktar sig specifikt mot Alternating Direction Method of Multipliers (ADMM). Det centrala målet är att omkonstruera ADMM för att övervinna de dubbla utmaningarna med konvergenslatens och kommunikationsineffektivitet i peer-to-peer-nätverk.

Den första delen av avhandlingen fokuserar på decentralisering, konvergenshastighet och beräkningskostnad. Medan centraliserade ADMM-algoritmer lider av skalbarhetsproblem och flaskhalsar, förlitar sig existerande decentraliserade metoder antingen på omfattande beräkningar och kommunikation eller helt sekventiella operationer, vilket leder till långsammare konvergens. För att hantera dessa problem introducerar vi två snabbt konvergerande ADMM-algoritmer. Vår teoretiska analys bekräftar att dessa algoritmer behåller de klassiska konvergensegenskaperna hos centraliserad ADMM samtidigt som de bibehåller låg komplexitet på $O(1)$ per nod. Numeriska simuleringar visar vidare att våra algoritmer konvergerar betydligt snabbare än befintliga decentraliserade implementeringar.

Den andra delen av avhandlingen behandlar det klassiska problemet med eftersläntrare hos ADMM-algoritmer, där snabbare noder tvingas vänta på de långsammaste noderna på grund av det konventionella kravet på synkronisering i ADMM. Här introducerar vi tre algoritmer för att låta systemet förbli produktivt även under scenarier med en enda felpunkt eller extrem hårdvaruvarians. Den första algoritmen uppnår tolerans mot långsamma noder (eftersläntrare) genom att låta noderna fortsätta till nästa iteration även när en eller flera noder inte har tillhandahållit en uppdatering under en eller flera iterationer. Den andra algoritmen är en decentraliserad variant av den första algoritmen. Den tredje algoritmen utökar den andra algoritmen med ett tidsspårningsschema. Genom teoretiska analyser fastställer vi konvergensegenskaperna hos våra algoritmer och visar att våra decentraliserade algoritmer uppnår en beräkningskomplexitet på $O(1)$ för varje arbetsnod, medan vår centraliserade algoritm uppnår en beräkningskomplexitet på $O(N)$ för centralnoden och $O(1)$ per nod för de övriga noderna. Genom numeriska simuleringar med olika inställningar visar vi att våra algoritmer konvergerar betydligt snabbare än flera ADMM-algoritmer med väletablerade konvergensegenskaper.

Den sista delen av avhandlingen utökar dessa optimeringar till volatila system som kännetecknas av meddelandeförlust och dynamiska topologier. Här introducerar vi ytterligare två ADMM-algoritmer för dynamiska system, där nya noder kan ansluta sig mitt i processen samtidigt som systemet kan stöta på problem med eftersläntrare och meddelandeförlust. Dessa algoritmer är skapade för att uppnå snabb konvergens och flexibilitet genom att låta noder välja en stegstorlek som passar bäst för det egna systemet och genom att låta noderna gå vidare till nästa iteration även om inte alla noder har gjort en uppdatering för den aktuella iterationen. Ännu viktigare är att dessa algoritmer har ett identifieringssystem för att säkerställa konvergens trots meddelandeförlust. Algoritmerna upprätthåller även robusthet mot osäkerheter genom att ta bort behovet av att fördefiniera väntetiden eller det minsta antalet uppdateringar innan man går vidare till nästa iteration, genom ett tidsspårningsschema inspirerat av resultaten från den andra delen. Här har även en approximationsmekanism introducerats för att ge mer tid åt eftersläntrarna att utföra sina beräkningar medan de snabbare noderna går vidare till nästa iteration.

Sammanfattningsvis tillhandahåller denna avhandling accelererade algoritmer för snabb konvergens vid distribuerad optimering, med lösningar anpassade för såväl storskaliga som dynamiska nätverk.

Nyckelord

ADMM, optimering, decentralisering, asynkrona operationer, dynamiska nätverk, kommunikation, algoritmer, linjär regression, distribuerad inlärning

Acknowledgment

I would like to thank my partner, Joris Verhagen, for his unwavering support, for his sharp insights, and for standing by me throughout this journey. I would also like to express my gratitude to my supervisor, Ming Xiao, for his invaluable guidance and support throughout this research. Additionally, I would like to thank my co-supervisor, Mikael Skoglund, for his support and useful comments. Finally, I would like to thank my family for their love and support.

