Distributed Optimization Algorithms for
Networked Systems

by

Nikolaos Chatzipanagiotis

Department of Mechanical Engineering & Materials Science
Duke University

Date: ______________________

Approved:

__________________________
Michael M. Zavlanos, Supervisor

__________________________
Wilkins Aquino

__________________________
Henri Gavin

__________________________
Alejandro Ribeiro

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Mechanical Engineering & Materials
Science
in the Graduate School of Duke University
2015
Abstract

Distributed Optimization Algorithms for Networked Systems

by

Nikolaos Chatzipanagiotis

Department of Mechanical Engineering & Materials Science
Duke University

Date: ______________________

Approved:

_________________________
Michael M. Zavlanos, Supervisor

_________________________
Wilkins Aquino

_________________________
Henri Gavin

_________________________
Alejandro Ribeiro

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering & Materials Science in the Graduate School of Duke University

2015
Abstract

Distributed optimization methods allow us to decompose certain optimization problems into smaller, more manageable subproblems that are solved iteratively and in parallel. For this reason, they are widely used to solve large-scale problems arising in areas as diverse as wireless communications, optimal control, machine learning, artificial intelligence, computational biology, finance and statistics, to name a few. Moreover, distributed algorithms avoid the cost and fragility associated with centralized coordination, and provide better privacy for the autonomous decision makers. These are desirable properties, especially in applications involving networked robotics, communication or sensor networks, and power distribution systems.

In this thesis we propose the Accelerated Distributed Augmented Lagrangians (ADAL) algorithm, a novel decomposition method for convex optimization problems. The method is based on the augmented Lagrangian framework and addresses problems that involve multiple agents optimizing a separable convex objective function subject to convex local constraints and linear coupling constraints. We establish the convergence of ADAL and also show that it has a worst-case $O(1/k)$ convergence rate, where $k$ denotes the number of iterations.

Moreover, we show that ADAL converges to a local minimum of the problem for cases with non-convex objective functions. This is the first published work that formally establishes the convergence of a distributed augmented Lagrangian method for non-convex optimization problems. An alternative way to select the stepsizes
used in the algorithm is also discussed.

Furthermore, we consider cases where the distributed algorithm needs to operate in the presence of uncertainty and noise and show that the generated sequences of primal and dual variables converge to their respective optimal sets almost surely. In particular, we are concerned with scenarios where: i) the local computation steps are inexact or are performed in the presence of uncertainty, and ii) the message exchanges between agents are corrupted by noise. In this case, the proposed scheme can be classified as a distributed stochastic approximation method. Compared to existing literature in this area, our work is the first that utilizes the augmented Lagrangian framework. Moreover, the method allows us to solve a richer class of problems as compared to existing methods on distributed stochastic approximation that consider only consensus constraints.

Extensive numerical experiments have been carried out in an effort to validate the novelty and effectiveness of the proposed method in all the areas of the aforementioned theoretical contributions. We examine problems in convex, non-convex, and stochastic settings where uncertainties and noise affect the execution of the algorithm. For the convex cases, we present applications of ADAL to certain popular network optimization problems, as well as to a two-stage stochastic optimization problem. The simulation results suggest that the proposed method outperforms the state-of-the-art distributed augmented Lagrangian methods that are known in the literature. For the non-convex cases, we perform simulations on certain simple non-convex problems to establish that ADAL indeed converges to non-trivial local solutions of the problems; in comparison, the straightforward implementation of the other distributed augmented Lagrangian methods on the same problems does not lead to convergence. For the stochastic setting, we present simulation results of ADAL applied on network optimization problems and examine the effect that noise and uncertainties have in the convergence behavior of the method.
As an extended and more involved application, we also consider the problem of relay cooperative beamforming in wireless communications systems. Specifically, we study the scenario of a multi-cluster network, in which each cluster contains multiple single-antenna source destination pairs that communicate simultaneously over the same channel. The communications are supported by cooperating amplify-and-forward relays, which perform beamforming. Since the emerging problem is non-convex, we propose an approximate convex reformulation. Based on ADAL, we also discuss two different ways to obtain a distributed solution that allows for autonomous computation of the optimal beamforming decisions by each cluster, while taking into account intra- and inter-cluster interference effects.

Our goal in this thesis is to advance the state-of-the-art in distributed optimization by proposing methods that combine fast convergence, wide applicability, ease of implementation, low computational complexity, and are robust with respect to delays, uncertainty in the problem parameters, noise corruption in the message exchanges, and inexact computations.
‘To my parents and friends’
Contents

Abstract iv
List of Figures xi
Acknowledgements xvii

1 Introduction 1
  1.1 Motivation .................................................. 1
    1.1.1 Distributed Optimization ............................... 1
    1.1.2 Stochastic Settings .................................... 4
  1.2 Relevant Literature ........................................ 5
    1.2.1 Distributed Optimization ................................ 5
    1.2.2 Stochastic Settings .................................... 7
  1.3 Thesis contents ........................................... 8
  1.4 Contributions ............................................. 11

2 Preliminaries 14
  2.1 Convex Analysis ........................................... 15
  2.2 Optimality Conditions .................................... 18
    2.2.1 Unconstrained Optimization ............................ 18
    2.2.2 Constrained Optimization .............................. 19
  2.3 Duality Theory ........................................... 22
    2.3.1 Lagrangian Duality ................................... 22
List of Figures

4.1 Random network of 50 sources (red dots) and 2 sinks (green squares). The rate of flow $t_{ij}$ through arc $(i, j)$ defines the thickness and color of the corresponding drawn line. Thin, light blue lines indicate weaker links. ................................................................. 114

4.2 Evolution of the sum of rates $\sum_{i \in S} s_i$ during implementation of ADAL. The horizontal line depicts the value obtained by solving the centralized problem. We observe that the distributed utility function converges to the optimal solution very fast. Also included is the subfigure illustrating the evolution of individual rates for every source. ........ 115

4.3 Constraint violation convergence of ADAL for different network sizes of 25, 50, 100 and 400 source nodes. The ratio of sources-to-destinations is kept at 25/1 and the maximum degree $q = 6$ for all cases. ........ 115

4.4 Constraint violation convergence for: a) Different exit criteria from the inner loop of DQA. The line labeled 'open' accounts for repeating the DQA inner loop until $\|A_i x^k_i - A_i x_i \|_\infty < 10^{-2}$ for every $i = 1, \ldots, N$. In the other instances we force exit if either the aforementioned criterion is satisfied or if the indicated amount of iterations $M$ has been surpassed. For $M = 1$ we obtain the ADAL method. The results correspond to a network of 50 sources and 2 sinks with $q = 6$, b) Different network densities for the ADAL and DQA methods. The results correspond to networks of 50 sources and 2 sinks with $q = 5, 9, 14$, respectively. In both figures, the horizontal axis depicts the inner loop iterations for the case of DQA. Also, note that the step shape of the DQA graphs in the figures is caused by the dual updates at each outer loop iteration. ................................................................. 116

4.5 Comparison between the ASM, DQA and ADAL methods, for a network of 50 sources, 2 sinks and $q = 7$. ................................. 116
4.6 Simulation results for different values of $\rho$ on a NUM problem with 50 sources and 2 sinks. a) Objective function convergence, b) Maximum constraint violation convergence.

4.7 Two typical LNF cases considered, with $N = 50$ nodes. Blue dots denote source nodes, while green dots correspond to sinks and red dots to relays. The flow $t_{ij}$ through arc $(i, j)$ defines the thickness of the corresponding drawn line. Thin lines indicate weaker links. a) For this case, $S = D = 5$, $R = 40$ and the source nodes are positioned to be as far away from the destinations as possible. b) For this case, $S = D = 7$, $R = 36$ and the source nodes are only half the network diameter apart from the destinations.

4.8 Evolution of the sum of flows $\sum_{(i,j)\in A} t_{ij}$ after implementation of ADAL, ASM and DQA on: a) the case depicted in Fig. 4.7(a) and b) the case depicted in Fig. 4.7(b). The horizontal lines depict the objective function values obtained after solving the corresponding centralized problem. Note how the ASM oscillates between positive and negative values of the total flow, which normally should not be the case (since $t_{ij} \geq 0$). This is caused by the fact that we have used stepsize $\sigma = 1.9$ (cf. (2.41)) in our simulations. This choice of $\sigma$ returned the fastest convergence for ASM, even though it lead to this “counter-intuitive” behavior.

4.9 Evolution of the maximum residual after implementation of ADAL, ASM and DQA on: a) the case depicted in Fig. 4.7(a) and b) the case depicted in Fig. 4.7(b).

4.10 a) A network with $N = 400$ and $S = D = 20$. b) Evolution of $\phi(t^k, \lambda^k)$ for ADAL applied on the aforementioned network.

4.11 Convergence results for ADAL, ASM and DQA applied on the 400 node network depicted in Fig. 4.10(a): a) Objective function convergence, and b) Constraint violation convergence.

4.12 Simulation results of ADAL for different values of $\rho$, on a 100 node problem with 10 sources and 10 destinations. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence.

4.13 Simulation results of ADMM for different values of $\rho$, on a 100 node problem with 10 sources and 10 destinations. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence.
4.14 Convergence results for the Gauss-Seidel counterparts of ADAL, ASM, and DQA on the network depicted in Fig. 4.7(b): a) Objective function convergence, and b) Constraint violation convergence. 124

4.15 Typical structure of the networks considered in simulations. The blue, green, and red dots correspond to source, sink and relay nodes, respectively. The blue lines correspond to edges between nodes. A network of 50 nodes and 6 commodities is depicted. 138

4.16 Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 5$: (i) Objective function convergence (the horizontal line corresponds to the value of the centralized solution), and (ii) Maximum constraint violation evolution. 139

4.17 Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 10$: (i) Objective function convergence (the horizontal line corresponds to the value of the centralized solution), and (ii) Maximum constraint violation evolution. 140

4.18 Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with $N = 60$ nodes, $K = 6$ commodities, $S = 100$ scenarios and maximum node degree $q = 6$: (i) Objective function convergence, and (ii) Maximum constraint violation evolution. 141

4.19 Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with $N = 90$ nodes, $K = 6$ commodities, $S = 40$ scenarios and maximum node degree $q = 6$: (i) Objective function convergence, and (ii) Maximum constraint violation evolution. 142

4.20 Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with quadratic cost functions and $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 5$: (i) Objective function convergence (the horizontal line corresponds to the value of the centralized solution), and (ii) Maximum constraint violation evolution. 143

4.21 Simulation results for ADAL applied to problem (4.11): a) Convergence of the primal variables, and b) Convergence of the residual and the dual variable. 144
4.22 Simulation results for ADAL and the centralized solver applied to problem (4.12). The results correspond to 50 different initialization instances. At each instance, the initialization point is the same for both ADAL and the centralized solver. The red and blue squares indicate the objective function value at the point of convergence for the centralized method and ADAL, respectively. A blue dashed line indicates that ADAL converged to a better (or the same) local minimum, while a red dashed line indicates the opposite.

4.23 Simulation results of ADAL applied to problem (4.13) for different values of the penalty parameter $\rho = 1, 3, 10, 20$: a) Objective function convergence, and b) Constraint violation convergence.

4.24 a) Simulation results for ADAL and the centralized solver applied to problem (4.13). The results correspond to 50 different initialization instances. At each instance, the initialization point is the same for both ADAL and the centralized solver. The red and blue squares indicate the objective function value at the point of convergence for the centralized method and ADAL, respectively. A blue dashed line indicates that ADAL converged to a better (or the same) local minimum, while a red dashed line indicates the opposite. b) Evolution of the Lyapunov function $\phi(x^k, \lambda^k)$.

4.25 Simulation results for ADAL and the centralized solver applied to problem (4.13). The results correspond to 50 different choices of the problem parameters $A$ and $b$. At each instance, the initialization point is the same for both ADAL and the centralized solver. The red and blue squares indicate the objective function value at the point of convergence for the centralized method and ADAL, respectively. A blue dashed line indicates that ADAL converged to a better (or the same) local minimum, while a red dashed line indicates the opposite.

4.26 A randomly generated network for the NUM problem with 50 sources (red dots) and 4 sinks (green squares): a) All available arcs $(i, j) \in A$ are depicted as blue lines, b) Flow routing decisions after solving the problem with SADAL. The rate of flow $t_{ij}$ through arc $(i, j)$ defines the thickness and color of the corresponding drawn line. Thicker, darker blue lines indicate larger flows.

4.27 Comparative simulation results for different noise levels. We consider the noise-free case (deterministic) where ADAL is applied, and two noise scenarios, labeled “easy” and “hard” (larger noise), where SADAL is applied. a) Objective function convergence, b) Maximum constraint violation convergence.
4.28 Simulation results for different choices of the stepsize sequence $\tau_k = 1/(q\nu^k)$. We consider scenarios where $\nu^k$ is increased by one every 15, 60, and 100 iterations, and also a constant stepsize choice $\tau_k = 1/q$. The results correspond to the “hard” noise scenario. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence. 154

4.29 Simulation results for different values of $\rho$. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence. 155

5.1 Simultaneous communication between 2 sources and 2 destinations with the help of 3 relays. The signal transmitted from source 1 (S1) is intended for destination 1 (D1), while the signal transmitted from source 2 (S2) is intended for destination 2 (D2). Signals from S1 and S2 that reach D2 and D1, respectively, are considered interference. Also shown are the channel gains $f_{ij}$ between sources and relays, and $g_{ij}$ between relays and destinations. 160

5.2 A $N = 2$ multi-cluster relay beamforming scenario, with $M = 2$ source-destination pairs (green and blue dots respectively) and $L = 3$ dedicated relays (red dots) for each cluster. Note that only a portion of all the available channels is drawn in order to avoid congestion. 166

5.3 Two different spatial configurations of the multi-cluster network beamforming problem with: a) 5 clusters, and b) 5 clusters but larger interference levels compared to case (a), due to denser spatial positioning of the users. The blue and green circles correspond to sources and users, respectively, while the red dots depict the relays. 183

5.4 Comparison of the Direct and Indirect methods for: a) the case depicted in Fig. 5.3(a), and b) the case depicted in Fig. 5.3(b). For the Indirect method we employed two different considerations of the constraint set (5.18), one for a line formation communication network between clusters (Indirect $L$) and one for an all-to-all communication network (Indirect $D$) as explained in text. 185

5.5 Comparison of the two different distributed algorithms, ADAL and ADMM. The blue lines correspond to a problem with the setup of Fig. 5.3(a), while the pink lines correspond to a problem with the setup of Fig. 5.3(b). The results correspond to application of the Direct method. 186
5.6 Comparative convergence results for the ADAL and ADMM algorithms on a scenario with 15 clusters and $\gamma = 10\text{dB}$: a) Total transmitted power at the relays and b) Constraint feasibility evolution for the coupling constraints $\sum_{n \in N} \zeta_n = 0$. The results correspond to application of the Direct method.
Acknowledgements

I would like to express my sincere gratitude to all the people that have helped me, willingly or not, to complete this thesis. It has been a wonderful journey, both literally and figuratively.

First and foremost, I am indebted to my advisor, professor Michael M. Zavlanos from Duke University, who provided me with the opportunity, support, inspiration, guidance and mentoring throughout my five year tenure as a PhD student in Stevens and Duke. I couldn’t have wished for a better environment.

My deepest gratitude and respect go to professors Darinka Dentcheva from Stevens Institute of Technology, and Athina Petropulu from Rutgers University with whom we have had a very fruitful, creative, and, perhaps more importantly, enjoyable collaboration during all these years. It seems as yesterday when professor Dentcheva told me about augmented Lagrangians five years ago. At that time I had no idea what was going on, yet it felt so interesting and exciting.

My sincere thanks go to professors Wilkins Aquino, Henri Gavin, and Silvia Ferarri from Duke University and Alejandro Ribeiro from the University of Pennsylvania, for accepting to be members of my thesis committee and for their valuable comments that greatly helped improve this dissertation. Special reference is due to professor Ribeiro whose research results I have extensively used in my own work.

Aside from the faculty, there are many students and post doctoral scholars who have made my experience as a graduate student most gratifying. A very big thank
you goes to our research group of extraordinary people in Duke: Reza Khodayi-Mehr, Charles Freundlich, Yiannis Kantaros, Nithesh Reddy Nelvoy, and Soomin Lee for all the help and all the laughs we have had and are going to have in the future. A very special reference and gratitude is due to Dionysios Kalogerias from Rutgers University with whom I have had the best collaboration, both in terms of productivity and fun. From the old days of Stevens, I would like to thank Aspasia Kalomoiris and Dimitris Efthymiou. It is also essential to thank all my friends in that wonderful place that is called Greece. They made my vacations fun and relaxing, thus giving me the energy and willingness to work harder during the rest of the year.

Last but not least, I would like to thank my parents for their unconditional love and support. Without them, none of this would have been possible. A very spacial thank you goes to Vlotina for being who she is.
1.1 Motivation

1.1.1 Distributed Optimization

The problems that arise across all applied disciplines nowadays are characterized by their ever increasing size and complexity. Applications in areas as diverse as wireless communications, sensor networks, networked robotics, machine learning, artificial intelligence, power systems, computational biology, logistics, finance and statistics are, more often than not, accompanied by very large datasets and the need to obtain, store, and retrieve these data in a decentralized manner. This fact coupled with the ongoing advancements in the parallel processing capabilities of contemporary computers, have recently revealed the need for developing efficient decomposition methods to carry out the necessary computations. Decomposition refers to the general approach of breaking a problem into smaller, more manageable subproblems that are then solved separately, either in parallel or sequentially.

Another common occurrence in a wide range of modern day problems is that they often exhibit a network structure. Cellular phone networks, power grids, and the Internet are the prime examples, while some other typical cases include sensor
networks, multi-agent robotics applications, biological networks, transportation and logistics problems to name a few. Problems falling into this category are characterized as *Networked Control Systems*. They usually consist of processors, sensors, actuators and controllers which operate in a distributed fashion over geographically disparate locations and coordinate by exchanging information over an underlying communication network. On top of the aforementioned scalability issues, an additional characteristic of networked control systems is that they are often comprised of multiple autonomous agents. These agents need to establish some form of cooperation in order to achieve some global goal, while taking decisions separately based on heterogeneous preferences. Moreover, the network structure of such problems entails that the constituent agents have access only to local information. This may be due to security and privacy issues, or simply due to the size of the network being prohibitively large, such that the aggregation of global information to a central node is practically impossible or highly inefficient.

The aforementioned issues of scalability and local information availability create the necessity to develop *distributed* computation methods. Here, we use the term distributed to indicate that a scheme combines both features of decomposability and locality of information. In summary, the advantages of distributed computation methods can be listed as follows: i) they offer some remedy to the scalability issues that plague “big-data” applications, ii) they avoid the cost, and fragility of collecting all the required information at a centralized location and alleviate the need for central coordination, iii) they can provide better privacy and security for the autonomous decision makers, iv) they can adapt to changes in the problem’s parameters much faster than their centralized counterparts, and v) they are more robust to failures of the underlying communication network. Of course, distributed methods do not violate the “no free lunch” theorem; the advantage of breaking the problem into smaller subproblems comes at the cost of having to coordinate the distributed decision mak-
ers such that they adhere to the global objective and the coupling constraints (if any).

This need for coordination is the reason why distributed methods are almost always iterative procedures. In other words, we avoid solving a big, centralized problem at once, by opting to solve smaller, decentralized subproblems many times.

One example where the idea of decomposition appears is in the context of solving linear equations. Indeed, Gaussian elimination, Schur complement methods, column generation, and the matrix inversion lemma can be viewed as decomposition approaches; the concept is to use efficient methods to solve subproblems, and combine the results in an appropriate way such as to solve the larger problem. Nevertheless, the terms decomposition and distributed are almost always associated with optimization theory and applications. In fact, decomposition is a rather old idea in the optimization literature and traces back to at least the early 1960s, while still being vibrant until today.

The association of distributed computation and optimization theory is reasonable since many of the aforementioned problems that are amenable to distributed computations can be posed within an optimization framework. Some indicative examples where distributed optimization techniques are applied to modern day problems include and are not limited to: wired and wireless communication networks Shakkottai and Srikant (2007); Chiang et al. (2007); Ribeiro and Giannakis (2010); Ribeiro et al. (2008); Chen et al. (2006); Chatzipanagiotis et al. (2014b), multi-agent robotic networks Nedic and Ozdaglar (2009b); Zavlanos et al. (2010); Lobel et al. (2011); Raffard et al. (2004) machine learning Boyd et al. (2011); Forero et al. (2010), statistics Lu et al. (2012); Bioucas-Dias and Figueiredo (2010), power distribution systems Gatsis and Giannakis (2013); Xu et al. (2006); Jenabi et al. (2013), image processing Afonso et al. (2010); Figueiredo and Bioucas-Dias (2010); Steidl and Teuber (2010), model predictive control Camponogara and Scherer (2011); Giselsson and Rantzer (2010); Mota et al. (2012); Conte et al. (2012), logistics Tang et al. (2013); Lidestam and
Apart from the aforementioned issues of scalability, locality of information, and the need for autonomous behavior in networked systems applications, another important consideration for real-world problems is the, practically unavoidable, presence of uncertainty and randomness in all the facets of the problems’ formulation. A typical example of how uncertainty affects decision making occurs in wireless communication applications, where the channel gains are usually modeled as randomly varying both in space and in time to account for the complex and unpredictable nature of wave propagation. This ubiquitous presence of uncertainty in practical applications has lead to the development of optimization theory that involves stochastic models, commonly known as stochastic programming; see Shapiro et al. (2009) for an excellent overview of how to formulate, analyze, and solve such problems. Due to the presence of random parameters in the models, the theory of stochastic programming combines concepts from optimization, probability, statistics and functional analysis.

The fields of distributed and stochastic optimization are inherently coupled. On the one hand, decomposition approaches are essential in stochastic programming problems, due to the complexity of the latter; perhaps the biggest caveat of stochastic programming formulations is that they will quickly blow up in size and become intractable. On the other hand, stochasticity appears in the application of any distributed method. This is because the autonomous subproblems need to compute, communicate, and coordinate with each other, and all these processes are affected by noise and disturbances in practical settings. For example, the messages that are exchanged between the distributed processors can be corrupted, delayed, or even lost. Hence, there is a natural, mutually interactive relationship between distributed and stochastic optimization theories and applications.
1.2 Relevant Literature

1.2.1 Distributed Optimization

The origins of distributed optimization methods have to be attributed to the celebrated decomposition principles of Dantzig and Wolfe (1960) and Benders (1962) for large-scale linear optimization problems. Since then a considerable amount of literature has been devoted to decomposition methods. The classic decomposition approach is that of dual decomposition and is based on Lagrangian duality theory. Specifically, the dual decomposition method exploits the fact that, in many cases, the dual problem may be more suitable for decentralized solution than the original one. The idea for the dual decomposition scheme first appears in Everett (1963) and has enjoyed a vast amount of literature ever since; see for example Lasdon (1970); Geoffrion (1972, 1971); Hiriart-Urruty and Lemarchal (1993); Bertsekas and Tsitsiklis (1997); Nedic and Ozdaglar (2009a); Necoara and Suykens (2008).

Dual methods are simple and popular, however, they suffer from certain well-documented disadvantages. A practical issue is that they exhibit slow convergence rates. If the Lagrangian relaxation problem has a non-unique solution, then the dual function is non-differentiable. This property necessitates the application of advanced techniques of non-smooth optimization in order to ensure numerical stability and efficiency of the procedure. These drawbacks are alleviated by the application of regularization techniques such as bundle methods and by the augmented Lagrangian framework, which is akin to the regularization of the dual function; see Ruszczyński (2006); Nocedal and Wright (2006); Bertsekas (1982); Bertsekas and Tsitsiklis (1997); Shor (1985).

Nevertheless, a significantly fewer number of works is available in the extant literature on distributed methods using augmented Lagrangians, which is mainly due to the fact that augmented Lagrangians lack the decomposability properties
of the ordinary Lagrangian. However, the convergence speed and the numerical advantages of augmented Lagrangian methods (see, e.g., Hestenes (1969); Powell (1969); Rockafellar (1973, 1976b,c)) provide a strong motivation for creating distributed versions of them. Early specialized techniques that allow for decomposition of the augmented Lagrangian can be traced back to the works Stephanopoulos and Westerberg (1975); Tatjewski (1989); Watanabe et al. (1978); Gabay and Mercier (1976); Fortin and Glowinski (1983); Chen and Teboulle (1994). More recent literature involves the Diagonal Quadratic Approximation (DQA) algorithm Mulvey and Ruszczynski (1992); Ruszczyński (1995); Berger et al. (1994) and the Alternating Direction Method of Multipliers (ADMM) Eckstein and Bertsekas (1992, 1990); Boyd et al. (2011). The DQA method replaces each minimization step in the centralized augmented Lagrangian algorithm (Method of Multipliers) by a separable approximation of the augmented Lagrangian function. The ADMM algorithm for convex problems is discussed for example in Eckstein and Bertsekas (1992); Fortin and Glowinski (1983); Rockafellar (1976b,c). The ADMM method is based on splitting methods for monotone operators such as Douglas-Rachford (see, Fortin and Glowinski (1983)). In Eckstein and Bertsekas (1992), the authors discuss the relations of the Douglas-Rachford splitting method and ADMM to the proximal point algorithm. In Eckstein and Bertsekas (1990), the Alternating Step Method (ASM) is developed by applying the ADMM on a certain class of optimization problems.

Recently, certain alternative approaches for distributed optimization have been proposed. These include second order Newton methods Bickson et al. (2009); Wei et al. (2010); Zargham et al. (2011), where the challenge now lies in computing the Newton direction and stepsize, which require global information. This is resolved in Bickson et al. (2009) using a belief propagation algorithm, while in Zargham et al. (2011) a Taylor expansion of the inverse Hessian is proposed to compute the Newton direction using local information. Another approach for distributed implementa-
tion utilizes projection methods Lee and Nedic (2013a,b). Parallel coordinate de-
scent methods have also been explored in Mukherjee et al. (2013); Liu et al. (2015); 
Richtarik and Takac (2015), as well as Nesterov-like methods in Nesterov (2014); 
Jakovetic et al. (2014). Continuous-time distributed algorithms have also been pro-
posed in Mateos-Nunez and Cortes (2013); Kia et al. (2015); Richert and Cortes 

1.2.2 Stochastic Settings

The literature on stochastic optimization theory and application is huge, and any 
attempt to summarize all aspects of the particular field would fall outside the scopes 
of this thesis. Hence, we will limit the discussion on the body of work that is di-
rectly relevant to the contents of this thesis. A comprehensive overview of stochastic 
programming is provided in Shapiro et al. (2009). Here, we are mostly concerned 
with the class of problems that is known as Two-Stage Stochastic Programming prob-
lems. In a standard two-stage stochastic programming model, decision variables are 
divided into two groups; namely, first stage and second stage variables. The basic 
idea is that a decision must be made “here-and-now” for the first stage decision vari-
ables, before a realization of the corresponding random variables becomes known, so 
that they optimize the expectation of the optimal value of the second-stage prob-
lem. In turn, at the second stage, the optimal value for each specific realization of 
the random parameters is determined, after solving the corresponding deterministic 
opimization problem. Existing decomposition methods for such problems include 
Yakowitz (1994); Miller and Ruszczyski (2011); Fbin and Szke (2007); Birge (1985); 
Singh et al. (2009); Ruszczyski (1993); Mulvey and Ruszczyński (1995).

Part of our work is also directly related to the literature of stochastic approxima-
tion (SA) techniques. Generally speaking, the term SA characterizes those stochas-
tic methods that attempt to iteratively solve convex optimization problems based
SA has been an active area of research since the 1950s, beginning with the seminal work of Robbins and Monro (1951), which introduced the Robbins-Monro algorithm for unconstrained optimization problems and proved that the method generates iterates that converge to the optimal solution in the mean square sense. Since then, a significant amount of SA literature has emerged, with some of the most representative works being Kiefer and Wolfowitz (1952); Polyak and Juditsky (1992); Ermoliev (1969, 1983); Ljung (1977); Ruszczynski and Syski (1983); Spall (1992, 2000); Wang and Spall (1999, 2003); Nemirovski et al. (2009); Jiang and Xu (2008); Yousefian et al. (2012). Certain works follow the so-called “limiting mean ODE” (ordinary differential equations) technique to prove the convergence of SA schemes; see for example Kushner and Yin (2003); Borkar and Meyn (2000); Borkar (2008); Benam et al. (2005); Roth and Sandholm (2013).

On the other hand, there exists a significantly smaller number of works that considers distributed stochastic approximation schemes. The existing literature on such distributed approaches is mostly concerned with consensus constrained optimization problems, wherein a set of agents with separable objective functions need to agree on a common decision vector; see, e.g., Tsitsiklis et al. (1986); Kushner and Yin (1987); Stankovic et al. (2011); Srivastava and Nedic (2011); Nedic (2011); Sundhar Ram et al. (2010); Bianchi and Jakubowicz (2013); Bianchi et al. (2013); Yousefian et al. (2013).

1.3 Thesis contents

In Section 1.1 we motivated the need to advance the state-of-the-art in distributed optimization methods. The goal is to develop new distributed techniques that combine the favorable properties of being fast, applicable to a wide range of problems, simple and straightforward to implement, and computationally inexpensive. Stochastic extensions of distributed algorithms must also be explored, since often in real world
applications some problem parameters are random or not precisely known, and also the iterative execution of the algorithms suffers from the presence of noise corruption and uncertainties, e.g., these issues arise in typical routing problems in wireless communications.

In this thesis we seek to fulfill the above requirements by proposing a distributed optimization method that is provably able to address all these issues. Along these lines, in Chapter 3 we present the Accelerated Distributed Augmented Lagrangians (ADAL) method, a novel distributed optimization algorithm for a certain class of both convex and non-convex, constrained optimization problems. ADAL is a primal-dual iterative scheme based on the augmented Lagrangian framework and addresses problems that involve multiple agents optimizing a separable objective function subject to convex local constraints and linear global coupling constraints. Each iteration of the proposed method consists of three steps. First, every agent solves a local convex optimization problem based on a separable approximation of the AL, that utilizes only locally available variables. Then, the agents update and communicate their primal variables to neighboring agents. Finally, they update their dual variables based on the values of the communicated primal variables. The computations at each step are performed in parallel.

In Section 3.2 we establish the convergence of ADAL for convex problems and discuss relations with the existing distributed methods that are also based on augmented Lagrangians, i.e., the ADMM and DQA. We prove that ADAL generates sequences of primal and dual variables that converge to their respective optimal sets. A proof on the convergence rate of the proposed scheme is also included in Section 3.3, wherein it is established that ADAL has a worst-case $O(1/k)$ convergence rate, where $k$ denotes the number of iterations.

In Section 3.4 we develop a theoretical analysis that establishes the convergence of ADAL for problems where the objective functions are possibly non-convex. To the
best of our knowledge this is the first work that shows convergence of a distributed augmented Lagrangian method applied to non-convex optimization problems. Moreover, we propose a more general and decentralized rule to select the stepsizes involved in the method. This improves on the original analysis of the ADAL method for the convex setting, cf. Section 3.2, where the stepsize selection was based on global information at initialization. It is worth noting that these two contributions discussed in Section 3.4 are independent from each other, meaning that convergence of the non-convex ADAL method can still be shown using the stepsizes from the convex case analysis of Section 3.2, and, similarly, convergence of the convex case can be shown using the stepsizes proposed in this paper.

In Section 3.5 we extend the theoretical framework to also include formulations where the execution of ADAL is plagued by noise corruption and uncertainties. In particular, we consider the scenario where: i) the agents have access only to noisy approximations of their objective functions at each iteration or, equivalently, the agents can calculate only approximate subgradients of their objective functions, and ii) noise corrupts the primal and dual message exchanges between agents during the iterative execution of the method. We show that, under appropriate assumptions, the stochastic ADAL generates sequences of primal and dual variables that converge to their respective optimal sets with probability 1.

In Chapter 4 we present numerical results of ADAL applied on certain popular, non-trivial network optimization problems in convex, non-convex, and stochastic settings. For the convex cases, ADAL is shown to outperform the current state-of-the-art methods in distributed convex optimization, i.e., the ADMM and DQA. For the non-convex cases, we examine certain toy problems where the straightforward application of ADMM fails to converge, and establish that ADAL indeed converges to a non-trivial local minimum of the problems in practice. For the stochastic cases where the iterative execution of ADAL suffers from noise and uncertainties, we present
simulation results for different levels of noise corruption and compare them with the deterministic solution from ADAL.

In Chapter 5 we discuss in detail the application of ADAL on a specific problem in wireless communications. Specifically, we are concerned with cooperative beamforming, a smart signal processing technique that exploits the constructive interference phenomenon in wave propagation to allow for directional signal transmission or reception. We consider a multi-cluster network, in which multiuser peer-to-peer relay communications take place inside each cluster, while the intra-cluster communications cause inter-cluster interference. Since the original formulation of the problem leads to an NP-hard non-convex problem, we employ semidefinite relaxation (SDR) techniques to obtain a computationally efficient approximate solution. Then, based on the ADAL algorithm, we propose a distributed approach to solve the relaxed problem, which allows for each cluster to compute its optimal beamforming weights based on information exchanges with neighboring clusters only. We propose two different ways to apply ADAL to the multi-cluster beamforming problem that allow us to model different message exchange patterns between the individual clusters, hence providing flexibility in practical applications.

1.4 Contributions

The novel contributions of this thesis can be summarized as follows.

- We propose the ADAL algorithm, a novel distributed method for convex constrained optimization problems with a certain separability structure. We formally establish its convergence properties following a novel proof path (the existing proofs for the ADMM and DQA follow different paths).

- Numerical experiments indicate that, at least for the problems considered, the proposed method outperforms existing approaches by a significant margin.
- We prove that the proposed method has a $O(1/k)$ convergence rate in the worst case, where $k$ denotes the number of iterations. This result can be useful in problems where we need to terminate the iterative procedure at a pre-determined threshold of suboptimality/constraint violation.

- We prove that ADAL also exhibits converges properties in non-convex settings. This is the first published work that formally establishes the convergence of a distributed augmented Lagrangian method for non-convex optimization problems.

- Two alternative ways to select the stepsizes in the proposed method are established.

- We prove that, with only slight adjustments, the ADAL method converges even in stochastic setting where noise corruption and uncertainties are present. This means that the stochastic version of ADAL can be seen as a *distributed stochastic approximation* algorithm. The contribution relative to this field is twofold. First, the stochastic ADAL can address more general constraint sets compared to the relevant existing literature in SA which is concerned only with consensus constraints. Second, to the best of our knowledge this is the only work that addresses stochastic approximation methods for distributed optimization using augmented Lagrangians. The main challenge that was addressed here is that AL methods are primal-dual schemes, which means that there exist multiple potential sources of noise corruption and uncertainty whose effects propagate in-between the two domains.

- We formulate the multi-cluster relay beamforming problem such that it is amenable to a decentralized solution, and propose two possible way to achieve it. To the best of our knowledge, there is no prior work showing this is pos-
sible. The closest scenarios considered in the literature are those of multi-cell
downlink beamforming Tolli et al. (2009); Shen et al. (2012), which do not
involve relays and thus the formulation is considerably simpler; the two AF
communication stages of the relay problem that we consider in this paper give
rise to several additional interference terms that have to be taken into account.
In this chapter, we review some central notions and results from convex analysis and optimization, and Lagrangian duality theory. We also discuss certain optimization algorithms that are relevant to our work. The purpose of this is to provide the necessary background and motivation for the theoretical developments that will be introduced in the following chapters. We note that all the results reviewed in this chapter are already known in existing literature; the text is comprised mostly of excerpts from the books Ruszczyński (2006); Bertsekas and Tsitsiklis (1997); Boyd and Vandenberghe (2004); Nocedal and Wright (2006).

Since the literature on the aforementioned disciplines is vast, in what follows we will limit the discussion only on the topics that are pertinent and absolutely essential.
to our line of work. We recall that the main focus of our investigation is the problem

$$\min \sum_{i=1}^{N} f_i(x_i)$$

subject to $$\sum_{i=1}^{N} A_i x_i = b,$$

$$x_i \in \mathcal{X}_i, \quad i = 1, 2, \ldots, N.$$  \hspace{1cm} (2.1)

Here, for every $$i \in \mathcal{I} = \{1, 2, \ldots, N\}$$, the $$\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$$ denotes a nonempty closed, convex subset of $$n_i$$-dimensional Euclidean space, the $$f_i : \mathbb{R}^{n_i} \to \mathbb{R}$$ is a convex function, the $$A_i$$ is a matrix of dimension $$m \times n_i$$, and $$b \in \mathbb{R}^m$$.

The following definition introduces the notion of the degree of a constraint for problems (2.1). It will play an important role in the discussion of distributed algorithms for (2.1).

**Definition 1.** For a given problem of the form (2.1), we define the degree $$q_j$$ as a measure of sparsity of the constraint $$j$$. For each constraint $$j = 1, \ldots, m$$, the degree $$q_j$$ denotes the number of individual decision makers $$i$$ associated with this constraint. That is, $$q_j$$ is the number of all $$i \in \mathcal{I} : [A_i]_j \neq 0$$. Here, $$[A_i]_j$$ denotes the $$j$$-th row of matrix $$A_i$$ and 0 stands for a zero vector of proper dimension. We define the maximum degree $$q$$ to be the maximum over all $$q_j$$, i.e.

$$q = \max_{1 \leq j \leq m} q_j.$$  \hspace{1cm} (2.2)

2.1 Convex Analysis

The notion of a convex set is central to optimization theory. A convex set is such that, for any two of its points, the entire segment joining these points is contained in the set.
Definition 2. A set $\mathcal{X} \subset \mathbb{R}^n$ is called convex if for every $x \in \mathcal{X}$ and $y \in \mathcal{X}$ it contains all points

$$ax + (1 - a)y, \quad 0 < a < 1.$$  

A particular class of convex sets, convex cones, play a significant role in optimization theory.

Definition 3. A set $\mathcal{K} \subset \mathbb{R}^n$ is called a cone if for every $x \in \mathcal{K}$ and all $a > 0$ one has $ax \in \mathcal{K}$. A convex cone is a cone that is also a convex set.

Definition 4. The set

$$\text{cone}(\mathcal{X}) \triangleq \{ \gamma x : x \in \mathcal{X}, \gamma \geq 0 \}$$

is called the cone generated by $\mathcal{X}$. If $\mathcal{X}$ is convex, then $\text{cone}(\mathcal{X})$ is also convex. For a convex set $\mathcal{X}$ and a point $x \in \mathcal{X}$, the set

$$\mathcal{K}_\mathcal{X}(x) \triangleq \text{cone}(\mathcal{X} - x)$$

is called the cone of feasible directions of $\mathcal{X}$ at $x$, and is convex according to its definition.

Definition 5. Let $\mathcal{K}$ be a cone in $\mathbb{R}^n$. The set

$$\mathcal{K}^o \triangleq \{ y \in \mathbb{R}^n : \langle y, x \rangle \leq 0 \quad \forall \ x \in \mathcal{K} \}$$

(2.3)

is called the polar cone of $\mathcal{K}$. Given a convex cone $\mathcal{K}$, its polar cone $\mathcal{K}^o$ is convex and closed.

Definition 6. Consider a convex closed set $\mathcal{X} \subset \mathbb{R}^n$ and a point $x \in \mathcal{X}$. The set

$$\mathcal{N}_\mathcal{X}(x) \triangleq [\text{cone}(\mathcal{X} - x)]^o$$

(2.4)

$$= \{ h \in \mathbb{R}^n : \langle h, y - x \rangle \leq 0, \quad \forall \ y \in \mathcal{X} \}.$$  

is called the normal cone to the set $\mathcal{X}$ at the point $x$. Since it is a polar cone, the normal cone is closed and convex.
Having covered the essential notions for convex sets, we now discuss some basic facts about the calculus of convex functions.

**Definition 7.** The domain of a function \( f : \mathbb{R}^n \to \mathbb{R} \), is the set
\[
\text{dom} f \triangleq \{ x : f(x) < +\infty \}. \tag{2.5}
\]

**Definition 8.** The epigraph a function \( f : \mathbb{R}^n \to \mathbb{R} \), is the set
\[
\text{epi} f \triangleq \{ (x, u) \in \mathbb{R}^n \times \mathbb{R} : u \geq f(x) \}. \tag{2.6}
\]

**Definition 9.** A function \( f \) is called convex if \( \text{epi} f \) is a convex set. Alternatively, a function \( f \) is convex if and only if for all \( x \) and \( y \) and for \( 0 \leq a \leq 1 \) we have
\[
f(a x + (1-a) y) \leq a f(x) + (1-a) f(y).
\]
A function \( f \) is called concave if \(-f\) is convex.

**Definition 10.** A function \( f \) is called proper if \( f(x) > -\infty \) for all \( x \in \text{dom} f \) and \( f(x) < +\infty \) for at least one \( x \).

For general convex functions the notion of a subgradient generalizes the concept of the gradient.

**Definition 11.** Let \( f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\} \) be a proper convex function and let \( x \in \text{dom} f \). A vector \( u \in \mathbb{R}^n \) that satisfies
\[
f(y) \geq f(x) + \langle u, y - x \rangle, \quad \forall y \in \mathbb{R}^n
\]
is called a subgradient of \( f \) at \( x \).

The set of all subgradients of \( f \) at \( x \) is called the subdifferential of \( f \) at \( x \) and is denoted by \( \partial f(x) \). The subdifferential \( \partial f(x) \) is a nonempty, convex, closed and bounded set if \( f \) is convex and \( x \in \text{int dom} f \), where \( \text{int} \) denotes the interior of a set. Moreover, for every convex proper function \( f \) there exists a point \( x_0 \in \text{dom} f \) such that \( f \) is subdifferentiable at \( x_0 \).
Theorem 1. Assume that \( f = f_1 + f_2 \), where \( f_1, f_2 : \mathbb{R}^n \to \mathbb{R} \cup \{\pm \infty\} \) are convex proper functions. If there exists a point \( x_0 \in \text{int dom} \, f \) such that \( f_1 \) is continuous at \( x_0 \), then

\[
\partial f(x) = \partial f_1(x) + \partial f_2(x), \quad \forall \, x \in \text{dom} \, f.
\]

2.2 Optimality Conditions

In this section we briefly discuss the conditions that have to be satisfied at points which are local minima of optimization problems. The general form of an optimization problem for a function \( f : \mathbb{R}^n \to \mathbb{R} \) over a constraint set \( \mathcal{X} \subseteq \mathbb{R}^n \) is

\[
\min_{x \in \mathcal{X}} f(x). \tag{2.8}
\]

2.2.1 Unconstrained Optimization

Whenever \( \mathcal{X} = \mathbb{R}^n \), the optimization problem (2.8) is called unconstrained, and constrained in any other case. A point \( x^* \) is a local minimum of (2.8) if there exists \( \epsilon > 0 \) such that

\[
f(y) \geq f(x^*), \quad \forall \, y \in \mathcal{X} \, \text{ such that } \|y - x^*\| \leq \epsilon. \quad \tag{2.9}
\]

The point \( x^* \) is called a strict local minimum if the above inequality is strict. If \( f(y) \geq f(x^*) \) for all \( y \in \mathcal{X} \), the point \( x^* \) is called the global minimum of (2.8).

A local minimum \( x^* \) of an unconstrained optimization problem must satisfy the following necessary conditions of optimality.

Theorem 2 (First-Order Necessary Conditions). Assume that \( f : \mathbb{R}^n \to \mathbb{R} \) is differentiable at a point \( x^* \). If \( f \) attains its unconstrained local minimum at \( x^* \), then

\[
\nabla f(x^*) = 0. \tag{2.10}
\]

If \( f \) is convex and (2.10) is satisfied, then \( x^* \) is an unconstrained global minimum of \( f \).
Any point that satisfies (2.10) is called a stationary point. All local minima are stationary points, but a stationary point is not necessarily a local minimum, unless $f$ is convex. In other words, for convex functions the first order necessary conditions for unconstrained optimality are also sufficient. Moreover, all local minima of convex functions are global minima.

In the case that $f$ is convex but not differentiable, the first-order necessary conditions for unconstrained optimality take the form of the following inclusion.

**Theorem 3.** Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm \infty\}$ be a proper convex function. A point $x^*$ is an unconstrained global minimum of $f$ if and only if

$$0 \in \partial f(x^*).$$

(2.11)

These results, which are based on elementary calculus, provide the foundations for unconstrained optimization algorithms. In one way or another, all algorithms seek a point where (2.10) or (2.11) hold.

### 2.2.2 Constrained Optimization

If an optimization problem (2.8) has constraints $x \in \mathcal{X}$, then the optimality conditions must involve the analysis of ways to perturb $x^*$ while remaining in $\mathcal{X}$. The fundamental concept in this analysis is that of tangent directions and the tangent cone.

**Definition 12.** A direction $d$ is called tangent to the set $\mathcal{X} \subset \mathbb{R}^n$ at the point $x \in \mathcal{X}$ if there exist sequences of points $x^k$ and scalars $\tau_k > 0$. $k = 1, 2, \ldots$, such that $\tau_k \downarrow 0$ and

$$d = \lim_{k \to \infty} \frac{x^k - x}{\tau_k}.$$  

(2.12)

The set $\mathcal{T}_\mathcal{X}(x)$ of all tangent directions for $\mathcal{X}$ at $x$ is a closed cone and is called the tangent cone.
Tangent cones are crucial for developing the first-order necessary conditions of optimality for constrained optimization problems of the general form (2.8).

**Theorem 4 (First-Order Necessary Conditions).** Assume that $x^*$ is a local minimum of problem (2.8) and that $f$ is differentiable at $x^*$. Then, the following inclusion must be satisfied

$$-\nabla f(x^*) \in [\mathcal{T}_x(x^*)]^\circ,$$

(2.13)

If, in addition, the function $f$ and the set $\mathcal{X}$ are convex, then $x^*$ is a global minimum of (2.8).

The above optimality conditions indicate that solving nonlinear constrained optimization problems essentially entails deciphering the fundamental relation (2.13) for the possible different forms of the feasible sets $\mathcal{X}$. With that in mind, we know that in general tangent cones may be nonconvex, which makes the analysis of optimality conditions difficult. Nevertheless, we can identify special cases where tangent cones are guaranteed to be convex and we can provide their analytical description.

**Lemma 5.** Let $\mathcal{X} \subset \mathbb{R}^n$ be a convex set, and let $x \in \mathcal{X}$. Then

$$\mathcal{T}_x(x) = \text{closure}(\mathcal{K}_x(x)),$$

i.e., the tangent cone of $\mathcal{X}$ at $x$ is equal to the closure of the cone of feasible directions, cf. 4. Moreover, the polar to the tangent cone is the normal cone of $\mathcal{X}$ at $x$

$$[\mathcal{T}_x(x)]^\circ = \mathcal{N}_x(x).$$

To make all this theory less abstract, let us consider the following class of nonlinear optimization problems:

$$\min \ f(x)$$

s.t. \  $g_i(x) \leq 0, \ i = 1, \ldots, m, \quad (2.14)$

\hspace{1em} $h_i(x) = 0, \ i = 1, \ldots, p,$

\hspace{1em} $x \in \mathcal{X}_0.$
We assume that the functions \( f : \mathbb{R}^n \to \mathbb{R}, g_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m, \) and \( h_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, p, \) are continuously differentiable, and that the set \( \mathcal{X}_0 \) is convex and closed. We also assume that the problem satisfies a constraint qualification condition; cf. Ruszczyński (2006) for a deeper and rigorous discussion on metric regularity and constraint qualification conditions. Then, we have the following result regarding the first-order necessary optimality conditions for problems of the form (2.14).

**Theorem 6.** Let \( x^* \) be a local minimum of (2.14) and assume that at \( x^* \) a constraint qualification condition is satisfied. Then, there exist multipliers \( \lambda_i^* \geq 0, i = 1, \ldots, m, \) and \( \mu_i^* \in \mathbb{R}, i = 1, \ldots, p, \) such that

\[
0 \in \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla h_i(x^*) + \mathcal{N}_{\mathcal{X}_0}(x^*),
\]

(2.15)

and

\[
\lambda_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m.
\]

(2.16)

The multipliers \( \lambda_i^* \geq 0, i = 1, \ldots, m, \) and \( \mu_i^* \in \mathbb{R}, i = 1, \ldots, p, \) are commonly referred to as Lagrange multipliers or dual variables. In the case that the functions \( f \) and \( g_i, i = 1, \ldots, m, \) are convex (but not necessarily differentiable), the functions \( h_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, p, \) are affine, the set \( \mathcal{X}_0 \) is convex, and Slater’s condition is satisfied (cf. Ruszczyński (2006)), we can use subdifferential calculus to express the first-order necessary optimality conditions for (2.14) as follows.

**Theorem 7.** Assume that \( x^* \) is a local minimum of (2.14), the function \( f \) is continuous at some point \( x_0, \) and Slater’s condition is satisfied. Then, there exist multipliers \( \lambda_i^* \geq 0, i = 1, \ldots, m, \) and \( \mu_i^* \in \mathbb{R}, i = 1, \ldots, p, \) such that

\[
0 \in \partial f(x^*) + \sum_{i=1}^{m} \lambda_i^* \partial g_i(x^*) + \sum_{i=1}^{p} \mu_i^* \partial h_i(x^*) + \mathcal{N}_{\mathcal{X}_0}(x^*),
\]

(2.17)

and

\[
\lambda_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m.
\]

(2.18)
Conversely, if for some feasible point $x^*$ of (2.14), and some $\lambda_i^* \geq 0$, $i = 1, \ldots, m$, and $\mu_i^* \in \mathbb{R}$, $i = 1, \ldots, p$, the conditions (2.17)-(2.18) are satisfied, then $x^*$ is a global minimum of problem (2.14).