Sincerely,

Jeannie He
Stockholm, May 25, 2026

List of included papers

Contributions

- **Paper A** **Fast-Converging Decentralized ADMM for Consensus Optimization**, Jeannie He, Ming Xiao, and Mikael Skoglund. In Proc. IEEE Conf. on Artif. Intel. (CAI), 2024
- **Paper B** **Fast-converging decentralized alternating direction method of multipliers for consensus optimization**, Jeannie He, Ming Xiao, and Mikael Skoglund. In EURASIP Journal on Advances in Signal Processing, 2025
- **Paper C** **Straggler-Resilient Asynchronous Decentralized ADMM for Consensus Optimization**, Jeannie He, Ming Xiao, and Mikael Skoglund. In Proc. 58th Annu. Conf. Inf. Sci. Syst. (CISS), 2024
- **Paper D** **Straggler-resilient asynchronous ADMM for distributed consensus optimization**, Jeannie He, Ming Xiao, Mikael Skoglund, and Harold Vincent Poor. In IEEE Transactions on Signal Processing, 2025
- **Paper E** **Robust Asynchronous Decentralized ADMM for Distributed Optimization in Dynamic Networks**, Jeannie He, Ming Xiao, Mikael Skoglund, and Harold Vincent Poor.

Contents

Thesis overview

1 Introduction

In the contemporary landscape of high-dimensional data analysis and the Internet of Things, distributed systems have experienced exponential growth in both scale and topological complexity. Modern networked infrastructures—ranging from cyber-physical systems like smart grids and autonomous vehicle swarms to heterogeneous sensor arrays—generate massive, decentralized datasets that necessitate collective, large-scale processing. The efficacy of modern machine learning paradigms is fundamentally predicated on the availability of these extensive datasets and the development of distributed training algorithms capable of partitioning the computational load across multiple subunits. Such frameworks efficiently exploit the parallelization potential of multi-node, high-performance computing architectures [6].

Central to the operation of these systems is the paradigm of distributed optimization, where agents collaboratively solve a problem by minimizing a global objective function while maintaining local data sovereignty. In this domain, the Alternating Direction Method of Multipliers (ADMM) has emerged as a widely adopted framework commended for its simplicity, well-established convergence properties, and applicability to various optimization problems. Originally, ADMM is designed for centralized architectures [7]. However, as node density and data volumes increase, the centralized paradigm encounters a bottleneck of scale, where a single coordinating entity becomes a critical point of failure and a primary source of communication congestion. Consequently, several decentralized implementations of ADMM have been proposed, e.g. [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. Despite these efforts, existing decentralized ADMM algorithms suffer from several limitations. These include:

- High computational and communication costs: In a majority of decentralized implementations of ADMM, e.g. [8, 9, 19, 20], synchronization responsibility is duplicated across all nodes, leading to redundant computations and high communication overhead.

- Slow convergence: Methods relying on sequential update orders [21, 22] face a total convergence time that increases linearly with the number of worker nodes.
- Hyperparameter sensitivity: Algorithms relying on waiting-time hyperparameters [23] or fixed iteration gaps [24, 25] are unsuitable for dynamic environments where node performance is unpredictable.
- Single-point-of-failures: As will be explained in Section 1.2, a majority of existing implementations of ADMM have the drawback of halting entirely if a single node becomes unavailable.
- Network instability: As existing research has focused on environments where a message transaction between two nodes can never be lost, there is a research gap on how to implement ADMM in real-world environments characterized by information loss.

1.1 Classical Alternating Direction Method of Multipliers

To enable distributed machine learning across disparate datasets, the Alternating Direction Method of Multipliers (ADMM) has emerged as a robust and widely adopted framework [7, 26]. ADMM is typically used to solve standard consensus optimization problems, which allow multiple entities to contribute to a shared model without centralizing raw data—a process often hindered by privacy, security, or high transfer costs [6, 27].

The global optimization task is defined as follows:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{N}} f_i(x_i) \\ \text{s.t.} \quad & x_i - z = 0, \quad \forall i \in \mathcal{N}, \end{aligned} \tag{1.1.1}$$

where $f_i(x_i)$ is the local loss function at node i, $x_i \in \mathbb{R}^n$ is the local primal variable, and $z \in \mathbb{R}^n$ is the global consensus variable [7].

To solve this constrained problem, ADMM utilizes the Augmented Lagrangian function, $\mathcal{L}_\rho$, which combines the standard Lagrangian with a quadratic penalty term to improve convergence robustness [7]:

$$\mathcal{L}_\rho(x, y, z) = \sum_{i \in \mathcal{N}} \left(f_i(x_i) + \langle y_i, x_i - z \rangle + \frac{\rho}{2} \|x_i - z\|^2 \right) \tag{1.1.2}$$

where $y_i \in \mathbb{R}^n$ are the local dual variables (Lagrange multipliers) and $\rho > 0$ is the penalty parameter[7].