2.3 Duality Theory

In this section we briefly discuss the basic elements of Lagrangian duality theory in nonlinear optimization. Duality is a fundamental concept in optimization theory, even moreso for convex problems. Given our particular focus on distributed optimization methods, duality theory is even more important; it provides us with the archetypical distributed approach, the dual decomposition method, cf. Ruszczyński (2006); Bertsekas and Tsitsiklis (1997).

2.3.1 Lagrangian Duality

Consider the general nonlinear optimization problem (2.14) and associate Lagrange multipliers $\lambda = [\lambda_1, \ldots, \lambda_m]^T \in \mathbb{R}_+^m$ to the inequality constraints, and also $\mu = [\mu_1, \ldots, \mu_p]^T \in \mathbb{R}^p$ to the equality constraints. Then, the Lagrangian for this problem is defined as

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \mu_i h_i(x). \quad (2.19)$$

The Lagrangian is a function of both the primal variables $x \in \mathcal{X}_0$, and the dual variables $(\lambda, \mu) \in \Lambda_0$, where $\Lambda_0 = \mathbb{R}_+^m \times \mathbb{R}^p$. The primal function associated with (2.14) is defined as

$$L_P(x) = \sup_{(\lambda, \mu) \in \Lambda_0} L(x, \lambda, \mu), \quad (2.20)$$

and the dual function as

$$L_D(\lambda, \mu) = \inf_{x \in \mathcal{X}_0} L(x, \lambda, \mu). \quad (2.21)$$
The primal problem associated with (2.14) is defined as
\[ \min_{x \in \mathcal{X}_0} L_P(x), \tag{2.22} \]
and the dual problem as
\[ \max_{(\lambda, \mu) \in \Lambda_0} L_D(\lambda, \mu). \tag{2.23} \]

Duality theory is essentially the study of the relations between the primal and dual problems. Two very basic facts are that the primal problem (2.22) is equivalent to the original problem (2.14), and that the dual function (2.21) is always concave; see Ruszczyński (2006) for more details. Note that since the dual function is always concave, the corresponding dual problem is always a convex optimization problem.

Another central concept is that of a saddle point.

**Definition 13.** A point \( (\bar{x}, (\bar{\lambda}, \bar{\mu})) \in \mathcal{X}_0 \times \Lambda_0 \) is called a saddle point of the Lagrangian, if for all \( x \in \mathcal{X}_0 \), and all \( (\lambda, \mu) \in \Lambda_0 \) the following inequalities are satisfied
\[ L(\bar{x}, \lambda, \mu) \leq L(\bar{x}, \bar{\lambda}, \bar{\mu}) \leq L(x, \bar{\lambda}, \bar{\mu}). \tag{2.24} \]

In other words, a saddle point is a point where the maximum of the Lagrangian with respect to \( (\lambda, \mu) \in \Lambda_0 \) and the minimum with respect to \( x \in \mathcal{X}_0 \) are attained
\[ \max_{(\lambda, \mu) \in \Lambda_0} L(\bar{x}, \lambda, \mu) = L(\bar{x}, \bar{\lambda}, \bar{\mu}) = \min_{x \in \mathcal{X}_0} L(x, \bar{\lambda}, \bar{\mu}). \tag{2.25} \]

In the case that the optimization problem is convex, then the optimal solution \( x^* \) and its corresponding Lagrange multipliers constitute such a saddle point, as stated in the following theorem.

**Theorem 8.** Assume that for problem (2.14) the functions \( f \) and \( g_i, i = 1, \ldots, m, \) are convex, the functions \( h_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, p, \) are affine, the set \( \mathcal{X}_0 \) is convex, and Slater’s condition is satisfied. Then a point \( x^* \) satisfies the first order optimality conditions of Theorem 7 with Lagrange mutlipliers \( (\lambda^*, \mu^*) \) if and only if \( (x^*, (\lambda^*, \mu^*)) \) is a saddle point of the associated Lagrangian.
There is also a straightforward relation between saddle points and the solutions of the primal 2.22 and the dual problem 2.23.

**Theorem 9.** If the Lagrangian of a problem has a saddle point \((\bar{x}, (\bar{\lambda}, \bar{\mu}))\), then \(\bar{x}\) is a solution to the primal problem, \((\bar{\lambda}, \bar{\mu})\) is a solution to the dual problem, and the following duality relation holds

\[
\min_{x \in \mathcal{X}_0} L_P(x) = \max_{(\lambda, \mu) \in \Lambda_0} L_D(\lambda, \mu). \tag{2.26}
\]

Moreover, if the duality relation (2.26) is satisfied with finite values of the primal and dual functions, then for every solution \(x^*\) of the primal problem, and for every solution \((\lambda^*, \mu^*)\) of the dual problem, the point \((x^*, (\lambda^*, \mu^*))\) is a saddle point of the associated Lagrangian.

Combining the aforementioned results, it follows that we can look for a solution to the primal problem 2.22 (which is equivalent to a solution of the original problem (2.14)) by first solving the dual problem to get the \((\bar{\lambda}, \bar{\mu})\), and then determining the primal optimal solution \(x^*\) from the saddle point conditions.

**Theorem 10.** Assume that the duality relation (2.26) holds true. If \((\bar{\lambda}, \bar{\mu}) \in \Lambda_0\) is a feasible point of the dual problem, then every point \(x^* \in \mathcal{X}_0\) that satisfies

\[
(i) \quad L(x^*, \bar{\lambda}, \bar{\mu}) = \min_{x \in \mathcal{X}_0} L(x, \bar{\lambda}, \bar{\mu}),
\]

\[
(ii) \quad \text{the point } x^* \text{ is feasible},
\]

\[
(iii) \quad \bar{\lambda}_i g_i(x^*) = 0, \quad \forall \ i = 1, \ldots, m,
\]

is a solution of the problem (2.14).

Note that a saddle point of the Lagrangian might not always exist; this is typical in nonconvex optimization problems. In that case, the duality relation (2.26) no longer holds true, and the dual problem can be used to obtain a lower bound to the optimal value of the primal problem, due to the relation \(L_D(\lambda, \mu) \leq L_P(x)\).
Definition 14. The difference
\[ \delta = \min_{x \in X_0} L_P(x) - \max_{(\lambda, \mu) \in \Lambda_0} L_D(\lambda, \mu) \geq 0, \] (2.27)
is referred to as the duality gap.

In what follows we adapt the aforementioned results from duality theory to our particular problem of interest (2.1). Let us first recall the problem

\[
\min \sum_{i=1}^{N} f_i(x_i)
\]
subject to \( \sum_{i=1}^{N} A_i x_i = b, \)
\[ x_i \in X_i, \quad i = 1, 2, \ldots, N. \]

Here, for every \( i \in \mathcal{I} = \{1, 2, \ldots, N\}, \) the \( X_i \subseteq \mathbb{R}^{n_i} \) denotes a nonempty closed, convex subset of \( n_i \)-dimensional Euclidean space, the \( f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R} \) is a convex function, the \( A_i \) is a matrix of dimension \( m \times n_i \), and \( b \in \mathbb{R}^m \). We denote

\[ f(x) = \sum_{i=1}^{N} f_i(x_i), \]

where \( x = [x_1^T, \ldots, x_N^T]^T \in \mathbb{R}^n \), with \( n = \sum_{i=1}^{N} n_i \). Furthermore, we denote \( A = [A_1 \ldots A_N] \in \mathbb{R}^{m \times n} \). The constraint \( \sum_{i=1}^{N} A_i x_i = b \) of problem (2.1) takes on the form \( Ax = b \). We associate Lagrange multipliers \( \lambda \in \mathbb{R}^m \) with that constraint. Then, the Lagrangian for (2.1) is

\[ L(x, \lambda) = f(x) + \langle \lambda, Ax - b \rangle, \]

where we have used the bra-ket notation \( \langle \lambda, Ax - b \rangle = \lambda^T (Ax - b) \). The dual function takes the form

\[ g(\lambda) = \inf_{x \in X} L(x, \lambda), \]
and the dual problem is given by

$$\max_{\lambda \in \mathbb{R}^n} g(\lambda).$$  \hspace{1cm} (2.28)

**Proposition 11.** Assume that (2.1) has an optimal solution and at least one of the following conditions are satisfied:

(i) \( \text{interior}\left(\mathcal{K}_{\mathcal{X}}(x^0)\right) \cap \{d : Ad = 0\} \neq \emptyset \), at some point \( x^0 \) such that \( Ax^0 = b \).

(i) \( \mathcal{X} \) is a polyhedral set.

Then, the dual problem (2.28) has an optimal solution, and

(i) for every optimal solution \( x^* \) of (2.1) and every optimal solution \( \lambda^* \) of (2.28)

$$f(x^*) = g(\lambda^*)$$

(i) for every optimal solution \( \lambda^* \) of (2.28), a point \( x^* \in \mathcal{X} \) is a solution of (2.1)

if and only if

$$L(x^*, \lambda^*) = \min_{x \in \mathcal{X}} L(x, \lambda^*)$$

$$Ax^* = b.$$ 

2.3.2 **Dual Subgradient Method**

In this section we briefly review the basic dual method to solve constrained optimization problems. The main idea in this approach is to find the optimal Lagrange multipliers by iteratively solving the dual problem, and then to recover the primal optimal solution utilizing the result from Theorem 10. Note that since the dual function is always concave, the corresponding dual problem is always a convex optimization problem. The basic dual method method is commonly referred to as the dual subgradient method, and it is applicable to general constrained optimization
problems of the form (2.14). Nevertheless, for the sake of clarity and readability, in what follows we will focus the discussion on the class of problems (2.1) that we are particularly interested in.

Based on the analysis of section 2.3, problem (2.1) has an optimal solution if and only if the dual problem (2.28) has an optimal solution, and the optimal values of the primal and dual problems are the same. In order to solve the dual problem (2.28), we observe that the dual function $g(\lambda)$ is concave, but not necessarily differentiable; the differentiability of $g(\lambda)$ is guaranteed if $f$ is strictly convex. The non-differentiability of the dual function is directly related to the properties of convex conjugate functions; see Rockafellar (1970) for a detailed analysis. Hence, in the general case, the solution of the dual problem requires application of methods from nonsmooth optimization. The following result helps us towards this goal, by providing a way to calculate a subgradient of the dual function $g(\lambda)$.

**Lemma 12.** Assume that for $\lambda' \in \mathbb{R}^m$ we can find $x' \in \mathcal{X}$ such that $\Lambda_D(x') = \Lambda(x', \lambda')$. Then, for all $\lambda \in \mathbb{R}^m$ we have

$$\Lambda_D(\lambda) \leq \Lambda_D(x') + \langle Ax' - b, \lambda - \lambda' \rangle.$$

Recalling the definition of a subgradient for a convex function (2.7), we observe that $Ax' - b$, henceforth called the *residual*, is indeed a subgradient for the dual function of problem (2.1). In the dual subgradient method we exploit exactly this fact to devise an iterative algorithm that performs subgradient ascent in the dual domain.

The dual subgradient method is summarized in Alg. 1. The coefficient $\tau_k$ is a positive stepsize, that in general has to satisfy certain conditions such as being summable, but not square summable; see Ruszczyński (2006) for more details on stepsize selection in subgradient methods. The term $\gamma_k$ is a positive scaling coefficient. The key practical problem associated with the dual method is *primal recovery*,
Algorithm 1 Dual Subgradient Method
Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$.

1. For a fixed vector $\lambda^k$, calculate $\hat{x}^k$ as a solution of the problem:

$$\min_{x \in \mathcal{X}} L(x, \lambda^k). \quad (2.29)$$

2. Update the dual variables according to

$$\lambda^{k+1} = \lambda^k + \tau_k \gamma_k (A \hat{x}^k - b), \quad (2.30)$$

Increase $k$ by one and return to Step 1.

that is determining the optimal solution of the original problem, after solving the dual problem. Theorem 10 provides the answer in theory, but its practical application may encounter difficulties when the solution of problem (2.29) is not unique given the optimal value of $\lambda^*$. This situation is typical in linear programming. Nevertheless, for our purposes the dual subgradient method is very important since it can lead to a distributed method in a straightforward manner, as discussed next.

2.3.3 Dual Decomposition

In this section we review the archetypical distributed optimization method, the dual decomposition. Dual decomposition is an old idea in optimization, and traces back at least to the early 1960s. Related ideas appear in the well known work by Dantzig and Wolfe (1960) and Benders (1962) on large-scale linear programming. The general idea of dual decomposition appears to be originally due to Everett (1963), and is explored in many early references such as Geoffrion (1972); Lasdon (1970).

The approach is based on the simple fact that the dual function (2.21) of problem (2.1) has a separable structure. To see why this is true, consider the Lagrangian of
problem (2.1)

\[ L(x, \lambda) = \sum_{i=1}^{N} f_i(x_i) + \sum_{i=1}^{N} \langle \lambda, A_i x_i \rangle - \langle b, \lambda \rangle \]

\[ = \sum_{i=1}^{N} L_i(x_i, \lambda) - \langle b, \lambda \rangle, \]

where we define the local Lagrangian for every \( i = 1, \ldots, N \)

\[ L_i(x_i, \lambda) = f_i(x_i) + \langle \lambda, A_i x_i \rangle. \]

The dual function has the form

\[ g(\lambda) = \sum_{i=1}^{N} \inf_{x_i \in X_i} L_i(x_i, \lambda^k) - \langle b, \lambda \rangle \]

\[ = \sum_{i=1}^{N} g_i(\lambda) - \langle b, \lambda \rangle, \]

where \( X = X_1 \times X_2 \cdots \times X_N \) and

\[ g_i(\lambda) = \inf_{x_i \in X_i} \left[ f_i(x_i) + \langle \lambda, A_i x_i \rangle \right]. \]

Finally, the dual problem can be expressed as

\[ \max_{\lambda \in \mathbb{R}^m} \sum_{i=1}^{N} g_i(\lambda) - \langle b, \lambda \rangle. \]

It follows that the first step of the dual subgradient method, i.e., the calculation of the dual function, can be decomposed into \( N \) smaller subproblems, each for the subvector \( x_i \), that can be solved independently of each other and also in parallel. Moreover, note that the dual update step (2.30) of the dual subgradient method is decomposable with respect to the rows of \( A \) by construction. Note that the decomposition property holds true for a more general class of problems than just (2.1),
Algorithm 2 Dual Decomposition Method

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$.

1. For a fixed vector $\lambda^k$, calculate $\hat{x}^k_i$ for every $i = 1, \ldots, N$ as a solution of the local problem:

   $$\min_{x_i \in X_i} L_i(x_i, \lambda^k).$$  \hspace{1cm} (2.31)

2. Update the dual variables $\lambda_j$ for every $j = 1, \ldots, m$ according to

   $$\lambda_j^{k+1} = \lambda_j^k + \tau_k \gamma_k \left( \sum_{i=1}^{N} [A_i]_j \hat{x}^k_i - b \right),$$  \hspace{1cm} (2.32)

Increase $k$ by one and return to Step 1.

however, for our purposes we have limited the discussion on (2.1) to avoid unnecessary confusions; For a more detailed discussion on the matter, the books Bertsekas and Tsitsiklis (1997); Ruszczyński (2006) are excellent references. The dual decomposition method is summarized in Alg. 2

In the general case, each iteration of the dual decomposition method requires message exchanges (transmissions and receptions) to perform the local updates of the primal and dual variables. After the local computations (2.31) have been performed, the $\hat{x}^k_i$ are communicated for the dual updates (2.32). Similarly, once the dual variables are updated, they are communicated for the primal updates in the next iteration.

2.4 Augmented Lagrangians

The dual subgradient and decomposition methods described in sections 2.3.2 and 2.3.3, respectively, suffer from well-documented disadvantages, such as slow convergence rates, and, also, non-uniqueness of solutions, which necessitates the application of advanced techniques of non-smooth optimization in order to ensure numerical stability and efficiency of the procedure. These drawbacks are alleviated by the application of regularization techniques such as bundle methods and by the augmented
Lagrangian framework, which is akin to the regularization of the dual function. In what follows we focus on augmented Lagrangian techniques and review some of their basic concepts. Moreover, we describe two existing distributed algorithms that are based on augmented Lagrangians, the \textit{Diagonal Quadratic Approximation} (DQA) algorithm Mulvey and Ruszczyński (1992); Ruszczyński (1995); Berger et al. (1994), and the \textit{Alternating Direction Method of Multipliers} (ADMM) Bertsekas and Tsitsiklis (1997); Eckstein and Bertsekas (1992, 1990); Eckstein (1993); Boyd et al. (2011).

\subsection{The augmented Lagrangian Method}

Augmented Lagrangian methods were developed in part to improve on the behavior of the dual subgradient method, and in particular, to yield convergence without assumptions like strict convexity of the objective function $f$ (recall from the analysis in section 2.3.2 that the differentiability of the dual function is guaranteed only if $f$ is strictly convex). The main idea is to add a regularization term to the ordinary Lagrangian. The augmented Lagrangian associated with problem (2.1) has the form:

$$\Lambda_\rho(x, \lambda) = f(x) + \langle \lambda, \sum_{i=1}^{N} A_i x_i - b \rangle + \frac{\rho}{2} \| \sum_{i=1}^{N} A_i x_i - b \|^2,$$

where $\rho > 0$ is a penalty parameter, cf. Ruszczyński (2006); Nocedal and Wright (2006) for more details. The augmented Lagrangian can be viewed as the ordinary Lagrangian associated with the problem

$$\min \sum_{i=1}^{N} f_i(x_i) + \frac{\rho}{2} \| \sum_{i=1}^{N} A_i x_i - b \|^2$$

subject to $\sum_{i=1}^{N} A_i x_i = b,$

$x_i \in X_i, \quad i = 1, 2, \ldots, N.$
Algorithm 3 Augmented Lagrangian Method (ALM)

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$.

1. For a fixed vector $\lambda^k$, calculate $x^{k+1}$ as a solution of the problem:

$$
\min_{x \in \mathcal{X}} \Lambda_\rho(x, \lambda^k).
$$

2. If the constraints $\sum_{i=1}^N A_i x_i^{k+1} = b$ are satisfied, then stop (optimal solution found). Otherwise, set:

$$
\lambda^{k+1} = \lambda^k + \rho \left( \sum_{i=1}^N A_i x_i^{k+1} - b \right),
$$

increase $k$ by one and return to Step 1.

This problem is clearly equivalent to the original problem (2.1), since for any feasible $x$ the penalty term added to the objective is zero.

A loose, intuitive explanation motivating the use of augmented Lagrangians is that the addition of the quadratic regularization term serves to “convexify” the problem. In particular, the benefit of including the penalty term is that there always exists some $\rho_0 \geq 0$ such that for any $\rho > \rho_0$ the augmented Lagrangian is a strictly convex function, even if the function $f$ is not strictly convex, which in turn makes the dual function differentiable. This means that we can solve the dual problem with a method that is essentially a version of gradient ascent in the dual domain, thus allowing us to avoid the problems associated with the subgradient ascent used in the classical dual method. Moreover, we can now easily determine the stepsize for the dual ascent; it is simply the penalty parameter $\rho$. The iterative algorithm that utilizes augmented Lagrangians is called the augmented Lagrangian method, also known as the Method of Multipliers and is summarized in Alg. 3.

An alternative, and fascinating, viewpoint on the ALM is that it can be regarded as the proximal point method applied to the dual problem. The proximal point method is an iterative algorithm to solve nondifferentiable optimization prob-
lems. The interested reader about the proximal point method and its connections to the augmented Lagrangian method is directed to Rockafellar (1976c); Ruszczyński (2006); Bertsekas and Tsitsiklis (1997); Iusem (1999).

The basic convergence properties of the ALM applied on our problem of interest (2.1) can be summarized in the following proposition.

**Proposition 13.** Let the assumption of Proposition 11 be satisfied. Then, the sequence of Lagrange multipliers $\lambda^k$ generated by the augmented Lagrangian method converges to a solution $\lambda^*$ of the dual problem (2.28). Moreover, if the $f_i$, $i = 1, \ldots, N$, are polyhedral functions, the sets $X_i$, $i = 1, \ldots, N$ are convex and polyhedral, and the primal problem (2.1) has a solution, then the augmented Lagrangian method converges in finitely many iterations.

It is worth noting that the augmented Lagrangian method exhibits convergence properties also in a non-convex settings assuming that the functions $f_i$, $i = 1, \ldots N$ are twice continuously differentiable and the strong second-order conditions of optimality are satisfied. We refer to Rockafellar (1976a); Bertsekas and Tsitsiklis (1997) for the analysis of the augmented Lagrangian method in the convex case and to Rockafellar (1973); Ruszczyński (2006) for the non-convex case.

A drawback of the Augmented Lagrangian Method as compared to the dual subgradient method stems from the fact that (2.34) is not directly amenable to decomposition. This is due to the quadratic penalty term in (2.33) that introduces coupling between the decision variables $x_i$, $i = 1, \ldots, N$. Nevertheless, the convergence speed and the numerical advantages of augmented Lagrangian methods provide a strong motivation for creating decomposed versions of them. Early specialized techniques that allow for decomposition of the augmented Lagrangian can be traced back to the works Stephanopoulos and Westerberg (1975); Tatjewski (1989); Watanabe et al. (1978); Gabay and Mercier (1976); Fortin and Glowinski (1983);
Algorithm 4 Alternating Directions Method of Multipliers (ADMM)

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$ and initial primal variables $x^1, z^1$.

1. For a fixed vector $\lambda^k$, and $z^k$, calculate $x^{k+1}$ as a solution of the problem:

$$
\min_x \Lambda_{\rho}^{ADMM}(x, z^k, \lambda^k).
$$

2. For a fixed vector $\lambda^k$, and $x^{k+1}$, calculate $z^{k+1}$ as a solution of the problem:

$$
\min_z \Lambda_{\rho}^{ADMM}(x^{k+1}, z, \lambda^k).
$$

3. If the constraints $Ax^{k+1} + Bz^{k+1} = c$ are satisfied, then stop (optimal solution found). Otherwise, set:

$$
\lambda^{k+1} = \lambda^k + \rho \left( Ax^{k+1} + Bz^{k+1} - c \right),
$$

increase $k$ by one and return to Step 1.

Chen and Teboulle (1994). More recent literature involves the works Mulvey and Ruszczyński (1992); Ruszczyński (1995); Berger et al. (1994); Kiwiel et al. (1999). The two most popular distributed augmented Lagrangian methods are the Diagonal Quadratic Approximation (DQA) algorithm Mulvey and Ruszczyński (1992); Ruszczyński (1995); Berger et al. (1994) and the Alternating Direction Method of Multipliers (ADMM) Bertsekas and Tsitsiklis (1997); Eckstein and Bertsekas (1992, 1990); Eckstein (1993); Boyd et al. (2011). In what follows we briefly review these two algorithms.

2.4.2 The Alternating Directions Method of Multipliers

The alternating directions method of multipliers (ADMM) is an iterative distributed optimization algorithm that was originally proposed in the mid-1970s by Glowinski and Marrocco (1975) and Gabay and Mercier (1976). In particular, Gabay and Mercier (1976) showed that ADMM is a special case of a method called Douglas-Rachford splitting Douglas and Rachford (1956) for monotone operators, and Eck-
stein and Bertsekas (1992) showed in turn that the Douglas-Rachford splitting is a special case of the proximal point algorithm Rockafellar (1976c). Recently, the ADMM has received a huge amount of interest; see e.g., Bertsekas and Tsitsiklis (1997); Eckstein and Bertsekas (1992, 1990); Boyd et al. (2011); Boley (2013); Monteiro and Svaiter (2013); Shi et al. (2014); Wei and Ozdaglar (2012, 2015); Jakovetic et al. (2015); Deng and Yin (2013); Iutzeler et al. (2015); Hong and Luo (2015); Lin et al. (2015).

In general, the ADMM solves problems of the form

$$\min_{x, z} f(x) + g(z)$$

subject to \( Ax + Bz = c \), \( 2.39 \)

where \( x \in \mathbb{R}^n \), \( z \in \mathbb{R}^m \), and \( A \in \mathbb{R}^{p \times n} \), \( B \in \mathbb{R}^{p \times m} \), \( c \in \mathbb{R}^p \). The augmented Lagrangian for this problem is

$$\Lambda^{ADM}_{\rho}(x, z, \lambda) = f(x) + g(z) + \langle \lambda, Ax + Bz - c \rangle + \frac{\rho}{2} \| Ax + Bz - c \|^2.$$  

At each iteration of ADMM the augmented Lagrangian is first minimized with respect to \( x \), then with respect to \( z \), and finally a dual update similar to the ALM takes place. The method is summarized in Alg. 4.

Obviously, problem (2.39) is different than the kind of problems we are interested in (2.1). In Eckstein and Bertsekas (1990); Eckstein (1993), the authors apply a generalized version of ADMM on problems of the form (2.1), and derive a simplified algorithmic form of ADMM, which they call the Alternating Step Method (ASM). The ASM algorithm uses the following form of local augmented Lagrangian \( \bar{\Lambda}^i_{\rho}(x_i, x^k, \lambda^k) \)

$$\bar{\Lambda}^i_{\rho}(x_i, x^k, \lambda) = f_i(x_i) + \langle \lambda, A_ix_i \rangle + \frac{\rho}{2} \sum_{i=1}^{m} \left( [A_i]_l x_i - [A_i]_l x^k_i + \frac{1}{q_l} \left( \sum_{j \in I} [A_j]_l x^k_j - b_l \right) \right)^2,$$

where \( q_l \) denotes the degree of constraint \( l \), as per Definition 1. The ASM method is summarized in Alg. 5.
Algorithm 5 Alternating Step Method (ASM)

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$ and initial primal variables $x^1$.

1. For fixed Lagrange multipliers $\lambda^k$ and for every $i = 1, \ldots, N$, determine $\hat{x}_i^k$ as the solution of the following problem:

$$
\min_{x_i \in X_i} \bar{\Lambda}^i_{p}(x_i, x^k, \lambda^k)
$$

(2.40)

2. For every $i = 1, \ldots, N$, set

$$
x_i^{k+1} = x_i^k + \sigma(\hat{x}_i^k - x_i^k),
$$

(2.41)

where $\sigma$ is a non-negative stepsize satisfying $\sigma \in (0, 2)$.

3. If the constraint $\sum_{i=1}^{N} A_i \hat{x}_i^k = b$ is satisfied, then stop (optimal solution found). Otherwise, set:

$$
\lambda_j^{k+1} = \lambda_j^k + \frac{\rho \sigma}{q_j} \left( \sum_{i=1}^{N} [A_i \hat{x}_i^k]_j - b_j \right),
$$

(2.42)

increase $k$ by one and return to Step 1.

Some interesting facts to note about the ASM method are the following. The ASM allows for adapted stepsizes in Step 3, containing the degrees of each constraint. Furthermore, some terms involving $q_j$ appear into the quadratic penalty term. The method introduces the relaxation factor $\sigma \in (0, 2)$ from the theory of the generalized ADMM Eckstein and Bertsekas (1992); Eckstein (2012) and utilizes it as a stepsize for the primal update. Note that, for $\sigma = 1$, we obtain the classical ADMM Eckstein and Bertsekas (1992); Eckstein (2012). We refer to Boyd et al. (2011); Eckstein and Bertsekas (1992); Eckstein (2012) for a discussion on the general properties of the ADMM and its applications.

2.4.3 The Diagonal Quadratic Approximation Method

In Ruszczyński (1995), the Diagonal Quadratic Approximation (DQA) method based on the augmented Lagrangian function is developed for problems of the form (2.1)
and its convergence is analyzed. The method has found applications in stochastic programming, engineering, and finance. The idea of DQA is to produce a separable approximation of the primal step of the centralized augmented Lagrangian Method (2.34)-(2.35), which iteratively converges to the actual primal step of the Augmented Lagrangian Method. This is achieved by introducing an inner loop of minimization and correction steps.

For $i = 1, \ldots, N$, the local augmented Lagrangian function $\Lambda_i^\rho : \mathbb{R}^{n_i} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is defined according to

$$
\Lambda_i^\rho(x_i, x^k, \lambda) = f_i(x_i) + \langle \lambda, A_i x_i \rangle + \frac{\mu}{2} \| A_i x_i + \sum_{j \neq i} A_j x^k_j - b \|_2^2.
$$

The method uses a parameter $\tau \in (0, 1)$, which is utilized as a stepsize in updating the primal variables. It works as follows. The DQA is summarized in Alg. 6.

Convergence of the DQA method is guaranteed if the stepsize $\tau$ satisfies $0 < \tau < \frac{1}{q}$, where $q$ is defined in (2.2). The inner-loop termination criterion in Step 2 of DQA requires that $A_i x_i^{k,s} = A_i x_i^{k,s}$ for every $i \in \mathcal{I}$, which in practice is achieved within a given numerical accuracy $\epsilon$. This entails that the augmented Lagrangian is calculated with an error bounded by $\frac{1}{\rho} \epsilon q$ (see (Ruszczyński, 1995, Lemma 1)).
Algorithm 6 Diagonal Quadratic Approximation (DQA)

Set $k = 1$, $s = 1$ and define initial Lagrange multipliers $\lambda^1$ and initial primal variables $x^{1,1}$.

1. For fixed Lagrange multipliers $\lambda^k$ and for every $i \in \mathcal{I}$, determine $\hat{x}^{k,s}_i$ as the solution of:

   $$\min_{x_i \in \mathcal{X}_i} \Lambda_i^s(x_i, x^{k,s}_i, \lambda^k).$$  

   \hspace{1cm} (2.44)

2. For every $i \in \mathcal{I}$, if $A_i \hat{x}^{k,s}_i = A_i x^{k,s}_i$, then go to step 3; otherwise, set

   $$x^{k,s+1}_i = x^{k,s}_i + \tau(\hat{x}^{k,s}_i - x^{k,s}_i),$$  

   \hspace{1cm} (2.45)

   increase $s$ by 1 and go to Step 1.

3. Set $x^{k,s} = x^{k+1}$. If the constraint $\sum_{i=1}^N A_i x^{k+1}_i = b$ is satisfied, then stop (optimal solution found). Otherwise, set :

   $$\lambda^{k+1} = \lambda^k + \rho \left( \sum_{i=1}^N A_i x^{k+1}_i - b \right),$$  

   \hspace{1cm} (2.46)

   and $s = 1$, $x^{k+1,1} = x^{k+1}$, increase $k$ by one and return to Step 1.
In this chapter we develop the main theoretical contributions that are part of this dissertation. Specifically, we present a novel distributed optimization algorithm that is based on augmented Lagrangians, which we refer to as the *accelerated distributed augmented Lagrangians (ADAL)* method. Based mostly on the theoretical notions and tools that were discussed in Chapter 2, we analyze the convergence properties of ADAL, both in deterministic and stochastic settings. We note that all results presented in this chapter are novel and have been included in the published works Chatzipanagiotis et al. (2014a); Chatzipanagiotis and Zavlanos (2015c); Chatzipanagiotis et al. (2012a), the paper Chatzipanagiotis and Zavlanos (2015a) which has been conditionally accepted, and the paper Chatzipanagiotis and Zavlanos (2015b) which is currently under review.
First, we recall that the main focus of our investigation is the problem

\[
\min_{i=1}^{N} f_i(x_i)
\]

subject to \( \sum_{i=1}^{N} A_i x_i = b \),

\[ x_i \in \mathcal{X}_i, \quad i = 1, 2, \ldots, N. \] (3.1)

Here, for every \( i \in \mathcal{I} = \{1, 2, \ldots, N\} \), the \( \mathcal{X}_i \subseteq \mathbb{R}^{n_i} \) denotes a nonempty closed, convex subset of \( n_i \)-dimensional Euclidean space, the \( f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R} \) is a convex function, the \( A_i \) is a matrix of dimension \( m \times n_i \), and \( b \in \mathbb{R}^m \).

Let us also recall the definition of the \textit{degree} of a constraint for problems (3.1), cf. Definition 1. Specifically, for each constraint \( j = 1, \ldots, m \), the degree \( q_j \) denotes the number of individual decision makers \( i \) associated with this constraint. That is, \( q_j \) is the number of all \( i \in \mathcal{I} : [A_i]_{j} \neq 0 \). Here, \([A_i]_{j}\) denotes the \( j \)-th row of matrix \( A_i \) and \( 0 \) stands for a zero vector of proper dimension. We define the maximum degree \( q \) to be the maximum over all \( q_j \), i.e.

\[
q = \max_{1 \leq j \leq m} q_j. \tag{3.2}
\]

The maximum degree \( q \) will play a critical role in the convergence properties of ADAL.

3.1 The ADAL Algorithm

The ADAL method is based on defining the \textit{local augmented Lagrangian} function \( \Lambda^i_\rho : \mathbb{R}^{n_i} \times \mathbb{R}^{n} \times \mathbb{R}^{m} \rightarrow \mathbb{R} \) for every agent \( i = 1, \ldots, N \) at each iteration \( k \), according to

\[
\Lambda^i_\rho(x_i, x^k, \lambda) = f_i(x_i) + \langle \lambda, A_i x_i \rangle + \frac{\rho}{2} \| A_i x_i + \sum_{j \in \mathcal{I}} A_j x^k_j - b \|^2. \tag{3.3}
\]
Algorithm 7 Accelerated Distributed Augmented Lagrangians (ADAL)

Set \( k = 1 \) and define initial Lagrange multipliers \( \lambda^1 \) and initial primal variables \( x^1 \).

1. For fixed Lagrange multipliers \( \lambda^k \), determine \( \hat{x}_i^k \) for every \( i \in \mathcal{I} \) as the solution of the following problem:
   \[
   \min_{x_i \in \mathcal{X}_i} \Lambda^i_{\rho}(x_i, x^k, \lambda^k).
   \] (3.4)

2. Set for every \( i \in \mathcal{I} \)
   \[
   x_i^{k+1} = x_i^k + \tau (\hat{x}_i^k - x_i^k).
   \] (3.5)

3. If the constraints \( \sum_{i=1}^N A_i x_i^{k+1} = b \) are satisfied and \( A_i \hat{x}_i^k = A_i x_i^k \), then stop (optimal solution found). Otherwise, set:
   \[
   \lambda^{k+1} = \lambda^k + \rho \tau \left( \sum_{i=1}^N A_i x_i^{k+1} - b \right),
   \] (3.6)
   increase \( k \) by one and return to Step 1.

ADAL has two parameters: a positive penalty parameter \( \rho \) and a stepsize parameter \( \tau \in (0, 1) \). Each iteration of ADAL is comprised of three steps: i) a minimization step of all the local augmented Lagrangians, ii) an update step for the primal variables, and iii) an update step for the dual variables. The computations at each step are performed in a parallel fashion, so that ADAL resembles a Jacobi-type algorithm; see Bertsekas and Tsitsiklis (1997) for more details on Jacobi and Gauss-Seidel type algorithms. The ADAL method is summarized in Alg. 7.

At the first step of each iteration, each agent minimizes its local AL subject to its local convex constraints. This computation step requires only local information. To see this, note that the variables \( A_j x_j^k \), appearing in the penalty term of the local AL \( (3.3) \), correspond to the local primal variables of agent \( j \) that were communicated to agent \( i \) for the optimization of its local Lagrangian \( \Lambda^i_{\rho} \). With respect to agent \( i \), these are considered fixed parameters. The penalty term of each \( \Lambda^i_{\rho} \) can be equivalently
expressed as
\[
\|A_i x_i + \sum_{j \in I}^{|I|} A_j x_j^k - b\|^2 = \sum_{l=1}^{m} \left( \left[ A_i x_i \right]_l + \sum_{j \in I} \left[ A_j x_j^k \right]_l - b_l \right)^2,
\]
where \([A_i]_j\) denotes the \(j\)-th row of matrix \(A_i\). The above penalty term is involved only in the minimization computation (3.4). Hence, for those \(l\) such that \([A_i]_l = 0\), the terms \(\sum_{j \in I} \left[ A_j x_j^k \right]_l - b_l\) are just constant terms in the minimization step, and can be excluded. Here, \([A_i]_l\) denotes the \(l\)-th row of \(A_i\) and \(0\) stands for a zero vector of proper dimension. This implies that subproblem \(i\) needs access only to the decisions \([A_j x_j^k]\)_l from all subproblems \(j \neq i\) that are involved in the same constraints \(l\) as \(i\). Moreover, regarding the term \(\langle \lambda, A_i x_i \rangle\) in (3.3), we have that \(\langle \lambda, A_i x_i \rangle = \sum_{j=1}^{m} \lambda_j [A_i x_i]_j\). Hence, we see that, in order to compute (3.4), each subproblem \(i\) needs access only to those \(\lambda_j\) for which \([A_i]_j \neq 0\). Intuitively speaking, each agent needs access only to the information that is relevant to the constraints that this agent is involved in.

After the local optimization steps have been carried out, the second step consists of each agent updating its primal variables by taking a convex combination with the corresponding values from the previous iteration. This update depends on a stepsize \(\tau\) which must satisfy \(\tau \in (0, \frac{1}{q})\) in order to ensure convergence of the algorithm, see section 3.2 for details.

The third and final step of each ADAL iteration consists of the dual update. This step is distributed by structure, since the Lagrange multiplier of the \(j\)-th constraint is updated according to \(\lambda_j^{k+1} = \lambda_j^k + \rho \tau \left( \sum_{i=1}^{N} \left[ A_i x_i^{k+1} \right]_j - b_j \right)\), which implies that the update of \(\lambda_j\) needs only information from those \(i\) for which \([A_i]_j \neq 0\). We can define, without loss of generality, a set \(\mathcal{M} \subseteq \{1, \ldots, m\}\) of agents that perform the dual updates, such that an agent \(j \in \mathcal{M}\) is responsible for the update of the dual variables corresponding to a subset of the coupling constraint set \(Ax = b\) (without overlapping agents). For example, if the cardinality of \(\mathcal{M}\) is equal to the number
of constraints \( m \), then each agent \( j \in \mathcal{M} \) is responsible for the update of the dual variable of the \( j \)-th constraint. In practical settings, \( \mathcal{M} \) can be a subset of \( \mathcal{I} \), or it can be a separate set of agents, depending on the application.

### 3.2 Convergence

In this section we prove that the ADAL algorithm generates sequences of primal and dual variables that converge to their respective optimal sets. The contents of this section are part of the published work Chatzipanagiotis et al. (2014a).

In order to prove convergence of ADAL, we need the following three assumptions:

(A1) The functions \( f_i : \mathbb{R}^{n_i} \to \mathbb{R}, i \in \mathcal{I} = \{1, 2, \ldots, N\} \) are convex and \( \mathcal{X}_i \subseteq \mathbb{R}^{n_i}, i = 1, \ldots, N \) are nonempty closed convex sets.

(A2) The Lagrange function \( L \) has a saddle point \((x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m\):

\[
L(x^*, \lambda) \leq L(x, \lambda) \leq L(x, \lambda^*) \quad \forall x \in \mathcal{X}, \forall \lambda \in \mathbb{R}^m. \tag{3.7}
\]

(A3) All subproblems (3.4) are solvable at any iteration \( k \in \mathcal{N} \).

Assumption (A2) implies that the point \( x^* \) is a solution of problem (3.1), the point \( \lambda^* \) is a solution of (2.28) and the strong duality relation holds, i.e., the optimal values of the primal and dual problems are equal.

Assumption (A3) is satisfied if for every \( i = 1, \ldots, N \), either the set \( \mathcal{X}_i \) is compact, or the function \( f_i(x_i) + \frac{\rho}{2}\|A_i x_i - \bar{b}\|^2 \) is inf-compact for any vector \( \bar{b} \). The latter condition, means that the level sets of the function are compact sets, implying that set \( \{x \in \mathcal{X}_i : f_i(x_i) + \frac{\rho}{2}\|A_i x_i - \bar{b}\|^2 \leq \alpha \} \) is compact for any \( \alpha \in \mathbb{R} \).

Define the residual \( r(x) \in \mathbb{R}^m \) as the vector containing the amount of all constraint violations with respect to primal variable \( x \), i.e. \( r(x) = \sum_{i=1}^{N} A_i x_i - \bar{b} \).

To avoid cluttering the notation, we will use the simplified notation \( \sum_i \) to denote summation over all \( i \in \mathcal{I} \), i.e. \( \sum_i = \sum_{i=1}^{N} \), unless explicitly noted otherwise. Also,
we define the auxiliary variables:
\[ \hat{\lambda}^k = \lambda^k + \rho r(\hat{x}^k), \quad (3.8) \]
available at iteration \( k \).

The basic idea of the proof is to introduce the Lyapunov (merit) function
\[
\phi(x^k, \lambda^k) = \sum_{i=1}^{N} \rho \| A_i(x_i^k - x_i^*) \|^2 + \frac{1}{\rho} \| \lambda^k + \rho(1 - \tau)r(x^k) - \lambda^* \|^2. \quad (3.9)
\]
We will show in Theorem 17 that this merit function is strictly decreasing during the execution of the ADAL algorithm (3.4)-(3.6), given that the stepsize \( \tau \) satisfies the condition \( 0 < \tau < 1/q \). Then, in Theorem 18 we argue that this strict decrease property implies convergence of the primal and dual variables to their respective optimal values.

We begin the proof by utilizing the first order optimality conditions of all the subproblems (3.4) in order to derive some necessary inequalities.

**Lemma 14.** Assume (A1)-(A3). The following inequality holds:
\[
\frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k) \geq \rho \sum_i \left( (A_i\hat{x}_i^k - A_ix_i^*)^\top \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right). \quad (3.10)
\]
where \((x^*, \lambda^*)\) is a saddle point of the Lagrangian \( L \) and \( \hat{x}_i^k, \hat{x}_j^k \), and \( x_j^k \) are calculated at iteration \( k \).

**Proof.** The first order optimality conditions for problem (3.4) imply the following inclusion for the minimizer \( \hat{x}_i^k \)
\[
0 \in \partial f_i(\hat{x}_i^k) + A_i^\top \lambda^k + \rho A_i^\top \left( A_i\hat{x}_i^k + \sum_{j \neq i} A_jx_j^k - b \right) + N_{\lambda_i}(\hat{x}_i^k). \quad (3.11)
\]
We infer that subgradients \( s_i^k \in \partial f_i(\hat{x}_i^k) \) and normal elements \( z_i^k \in N_{\lambda_i}(\hat{x}_i^k) \) exist such that we can express (3.11) as follows:
\[
0 = s_i^k + A_i^\top \lambda^k + \rho A_i^\top \left( A_i\hat{x}_i^k + \sum_{j \neq i} A_jx_j^k - b \right) + z_i^k. \quad (3.12)
\]
Taking inner product with $\mathbf{x}_i^* - \hat{\mathbf{x}}_i^k$ on both sides of this equation and using the definition of a normal cone, we obtain

$$\langle \mathbf{s}_i^k + \mathbf{A}_i^T \lambda^k + \rho \mathbf{A}_i^T \left( \mathbf{A}_i \hat{\mathbf{x}}_i^k + \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b} \right), \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k \rangle = \langle -\mathbf{z}_i^k, \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k \rangle \geq 0. \quad (3.13)$$

Using the variables $\hat{\lambda}^k$ defined in (3.8), we substitute $\lambda^k$ in (3.13) and obtain:

$$0 \leq \langle \mathbf{s}_i^k + \mathbf{A}_i^T \hat{\lambda}^k - \rho \left( \sum_{j \neq i} \mathbf{A}_j \hat{\mathbf{x}}_j^k - \mathbf{b} \right) + \rho \left( \mathbf{A}_i \hat{\mathbf{x}}_i^k + \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b} \right), \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k \rangle$$

$$= \langle \mathbf{s}_i^k + \mathbf{A}_i^T \hat{\lambda}^k + \rho \left( \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \sum_{j \neq i} \mathbf{A}_j \hat{\mathbf{x}}_j^k \right), \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k \rangle \quad (3.14)$$

The assumptions (A1) and (A2) entail that the following optimality conditions are satisfied at the point $(\mathbf{x}^*, \lambda^*)$:

$$0 \in \partial f_i(\mathbf{x}_i^*) + \mathbf{A}_i^T \lambda^* + \mathcal{N}_{\chi_i}(\mathbf{x}_i^*) \quad \text{for all } i = 1, \ldots, N. \quad (3.15)$$

Inclusion (3.15) implies that subgradients $\mathbf{s}_i^* \in \partial f_i(\mathbf{x}_i^*)$ and normal vectors $\mathbf{z}_i^* \in \mathcal{N}_{\chi_i}(\mathbf{x}_i^*)$ exist, such that we can express (3.15) as:

$$0 = \mathbf{s}_i^* + \mathbf{A}_i^T \lambda^* + \mathbf{z}_i^*$$

Taking inner product with $\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*$ on both sides of this equation and using the definition of a normal cone, we infer

$$\langle \mathbf{s}_i^* + \mathbf{A}_i^T \lambda^*, \hat{\mathbf{x}}_i^k - \mathbf{x}_i^* \rangle \geq 0, \quad \text{for all } i = 1, \ldots, N. \quad (3.16)$$

Combining (3.14) and (3.16), we obtain the following inequalities for all $i = 1, \ldots, N$:

$$(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)^T \left( \mathbf{s}_i^* - \mathbf{s}_i^k + \mathbf{A}_i^T (\lambda^* - \hat{\lambda}^k) - \rho \mathbf{A}_i^T \left[ \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \sum_{j \neq i} \mathbf{A}_j \hat{\mathbf{x}}_j^k \right] \right) \geq 0. \quad (3.17)$$

Using the monotonicity of the subdifferential mapping, we take out the terms involving the subgradients $(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)^T (\mathbf{s}_i^* - \mathbf{s}_i^k) \leq 0$ and arrive at:

$$(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)^T \left[ \mathbf{A}_i^T (\lambda^* - \hat{\lambda}^k) - \rho \mathbf{A}_i^T \left( \sum_{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \sum_{j \neq i} \mathbf{A}_j \hat{\mathbf{x}}_j^k \right) \right] \geq 0 \quad \forall i = 1, \ldots, N. \quad (3.18)$$
Adding the inequalities for all \(i = 1, \ldots, N\) and rearranging terms, we get:

\[
(\lambda^* - \hat{\lambda}^k)^\top \left[ \sum_i A_i (\hat{x}_i^k - x_i^k) \right] \geq \rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^k)^\top \sum_j (A_j x_j^k - A_j \hat{x}_j^k) \right] \tag{3.19}
\]

Substituting \(\sum_{i=1}^N A_i x_i^k = b\) and \(\sum_{i=1}^N A_i \hat{x}_i^k - b = \frac{1}{\rho} (\hat{\lambda}^k - \lambda^k)\) in (3.19), we conclude that

\[
\frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^* - \hat{\lambda}^k) \geq \rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^k)^\top \sum_j (A_j x_j^k - A_j \hat{x}_j^k) \right]
\]

as required. \(\square\)

In the following two lemmata, we exploit the result from Lemma 1 and perform some necessary manipulations that will allow us to prove the strict decrease property of the Lyapunov (merit) function in Theorem 17 later on.