By minimizing this Augmented Lagrangian in an alternating fashion, the problem is solved via the following iterative update steps:

$$\begin{cases} x_i^{k+1} := \arg\min_x f_i(x) + \langle y_i^k, x - z^k \rangle + \frac{\rho}{2}\|x - z^k\|^2, \\ y_i^{k+1} := y_i^k + \rho(x_i^{k+1} - z^k), \\ z^{k+1} := \frac{1}{N}\sum_{i=1}^N \left(x_i^{k+1} + \frac{y_i^{k+1}}{\rho} \right). \end{cases} \tag{1.1.3}$$

Conventionally, these updates are implemented in a centralized manner with a Gather and Broadcast phase. In the Gather phase, worker nodes compute their local variables (x_i^k, y_i^k) and transmit them to a central node. In the Broadcast phase, the central node aggregates these into the global variable z^k and broadcasts it back to the network. This process, hereafter referred to as classical centralized ADMM (CC-ADMM), repeats until a stopping criterion is met [7].

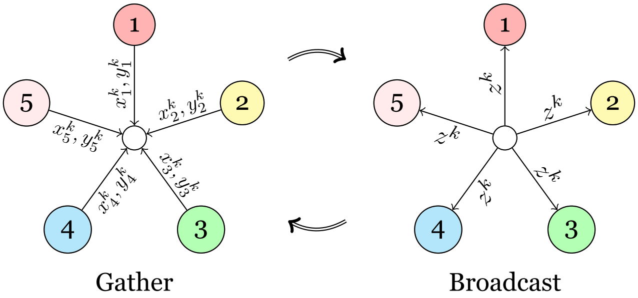

Figure 1.1.1: An illustration of CC-ADMM iterative process with two phases: Gather and Broadcast. The circles with numbers are the worker nodes $i = 1, \dots N$. The circle at the center is the central node. The arrows with variables show the message flow amongst the nodes.

1.2 Existing Implementations of ADMM and Their Limitations

1.2.1 Centralized ADMM

In several studies (e.g., [28, 29, 30, 31, 32, 33, 34, 35, 36]), ADMM has been implemented in a centralized manner where a central coordinator manages the synchronization of primal and dual variables. While effective for small-scale deployments, these implementations are frequently criticized for their high dependency on the capacity, physical location [37], and constant availability of the central node [38]. In these architectures, the central node becomes a performance bottleneck as the computational and communication workload scales linearly with the network size, making the system prone to total failure if the central node becomes unavailable.

1.2.2 Decentralized ADMM

To mitigate the risks of centralization, researchers proposed various decentralized ADMM implementations. For instance, several authors (e.g., [8, 9, 10, 11, 26, 19, 12, 13, 14, 15, 20, 39, 40, 41, 42, 16, 17, 18]) developed algorithms where worker nodes independently broadcast and aggregate outputs to achieve synchronization. While this removes the reliance on a single central node, it introduces the issue of "replicated work," where the responsibility for aggregation is duplicated across every node rather than shared. This redundancy, coupled with the requirement for fully connected networks, leads to high communication costs and limits the algorithm's scalability.

Alternative structures, such as the hybrid approach in [43], utilize fusion centers for aggregation. However, this setup requires complex topology-specific matrices and still suffers from replicated communications and computations as both nodes and centers redundantly manage output synchronization.

1.2.3 Sequential ADMM

To address the inefficiency of replicated work, algorithms such as Random-Walk ADMM (RW-ADMM) [21] and Incremental ADMM (I-ADMM) [22] were introduced, allowing nodes to take turns updating variables in a sequential manner. A similar approach in [44] lets agents update variables based on outputs from edge computing nodes. While these methods resolve redundant work, they are susceptible to slow convergence because only one worker is active at a time.

Efforts to parallelize these processes, such as Parallel Random-Walk ADMM (PW-ADMM) [45], utilize multiple updating threads but often lack synchronization mechanisms, leading to precision loss or a continued reliance on sequential logic. To overcome this, [1] proposed an algorithm that enables parallel local computation while sharing the workload of aggregation across nodes.

1.2.4 Asynchronous ADMM

Aside from the aforementioned problems, all algorithms mentioned above are also prone to the straggler problem, where the global convergence speed is dictated by the slowest node in the network. To mitigate the problem, authors [23] proposed Asynchronous Proximal Gradient ADMM (Async-PGADMM), where a central node periodically updates the global variable based on received updates. However, this approach remains centralized and is highly sensitive to the waiting time hyperparameter, causing poorly tuned intervals to lead to redundant computations or severely delayed convergence.

While other asynchronous variants (e.g., [24, 25, 46, 47, 48]) exist, the convergence property of their algorithms heavily relies on letting the nodes move on to the next iteration only when the iteration gap between the current iteration and the last iteration at which the nodes have made an update is sufficiently small. To ensure that the iteration gap is sufficiently small, a hyperparameter known as the maximum iteration gap τ is therefore used in these algorithms. Consequently, these algorithms still face critical drawbacks:

- Single-Point-of-Failure: By setting a limit on the maximum iteration gap, systems implementing algorithms from [24, 25, 46, 47, 48] are highly sensitive to single-point-of-failures, where the entire process would stop when a single node becomes unavailable for more than τ iterations.
- Computational Complexity: In [24, 25, 46], the per-node workload grows quadratically with network size.
- Coordination Bottlenecks: Implementations in [47, 48] still depend on the capacity of a central node, which scales poorly with network growth.