**Lemma 15.** Under assumptions (A1)–(A3), the following estimate holds:

\[
\rho \sum_i \left[ (A_i x_i^k - A_i x_i^*)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \frac{1}{\rho} (\lambda^k - \lambda^*)^\top (\lambda^* - \hat{\lambda}^k) \\
\geq \sum_i \rho \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{\rho} \|\hat{\lambda}^k - \lambda^k\|^2 + (\hat{\lambda}^k - \lambda^k)^\top [r(x^k) - r(\hat{x}^k)].
\]

(3.20)

**Proof.** Consider the result of Lemma 14 and add the term \(\rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^*)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right]\) to both sides of inequality (3.10), which gives us

\[
\rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^*)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^* - \hat{\lambda}^k) \\
\geq \rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^*)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \rho \sum_i \left[ (A_i \hat{x}_i^k - A_i x_i^*)^\top \sum_j (A_j x_j^k - A_j \hat{x}_j^k) \right].
\]
Grouping the terms at the right-hand side of the inequality by their common factor, we transform the estimate as follows:

\[
\rho \sum_i \left[ (A_i \dot{x}_i^k - A_i x_i^k)^\top (A_i^{*} x_i^k - A_i \dot{x}_i^k) \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k)
\]

\[
\geq \rho \sum_i \left[ (A_i \dot{x}_i^k - A_i x_i^k)^\top \left( \sum_j (A_j x_j^k - A_j \dot{x}_j^k) \right) \right]
\]

(3.21)

Recall that \( \sum_j A_j (x_j^k - \dot{x}_j^k) = r(x^k) - r(\dot{x}^k) \), which means that this term is a constant factor with respect to the summation over \( i \) in the right hand side of (3.21). Moreover, \( \sum_i A_i x_i^* = b \). Substituting these terms at the right-hand side of (3.21), yields

\[
\rho \sum_i \left[ (A_i \dot{x}_i^k - A_i x_i^k)^\top (A_i^{*} x_i^k - A_i \dot{x}_i^k) \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k)
\]

\[
\geq \rho \left[ \sum_i (A_i \dot{x}_i^k - A_i x_i^k)^\top \right] \left[ r(x^k) - r(\dot{x}^k) \right]
\]

(3.22)

Next, we represent

\[
(A_i \dot{x}_i^k - A_i x_i^*) = (A_i x_i^k - A_i x_i^*) + (A_i \dot{x}_i^k - A_i x_i^k)
\]

and \( \hat{\lambda}^k - \lambda^* = (\lambda^k - \lambda^*) + (\hat{\lambda}^k - \lambda^k) \),

in the left-hand side of (3.22). We obtain

\[
\rho \sum_i \left[ (A_i x_i^k - A_i x_i^*)^\top (A_i^{*} x_i^k - A_i \dot{x}_i^k) \right] + \frac{1}{\rho} (\lambda^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k)
\]

\[
\geq \rho \sum_i \| A_i (x_i^k - \dot{x}_i^k) \|^2 + \frac{1}{\rho} \| \hat{\lambda}^k - \lambda^k \|^2 + (\hat{\lambda}^k - \lambda^k)^\top [r(x^k) - r(\dot{x}^k)]
\]

which completes the proof.

\[
\square
\]

In the next lemma, we obtain a modified version of (3.20) whose right-hand side is nonnegative. This result is utilized in Theorem 17 to show that the Lyapunov
(merit) function (3.9) is strictly decreasing. We shall use the following “pseudo-dual” variable

\[ \hat{\lambda}^k = \lambda^k + \rho(1 - \tau)\rho(x^k). \] (3.23)

**Lemma 16.** Under the assumptions (A1)–(A3), the following estimate holds

\[
\rho \sum_i \left[ (A_i x_i^k - A_i \hat{x}_i^k)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^k)^\top (\lambda^k - \hat{\lambda}^k)
\geq \frac{\rho}{2} \sum_i \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \left( \frac{\tau}{\rho} - \frac{\tau^2 q}{2 \rho} \right) \| \lambda^k - \hat{\lambda}^k \|^2, \tag{3.24}
\]

where \( \hat{\lambda}^k \) are defined in (3.23).

**Proof.** Adding \( \frac{1}{\rho}[\rho(1 - \tau)\rho(x^k)]^\top (\lambda^k - \hat{\lambda}^k) = -(1 - \tau)\rho(x^k)^\top \rho(\hat{x}^k) \) to both sides of inequality (3.20) we get:

\[
\rho \sum_i \left[ (A_i x_i^k - A_i \hat{x}_i^k)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^k)^\top (\lambda^k - \hat{\lambda}^k)
\geq \rho \sum_i \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \frac{1}{\rho} \| \hat{\lambda}^k - \lambda^k \|^2 + (\hat{\lambda}^k - \lambda^k)^\top [\rho(x^k) - \rho(\hat{x}^k)]
- (1 - \tau)\rho(x^k)^\top \rho(\hat{x}^k). \tag{3.25}
\]

Consider the term \((\hat{\lambda}^k - \lambda^k)^\top [\rho(x^k) - \rho(\hat{x}^k)] - (1 - \tau)\rho(x^k)^\top \rho(\hat{x}^k)\) at the right hand side of (3.25). We manipulate it to yield:

\[
(\hat{\lambda}^k - \lambda^k)^\top [\rho(x^k) - \rho(\hat{x}^k)] - (1 - \tau)\rho(x^k)^\top \rho(\hat{x}^k) = \tag{3.26}
\]

\[
= \rho \rho(\hat{x}^k)^\top [\rho(x^k) - \rho(\hat{x}^k)] - (1 - \tau)\rho(x^k)^\top \rho(\hat{x}^k)
= \rho \rho(\hat{x}^k)^\top [\rho(x^k) - \rho(\hat{x}^k)] - (1 - \tau)\rho \rho(x^k)^\top [\rho(x^k) - \rho(\hat{x}^k)]^\top \rho(\hat{x}^k)
= \tau \rho \rho(\hat{x}^k)^\top [\rho(x^k) - \rho(\hat{x}^k)] - (1 - \tau)\rho \rho(\hat{x}^k)^\top [\rho(x^k) - \rho(\hat{x}^k)]^\top \rho(\hat{x}^k)
= \tau(\hat{\lambda}^k - \lambda^k)^\top \sum_i A_i (x_i^k - \hat{x}_i^k) - (1 - \tau)\frac{1}{\rho} \| \hat{\lambda}^k - \lambda^k \|^2.
\]
Substituting back in (3.25), we obtain:

\[
\rho \sum_i \left[ (A_i x_i^k - A_i x_i^* \tau) (A_i x_i^k - A_i x_i^*)^\top \right] + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k)
\]

\[
\geq \rho \sum_i \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \frac{\tau}{\rho} \|\hat{\lambda}^k - \lambda^k\|^2 + \tau \sum_i (\hat{\lambda}^k - \lambda^k)^\top A_i (x_i^k - \hat{x}_i^k).
\]

(3.27)

Each of the terms \(\tau (\hat{\lambda}^k - \lambda^k)^\top A_i (x_i^k - \hat{x}_i^k)\) on the right hand side of (3.27) can be bounded below by considering

\[
\tau (\hat{\lambda}^k - \lambda^k)^\top A_i (x_i^k - \hat{x}_i^k) = \tau \sum_{j=1}^m (\hat{\lambda}_j^k - \lambda_j^k)^\top [A_i (x_i^k - \hat{x}_i^k)]_j
\]

\[
\geq - \frac{1}{2} \sum_{j=1}^m \left( \rho \left[ A_i (x_i^k - \hat{x}_i^k) \right]_j^2 + \frac{\tau^2}{\rho} (\hat{\lambda}_j^k - \lambda_j^k)^2 \right),
\]

where \([\cdot]_j\) denotes the \(j\)-th row of a matrix, and \(\lambda_j\) indicates the Lagrange multiplier of the \(j\)-th constraint. Note, however, that some of the rows of \(A_i\) might be zero. If \([A_i]_j = 0\), then it follows that \((\hat{\lambda}_j^k - \lambda_j^k)^\top [A_i (x_i^k - \hat{x}_i^k)]_j = 0\). Hence, denoting the set of nonzero rows of \(A_i\) as \(Q_i\), i.e., \(Q_i = \{j = 1, \ldots, m : [A_i]_j \neq 0\}\), we can obtain a tighter lower bound for each \(\tau (\hat{\lambda}^k - \lambda^k)^\top A_i (x_i^k - \hat{x}_i^k)\) as

\[
\tau (\hat{\lambda}^k - \lambda^k)^\top (A_i x_i^k - A_i \hat{x}_i^k) \geq - \frac{1}{2} \sum_{j \in Q_i} \left( \rho \left[ A_i (x_i^k - \hat{x}_i^k) \right]_j^2 + \frac{\tau^2}{\rho} (\hat{\lambda}_j^k - \lambda_j^k)^2 \right).
\]

(3.28)

Now, recall from (2.2) that \(q\) denotes the maximum number of non-zero blocks \([A_i]_j\) over all \(j\) (in other words, \(q\) is the maximum number of locations \(i\) that are involved in the constraint \(j\)). Then, summing inequality (3.28) over all \(i\), we observe that each quantity \((\hat{\lambda}_j^k - \lambda_j^k)^2\) is included in the summation at most \(q\) times.

This observation leads us to the bound

\[
\tau \sum_i (\hat{\lambda}^k - \lambda^k)^\top A_i (x_i^k - \hat{x}_i^k) \geq - \frac{1}{2} \left( \sum_i \rho \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \frac{\tau^2 q}{\rho} \|\hat{\lambda}^k - \lambda^k\|^2 \right).
\]

(3.29)
Finally, we substitute (3.29) into (3.27) to get

\[
\rho \sum_i \left[ (A_i x^k - A_i x^*_i)^\top (A_i x^k - A_i x^k_i) \right] + \frac{1}{\rho} (\lambda^k - \lambda^*)^\top (\lambda^k - \lambda^*) \\
\geq \sum_i \frac{\rho}{2} \|A_i (x^k_i - \hat{x}_i^k)\|_2^2 + \left( \frac{\tau}{\rho} - \frac{\tau^2 q}{2 \rho} \right) \|\lambda^k - \hat{\lambda}^k\|_2^2, \tag{3.30}
\]

which completes the proof. \qed

We are ready to prove the key result pertaining to the convergence of our method. We shall show that the function \(\phi\) defined in (3.9) is a Lyapunov function for ADAL.

**Theorem 17.** Assume (A1)–(A3). If the ADAL method uses stepsize \(\tau\) satisfying

\[
0 < \tau < \frac{1}{q},
\]

then, the sequence \(\{\phi(x^k, \lambda^k)\}\), with \(\phi(x^k, \lambda^k)\) defined in (3.9), is strictly decreasing.

**Proof.** We show that the dual update step (3.6) in the ADAL method results in the following update rule for the variables \(\lambda^k\), which are defined in (3.23):

\[
\lambda^{k+1} = \lambda^k + \tau \rho r(\hat{x}^k) \tag{3.31}
\]

Indeed,

\[
\begin{align*}
\lambda^{k+1} &= \lambda^k + \tau \rho r(x^{k+1}) \\
&= \lambda^k + \tau \rho \left[ (1 - \tau) r(x^k) + \tau r(\hat{x}^k) \right] \\
&= \lambda^k + \tau \left[ - (1 - \tau) \rho \left( r(\hat{x}^k) - r(x^k) \right) + \rho r(\hat{x}^k) \right] \\
&= \lambda^k - (1 - \tau) \rho \tau \left( r(\hat{x}^k) - r(x^k) \right) + \tau \rho r(\hat{x}^k) \tag{3.32}
\end{align*}
\]
Adding $(1 - \tau)\rho r(x^k)$ to both sides of (3.32) and rearranging terms, we obtain

$$\lambda^{k+1} + (1 - \tau)\rho \left[r(x^k) + \tau \left(r(\hat{x}^{k}) - r(x^k)\right)\right] = \lambda^k + (1 - \tau)\rho r(x^k) + \tau \rho r(\hat{x}^{k}).$$

This is equivalent to

$$\lambda^{k+1} + (1 - \tau)\rho r(x^{k+1}) = \lambda^k + (1 - \tau)\rho r(x^k) + \tau \rho r(\hat{x}^{k}),$$

which is the update rule (3.31).

We define the progress at each iteration $k$ of the ADAL method as

$$\theta_k(\tau) = \phi(x^k, \lambda^k) - \phi(x^{k+1}, \lambda^{k+1}).$$

We substitute $\tilde{\lambda}^k$ in the formula for calculating the function $\phi$ and use relation (3.31).

The progress $\theta_k(\tau)$ can be evaluated as follows:

$$\theta_k(\tau) = \sum_{i=1}^{N} \rho \|A_i(x_i^k - x_i^*)\|^2 + \frac{1}{\rho} \|\tilde{\lambda}^k - \lambda^*\|^2 - \sum_{i=1}^{N} \rho \|A_i(x_i^{k+1} - x_i^*)\|^2 - \frac{1}{\rho} \|\tilde{\lambda}^{k+1} - \lambda^*\|^2$$

$$= \sum_{i=1}^{N} \rho \|A_i(x_i^k - x_i^*)\|^2 + \frac{1}{\rho} \|\tilde{\lambda}^k - \lambda^*\|^2$$

$$- \sum_{i=1}^{N} \rho \|A_i(x_i^k - x_i^*)\| + \tau A_i(\hat{x}_i^k - x_i^k)\|^2 - \frac{1}{\rho} \|\tilde{\lambda}^k - \lambda^* + \tau \rho r(\hat{x}_i^k)\|^2$$

$$= 2\tau \left[\rho \sum_i \left[ (A_i x_i^k - A_i x_i^*)^\top (A_i x_i^k - A_i \hat{x}_i^k) \right] + \frac{1}{\rho} (\tilde{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k) \right]$$

$$- \tau^2 \left[\sum_i \rho \|A_i(\hat{x}_i^k - x_i^k)\| + \frac{1}{\rho} \|\tilde{\lambda}^k - \lambda^k\|^2 \right]. \quad (3.33)$$

We use Lemma 16 to substitute the positive term in (3.33) by its lower bound.
and obtain that the progress at each iteration is estimated as follows:

\[
\theta_k(\tau) \geq 2\tau \left[ \sum_i \frac{\rho}{2} \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \left( \frac{\tau^2 - \tau q}{\rho} \right) \| \lambda^k - \hat{\lambda}^k \|^2 \right] \\
- \tau^2 \left[ \sum_i \rho \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \frac{1}{\rho} \| \lambda^k - \hat{\lambda}^k \|^2 \right] \\
\geq \tau \left[ \sum_i (1 - \tau) \rho \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \left( \frac{\tau - \tau^2 q}{\rho} \right) \| \lambda^k - \hat{\lambda}^k \|^2 \right] \\
\geq 0 \quad \text{whenever } 0 < \tau < \frac{1}{q} < 1.
\]

Relation (3.34) implies that \( \theta_k > 0 \) during the execution of ADAL. Indeed, the coefficients of all terms at the right-hand side are positive due to the choice of parameters \( \rho > 0 \) and \( 0 < \tau < \frac{1}{q} \leq 1 \). This means that the lower bound on \( \theta_k(\tau) \) can be zero only if all terms are equal to zero, which means that \( \lambda^k - \hat{\lambda}^k = \rho r(\hat{x}^k) = 0 \), and \( A_i x_i^k = A_i \hat{x}_i^k \), for all \( i = 1, \ldots, N \). In such a case, the ADAL method stops at Step 3. Thus, \( \theta_k > 0 \) during the execution of ADAL, which in turn means that the merit function \( \phi(x^k, \lambda^k) \) is strictly decreasing. \( \square \)

**Theorem 18.** In addition to (A1)-(A3), assume that the sets \( X_i \) are bounded for all \( i = 1, \ldots N \). Then the ADAL method either stops at an optimal solution of problem (2.28) or generates a sequence of \( \lambda^k \) converging to an optimal solution of it. Any sequence \( \{x^k\} \) generated by the ADAL algorithm has an accumulation point and any such point is an optimal solution of problem (3.1).

**Proof.** If the method stops at Step 3 in some iteration \( k_0 \), then \( r(\hat{x}^{k_0}) = 0 \) and \( A_i \hat{x}_i^{k_0} = A_i x_i^{k_0} \). In this case, the optimality conditions (3.11) for problems (3.4) become

\[
0 \in \partial f_i(\hat{x}_i^{k_0}) + A_i^\top \lambda^{k_0} + \mathcal{N}_{X_i}(\hat{x}_i^{k_0}), \quad i = 1, \ldots, N
\]
which implies that \( \hat{x}_i^{k_0} \) is an optimal solution of the problem \( \min_{x_i \in \mathcal{X}_i} L_i(x_i, \lambda^{k_0}) \) with optimal Lagrange multiplier \( \lambda^{k_0} \) for all \( i = 1, \ldots, N \). Thus, \( \lambda^{k_0} \) is a solution of problem (2.28) and \( x^{k_0} \) is a solution of problem (3.1).

Now, we consider the case, in which the method generates an infinite sequence of iterates. Relation (3.34) implies that

\[
\phi(x^{k+1}, \lambda^{k+1}) \leq \phi(x^k, \lambda^k) - \tau \left[ \sum_i (1 - \tau) \rho \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \left( \frac{\tau}{\rho} - \frac{\tau^2 q}{\rho} \right) \|\lambda^k - \hat{\lambda}_i^k\|^2 \right]
\]

Iterating inequality (3.35) for \( k = 1, 2, \ldots \) and dividing by \( \tau > 0 \), we obtain:

\[
\sum_{k=1}^{\infty} \left[ \sum_{i=1}^{N} (1 - \tau) \rho \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \left( \frac{\tau}{\rho} - \frac{\tau^2 q}{\rho} \right) \|\lambda^k - \hat{\lambda}_i^k\|^2 \right] < \frac{1}{\tau} \phi(x^1, \lambda^1)
\]

This implies that the sequence \( \{A_i\hat{x}_i^k - A_i\hat{x}_i^k\} \) converges to zero as \( k \to \infty \). We substitute \( \hat{\lambda}_i^k - \lambda_i^k = \rho r(\hat{x}_i^k) \) at the left-hand side of (3.36) and infer that the sequence \( \{r(\hat{x}_i^k)\} \) converges to zero whenever \( k \to \infty \). By the monotonicity and boundedness properties of \( \phi(x^k, \lambda^k) \), we conclude that the sequence \( \{\lambda^k\} \) is convergent as well. We denote \( \lim_{k \to \infty} \lambda^k = \mu \).

Due to the boundedness of \( \mathcal{X}_i \) all sequences \( \{x_i^k\} \), \( i = 1, \ldots, N \), are bounded. This means that the sequences \( \{x_i^k\} \) have accumulation points \( \hat{x}_i \), which are also accumulation points of \( \{\hat{x}_i^k\} \) due to Step 2 of the ADAL algorithm. We can choose a subsequence \( K \subset \{1, 2, \ldots\} \) so that \( \{x_i^k\}_{k \in K} \) and \( \{\hat{x}_i^k\}_{k \in K} \) converge to \( \hat{x}_i \) for all \( i = 1, \ldots, N \). Denoting \( \hat{x} = (\hat{x}_1, \ldots, \hat{x}_N)^T \), we observe that the point \( \hat{x} \) is feasible due to the closedness of the sets \( \mathcal{X}_i \) and the continuity of \( r(\cdot) \).

For any \( i = 1, \ldots, N \), consider the sequence \( \{s_i^k\}_{k \in K} \), where \( s_i^k \) are the subgradients of \( f_i \) from the optimality condition (3.12) for problems (3.4). The subdifferential mapping \( x \mapsto \partial f(x) \) of any finite-valued convex function defined on \( \mathbb{R}^n \) is upper semi-continuous and has compact images. Therefore, the sequences \( \{s_i^k\}_{k \in K} \) have
convergent subsequences due to a fundamental result that goes back to Berge (1959).
We can choose $K_1 \subset K$ such that $\{s_i^k\}_{k \in K_1}$ converge to some $\tilde{s}_i \in \partial f_i(\tilde{x}_i)$ for all $i = 1, \ldots, N$.

Passing to the limit in equation (3.12), we infer that each sequence $\{z_i^k\}_{k \in K_1}$ converges to a point $\tilde{z}_i, i = 1, \ldots, N$. The mapping $x_i \mapsto N_{X_i}(x_i)$ has closed graph and, hence, $\tilde{z}_i \in N_{X_i}(\tilde{x}_i)$.

After the limit pass in (3.12) over $k \in K_1$, we conclude that

$$0 = \tilde{s}_i + A_i^\top \mu + \tilde{z}_i^k, \quad \forall i = 1, \ldots, N.$$  

This relation together with the feasibility of $\tilde{x}$ implies that $\tilde{x}$ is a solution of problem (3.1) and $\mu$ is a solution of problem (2.28).

Note that if $r(\tilde{x}^{k_0}) = 0$ at some iteration $k_0$ but $A_i \tilde{x}_i^{k_0} \neq A_i x_i^{k_0}$, then the ADAL method will iterate with the same Lagrange multipliers until the information is synchronized ($A_i \tilde{x}_i^k = A_i x_i^k$ for all $i = 1, \ldots, N$) resembling one inner loop of the DQA method.

**Remark 19.** The ADAL method and its convergence analysis are presented with a single penalty parameter $\rho \in \mathbb{R}_+$. Nevertheless, the constraints can be re-scaled beforehand, which amounts to using different penalty parameters $\rho_j, j = 1, \ldots, m$ for each constraint. The convergence follows in such a case by exactly the same line of arguments. However, the penalty parameter $\rho$ should be kept fixed from iteration to iteration.

**Remark 20.** Although, we have established which stepsizes guarantee theoretical convergence, we have undertaken numerical experiments, in which we multiply the primal and dual update stepsizes with some relaxation factors $\beta_p, \beta_d$, in an effort to accelerate convergence. Here, $\beta_p$ is applied to the primal variables update step and $\beta_d$ to the dual update step. Our numerical experiments indicate that, at least for
the applications considered here, we can employ relaxation factors $\beta_p \in [1, 2.5]$ and $\beta_d \in [1, q)$. The employment of such relaxations provides reasonable acceleration to the overall convergence speed.

### 3.3 Rate of Convergence

In this section we prove that ADAL has a worst-case $O(1/k)$ convergence rate, where $k$ denotes the number of iterations. The convergence rate is established in an ergodic sense, i.e., it refers to the ergodic average of the generated sequences of primal variables up to iteration $k$. The contents of this section are part of the published work Chatzipanagiotis and Zavlanos (2015c).

The main idea behind the following analysis is to define the ergodic average of the primal variables up to iteration $k$ as $y^k = \frac{1}{k} \sum_{p=0}^{k-1} \hat{x}^p$. Specifically, we show that the difference of the Lagrangian $L(y^k, \lambda^*)$ at iteration $k$, from the optimal Lagrangian $L(x^*, \lambda^*)$, i.e., the nonnegative quantity $L(y^k, \lambda^*) - L(x^*, \lambda^*) \geq 0$, decreases at a worst-case $O(1/k)$ rate. The basic step of the proof is to show that the relation

$$L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) \leq \frac{1}{2}\phi^k - \phi^{k+1}$$

holds for all iterations $k$.

In the following lemma, we utilize the first order optimality conditions for each local subproblem (3.4) to obtain a first result towards proving (3.37).

**Lemma 21.** Assume (A1)–(A3). Then, the following inequality holds:

$$L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) \leq -\left\langle \hat{x}^k - \lambda^*, r(\hat{x}^k) \right\rangle$$

$$+ \rho \sum_i \left\langle A_i(x_i^* - \hat{x}_i^k), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle.$$  

**Proof.** The first order optimality conditions for each local problem (3.4) imply the
following inclusion for the minimizer $\hat{x}_i^k$:

$$0 \in \partial f_i(\hat{x}_i^k) + \rho A_i^T \left( \lambda^k + A_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) + N_{\mathcal{X}_i}(\hat{x}_i^k).$$

We infer that subgradients $s_i^k \in \partial f_i(\hat{x}_i^k)$ and normal elements $z_i^k \in N_{\mathcal{X}_i}(\hat{x}_i^k)$ exist such that we can express the above inequality as

$$0 = s_i^k + \rho A_i^T \left( \lambda^k + A_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) + z_i^k.$$

Taking the inner product of both sides of this equation with $\hat{x}_i - \hat{x}_i^k$ and using the definition of a normal cone, cf. Def. 6, we obtain

$$\left\langle s_i^k + \rho A_i^T \left( \lambda^k + A_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right), \hat{x}_i - \hat{x}_i^k \right\rangle = \left\langle -z_i^k, \hat{x}_i^k - \hat{x}_i^k \right\rangle \geq 0.$$

Substituting $\hat{\lambda}^k$ from (3.8) in place of $\lambda^k$ in the above equation we get

$$0 \leq \left\langle s_i^k + A_i^T \left[ \hat{\lambda}^k + \rho \sum_{j \neq i} A_j (x_j^k - \hat{x}_j^k) \right], \hat{x}_i - \hat{x}_i^k \right\rangle. \quad (3.39)$$

By the definition of the subgradient of $f_i$ at $\hat{x}_i^k$, we have the relation

$$f_i(x_i) - f_i(\hat{x}_i^k) \geq s_i^k(x_i - \hat{x}_i^k), \quad \forall x_i \in \mathcal{X}_i. \quad (3.40)$$

Substituting (3.40) for $x_i = x_i^*$ into (3.39), we get

$$f_i(x_i^*) - f_i(\hat{x}_i^k) + \left\langle \hat{\lambda}^k, A_i(x_i^* - \hat{x}_i^k) \right\rangle + \rho \left\langle A_i(x_i^* - \hat{x}_i^k), \sum_{j \neq i} A_j (x_j^k - \hat{x}_j^k) \right\rangle \geq 0.$$

Summing over all $i$, we get

$$F(x^*) - F(\hat{x}^k) + \left\langle \hat{\lambda}^k, \sum_i A_i(x_i^* - \hat{x}_i^k) \right\rangle$$

$$+ \rho \sum_i \left\langle A_i(x_i^* - \hat{x}_i^k), \sum_{j \neq i} A_j (x_j^k - \hat{x}_j^k) \right\rangle \geq 0.$$
Substituting $\sum_i A_i(x^*_i - \hat{x}_i^k) = b - \sum_i A_i \hat{x}_i^k = -r(\hat{x}^k)$, and adding and subtracting $\langle \lambda^*, r(\hat{x}^k) \rangle$ to the above relation we get

$$F(x^*) - F(\hat{x}^k) - \left\langle \lambda^*, r(\hat{x}^k) \right\rangle - \left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle$$

$$+ \rho \sum_i \left\langle A_i(x^*_i - \hat{x}_i^k), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle \geq 0.$$ 

Rearranging terms in the above inequality, and noting that

$$F(\hat{x}^k) + \left\langle \lambda^*, r(\hat{x}^k) \right\rangle - F(x^*) = L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*),$$

we obtain

$$L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) \leq -\left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle$$

$$+ \rho \sum_i \left\langle A_i(x^*_i - \hat{x}_i^k), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle.$$ 

as required. \hfill \qed

In the next lemma, we further manipulate the result from Lemma 21 to obtain an expression that can help show the desired inequality (3.37).

**Lemma 22.** Assume (A1)–(A3). Then, the following inequality holds:

$$L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) + \frac{\rho}{2} \sum_i \|A_i(x^*_i - \hat{x}_i^k)\|^2 + \rho(\tau - \frac{\tau^2 q}{2}) \|r(\hat{x}^k)\|^2 \quad (3.41)$$

$$\leq \rho \sum_i \left\langle A_i(x^*_i - x^*_i), A_i(x^*_i - \hat{x}_i^k) \right\rangle - \left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle.$$

**Proof.** Consider the result (3.38) from Lemma 21, and rearrange the terms as

$$-\left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle \geq L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) + \rho \sum_i \left\langle A_i(\hat{x}_i^k - x^*_i), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle.$$ 

To avoid cluttering the notation, we temporarily disregard the $L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*)$ term, i.e., consider only the terms

$$-\left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle \geq \rho \sum_i \left\langle A_i(\hat{x}_i^k - x^*_i), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle.$$
Add the term \( \rho \sum_i \langle A_i(\hat{x}_i^k - x_i^k), A_i(x_i^k - \hat{x}_i^k) \rangle \) to both sides of the above inequality, and group the terms at the right-hand side by their common factor to get

\[
\rho \sum_i \left( \langle A_i(\hat{x}_i^k - x_i^k), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \hat{\lambda}_k - \lambda^*, r(\hat{x}_k) \rangle \right) \\
\geq \rho \sum_i \left( \langle A_i(\hat{x}_i^k - x_i^k), \sum_j A_j(x_j^k - \hat{x}_j^k) \rangle \right).
\]

The term \( \sum_j A_j(x_j^k - \hat{x}_j^k) = r(x^k) - r(\hat{x}^k) \) is a constant factor with respect to the summation over \( i \) in the right hand side of the above inequality. Moreover, \( \sum_i A_i x_i^k = b \). Hence, we have that

\[
\rho \sum_i \left( \langle A_i(\hat{x}_i^k - x_i^k), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \hat{\lambda}_k - \lambda^*, r(\hat{x}_k) \rangle \right) \\
\geq \rho \langle r(\hat{x}_k), r(x^k) - r(\hat{x}^k) \rangle.
\]

and (3.42) becomes

\[
\rho \sum_i \left( \langle A_i(\hat{x}_i^k - x_i^k), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \hat{\lambda}_k - \lambda^*, r(\hat{x}_k) \rangle \right) \\
\geq \rho \langle r(\hat{x}_k), r(x^k) - r(\hat{x}^k) \rangle.
\]

Next, we represent

\[ A_i \hat{x}_i^k - A_i x_i^k = (A_i x_i^k - A_i x_i^k) + (A_i \hat{x}_i^k - A_i x_i^k) \] and

\[ \hat{\lambda}_k - \lambda^* = (\lambda^k - \lambda^*) + (\hat{\lambda}_k - \lambda^k) = (\lambda^k - \lambda^*) + \rho r(\hat{x}_k), \]

in the left-hand side of (3.43). We obtain

\[
\rho \sum_i \left( \langle A_i(\hat{x}_i^k - x_i^k), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \hat{\lambda}_k - \lambda^*, r(\hat{x}_k) \rangle \right) \\
\geq \rho \sum_i \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \rho \|r(\hat{x}_k)\|^2 + \rho \langle r(\hat{x}_k), r(x^k) - r(\hat{x}^k) \rangle.
\]

Adding \( -(1 - \tau) \rho \langle r(x^k), r(x^k) \rangle \) to both sides of the above inequality, and recalling
the definition of $\tilde{\lambda}^k$ in (3.23), we arrive at

$$
\rho \sum_i \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \tilde{\lambda}^k - \lambda^*, r(\hat{x}^k) \rangle
\geq \rho \sum_i \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \rho \|r(\hat{x}^k)\|^2
$$

(3.44)

$$
+ \rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - (1 - \tau)\rho \langle r(x^k), r(\hat{x}^k) \rangle.
$$

Consider only the last two terms $\rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - (1 - \tau)\rho \langle r(x^k), r(\hat{x}^k) \rangle$
at the right hand side of (3.44). We can manipulate them to obtain

$$
\rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - (1 - \tau)\rho \langle r(x^k), r(\hat{x}^k) \rangle =
$$

$$
= \rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - (1 - \tau)\rho \langle r(x^k) - r(\hat{x}^k) + r(\hat{x}^k), r(\hat{x}^k) \rangle
$$

$$
= \tau \rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - (1 - \tau)\rho \|r(\hat{x}^k)\|^2.
$$

Substituting back in (3.44), we obtain

$$
\rho \sum_i \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \tilde{\lambda}^k - \lambda^*, r(\hat{x}^k) \rangle
\geq \rho \sum_i \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \tau \rho \|r(\hat{x}^k)\|^2 + \tau \rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle.
$$

(3.45)

Our aim now is to show that the right hand side of (3.45) is nonnegative at all times.

For this, consider the term $\tau \rho \langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle = \tau \rho \langle r(\hat{x}^k), \sum_i A_i(x_i^k - \hat{x}_i^k) \rangle$.

Each one of the summands in this term can be bounded below by considering

$$
\tau \rho \langle r(\hat{x}^k), A_i(x_i^k - \hat{x}_i^k) \rangle = \tau \rho \sum_{j=1}^m [r(\hat{x}^k)]_j[A_i(x_i^k - \hat{x}_i^k)]_j
$$

$$
\geq - \frac{1}{2} \sum_{j=1}^m \left( \rho \left[ A_i(x_i^k - \hat{x}_i^k) \right]_j^2 + \tau^2 \rho [r(\hat{x}^k)]_j^2 \right),
$$

where $[\cdot]_j$ denotes the $j$-th entry of a vector. Note, however, that some of the rows of $A_i$ might be zero. If $[A_i]_j = 0$, then it follows that $[r(\hat{x}^k)]_j[A_i(x_i^k - \hat{x}_i^k)]_j = 0$.
\[ \hat{x}_i^k \]_j = 0. Hence, denoting the set of nonzero rows of \( \mathbf{A}_i \) as \( \mathcal{Q}_i \), i.e., \( \mathcal{Q}_i = \{ j = 1, \ldots, m : [\mathbf{A}_i]_j \neq 0 \} \), we can obtain a tighter lower bound for each term
\[ \tau \rho \left\langle r(\hat{x}^k), \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right\rangle \]
as follows
\[ \tau \rho \left\langle r(\hat{x}^k), \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right\rangle \geq - \frac{1}{2} \sum_{j \in \mathcal{Q}_i} \left( \rho \left[ \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right]_j^2 + \tau^2 \rho \left[ r(\hat{x}^k) \right]_j^2 \right). \quad (3.46) \]

Now, recall from (2.2) that \( q \) denotes the maximum number of non-zero blocks \([\mathbf{A}_i]_j\) over all \( j \) (in other words, \( q \) is the maximum number of agents \( i \) that are involved in the constraint \( j \)). Then, summing inequality (3.46) over all \( i \), we observe that each quantity \([r(\hat{x}^k)]_j^2\) is included in the summation at most \( q \) times. This observation leads us to the bound
\[ \tau \rho \left\langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \right\rangle = \sum_i \tau \rho \left\langle r(\hat{x}^k), \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right\rangle \quad (3.47) \]
\[ \geq - \frac{\rho}{2} \sum_i \| \mathbf{A}_i(x_i^k - \hat{x}_i^k) \|^2 - \frac{\tau^2 q \rho}{2} \| r(\hat{x}^k) \|^2. \]

We substitute (3.47) back into (3.45) and arrive at
\[ \rho \sum_i \left\langle \mathbf{A}_i(x_i^k - x_i^*), \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right\rangle - \left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle \quad (3.48) \]
\[ \geq \frac{\rho}{2} \sum_i \| \mathbf{A}_i(x_i^k - \hat{x}_i^k) \|^2 + \rho (\tau - \frac{\tau^2 q}{2}) \| r(\hat{x}^k) \|^2. \]

Recall that until now we have disregarded the term \( L(\tilde{x}^k, \lambda^*) - L(x^*, \lambda^*) \) in (3.38) of Lemma 21. Reinstating this term in (3.48), we finally get
\[ L(\tilde{x}^k, \lambda^*) - L(x^*, \lambda^*) + \frac{\rho}{2} \sum_i \| \mathbf{A}_i(x_i^k - \hat{x}_i^k) \|^2 + \rho (\tau - \frac{\tau^2 q}{2}) \| r(\hat{x}^k) \|^2 \]
\[ \leq \rho \sum_i \left\langle \mathbf{A}_i(x_i^k - x_i^*), \mathbf{A}_i(x_i^k - \hat{x}_i^k) \right\rangle - \left\langle \hat{\lambda}^k - \lambda^*, r(\hat{x}^k) \right\rangle, \]
as required. \[\square\]
Now, we are ready to prove the key relation (3.37).

**Lemma 23.** Assume (A1)–(A3). Then, the following inequality holds:

\[
L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) \leq \frac{1}{2\tau}(\phi^k - \phi^{k+1}),
\]

(3.49)

where

\[
\phi^k = \rho \|A(x^k - x^k_*)\|^2 + \frac{1}{\rho} \|\tilde{\lambda}^k - \lambda^*\|^2,
\]

and

\[
\tilde{\lambda}^k = \lambda^k + \rho(1 - \tau)r(x^k).
\]

**Proof.** First, we show that the dual update step (3.6) in the ADAL method results in the update rule

\[
\tilde{\lambda}^{k+1} = \tilde{\lambda}^k + \tau \rho r(\hat{x}^k).
\]

(3.50)

for the variables \(\tilde{\lambda}^k\). Indeed, we have

\[
\lambda^{k+1} = \lambda^k + \tau \rho r(x^{k+1})
\]

\[
= \lambda^k + \tau \rho \left[(1 - \tau)r(x^k) + \tau r(\hat{x}^k)\right]
\]

\[
= \lambda^k + \tau \left[- (1 - \tau)\rho\left(r(\hat{x}^k) - r(x^k)\right) + \rho r(\hat{x}^k)\right]
\]

\[
= \lambda^k - (1 - \tau)\rho \tau \left(r(\hat{x}^k) - r(x^k)\right) + \tau \rho r(\hat{x}^k).
\]

Adding \((1 - \tau)\rho r(x^k)\) to both sides of the above equation and rearranging terms we obtain

\[
\lambda^{k+1} + (1 - \tau)\rho \left[r(x^k) + \tau \left(r(\hat{x}^k) - r(x^k)\right)\right]
\]

\[
= \lambda^k + (1 - \tau)\rho r(x^k) + \tau \rho r(\hat{x}^k).
\]

This is equivalent to

\[
\lambda^{k+1} + (1 - \tau)\rho r(x^{k+1}) = \lambda^k + (1 - \tau)\rho r(x^k) + \tau \rho r(\hat{x}^k),
\]
which is exactly the update rule (3.50). With that in mind, we have that

$$
\phi^k - \phi^{k+1} = \sum_{i=1}^{N} \rho \| \mathbf{A}_i (\mathbf{x}_i^k - \mathbf{x}_i^*) \|^2 + \frac{1}{\rho} \| \hat{\mathbf{x}}^k - \mathbf{\lambda}^* \|^2 \\
- \sum_{i=1}^{N} \rho \| \mathbf{A}_i (\mathbf{x}_i^{k+1} - \mathbf{x}_i^*) \|^2 - \frac{1}{\rho} \| \hat{\mathbf{x}}^{k+1} - \mathbf{\lambda}^* \|^2 \\
= 2\tau \left[ \rho \sum_i \langle \mathbf{A}_i (\mathbf{x}_i^k - \mathbf{x}_i^*), \mathbf{A}_i (\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \rangle - \langle \hat{\mathbf{\lambda}}^k - \mathbf{\lambda}^*, \mathbf{r}(\hat{\mathbf{x}}^k) \rangle \right] \\
- \tau^2 \left[ \sum_i \rho \| \mathbf{A}_i (\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k) \|^2 + \rho \| \mathbf{r}(\hat{\mathbf{x}}^k) \|^2 \right].
$$

Rearranging terms in the above equation we get that

$$
\rho \sum_i \langle \mathbf{A}_i (\mathbf{x}_i^k - \mathbf{x}_i^*), \mathbf{A}_i (\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \rangle - \langle \hat{\mathbf{\lambda}}^k - \mathbf{\lambda}^*, \mathbf{r}(\hat{\mathbf{x}}^k) \rangle \\
= \frac{1}{2\tau} (\phi^k - \phi^{k+1}) + \frac{\tau \rho}{2} \left( \sum_i \| \mathbf{A}_i (\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k) \|^2 + \| \mathbf{r}(\hat{\mathbf{x}}^k) \|^2 \right).
$$

From the relation (3.41) of Lemma 22, we can substitute the term \( \rho \sum_i \langle \mathbf{A}_i (\mathbf{x}_i^k - \mathbf{x}_i^*), \mathbf{A}_i (\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \rangle - \langle \hat{\mathbf{\lambda}}^k - \mathbf{\lambda}^*, \mathbf{r}(\hat{\mathbf{x}}^k) \rangle \) on the left hand side of the above equality, to arrive at

$$
L(\hat{\mathbf{x}}^k, \mathbf{\lambda}^*) - L(\mathbf{x}^*, \mathbf{\lambda}^*) + \frac{\rho}{2} \sum_i \| \mathbf{A}_i (\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \|^2 + \rho (\tau - \frac{\tau^2 \rho}{2}) \| \mathbf{r}(\hat{\mathbf{x}}^k) \|^2 \\
\leq \frac{1}{2\tau} (\phi^k - \phi^{k+1}) + \frac{\tau \rho}{2} \left( \sum_i \| \mathbf{A}_i (\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k) \|^2 + \| \mathbf{r}(\hat{\mathbf{x}}^k) \|^2 \right),
$$

or, equivalently, at

$$
L(\hat{\mathbf{x}}^k, \mathbf{\lambda}^*) - L(\mathbf{x}^*, \mathbf{\lambda}^*) + \frac{\rho(1 - \tau)}{2} \sum_i \| \mathbf{A}_i (\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \|^2 + \frac{\rho(\tau - \tau^2 \rho)}{2} \| \mathbf{r}(\hat{\mathbf{x}}^k) \|^2 \\
\leq \frac{1}{2\tau} (\phi^k - \phi^{k+1}).
$$
Since $0 < \tau < \frac{1}{q}$, we have that $1 - \tau > 0$ and $\tau - \tau^2 q > 0$. Thus, the term
\[
\frac{\rho(1-\tau)}{2} \sum_i \| A_i(x_i^k - \hat{x}^k_i) \|^2 + \frac{\rho(\tau-\tau^2 q)}{2} \| r(\hat{x}^k) \|^2
\]
is always nonnegative, which gives us
\[
L(\hat{x}^k, \lambda^*) - L(x^*, \lambda^*) \leq \frac{1}{2\tau}(\phi^k - \phi^{k+1}).
\]
as required.

We are now in a position to prove the $O(1/k)$ ergodic convergence rate of ADAL. The idea is to utilize (3.37) together with the properties of convex functions in order to obtain the required bound on the ergodic average of the primal variable sequence. The main result is stated in the following theorem.

**Theorem 24.** Assume (A1)–(A3). Let $y^k = \frac{1}{k} \sum_{p=0}^{k-1} \hat{x}^p$ denote the ergodic average of the primal variable sequence generated by ADAL up to iteration $k$. Then, for every iteration $k$, it holds that
\[
0 \leq L(y^k, \lambda^*) - L(x^*, \lambda^*) \leq \frac{1}{2k\tau} \phi^0. \tag{3.51}
\]

**Proof.** The first inequality $0 \leq L(y^k, \lambda^*) - L(x^*, \lambda^*)$ follows directly from the definition of a saddle point. To prove the other inequality in (3.51) we proceed as follows. Summing (3.49) for all $p = 0, 1, \ldots, k-1$, we get
\[
\sum_{p=0}^{k-1} F(\hat{x}^p) + \sum_{p=0}^{k-1} \langle \lambda^*, r(\hat{x}^p) \rangle - \sum_{p=0}^{k-1} F(x^*) \leq \frac{1}{2\tau}(\phi^0 - \phi^k). \tag{3.52}
\]
By the convexity of $F$, we have that
\[
\sum_{p=0}^{k-1} \frac{1}{k} F(\hat{x}^p) \geq F\left( \sum_{p=0}^{k-1} \frac{1}{k} \hat{x}^p \right),
\]
which implies that
\[
\sum_{p=0}^{k-1} F(\hat{x}^p) \geq kF(y^k).
\]
The analogous relation holds for $\sum_{p=0}^{k-1} r(\hat{x}^p) \geq k r(y^k)$, since it is a linear (convex) mapping. Moreover, we have that $\sum_{p=0}^{k-1} F(x^*) = k F(x^*)$. Hence, (3.52) can be expressed as

$$k F(y^k) + k \langle \lambda^*, r(y^k) \rangle - k F(x^*) \leq \frac{1}{2\tau} \left( \phi^0 - \phi^k \right),$$

or, equivalently,

$$F(y^k) + \langle \lambda^*, r(y^k) \rangle - F(x^*) + \frac{1}{2k\tau} \phi^k \leq \frac{1}{2k\tau} \phi^0.$$

Since $\phi^k \geq 0$, we infer that $L(y^k, \lambda^*) - L(x^*, \lambda^*) \leq \frac{1}{2k\tau} \phi^0$, as required. 

3.4 Convergence for Non-convex Problems and Alternative Stepsize Choices

In this section we extend the theoretical analysis of ADAL to address non-convex settings. Under assumptions that are common in the study of non-convex optimization methods, we prove the convergence of ADAL to a local minimum of problem (3.1) when the local cost functions $f_i$ are possibly non-convex. To the best of our knowledge, this is the first published work that formally establishes the convergence of a distributed augmented Lagrangian method for non-convex optimization problems. There are some numerical experiments indicating that ADMM can converge in scenarios with non-convex objective functions; see Marchesini et al. (2013); Zhang and Kwok (2014); Forero et al. (2011); Shen et al. (2014) for some examples. However, there is no published theoretical guarantee for the convergence of ADMM on non-convex problems. In fact, in Section 4.4 we will discuss a very simple problem where a straightforward implementation of ADMM actually diverges, while the proposed ADAL scheme converges to the optimal solution.

Furthermore, in this section we propose a way to select the stepsizes used in the ADAL algorithm that is more general compared to the choice described in the
previous sections. Specifically, it was shown in Section 3.2 that ADAL converges to the optimal solution of (3.1) if the stepsizes satisfy a certain condition that requires knowledge of the global structure of the constraint set at initialization. Here, we lift this requirement for global information and, instead, define $m$ stepsizes associated with each one of the $m$ coupling constraints in (3.1); each stepsize must adhere to a condition that requires only local information from the corresponding constraint. It is worth noting that these two contributions are independent from each other, meaning that convergence of the non-convex ADAL method can still be shown using the stepsizes from Section 3.2, and, similarly, convergence of the convex ADAL method can be shown using the stepsizes proposed in the current section. The contents of this section are part of the work Chatzipanagiotis and Zavlanos (2015b) that is currently under review.

The problems of interest in the non-convex case are:

$$\min \sum_{i=1}^{N} f_i(x_i)$$

subject to $\sum_{i=1}^{N} A_i x_i = b$, (3.53)

where, for every $i \in \mathcal{I} = \{1, 2, \ldots, N\}$, the function $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is twice continuously differentiable, $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ denotes a nonempty closed, convex subset of $n_i$-dimensional Euclidean space, and $A_i$ is a matrix of dimension $m \times n_i$.

### 3.4.1 Alternative Stepsize Choices

An alternative structure for the ADAL algorithm that utilizes locally defined stepsizes $\tau_j$ for every constraint $j = 1, \ldots, m$ is given in Alg. 8. The local optimization computations in step 1 of the algorithm remain the same as in the previous version,
Algorithm 8 Accelerated Distributed Augmented Lagrangians (ADAL)

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$ and initial primal variables $x^1$.

1. For every $i \in \mathcal{I}$, determine $\hat{x}^k_i$ as the solution of the following problem:

$$
\min_{x_i \in \mathcal{X}_i} \Lambda^i_\rho(x_i, Ax^k_i, \lambda^k).
$$

(3.54)

2. Set for every $i \in \mathcal{I}$

$$
A_i x^{k+1}_i = A_i x^k_i + T \left( A_i \hat{x}^k_i - A_i x^k_i \right).
$$

(3.55)

3. Set:

$$
\lambda^{k+1} = \lambda^k + \rho T \left( \sum_{i=1}^{N} A_i x^{k+1}_i - b \right),
$$

(3.56)

increase $k$ by one and return to Step 1.

cf. Alg. 7. The differences are located in steps 2 and 3, and are explained in detail in what follows.

After the local optimization steps have been carried out, the second step consists of each agent updating its primal variables by taking a convex combination with the corresponding values from the previous iteration. This update now depends on a vector of stepsizes $\tau \in \mathbb{R}^m$, where each entry $\tau_j$ is the stepsize corresponding to constraint $j$. For notational purposes, we define the diagonal, square matrix $T$ of dimension $m$ according to

$$
T = \text{diag}(\tau_1, \ldots, \tau_m),
$$

(3.57)

so that the diagonal entries of $T$ are the stepsizes for each constraint. To select the appropriate values for $\tau$, we first need to define the degree of a constraint for problems of the form (3.53). Specifically, for each constraint $j = 1, \ldots, m$, let the degree $q_j$ denote the number of individual decision makers $i$ associated with this constraint. That is, $q_j$ is the number of all $i \in \mathcal{I}$ such that $[A_i]_j \neq 0$. Then, to guarantee the convergence of ADAL we need to select $\tau_j \in (0, \frac{1}{q_j})$, according to the
Note that, at the local update steps (3.55), each agent \( i \) does not update the primal variables \( x_i \), but rather the products \( A_i x_i^k \). Using a more rigorous notation, we could define an auxiliary variable \( y_i^k = A_i x_i^k \), so that the update (3.55) takes the form \( y_i^{k+1} = y_i^k + T(A_i \hat{x}_i^k - y_i^k) \). To avoid introducing additional notation, we have chosen not to introduce the variables \( y_i^k \) and, instead, we directly update the terms \( A_i x_i^k \), slightly abusing notation.

In the ADAL version presented in Section 3.1, the second step of the algorithm has the form

\[
x_i^{k+1} = x_i^k + \tau(x_i^k - x_i^k),
\]

i.e., the primal variables are updated directly, as opposed to updating the products \( A_i x_i^k \) in the current version of ADAL, cf. (3.55). Additionally, in the previous ADAL version, the stepsize \( \tau \) is a scalar that must satisfy \( \tau \in (0, \frac{1}{q}) \), where \( q = \max_{1 \leq j \leq m} q_j \).

Intuitively, \( q \) is the number of agents coupled in the “most populated” constraint of the problem. Obtaining the parameter \( q \) clearly requires global information of the structure of the constraint set at initialization, which may hinder the distributed nature of the algorithm. To remedy this problem, in the following section we show that the ADAL method still converges to an optimal solution of problem (3.53) using the update step (3.55) with a vector stepsize \( T \) that can be locally determined. To see why (3.55) requires only local information, note that every agent \( i \) needs to know only the \( q_j \)‘s that correspond to the constraints that this agent is involved in. Analogous arguments hold for the dual update step (3.56), also.

### 3.4.2 Convergence for Non-convex Problems

In order to prove convergence of ADAL to a local minimum of (3.53), we need the following four assumptions:
(A1) The sets $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$, $i = 1, \ldots, N$ are nonempty, closed and convex.

(A2) The functions $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $i \in I = \{1, 2, \ldots, N\}$ are twice continuously differentiable on $\mathcal{X}_i$.

(A3) The subproblems (3.54) are solvable.

(A4) There exists a point $\mathbf{x}^*$ satisfying the strong second order sufficient conditions of optimality for problem (3.53) with Lagrange multipliers $\mathbf{\lambda}^*$.