1.3 Research Questions

In this thesis, we aim to solve the problems with existing implementations of ADMM by answering the following research questions:

1. Fast-Convergence and Decentralization: Is it possible to, and how can we enable fast-convergence and decentralization without causing excessive replicated computational workload and communication costs?
2. Straggler-Resilience in Any Topology Is it possible to, and how can we enable straggler-resilience in any topology without causing excessive replicated computational workload and communication costs?
3. Robustness Against Message-Dropouts: Is it possible to, and how do we enforce fast-convergence despite message-dropouts?

1.4 Statement of Own Contributions

For all papers, Jeannie H. has designed the algorithms, written the main content, and executed the experiments. The other authors have contributed to the improvements of the algorithms and the manuscript.

1.5 Use of Generative AI

Generative AI was used for grammar check.

2 Conclusions and future work

This doctoral thesis has addressed the critical limitations of decentralized optimization, specifically focusing on the ADMM framework. As distributed systems scale toward massive, heterogeneous architectures - driven by the rise of Big Data and distributed machine learning - there is an increasing demand for algorithms that are both fast and robust. To meet this demand, this work introduces various ADMM variants suitable for various scenarios.

2.1 Key Contributions

This thesis provides a systematic advancement of the ADMM framework, transitioning it from a rigid, synchronous, and often centralized architecture to a fluid, asynchronous, and fully decentralized one. The following sections detail the primary scientific contributions and the empirical results that validate their performance.

2.1.1 Contribution I: Scalable and Fast-Converging Decentralized ADMM

The first contribution addresses the performance gap between centralized and decentralized optimization. While centralized systems are fast, they suffer from bandwidth bottlenecks; conversely, decentralized systems often trade speed for reduced communication overhead.

- Innovation: We introduced a fast-converging decentralized ADMM algorithm, and a variant with rotational order [1, 2]. In these algorithms, we let worker nodes collectively compute the global variable while other nodes continue computing their local variables. This way, we enable fast convergence and decentralization without introducing the redundant replicated work seen in prior decentralized models.

- Key finding: Our analysis confirms that our proposed algorithms retain the linear convergence properties of centralized ADMM while ensuring that each node only maintains a local computational complexity of $O(1)$. Numerical simulations demonstrate that our algorithms converge significantly faster than state-of-the-art decentralized implementations, effectively matching the speed of a centralized hub while operating in a peer-to-peer topology.

2.1.2 Contribution II: Straggler-Resilience and Asynchronicity

To solve the straggler problem — where slower nodes halt the entire network - we developed a suite of asynchronous algorithms with the following contributions.

- Innovation: In [3, 4], we introduced a straggler-resilient asynchronous decentralized ADMM algorithm [3, 4], and its variant with a time-tracking scheme. This is achieved by allowing nodes to collectively compute the global variable in a first-come-first-served order, as well as by allowing faster nodes to proceed to subsequent iterations even when slower nodes have not yet provided updates.
- Key finding: Our analysis confirms that our algorithms retain the linear convergence properties of centralized ADMM while ensuring that each node in the decentralized algorithms only has a local computational complexity of $O(1)$. Our experiments show that our algorithms remain productive in single-point-of-failure scenarios. In networks with high hardware variance, our asynchronous methods reached convergence several times faster than the algorithms from [47, 48, 23, 7].
- Practical application: As noted in [4], the variant with a time-tracking scheme is designed for dynamic environments, where it is difficult to predefine an appropriate minimum number of active nodes before proceeding to the subsequent iteration. Meanwhile, the other variant may be prioritized in scenarios where this hyperparameter is relatively easy to set and simplicity is preferred.

2.1.3 Contribution III: Robustness to Message Dropouts and Dynamic Networks

The final contribution extends the framework to the communication layer, addressing the reality of unreliable networks and moving populations of agents.

- Innovation: We introduced an ADMM algorithm with resilience against stragglers and message dropouts to ensure fast convergence despite stragglers and potential message dropouts. To achieve this, the algorithm combines asynchronous and decentralized operations with a contribution tracking scheme designed to ensure data consistency. To enforce higher robustness against stragglers, we further proposed another algorithm that extends this algorithm with an approximation scheme. This is to give slower nodes more time to contribute, and thereby avoid the risk of converging towards poor local minima.