The assumptions (A1), (A2), and (A4) are common and are used in the convergence proof of the standard augmented Lagrangian method (ALM) for non-convex optimization problems, cf. Ruszczyński (2006). Assumption (A4) implies that there exist lagrange multipliers $\mathbf{\lambda}^* \in \mathbb{R}^m$ that satisfy the first order optimality conditions for problem (3.53) at the feasible point $\mathbf{x}^*$, provided that a constraint qualification condition is satisfied at $\mathbf{x}^*$, i.e.,

$$\nabla F(\mathbf{x}^*) + A^\top \mathbf{\lambda}^* \in \mathcal{N}_{\mathcal{X}}(\mathbf{x}^*),$$

where we recall that $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top]^\top \in \mathbb{R}^n$, $F(\mathbf{x}) = \sum_i f_i(\mathbf{x}_i)$, and $A = [A_1 \ldots A_N] \in \mathbb{R}^{m \times n}$. Here, we use $\mathcal{N}_{\mathcal{X}}(\mathbf{x})$ to denote the normal cone to the set $\mathcal{X}$ at point $\mathbf{x}$, see also Def. 6,

$$\mathcal{N}_{\mathcal{X}}(\mathbf{x}) = \{ \mathbf{h} \in \mathbb{R}^n : \langle \mathbf{h}, \mathbf{y} - \mathbf{x} \rangle \leq 0, \quad \forall \mathbf{y} \in \mathcal{X} \}.$$

The strong second order sufficient conditions of optimality for problem (3.53) at a point $\mathbf{x}^*$ imply that

$$\langle \mathbf{s}, \nabla^2 F(\mathbf{x}^*) \mathbf{s} \rangle > 0, \quad \text{for all } \mathbf{s} \neq \mathbf{0}, \quad \text{such that } A \mathbf{s} = \mathbf{0},$$


Assumption (A3) is satisfied if for every $i = 1, \ldots, N$, either the set $\mathcal{X}_i$ is compact, or the function $f_i(\mathbf{x}_i) + \frac{\xi}{2} \| A_i \mathbf{x}_i - \mathbf{c} \|^2$ is inf-compact for any vector $\mathbf{c}$. The latter
condition, means that the level sets of the function are compact sets, implying that the set \( \{ x_i \in X_i : f_i(x_i) + \frac{\rho}{2} \| A_i x_i - c \|^2 \leq \alpha \} \) is compact for any \( \alpha \in \mathbb{R} \).

Define the residual \( r(x) \in \mathbb{R}^m \) as the vector containing the amount of all constraint violations with respect to primal variable \( x \), i.e.,

\[
  r(x) = \sum_{i=1}^{N} A_i x_i - b.
\]  

(3.58)

Define also the auxiliary variables

\[
  \hat{\lambda}^k = \lambda^k + \rho r(\hat{x}^k),
\]  

(3.59)

and

\[
  \bar{\lambda}^k = \lambda^k + \rho(I - T)r(x^k),
\]  

(3.60)

where \( I \) is the identity matrix of size \( m \).

The basic idea to show convergence of our method is to introduce the Lyapunov (merit) function

\[
  \phi(x^k, \lambda^k) = \sum_{i=1}^{N} \rho \left\| A_i x_i^k - A_i x_i^* \right\|_{T^{-1}}^2 + \frac{1}{\rho} \left\| \hat{\lambda}^k - \lambda^* \right\|_{T^{-1}}^2 ,
\]  

(3.61)

where we use the notation \( \|x\|_M = \sqrt{x^T M x} \). We will show in Theorem 30 that this Lyapunov function is strictly decreasing during the execution of the ADAL algorithm (3.54)-(3.56), given that the stepsizes \( \tau_j \) satisfy the condition \( 0 < \tau_j < 1/q_j \) for all \( j = 1, \ldots, m \). Then, in Theorem 32 we show that the strictly decreasing property of the Lyapunov function (3.61) implies the convergence of the primal and dual variables to their respective optimal values defined at a local minimum of problem (3.53).

We begin the proof by utilizing the first order optimality conditions of all the subproblems (3.54) in order to derive some necessary inequalities.
Lemma 25. Assume (A1)–(A4). Then, the following inequality holds:

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^T (x_i^k - x_i^*) + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^T (\lambda^k - \hat{\lambda}^k) \geq 0
\]  \hspace{1cm} (3.62)

where \( \lambda^k, \hat{\lambda}^k, \hat{x}_i^k, \) and \( x_j^k \) are calculated at iteration \( k \).

Proof. The first order optimality conditions for problem (3.54) imply the following inclusion for the minimizer \( \hat{x}_i^k \)

\[
0 \in \nabla f_i(\hat{x}_i^k) + A_i^T \lambda^k + \rho A_i^T \left( \hat{A}_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) + N_{\mathcal{X}_i}(\hat{x}_i^k). \]  \hspace{1cm} (3.63)

We infer that there exist normal elements \( z_i^k \in N_{\mathcal{X}_i}(\hat{x}_i^k) \) such that we can express (3.63) as follows:

\[
0 = \nabla f_i(\hat{x}_i^k) + A_i^T \lambda^k + \rho A_i^T \left( \hat{A}_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) + z_i^k. \]  \hspace{1cm} (3.64)

Taking inner product with \( x_i^* - \hat{x}_i^k \) on both sides of this equation and using the definition of a normal cone, we obtain

\[
\langle \nabla f_i(\hat{x}_i^k) + A_i^T \lambda^k + \rho A_i^T \left( \hat{A}_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right), x_i^* - \hat{x}_i^k \rangle = \langle -z_i^k, x_i^* - \hat{x}_i^k \rangle \geq 0.
\]  \hspace{1cm} (3.65)

Using the variables \( \hat{\lambda}^k \) defined in (3.59), we substitute \( \lambda^k \) in (3.65) and obtain:

\[
0 \leq \langle \nabla f_i(\hat{x}_i^k) + A_i^T \left[ \hat{\lambda}^k - \rho \left( \sum_{j} A_j \hat{x}_j^k - b \right) + \rho \left( \hat{A}_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) \right] \rangle, x_i^* - \hat{x}_i^k \rangle
\]

\[
= \langle \nabla f_i(\hat{x}_i^k) + A_i^T \left[ \hat{\lambda}^k + \rho \left( \sum_{j \neq i} A_j x_j^k - \sum_{j \neq i} A_j \hat{x}_j^k \right) \right], x_i^* - \hat{x}_i^k \rangle
\]  \hspace{1cm} (3.66)
The assumption (A4) entails that the following first-order optimality conditions are satisfied at the point \((x^*, \lambda^*)\), i.e.,

\[
0 \in \nabla f_i(x^*_i) + A_i^\top \lambda^* + \mathcal{N}_{\mathcal{X}}(x^*_i) \quad \text{for all } i = 1, \ldots, N. \tag{3.67}
\]

After using the definition of the normal cone and taking inner product with \(\hat{x}_i^k - x_i^*\) on both sides of this equation (as before), we obtain the equivalent expression for the above inclusion

\[
\left\langle \nabla f_i(x^*_i) + A_i^\top \lambda^* , \hat{x}_i^k - x_i^* \right\rangle \geq 0, \quad \text{for all } i = 1, \ldots, N. \tag{3.68}
\]

Adding together (3.66) and (3.68), we obtain the following inequalities for all \(i = 1, \ldots, N\):

\[
\left\langle \nabla f_i(x^*_i) - \nabla f_i(\hat{x}_i^k) + A_i^\top (\lambda^* - \hat{\lambda}^k) - \rho A_i^\top \left( \sum_{j \neq i} A_j x_j^k - \sum_{j \neq i} A_j \hat{x}_j^k \right), \hat{x}_i^k - x_i^* \right\rangle \geq 0.
\]

Adding the inequalities for all \(i = 1, \ldots, N\) and rearranging terms, we get:

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top (\hat{x}_i^k - x_i^*) + \left( \lambda^* - \hat{\lambda}^k \right)^\top \left( \sum_i A_i \hat{x}_i^k - A_i x_i^* \right) \geq \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( \sum_{j \neq i} (A_j x_j^k - A_j \hat{x}_j^k) \right).
\]

Substituting \(\sum_{i=1}^N A_i x_i^* = b\) and \(\sum_{i=1}^N A_i \hat{x}_i^k - b = \frac{1}{\rho} (\hat{\lambda}^k - \lambda^k)\) from (3.59), we conclude that

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top (\hat{x}_i^k - x_i^*) + \frac{1}{\rho} (\hat{\lambda}^k - \lambda^*)^\top (\lambda^k - \hat{\lambda}^k) \geq \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( \sum_{j \neq i} (A_j x_j^k - A_j \hat{x}_j^k) \right),
\]

as required. \(\square\)
Lemma 26. Under assumptions (A1)–(A4), the following relation holds:

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) \\
+ \frac{1}{\rho} \left( \lambda^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) 
\geq \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( A_i \hat{x}_i^k - A_i x_i^* \right) + \rho \rho \sum_j \left( A_j x_j^k - A_j x_j^* \right)^\top \left( A_j x_j^k - A_j \hat{x}_j^k \right),
\]

(3.69)

Proof. Consider the result of Lemma 25 and add the term \( \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) \) to both sides of inequality (3.69), which gives us

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) \\
+ \frac{1}{\rho} \left( \lambda^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) 
\geq \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( \sum_j (A_j x_j^k - A_j \hat{x}_j^k) \right) + \rho \rho \sum_j \left( A_j \hat{x}_j^k - A_j x_j^* \right)^\top \left( A_j x_j^k - A_j \hat{x}_j^k \right),
\]

Grouping the terms at the right-hand side of the inequality by their common factor, we transform the estimate as follows:

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) \\
+ \frac{1}{\rho} \left( \lambda^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) 
\geq \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \sum_j \left( A_j x_j^k - A_j \hat{x}_j^k \right),
\]

Recall that \( \sum_j A_j (x_j^k - \hat{x}_j^k) = r(x^k) - r(\hat{x}^k) \), which means that this term is a constant factor with respect to the summation over \( i \) in the right hand side of the previous relation. Moreover, \( \sum_i A_i \hat{x}_i^k - \sum_i A_i x_i^* = \sum_i A_i \hat{x}_i^k - b = r(\hat{x}^k) \). Substituting these
terms at the right-hand side of the previous relation, gives us

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i \hat{x}_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right)
\]

\[
+ \frac{1}{\rho} \left( \hat{\lambda}^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) \geq \rho r(\hat{x}^k)^\top \left( r(x^k) - r(\hat{x}^k) \right) = \left( \hat{\lambda}^k - \lambda^k \right)^\top \left( r(x^k) - r(\hat{x}^k) \right).
\]

Next, we substitute the expressions

\[
(A_i \hat{x}_i^k - A_i x_i^k) = (A_i x_i^k - A_i x_i^*) + (A_i \hat{x}_i^k - A_i x_i^*)
\]

and \( \hat{\lambda}^k - \lambda^* = (\lambda^k - \lambda^*) + (\hat{\lambda}^k - \lambda^k) \),

in the left-hand side of (3.70). We obtain

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right)
\]

\[
+ \frac{1}{\rho} \left( \hat{\lambda}^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) \geq \sum_i \rho \left\| A_i (x_i^k - \hat{x}_i^k) \right\|^2 + \frac{1}{\rho} \left\| \hat{\lambda}^k - \lambda^k \right\|^2 + \left( \hat{\lambda}^k - \lambda^k \right)^\top \left( r(x^k) - r(\hat{x}^k) \right).
\]

which completes the proof. \( \square \)

In the next lemma, we obtain a modified version of (3.69) whose right-hand side is nonnegative.

**Lemma 27.** Under the assumptions \((A1)-(A4)\), the following relation holds

\[
\sum_i \left( \nabla f_i(x_i^*) - \nabla f_i(\hat{x}_i^k) \right)^\top \left( \hat{x}_i^k - x_i^* \right) + \rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right)
\]

\[
+ \frac{1}{\rho} \left( \hat{\lambda}^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) \geq \rho \sum_i \left\| A_i (x_i^k - \hat{x}_i^k) \right\|^2 + \rho \left\| r(\hat{x}^k) \right\|^2_{T - \frac{1}{2}D}.
\]

73
Substituting back in (3.72), we obtain:

$$
\frac{1}{\rho} \left( \rho (I - T) r(x^k) \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) = \rho \left( (I - T) r(x^k) \right)^\top \left( - r(\hat{x}^k) \right),
$$

to both sides of inequality (3.69). Recalling the definition of $\bar{\lambda}^k$ from (3.60), we get:

$$
\rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) + \frac{1}{\rho} \left( \lambda^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) \\
\quad \geq \rho \sum_i \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \rho \| r(\hat{x}^k) \|^2 \\
\quad + \left( \lambda^k - \lambda^* \right)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho \left( (I - T) r(x^k) \right)^\top r(\hat{x}^k).
$$

Consider the term $\left( \lambda^k - \lambda^* \right)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho \left( (I - T) r(x^k) \right)^\top r(\hat{x}^k)$ in the right hand side of (3.72). We manipulate it to get:

$$
\left( \lambda^k - \lambda^* \right)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho \left( (I - T) r(x^k) \right)^\top r(\hat{x}^k) =
$$

$$
= \rho r(\hat{x}^k)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho \left( (I - T) r(x^k) \right)^\top r(\hat{x}^k)
$$

$$
= \rho r(\hat{x}^k)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho \left( (I - T) \left[ r(x^k) - r(\hat{x}^k) + r(\hat{x}^k) \right] \right)^\top r(\hat{x}^k)
$$

$$
= \rho \left( T r(\hat{x}^k) \right)^\top \left( r(x^k) - r(\hat{x}^k) \right) - \rho r(\hat{x}^k)^\top (I - T) r(\hat{x}^k)
$$

$$
= \rho \left( T r(\hat{x}^k) \right)^\top \left( \sum_i A_i (x_i^k - \hat{x}_i^k) \right) - \rho r(\hat{x}^k)^\top (I - T) r(\hat{x}^k).
$$

Substituting back in (3.72), we obtain:

$$
\rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) + \frac{1}{\rho} \left( \lambda^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) \\
\quad \geq \rho \sum_i \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \rho r(\hat{x}^k)^\top T r(\hat{x}^k) + \rho \left( T r(\hat{x}^k) \right)^\top \left( \sum_i A_i (x_i^k - \hat{x}_i^k) \right).
$$
Each of the terms $\rho \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i) \right)$ in the right hand side of (3.74) can be bounded below by considering

$$\rho \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i\mathbf{x}^k_i - \mathbf{A}_i\mathbf{\hat{x}}^k_i \right) = \sum_{j=1}^{m} \left( \tau_j [\mathbf{r}(\mathbf{x}^k)]_j \right) \left( [\mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i)]_j \right)$$

$$\geq -\frac{\rho}{2} \sum_{j=1}^{m} \left( \left[ \mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i) \right]_j^2 + \tau_j^2 [\mathbf{r}(\mathbf{x}^k)]_j^2 \right) ,$$

where $[\cdot]_j$ denotes the $j$-th entry of a vector. Note, however, that some of the rows of $\mathbf{A}_i$ might be zero. If $[\mathbf{A}_i]_j = \mathbf{0}$, then it follows that $[\mathbf{r}(\mathbf{x}^k)]_j [\mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i)]_j = 0$. Hence, denoting the set of nonzero rows of $\mathbf{A}_i$ as $\mathcal{Q}_i$, i.e., $\mathcal{Q}_i = \{ j = 1, \ldots, m : [\mathbf{A}_i]_j \neq \mathbf{0} \}$, we can obtain a tighter lower bound for each term $\rho \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i\mathbf{x}^k_i - \mathbf{A}_i\mathbf{\hat{x}}^k_i \right)$ as

$$\rho \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i\mathbf{x}^k_i - \mathbf{A}_i\mathbf{\hat{x}}^k_i \right) \geq -\frac{\rho}{2} \sum_{j \in \mathcal{Q}_i} \left( \left[ \mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i) \right]_j^2 + \tau_j^2 [\mathbf{r}(\mathbf{x}^k)]_j^2 \right).$$

(3.75)

Now, recall that $q_j$ denotes the number of non-zero blocks $[\mathbf{A}_i]_j$ over all $i = 1, \ldots, N$, in other words, $q_j$ is the number of decision makers $i$ that are involved in the constraint $j$. Then, summing inequality (3.75) over all $i$, we observe that each term $\tau_j^2 [\mathbf{r}(\mathbf{x}^k)]_j^2$ is included in the summation at most $q_j$ times.

This observation leads us to the bound

$$\rho \sum_i \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i\mathbf{x}^k_i - \mathbf{A}_i\mathbf{\hat{x}}^k_i \right) \geq -\frac{\rho}{2} \left( \sum_i \| \mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i) \|^2 + \sum_{j=1}^{m} q_j \tau_j^2 [\mathbf{r}(\mathbf{x}^k)]_j^2 \right) ,$$

or, equivalently,

$$\rho \sum_i \left( \mathbf{T} \mathbf{r}(\mathbf{x}^k) \right)^\top \left( \mathbf{A}_i\mathbf{x}^k_i - \mathbf{A}_i\mathbf{\hat{x}}^k_i \right) \geq -\frac{\rho}{2} \sum_i \| \mathbf{A}_i(\mathbf{x}^k_i - \mathbf{\hat{x}}^k_i) \|^2 - \frac{\rho}{2} \mathbf{r}(\mathbf{x}^k)^\top \mathbf{D} \mathbf{r}(\mathbf{x}^k) ,$$

(3.76)
where $\mathbf{D} = \text{diag}(q_1 \tau_1^2, \ldots, q_m \tau_m^2)$. Finally, we substitute (3.76) into (3.74) to get

$$
\rho \sum_i \left( \mathbf{A}_i \mathbf{x}_i^k - \mathbf{A}_i \mathbf{x}_i^\ast \right)^\top \left( \mathbf{A}_i \mathbf{x}_i^k - \mathbf{A}_i \mathbf{x}_i^k \right) + \frac{1}{\rho} \left( \mathbf{\lambda}^k - \mathbf{\lambda}^\ast \right)^\top \left( \mathbf{\lambda}^k - \mathbf{\lambda}^k \right)
$$

\[
\geq \frac{\rho}{2} \sum_i \| \mathbf{A}_i (\mathbf{x}_i^k - \mathbf{x}_i^k) \|^2 + \rho \| r(\mathbf{x}^k) \|^2_{T - \frac{1}{2} \mathbf{D}}.
\]

After reinstating the gradient terms that we have neglected thus far, we obtain the required result.

Next, our goal is to find a lower bound for the gradient terms appearing in (3.71). To do so, let $\mathbf{C}$ denote any diagonal matrix with strictly positive diagonal entries, and consider the function

$$
G_\rho(\mathbf{x}) = F(\mathbf{x}) + \frac{\rho}{2} \mathbf{x}^T \mathbf{A}^\top \mathbf{C} \mathbf{A} \mathbf{x},
$$

where we recall that $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top]^\top \in \mathbb{R}^n$, $F(\mathbf{x}) = \sum_i f_i(\mathbf{x}_i)$, and $\mathbf{A} = [\mathbf{A}_1 \ldots \mathbf{A}_N] \in \mathbb{R}^{m \times n}$. In the next lemmas, we will make use of the fact that, for sufficiently large $\rho$, the function $G_\rho(\mathbf{x})$ is strongly convex in a neighborhood around the optimal solution $\mathbf{x}^\ast$ of (3.53). For this, we will make use of the following result.

**Lemma 28** (Ruszczynski (2006), Lemma 4.28). Assume that a symmetric matrix $\mathbf{Q}$ of dimension $n$ and a matrix $\mathbf{B}$ of dimension $m \times n$ are such that

$$
\langle \mathbf{x}, \mathbf{Q} \mathbf{x} \rangle > 0, \quad \text{for all } \mathbf{x} \neq \mathbf{0} \quad \text{such that } \mathbf{B} \mathbf{x} = \mathbf{0}.
$$

Then, there exists $\rho_0$ such that for all $\rho > \rho_0$ the matrix $\mathbf{Q} + \rho \mathbf{B}^\top \mathbf{B}$ is positive definite.

Using Lemma 28, we can obtain an important relation involving the gradient terms that appear in (3.71).
Lemma 29. Assume \((A1)-(A4)\). Then, for any diagonal matrix \(C\) with strictly positive diagonal entries, the following relation holds
\[
\rho \sum_i \left( A_i x_i^k - A_i x_i^\ast \right)^\top \left( A_i x_i^k - A_i \hat{x}_i^k \right) + \frac{1}{\rho} \left( \hat{\lambda}^k - \lambda^\ast \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) 
\geq \frac{\rho}{2} \sum_i \| A_i (x_i^k - \hat{x}_i^k) \|^2 + \rho \| r(\hat{x}^k) \|^2_{T^{-1/2}D^{-1}C} + \kappa \| \hat{x}_i^k - x^\ast \|^2,
\]
provided that \(\rho\) is sufficiently large, and that for all iterations \(k\) the terms \(\lambda^k + \rho \sum_{j \neq i} (A_j x_j^k - A_j x_j^\ast)\) are sufficiently close to \(\lambda^\ast\) for all \(i = 1, \ldots, N\).

Proof. From assumption \((A4)\), we have that there exists a point \(x^\ast\) satisfying the strong second order sufficient conditions of optimality for problem (3.53). These conditions imply that
\[
\langle s, \nabla^2 F(x^\ast) s \rangle > 0, \quad \text{for all } s \neq 0, \text{ such that } As = 0.
\]

Now, combine this with the result of Lemma 28 for
\[
Q = \nabla^2 F(x^\ast), \quad \text{and} \quad B = C^{1/2}A.
\]

It follows that there exists \(\rho_0\) such that for all \(\rho > \rho_0\) the matrix \(\nabla^2 F(x^\ast) + \rho A^\top CA\) is positive definite. From assumption \((A2)\) this matrix is also continuous, hence we can choose sufficiently large \(\rho\) such that, for all \(x \in \mathcal{X}\) sufficiently close to \(x^\ast\), i.e., for \(\|x - x^\ast\| \leq \delta\), all the eigenvalues of \(\nabla^2 F(x^\ast) + \rho A^\top CA\) lie above some \(\kappa > 0\).

Since the positive definite matrix \(\nabla^2 F(x) + \rho A^\top CA\) is the Hessian of the function \(G_\rho(x) = F(x) + \frac{\rho}{2} x^T A^\top CA x\) and \(\mathcal{X}\) is a convex closed set, we infer that, for sufficiently large \(\rho\), there exists some \(\delta\) such that the function \(G_\rho(x)\) is strongly convex with modulus \(\kappa\) for every \(x\) belonging in the set \(\{ x \in \mathcal{X} : \|x - x^\ast\| \leq \delta\}\). From the definition of strongly convex functions, we get that the following holds for all \(x\) that are sufficiently close to \(x^\ast\):
\[
\left( \nabla G_\rho(x) - \nabla G_\rho(x^\ast) \right)^\top (x - x^\ast) \geq \kappa \|x - x^\ast\|^2.
\]
For the term \( (\nabla G_{\rho}(x) - \nabla G_{\rho}(x^*))^T (x - x^*) \), we have

\[
(\nabla G_{\rho}(x) - \nabla G_{\rho}(x^*))^T (x - x^*)
\]

\[
= (\nabla F(x) - \nabla F(x^*) + \rho A^T CX - \rho A^T CAx^*)^T (x - x^*)
\]

\[
= (\nabla F(x) - \nabla F(x^*) + \rho A^T C(Ax - b))^T (x - x^*)
\]

\[
= \sum_i \left( \nabla f_i(x_i) - \nabla f_i(x_i^*) \right)^T (x_i - x_i^*) + \rho r(x)^T Cr(x)
\]

where we have used the fact that \( Ax^* = b \), and \( r(x) = Ax - b \). It follows that

\[
\sum_i \left( \nabla f_i(x_i) - \nabla f_i(x_i^*) \right)^T (x_i - x_i^*) + \rho r(x)^T Cr(x) \geq \kappa \|x - x^*\|^2. \tag{3.79}
\]

Now, substitute \( x = \hat{x}^k \) in (3.79), and add it to (3.71). We get the following relation

\[
\rho \sum_i \left( A_i x_i^k - A_i x_i^k \right)^T \left( A_i x_i^k - A_i \hat{x}_i^k \right) + \frac{1}{\rho} \left( \lambda^k - \hat{\lambda}^k \right)^T \left( \lambda^k - \hat{\lambda}^k \right)
\]

\[
\geq \frac{\rho}{2} \sum_i \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \rho r(\hat{x}^k)^T \left( T - \frac{1}{2} D - C \right) r(\hat{x}^k) + \kappa \|\hat{x}^k - x^*\|^2,
\]

which is the required result.