- Key finding: Through theoretical analyses, we established the convergence properties of our algorithms. Across numerical experiments with various settings, our algorithms successfully reached convergence despite message dropouts. In particular, the algorithm with the approximation scheme outperformed all benchmark algorithms in terms of the object value reached within the same time frame. This is likely due to two main reasons. Firstly, faster execution allowed our algorithm to complete substantially more iterations within the allotted runtime, moving closer to the optimal solution. Secondly, the approximation mechanism in our algorithm enabled slower nodes to contribute despite being several times slower than the other nodes, reducing the likelihood of convergence toward poor local minima caused by stale or missing data.
- Practical application: By giving slower nodes more time to compute their updates, our algorithm with the approximation mechanism is more suitable when the contribution of slower nodes is essential. Meanwhile, the variant without the approximation mechanism may be preferred in scenarios where the contribution of slower nodes is negligible and simplicity is preferred.

2.1.4 Summary of Contributions

By addressing convergence speed, stragglers, and network instability, we have moved ADMM from a theoretical tool into a deployable solution for large-scale, real-world multi-agent systems. The main contributions of the papers in this thesis can be summarized as follows:

- Decentralization with fast-convergence: We bridged the performance gap between centralized and decentralized architectures. Through the proposal of FCD-ADMM and its rotational variants [1, 2], we demonstrated that it is possible to achieve fast convergence without the prohibitive messaging overhead or redundant computations typically associated with peer-to-peer networks.
- Straggler-resilience: By shifting to asynchronous operations in SRAD-ADMM, MDSRAD-ADMM and RMDSRAD-ADMM [3, 4, 5], we made it possible to enable fast-convergence despite stragglers.
- Dynamic and unreliable networking: Through [5] with MDSRAD-ADMM and RMDSRAD-ADMM, we proved that data consistency and convergence can be maintained even in the presence of severe message dropouts

and dynamic node entry. This contribution is essential for real-world deployments where network stability cannot be guaranteed.

The frameworks developed in this work can be used in various applications. For instance, they can be used for large-scale decentralized learning and multi-agent systems requiring collective decision-making. In healthcare, these algorithms allow hospitals to train high-performance diagnostic models without sharing sensitive patient data. In autonomous vehicle coordination, these algorithms can be used to help vehicles reach consensus in a decentralized manner. More specifically, the algorithms proposed in [1, 2] are suitable in environments where simplicity and decentralization is preferred; the algorithms proposed in [3, 4] are suitable in systems with fast and stable communication among the nodes yet prone to problems with stragglers; the algorithms proposed in [5] are suitable in dynamic and unstable systems with potential stragglers and single points of failures.

2.2 Answers to Research Questions

The contributions of this thesis include answers to the research questions presented in Chapter 1:

- Fast-convergence and decentralization: Through the proposal and evaluation of two fast-converging decentralized ADMM algorithms, we have demonstrated that it is indeed possible to achieve decentralization without causing prolonged time to convergence and without causing excessive computations and communication costs. By eliminating the need for a central coordinator while ensuring each node only maintains a local computational complexity of $O(1)$, we match the convergence speed of a centralized implementation of ADMM within a decentralized topology [1, 2].
- Resilience against stragglers and single-point-of-failures: Through the proposal and evaluation of various asynchronous ADMM algorithms, we have shown that straggler- and single-point-of-failure-resilience can be integrated into ADMM[3, 4, 5].
- Robustness against message-dropouts: We have shown that it is possible to enforce fast-convergence despite message-dropouts by proposing and evaluating two asynchronous decentralized ADMM algorithms with proven convergence properties despite message loss [5].

2.3 Future Research Directions

In this section, we present potential areas for future research.

2.3.1 Theoretical Analysis for Non-Convex Optimization

Although empirical tests conducted throughout this research indicate that the proposed algorithms perform reliably on non-convex problems, they were not explicitly designed for non-convex problems. A key area for future work is the development of a formal theoretical analysis to provide convergence guarantees for non-convex objective functions.

2.3.2 Numerical Experiments in Dynamic and Growing Networks

While the algorithms proposed in [3, 4, 5] are designed to support growing networks and handle varying network latencies, the numerical evaluations in this study focus primarily on the impact of message dropouts. Future studies could expand the experimental scope to involve a more elaborate experimental campaign to stress-test these secondary variables:

- Network delays modeling: Although our algorithms proposed in [5] are designed to work despite extended network delays, no network delays were simulated in our numerical experiments. Future work could verify this through experiments with simulated transmission latency.
- Growing network topologies: While our algorithms are designed to maintain convergence properties even when new nodes join a process mid-stream, we limited our current experiments to a fixed set of potential worker nodes $i \in \mathcal{N}$ to ensure fair comparisons across data distributions. A significant avenue for future work is the implementation of ”elastic” simulations where the network size fluctuates. This would allow for the analysis of ”warm-start” strategies and the investigation of how new entries impact the global consensus speed, particularly when the incoming node’s local data varies significantly from the existing cohort.
- Artificial straggler scenarios: In this study, we allowed straggler behavior to arise naturally from the interaction of asynchronous logic and message dropouts. Future research could involve artificially imposing fixed or fluctuating delays on specific clusters of nodes to simulate extreme hardware heterogeneity. This would allow for a deeper comparison with traditional schemes, testing the limits of our algorithms in scenarios where a subset of the network is perpetually underpowered or subjected to varying computational loads.

By expanding these numerical boundaries, future studies can provide an even more authentic representation of the algorithms’ performance in the highly unpredictable environments of mobile sensor networks and fluctuating cloud compute clusters.

2.3.3 Real-World Field Implementation and Edge Deployment

To bridge the gap between simulation and practice, the proposed algorithms should be deployed in real-world environments. This includes implementing the framework on low-power edge devices, such as mobile sensor networks or autonomous vehicle fleets.

2.3.4 Integration of Privacy and Security Mechanisms

As data privacy becomes a regulatory necessity, integrating deep learning directly into the ADMM update steps is a priority. Future research could explore adding noise to primal or dual updates - similar to the perturbation methods discussed in [22] - to prevent the leakage of sensitive information. Additionally, the algorithms can be extended to enforce resilience against Byzantine attacks by identifying and ignoring updates from compromised nodes.

Bibliography

[1] Jeannie He, Ming Xiao, and Mikael Skoglund. "Fast-Converging Decentralized ADMM for Consensus Optimization". In: *Proc. IEEE Conf. on Artif. Intel. (CAI)* (2024), pp. 575–580.

[2] Jeannie He, Ming Xiao, and Mikael Skoglund. "Fast-converging decentralized alternating direction method of multipliers for consensus optimization". In: *EURASIP Journal on Advances in Signal Processing* 2025.1 (2025), p. 27.

[3] Jeannie He, Ming Xiao, and Mikael Skoglund. "Straggler-Resilient Asynchronous Decentralized ADMM for Consensus Optimization". In: *Proc. 58th Annu. Conf. Inf. Sci. Syst. (CISS)* (Mar. 2024), pp. 1–6.

[4] Jeannie He, Ming Xiao, Mikael Skoglund, et al. "Straggler-resilient asynchronous ADMM for distributed consensus optimization". In: *IEEE Transactions on Signal Processing* (2025).

[5] Jeannie He, Ming Xiao, Mikael Skoglund, et al. "Robust Asynchronous Decentralized ADMM for Distributed Optimization in Dynamic Networks".

[6] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, et al. "A Survey on Distributed Machine Learning". In: *ACM Comput. Surv.* 53.2 (Mar. 2020), 30:1–30:33. ISSN: 0360-0300. DOI: `10.1145/3377454`. (Visited on 03/21/2025).

[7] Stephen Boyd, Neal Parikh, Eric Chu, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends ® in Machine learning* 3.1 (2011), pp. 1–122. DOI: `10.1561/2200000016`.

[8] Luyao Guo, Xinli Shi, Jinde Cao, et al. "Decentralized Inexact Proximal Gradient Method With Network-Independent Stepsizes for Convex Composite Optimization". In: *IEEE Trans. Signal Process.* 71 (Mar. 2023), pp. 786–801. DOI: `10.1109/TSP.2023.3250839`.

[9] Jiaojiao Zhang, Huikang Liu, Anthony Man-Cho So, et al. "A Penalty Alternating Direction Method of Multipliers for Convex Composite Optimization Over

Decentralized Networks”. In: *IEEE Trans. Signal Process.* 69 (June 2021), pp. 4282–4295. DOI: `10.1109/TSP.2021.3092347`.

[10] Yaohua Liu, Wei Xu, Gang Wu, et al. “Communication-Censored ADMM for Decentralized Consensus Optimization”. In: *IEEE Trans. Signal Process.* 67.10 (Mar. 2019), pp. 2565–2579.

[11] Meng Ma, Athanasios N. Nikolakopoulos, and Georgios B. Giannakis. “Fast Decentralized Learning Via Hybrid Consensus ADMM”. In: *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Apr. 2018), pp. 3829–3833. DOI: `10.1109/ICASSP.2018.8462038`.

[12] Bin Wang, Jun Fang, Huiping Duan, et al. “Graph Simplification-Aided ADMM for Decentralized Composite Optimization”. In: *IEEE Trans. Cybern.* 51.10 (2021), pp. 5170–5183. DOI: `10.1109/TCYB.2019.2953538`.

[13] Shreyansh Shethia, Akshita Gupta, Omanshu Thapliyal, et al. “Distributed Fast-Tracking Alternating Direction Method of Multipliers (ADMM) Algorithm with Optimal Convergence Rate”. In: *Proc. IEEE Int. Conf. Sys. Man Cyb. (SMC)* (Oct. 2021), pp. 976–981.

[14] Jiaqi Yan, Fanghong Guo, Changyun Wen, et al. “Parallel Alternating Direction Method of Multipliers”. In: *Information Sciences* 507 (Jan. 2020), pp. 185–196.

[15] Guo Chen and Jueyou Li. “A fully distributed ADMM-based dispatch approach for virtual power plant problems”. In: *Applied Mathematical Modelling* 58 (June 2018), pp. 300–312. ISSN: 0307-904X. DOI: `https://doi.org/10.1016/j.apm.2017.06.010`.

[16] Yicheng Chen, Rick S. Blum, and Brian M. Sadler. “Communication Efficient Federated Learning via Ordered ADMM in a Fully Decentralized Setting”. In: *Proc. 56th Annu. Conf. Inf. Sci. Syst. (CISS)* (Mar. 2022), pp. 96–100. DOI: `10.1109/CISS53076.2022.9751166`.

[17] Qing Yang and Hao Wang. “Exploring Blockchain for The Coordination of Distributed Energy Resources”. In: *Proc. 55th Annu. Conf. Inf. Sci. Syst. (CISS)* (Mar. 2021), pp. 1–6. DOI: `10.1109/CISS50987.2021.9400211`.

[18] Felix Rey, Zhoudan Pan, Adrian Hauswirth, et al. “Fully decentralized ADMM for coordination and collision avoidance”. In: *Proc. Europ. Contr. Conf.* (June 2018), pp. 825–830.

[19] Qing Ling, Yaohua Liu, Wei Shi, et al. “Weighted ADMM for Fast Decentralized Network Optimization”. In: *IEEE Trans. Signal Process.* 64.22 (Aug. 2016), pp. 5930–5942. DOI: `10.1109/TSP.2016.2602803`.

[20] Marie Maros and Joakim Jaldén. “ADMM for distributed dynamic beamforming”. In: *IEEE Transactions on Signal and Information Processing over Networks* 4.2 (Mar. 2017), pp. 220–235. DOI: `10.1109/TSIPN.2017.2681205`.

[21] Xianghui Mao, Kun Yuan, Yubin Hu, et al. “Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization”. In: *IEEE Trans. Signal Process.* 68 (Mar. 2020), pp. 2513–2528. DOI: `10.1109/TSP.2020.2983167`.

[22] Yu Ye, Hao Chen, Ming Xiao, et al. “Privacy-Preserving Incremental ADMM for Decentralized Consensus Optimization”. In: *IEEE Trans. Signal Process.* 68 (Oct. 2020), pp. 5842–5854.

[23] Mingyi Hong. “A Distributed, Asynchronous, and Incremental Algorithm for Nonconvex Optimization: An ADMM Approach”. In: *IEEE Trans. Control Netw. Syst.* 5.3 (Jan. 2018), pp. 935–945.

[24] ZhouDan Pan and Mark Cannon. “Asynchronous ADMM via a Data Exchange Server”. In: *IEEE Trans. Control Netw. Syst.* (Jan. 2024).

[25] Daniel Burk, Andreas Völz, and Knut Graichen. “Towards Asynchronous ADMM for Distributed Model Predictive Control of Nonlinear Systems”. In: *Proc. Europ. Contr. Conf. (ECC)* (June 2021), pp. 1957–1962.

[26] Chen Feng, Hong Xu, and Baochun Li. “An Alternating Direction Method Approach to Cloud Traffic Management”. In: *IEEE Trans. Parallel and Distributed Systems* 28.8 (2017), pp. 2145–2158. DOI: `10.1109/TPDS.2017.2658620`.

[27] Zheng Peng, Marit Van De Wiel, Hendrik Niemarkt, et al. “Validation of Three Late-Onset Sepsis Prediction Models in Hospitalized Infants”. In: *2024 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. June 2024, pp. 1–5. DOI: `10.1109/MeMeA60663.2024.10596841`. (Visited on 12/07/2024).

[28] Lidija Fodor, Dušan Jakovetić, Danijela Boberić Krstićev, et al. “A parallel ADMM-based convex clustering method”. In: *EURASIP Journal on Advances in Signal Processing* 2022.1 (Nov. 2022), p. 108. DOI: `10.1186/s13634-022-00942-8`.

[29] Feng Zhang, Erkang Xue, Ruixin Guo, et al. “DS-ADMM++: A Novel Distributed Quantized ADMM to Speed up Differentially Private Matrix Factorization”. In: *IEEE Trans. Parallel Distrib. Sys.* 33.6 (Sept. 2022), pp. 1289–1302.

[30] Shuai Zeng, Shuangsheng Wang, and Yi Zhang. “Optimization of Distributed ADMM Algorithm Based on Minimum Network Latency”. In: *2019 12th International Symposium on Computational Intelligence and Design (ISCID)* 2 (Dec. 2019), pp. 7–10.

[31] Kejun Huang and Nicholas D. Sidiropoulos. “Consensus-ADMM for General Quadratically Constrained Quadratic Programming”. In: *IEEE Trans. Signal Process.* 64.20 (2016), pp. 5297–5310. DOI: `10.1109/TSP.2016.2593681`.

[32] Arda Aytekin and Mikael Johansson. “Exploiting Serverless Runtimes for Large-Scale Optimization”. In: *Proc. IEEE 12th Int. Conf. Cloud Comput (CLOUD)* (July 2019), pp. 499–501.

[33] Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. “Improving the Privacy and Accuracy of ADMM-Based Distributed Algorithms”. In: *Proc. Int. Conf. Mach. Learn. (ICML)* (July 2018), pp. 5796–5805.

[34] Zheng Xu, Gavin Taylor, Hao Li, et al. “Adaptive Consensus ADMM for Distributed Optimization”. In: *Proc. Int. Conf. Mach. Learn. (ICML)* (2017), pp. 3841–3850.

[35] Xiangfeng Wang, Junchi Yan, Bo Jin, et al. “Distributed and Parallel ADMM for Structured Nonconvex Optimization Problem”. In: *IEEE Trans. Cybern.* 51.9 (Dec. 2019), pp. 4540–4552.

[36] Xiangfeng Wang, Junchi Yan, Bo Jin, et al. “Distributed and parallel ADMM for structured nonconvex optimization problem”. In: *IEEE Transactions on Cybernetics* 51.9 (Dec. 2019), pp. 4540–4552. DOI: `10.1109/TCYB.2019.2950337`.

[37] Parminder Singh, Mehedi Masud, M Shamim Hossain, et al. “Privacy-Preserving Serverless Computing using Federated Learning for Smart Grids”. In: *IEEE Trans. Indust. Inform.* (2021).

[38] Manshang Wang, Yifan Su, Laijun Chen, et al. “Distributed Optimal Power Flow of DC microgrids: A Penalty Based ADMM Approach”. In: *CSEE Journal of Power and Energy Systems* 7.2 (2019), pp. 339–347.

[39] Weiyu Li, Yaohua Liu, Zhi Tian, et al. “Communication-censored linearized ADMM for decentralized consensus optimization”. In: *IEEE Transactions on Signal and Information Processing over Networks* 6 (Dec. 2019), pp. 18–34. DOI: `10.1109/TSIPN.2019.2957719`.

[40] Zeming Hu, Yuanping Zhu, Jing Xu, et al. “Distributed weighted sum-rate maximization with multi-cell uplink-downlink throughput duality”. In: *EURASIP Journal on Advances in Signal Processing* 2015 (Mar. 2015), pp. 1–16. DOI: `10.1186/s13634-015-0213-2`.

[41] Peng Liu, Jiandong Li, Hongyan Li, et al. “Robust power allocation in two-tier heterogeneous networks”. In: *EURASIP Journal on Wireless Communications and Networking* 2014 (Apr. 2014), pp. 1–13. DOI: `10.1186/1687-1499-2014-69`.

[42] Tomaso Erseghe. “A distributed approach to the OPF problem”. In: *EURASIP Journal on Advances in Signal Processing* 2015 (May 2015), pp. 1–13. DOI: `10.1186/s13634-015-0226-x`.

[43] Meng Ma, Athanasios N Nikolakopoulos, and Georgios B Giannakis. “Hybrid ADMM: a unifying and fast approach to decentralized optimization”. In: *EURASIP Journal on Advances in Signal Processing* 2018 (Dec. 2018), pp. 1–17. DOI: `10.1186/s13634-018-0589-x`.

[44] Hao Chen, Yu Ye, Ming Xiao, et al. “Coded stochastic ADMM for decentralized consensus optimization with edge computing”. In: *IEEE Internet of Things Journal* 8.7 (2021), pp. 5360–5373.

[45] Yu Ye, Hao Chen, Zheng Ma, et al. “Decentralized Consensus Optimization Based on Parallel Random Walk”. In: *IEEE Communications Letters* 24.2 (2020), pp. 391–395. DOI: `10.1109/LCOMM.2019.2955442`.

[46] Jiafeng Zhang. “Asynchronous decentralized consensus ADMM for distributed machine learning”. In: *Proc. Int. Conf. High Perf. Big Data Intel. Sys.* (May 2019), pp. 22–28.

[47] Tsung-Hui Chang, Mingyi Hong, Wei-Cheng Liao, et al. “Asynchronous Distributed ADMM for Large-Scale Optimization—Part I: Algorithm and Convergence Analysis”. In: *IEEE Trans. Signal Process.* 64.12 (Mar. 2016), pp. 3118–3130.

[48] Ruiliang Zhang and James Kwok. “Asynchronous distributed ADMM for consensus optimization”. In: *Proc. Int. Conf. Mach. Learn. (ICML)* (June 2014), pp. 1701–1709.

Included papers