Note that, in order to substitute \( x = \hat{x}^k \) in (3.79), it is necessary that the \( \hat{x}^k \) are sufficiently close to \( x^* \) at iteration \( k \), i.e., that they belong to the set \( \{ \hat{x} \in \mathcal{X} : \| \hat{x} - x^* \| \leq \delta \} \). To see when this condition holds, note that the local AL for each \( i \) can be expressed as

\[
f_i(x_i) + \left< \lambda^k + \rho \sum_{j \neq i} (A_j x_j^k - A_j x_j^*), A_i x_i \right> + \frac{\rho}{2} \|A_i x_i + \sum_{j \neq i} A_j x_j^* - b\|^2 \tag{3.80}
\]

\[
+ \rho \left\langle \sum_{j \neq i} (A_j x_j^k - A_j x_j^*), \sum_{j \neq i} A_j x_j^* - b \right\rangle + \frac{\rho}{2} \left\| \sum_{j \neq i} (A_j x_j^k - A_j x_j^*) \right\|^2,
\]

where we have added the zero terms \( \sum_{j \neq i} A_j x_j^* - \sum_{j \neq i} A_j x_j^* \) in the penalty term of the AL, and expanded it. The last two terms can be disregarded when minimizing with
respect to $x_i$. Recalling a well known result on the sensitivity analysis of augmented Lagrangians, c.f. Bertsekas (1982); Shapiro and Sun (2004); Ruszczyński (2006), we have that, given assumptions (A1)–(A4), the $\hat{x}_i^k$ will be sufficiently close to $x_i^*$ if $\rho$ is sufficiently large and the terms $\lambda^k + \rho \sum_{j \neq i}(A_j x_j^k - A_j x_j^*)$ are sufficiently close to $\lambda^*$ for all $i = 1, \ldots, N$.

We are now ready to prove the key result pertaining to the convergence of our method. We will show that the function $\phi$ defined in (3.61) is a strictly decreasing Lyapunov function for ADAL. The results from Lemmas 27 and 29 will help us characterize the decrease of $\phi$ at each iteration.

**Theorem 30.** Assume (A1)–(A4). Assume also that $\rho$ is sufficiently large, and that the initial iterates $x^1, \lambda^1$ are chosen such that $\phi(x^1, \lambda^1)$ is sufficiently small. If the ADAL method uses stepsizes $\tau_j$ satisfying

$$0 < \tau_j < \frac{1}{q_j}, \quad \forall \ j = 1, \ldots, m,$$

then, the sequence $\{\phi(x^k, \lambda^k)\}$, with $\phi(x^k, \lambda^k)$ defined in (3.61), is strictly decreasing.

**Proof.** First, we show that the dual update step (3.56) in the ADAL method results in the following update rule for the variables $\bar{\lambda}_k$, which are defined in (3.60):

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \rho \text{Tr}(\hat{x}^k)$$

(3.81)
Indeed,

\[ \lambda^{k+1} = \lambda^k + \rho \text{Tr}(x^{k+1}) \]

\[ \lambda^{k+1} + \rho r(x^{k+1}) = \lambda^k + \rho \text{Tr}(x^{k+1}) + \rho r(x^{k+1}) \]

\[ \lambda^{k+1} + \rho (I - T)r(x^{k+1}) = \lambda^k + \rho r(x^{k+1}) \]

\[ \hat{\lambda}^{k+1} = \lambda^k + \rho (I - T)r(x^k) \]

\[ \hat{\lambda}^{k+1} = \hat{\lambda}^k + \rho \text{Tr}(\hat{x}^k) \]

as required.

We define the progress at each iteration \( k \) of the ADAL method as

\[ \theta_k(\tau) = \phi(x^k, \lambda^k) - \phi(x^{k+1}, \lambda^{k+1}). \]

We substitute \( \hat{\lambda}^k \) in the formula for calculating the function \( \phi \) and use relation (3.81).

The progress \( \theta_k(\tau) \) can be evaluated as follows:

\[
\theta_k(\tau) = \sum_{i=1}^{N} \rho \left\| A_i(x_i^k - x_i^*) \right\|_{T^{-1}}^2 + \frac{1}{\rho} \left\| \lambda^k - \lambda^* \right\|_{T^{-1}}^2 - \sum_{i=1}^{N} \rho \left\| A_i(x_i^{k+1} - x_i^*) \right\|_{T^{-1}}^2 - \frac{1}{\rho} \left\| \hat{\lambda}^{k+1} - \lambda^* \right\|_{T^{-1}}^2. \tag{3.82}
\]

First, consider the term \( \left\| A_i(x_i^{k+1} - x_i^*) \right\|_{T^{-1}}^2 \). We have that

\[
\rho \left\| A_i(x_i^{k+1} - x_i^*) \right\|_{T^{-1}}^2 = \rho \left( A_i x_i^{k+1} - A_i x_i^* \right)^\top T^{-1} \left( A_i x_i^{k+1} - A_i x_i^* \right) \]

\[
= \rho \left( A_i x_i^k - A_i x_i^* + T(A_i x_i^k - A_i x_i^k) \right)^\top T^{-1} \left( A_i x_i^k - A_i x_i^* + T(A_i x_i^k - A_i x_i^k) \right) \]

\[
= \rho \left( A_i x_i^k - A_i x_i^* \right)^\top T^{-1} \left( A_i x_i^k - A_i x_i^* \right) \tag{3.83}
\]

\[
+ \rho \left( T(A_i x_i^k - A_i x_i^k) \right)^\top T^{-1} \left( T(A_i x_i^k - A_i x_i^k) \right) \]

\[
+ 2\rho \left( A_i x_i^k - A_i x_i^* \right)^\top T^{-1} \left( T(A_i x_i^k - A_i x_i^k) \right) \]

\[
= \rho \left\| A_i(x_i^k - x_i^*) \right\|_{T^{-1}}^2 + \rho \left\| \dot{A}_i x_i^k - A_i x_i^k \right\|_{T}^2 + 2\rho \left( A_i x_i^k - A_i x_i^* \right)^\top T^{-1} \left( A_i x_i^k - A_i x_i^* \right). \]
Similarly, for the term \( \| \hat{\lambda}^{k+1} - \lambda^* \|_{T^{-1}}^2 \), we have

\[
\frac{1}{\rho} \left( \frac{1}{\lambda^{k+1} - \lambda^*} \right)^\top T^{-1} \left( \hat{\lambda}^{k+1} - \lambda^* \right)
= \frac{1}{\rho} \left( \hat{\lambda}^k - \lambda^* + \rho \text{Tr}(\hat{x}^k) \right)^\top T^{-1} \left( \hat{\lambda}^k - \lambda^* + \rho \text{Tr}(\hat{x}^k) \right)
= \frac{1}{\rho} \| \hat{\lambda}^k - \lambda^* \|_{T^{-1}}^2 + \rho \| \text{Tr}(\hat{x}^k) \|_T^2 + \frac{2}{\rho} \left( \hat{\lambda}^k - \lambda^* \right)^\top \rho \text{Tr}(\hat{x}^k).
\]

Hence, substituting (3.83) and (3.84) into (3.82), and recalling that \( \rho \hat{x}^k = \rho \hat{x}^k \) we get that the progress \( \theta_k(\tau) \) at each iteration is given by

\[
\theta_k(\tau) = 2\rho \sum_i \left( A_i x_i^k - A_i x_i^* \right)^\top \left( A_i x_i^k - A_i x_i^* \right) + \frac{2}{\rho} \left( \hat{\lambda}^k - \lambda^* \right)^\top \left( \lambda^k - \hat{\lambda}^k \right) - \rho \sum_i \| A_i \hat{x}_i^k - A_i x_i^k \|_T^2 - \rho \| \text{Tr}(\hat{x}^k) \|_T^2.
\]

The last two (quadratic) terms in (3.85) are always negative, due to \( T \) being positive definite by construction. Hence, in order to show that \( \phi \) is strictly decreasing, we need to show that the first two terms in (3.85) are always “more positive” than the last two terms. This is exactly what Lemma 29 and (3.78) enable us to do. In particular, using (3.78), we obtain a lower bound for the first two terms in (3.85), which gives us that

\[
\theta_k(\tau) \geq \rho \sum_i \| A_i \hat{x}_i^k - A_i x_i^k \|_T^2 + \rho \| \text{Tr}(\hat{x}^k) \|_T^2 - 2\kappa \| \hat{x}^k - x^* \|_T^2
= \rho \sum_i \| A_i \hat{x}_i^k - A_i x_i^k \|_{1-T}^2 - \rho \| \text{Tr}(\hat{x}^k) \|_{1-T}^2 - 2\kappa \| \hat{x}^k - x^* \|_T^2.
\]

The above relation suggests that we can choose \( T \) appropriately in order to guarantee that \( \phi \) is strictly decreasing. Specifically, it is sufficient to ensure that the matrices \( I - T \) and \( T - D - 2C \) are positive definite. From the condition \( I - T > 0 \),
we infer that the diagonal elements of $T$ must be strictly less than one. To ensure that $T - D - 2C > 0$, recall that $D = \text{diag}(q_1\tau_1^2, \ldots, q_m\tau_m^2)$ by construction. Also, according to Lemma 29, the matrix $C$ can be any diagonal matrix with strictly positive diagonal entries. Let $C = \frac{1}{2}TE$, where $E = \text{diag}(\epsilon_1, \ldots, \epsilon_m)$, and each $\epsilon_j$, $j = 1, \ldots, m$ is an arbitrarily small, positive number. Then, if we can choose $T$ such that

$$\tau_j - q_j\tau_j^2 - \epsilon_j\tau_j > 0, \quad \forall j = 1, \ldots, m,$$

the diagonal matrix $T - D - 2C$ is guaranteed to be positive definite. The above relation has solution

$$\tau_j < \frac{1 - \epsilon_j}{q_j}, \quad \forall j = 1, \ldots, m. \quad (3.87)$$

Hence, if we select $\tau_j$ according to (3.87), then $\theta_k > 0$ during the execution of ADAL, as required. Since the $\epsilon_j$ can be as small as we want, we obtain the the corresponding condition of the theorem.

Note that to arrive at (3.86), we have used the result of Lemma 29, which requires that the terms $\lambda^k + \rho \sum_{j \neq i}(A_jx_j^k - A_jx^*_j) - \lambda^*$ are sufficiently close to zero for all $i = 1, \ldots, N$ at iteration $k$; recall that the purpose of this condition is to guarantee that the $\hat{x}_i^k$ will fall into the strong convexity region of $G_\rho$, which allows us to use (3.79). Suppose also that the terms $A_i x_i^k - A_i x^*_i$ are sufficiently close to zero for all $i = 1, \ldots, N$ at iteration $k$. Adding $A_i x_i^k - A_i x^*_i$ to $\lambda^k + \rho \sum_{j \neq i}(A_jx_j^k - A_jx^*_j) - \lambda^*$, the assumption that the $\lambda^k + \rho \sum_{j \neq i}(A_jx_j^k - A_jx^*_j) - \lambda^*$ are sufficiently close to zero at iteration $k$ for all $i = 1, \ldots, N$ in Lemma 29, becomes equivalent to the condition that the terms $\lambda^k + \rho r(x^k) - \lambda^*$ and $A_i x_i^k - A_i x^*_i$ for all $i = 1, \ldots, N$ are sufficiently close to zero at iteration $k$.

Note that $\lambda^k + \rho r(x^k) - \lambda^*$ is sufficiently close to zero if and only if $\lambda^k + \rho(I - T)r(x^k) - \lambda^*$ is sufficiently close to zero at iteration $k$. This is because
(I - T) is a finite multiplicative factor on \( r(x^k) \) and \( r(x^k) \) is close to zero, since the \( A_i x_i^k \) are close to \( A_i x_i^* \) for all \( i = 1, \ldots, N \). Now, recall the definition of 
\[
\phi(x^k, \lambda^k) = \sum_{i=1}^{N} \rho \|A_i x_i^k - A_i x_i^*\|_{T^{-1}}^2 + \frac{1}{\rho} \|\lambda^k + \rho(I - T)r(x^k) - \lambda^*\|_{T^{-1}}^2,
\]
and observe that the terms in the right hand side of \( \phi(x^k, \lambda^k) \) are exactly the terms that we need to be sufficiently close to zero in order to apply the result of Lemma 29. Since \( 0 < T < 1 \), it follows that the terms \( \lambda^k + \rho(I - T)r(x^k) - \lambda^* \) and \( A_i x_i^k - A_i x_i^* \) for all \( i = 1, \ldots, N \) are sufficiently close to zero if \( \phi(x^k, \lambda^k) \) is sufficiently small. To see this, observe that all terms in the expression for \( \phi(x^k, \lambda^k) \) are individually upper bounded by the value of \( \phi(x^k, \lambda^k) \), e.g., 
\[
\|A_i x_i^k - A_i x_i^*\|_{T^{-1}} < \|A_i x_i^k - A_i x_i^*\|_{T^{-1}} \leq \frac{1}{\rho} \phi(x^k, \lambda^k).
\]
Hence, if we choose initial values \( x^1, \lambda^1 \) such that \( \phi(x^1, \lambda^1) \) is sufficiently small, then \( \theta_1 > 0 \), which implies that \( \phi(x^2, \lambda^2) < \phi(x^1, \lambda^1) \). Since, \( \phi(x^1, \lambda^1) \) is sufficiently small and \( \phi(x^2, \lambda^2) \) is even smaller, we can infer that the iterates \( \lambda^k + \rho \sum_{j \neq i} (A_j x_j^k - A_j x_j^*) - \lambda^* \) will be sufficiently close to zero for all iterations \( k \). Therefore, the result of Lemma 29 can be used, as required.

\[ \square \]

**Remark 31.** In the statement of Theorem 30, we assume that the initial iterates \( x^1, \lambda^1 \) are chosen such that \( \phi(x^1, \lambda^1) \) is sufficiently small. For comparison, in the convergence proof of the standard augmented Lagrangian method (ALM) described in Alg. 3, the assumption that the dual iterates \( \lambda^k \) are sufficiently close to \( \lambda^* \) for all iterations is used. Following a similar argument as in Theorem 30, this condition holds true if the initial values \( \lambda^1 \) are sufficiently close to \( \lambda^* \); see Ruszczyński (2006); Nocedal and Wright (2006); Bertsekas (1976) for more details. Here, we cannot simply require that the dual variables alone are close to their optimal values. Instead, we need to consider the terms \( \lambda^k + \rho \sum_{j \neq i} (A_j x_j^k - A_j x_j^*) \) for all \( i = 1, \ldots, N \), due to the structure of the local ALs, cf. (3.80), and the distributed nature of the algorithm. This difference gives rise to the condition that \( \phi(x^1, \lambda^1) \) is sufficiently small, which
replaces the requirement that $\lambda^1$ is sufficiently close to $\lambda^*$ as is the case in the ALM.

We are now ready to prove the main result of this section.

**Theorem 32.** Assume (A1)–(A4). Assume also that $\rho$ is sufficiently large, and that for all iterations $k$ the terms $\lambda^k + \rho \sum_{j \neq i}(A_j x_j^k - A_j x_j^*)$ are sufficiently close to $\lambda^*$ for all $i = 1, \ldots, N$. Then, the ADAL method generates sequences of primal variables $\hat{x}^k$ and dual variables $\lambda^k$ that converge to a local minimum $x^*$ of problem (3.53) and its corresponding optimal Lagrange multipliers $\lambda^*$, respectively.

**Proof.** Relation (3.86) implies that

$$
\phi(x^{k+1}, \lambda^{k+1}) \leq \phi(x^k, \lambda^k) - \rho \sum_{i} \|A_i(\hat{x}_i^k - x_i^k)\|^2_{1-T} - \rho \|r(\hat{x}^k)\|^2_{T-D-2C} - 2\kappa \|\hat{x}^k - x^*\|^2
$$

Summing the above inequality for $k = 1, 2, \ldots$, we obtain:

$$
\sum_{k=1}^{\infty} \left[ \rho \sum_{i} \|A_i\hat{x}_i^k - A_i x_i^k\|^2_{1-T} + \rho \|r(\hat{x}^k)\|^2_{T-D-2C} + 2\kappa \|\hat{x}^k - x^*\|^2 \right] < \phi(x^1, \lambda^1)
$$

Since $\phi(x^1, \lambda^1)$ is bounded, this implies that the sequences $\{r(\hat{x}^k)\}$, $\{\hat{x}_i^k - x_i^*\}$, and $\{A_i\hat{x}_i^k - A_i x_i^k\}$ for all $i = 1, \ldots, N$, converge to zero as $k \to \infty$. It follows that $\{r(x^k)\}$ converges to zero as well. By the monotonicity and boundedness properties of $\phi(x^k, \lambda^k)$, we conclude that the sequence $\{\lambda^k\}$ is also convergent. We denote $\lim_{k \to \infty} \lambda^k = \mu$.

From assumption (A2), the gradients of the functions $f_i$ are continuous on $X_i$. Therefore, the sequences $\{\nabla f_i(\hat{x}_i^k)\}$ converge to $\nabla f_i(x_i^*)$ for all $i = 1, \ldots, N$. Passing to the limit in equation (3.64), we infer that each sequence $\{z_i^k\}$ converges to a point $\tilde{z}_i$, $i = 1, \ldots, N$. The mapping $x_i \mapsto N_{X_i}(x_i)$ has closed graph and, hence, $\tilde{z}_i \in N_{X_i}(x_i^*)$. 

84
After the limit pass in (3.64), we conclude that

$$0 = \nabla f_i(x_i^*) + A_i^\top \mu_i + \tilde{z}_i, \quad \forall \ i = 1, \ldots, N.$$ 

Hence, \( \mu \) satisfies the first order optimality conditions for problem (3.53). Since \( x^* \) is a feasible point that satisfies the strong second order sufficient conditions of optimality for problem (3.53), we conclude that ADAL generates primal sequences \( \{\hat{x}^k\} \) that converge to a local minimum \( x^* \) of (3.53), and dual sequences \( \{\lambda^k\} \) that converge to their optimal values \( \lambda^* \) for the point \( x^* \), as required.

3.5 The Stochastic ADAL

In this section we extend ADAL to address the case where problem (3.1) needs to be solved distributedly in the presence of uncertainty and noise. In particular, we consider the scenario where: i) the agents have access only to noisy approximations of their objective functions at each iteration or, equivalently, the agents can calculate only approximate subgradients of their objective functions, and ii) noise corrupts the primal and dual message exchanges between agents during the iterative execution of the method. To address this stochastic framework, ADAL needs to be modified; we refer to the new algorithm as the *Stochastic Accelerated Distributed Augmented Lagrangians* (SADAL) method. We show that, under assumptions that are common in the relevant literature, SADAL generates sequences of primal and dual variables that converge to their optimal values almost surely (a.s.). The contents of this section are part of the work Chatzipanagiotis and Zavlanos (2015a) that has been conditionally accepted for publication.

The work presented in this section is also directly related to the literature of *stochastic approximation* (SA) techniques. Generally speaking, the term SA characterizes those stochastic methods that attempt to iteratively solve convex optimization problems based on noisy (sub)gradient observations. SA has been an active area of
research since the 1950s, beginning with the seminal work of Robbins and Monro (1951), which introduced the Robbins-Monro algorithm for unconstrained optimization problems and proved that the method generates iterates that converge to the optimal solution in the mean square sense. Since then, a significant amount of SA literature has emerged, with some of the most representative works being Kiefer and Wolfowitz (1952); Polyak and Juditsky (1992); Ermoliev (1969, 1983); Ljung (1977); Ruszczynski and Syski (1983); Spall (1992, 2000); Wang and Spall (1999, 2003); Nemirovski et al. (2009); Jiang and Xu (2008); Yousefian et al. (2012). Certain works follow the so-called “limiting mean ODE” (ordinary differential equations) technique to prove the convergence of SA schemes; see for example Kushner and Yin (2003); Borkar and Meyn (2000); Borkar (2008); Benam et al. (2005); Roth and Sandholm (2013). On the other hand, there exists a significantly smaller number of works that considers distributed stochastic approximation schemes. The existing literature on such distributed approaches is mostly concerned with consensus constrained optimization problems, wherein a set of agents with separable objective functions need to agree on a common decision vector; see, e.g., Tsitsiklis et al. (1986); Kushner and Yin (1987); Stankovic et al. (2011); Srivastava and Nedic (2011); Nedic (2011); Sundhar Ram et al. (2010); Bianchi and Jakubowicz (2013); Bianchi et al. (2013); Yousefian et al. (2013).

The contribution here is threefold. First, we propose a distributed stochastic approximation algorithm that can address more general constraint sets compared to the relevant existing literature in SA which is concerned only with consensus constraints. In fact, problems with consensus constraints can be expressed in the form of (3.1), such that they are a special case of the scenario under consideration here. Second, we allow for multiple noise terms, namely four, to appear in both the computation and communication stages of the proposed distributed iterative method. Typically, distributed stochastic approximation algorithms contain a single
source of noise, affecting either the computations or the communications, with only a few works considering two noise terms simultaneously Kushner and Yin (1987); Srivastava and Nedic (2011); Stankovic et al. (2011).

Finally, the proposed method is based on the augmented Lagrangian framework, for which only a limited amount of works exist that consider it in a stochastic setting. For example, in Erseghe et al. (2011); Schizas et al. (2009) the authors examine the stability properties of the *Alternating Direction Method of Multipliers* (ADMM) when applied to consensus problems that suffer from noise in the message exchanges. Moreover, in Ouyang et al. (2013) a convergence result is derived for the ADMM applied to problems of the form (3.1) with $N = 2$ and noise appearing in one of the two objective functions. In this dissertation we study a more general framework than the aforementioned stochastic ADMM works, and also provide a stronger convergence result (a.s.). Specifically, we consider the general class of problems (3.1), where $N$ is allowed to be larger than 2, and where noise appears in all the objective functions and all message exchanges at the same time. The main challenge here is that AL methods are primal-dual schemes, which means that the effects from multiple sources of noise corruption and uncertainty propagate in-between the two domains.

### 3.5.1 The ADAL in a Stochastic Setting

The SADAL algorithm allows for a distributed solution of (3.1) when: i) the local computation steps are inexact or are performed in the presence of uncertainty, and ii) the message exchanges between agents are corrupted by noise. The basic algorithmic structure of SADAL is essentially the same as that of ADAL. Nevertheless, to account for the presence of uncertainty and noise, appropriate adaptations need to be made. SADAL is summarized in Alg. 9.

In SADAL, each agent $i \in \mathcal{I}$ receives noise-corrupted versions of the actual primal and dual variables. We let $\tilde{\mathbf{x}}_{ij}^k$ and $\tilde{\lambda}_i^k$ denote the noise-corrupted versions of the
Algorithm 9 Stochastic Accelerated Distributed Augmented Lagrangians (SADAL)

Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$ and initial primal variables $x^1$.

1. For fixed Lagrange multipliers $\lambda^k$, determine $\hat{x}^k_i$ for every $i \in \mathcal{I}$ as the solution of the following problem:

$$
\min_{x_i} \Lambda^i_p(x_i, \tilde{x}^k_i, \tilde{\lambda}^k_i, \xi^k_i) \quad (3.89)
$$

s.t. $x_i \in X_i$

2. Set for every $i \in \mathcal{I}$

$$
x^{k+1}_i = x^k_i + \tau_k (\hat{x}^k_i - x^k_i) \quad (3.90)
$$

$$
y^{k+1}_i = x^k_i + \frac{1}{q} (\hat{x}^k_i - x^k_i) \quad (3.91)
$$

3. If the constraints $\sum_{i=1}^N A_i \tilde{x}^{k+1}_i = b$ are satisfied and $A_i \hat{x}^k_i = A_i x^k_i$, then stop (optimal solution found). Otherwise, set:

$$
\lambda^{k+1} = \lambda^k + \rho \tau_k \left( \sum_{i=1}^N A_i \tilde{y}^{k+1}_i - b \right) \quad (3.92)
$$

increase $k$ by one and return to Step 1.

The primal $x^k_i$ and the dual $\lambda^k$ variables, respectively, as received by agent $i$ at iteration $k$. Consequently, the local augmented Lagrangian function $\Lambda^i_p : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ of each agent $i \in \mathcal{I}$ is now formed based on these noise-corrupted variables, cf. (3.89), i.e., it takes the form

$$
\Lambda_i(x_i, \tilde{x}^k_i, \tilde{\lambda}^k_i, \xi^k_i) = F_i(x_i, \xi^k_i) + \langle \tilde{\lambda}^k_i, A_i x_i \rangle + \frac{\rho}{2} \| A_i x_i + \sum_{j \neq i} A_j \tilde{x}^k_{ij} - b \|^2,
$$

where $\tilde{x}^k_i = \{ \tilde{x}^k_{i1}, \ldots, \tilde{x}^k_{iN} \}$ denotes the collection of the noise corrupted variables $\tilde{x}^k_{ij}$. Note that every local AL is now defined with respect to the function $F_i(x_i, \xi^k_i)$, which is the noise-corrupted version of the true objective function $f_i$. Here, the term $\xi^k_i$ represents the uncertainty at iteration $k$.

Moreover, in SADAL the stepsize parameter $\tau$ has to be defined as $\tau_k$, cf. (3.90) and (3.92), which must be a decreasing, nonnegative sequence that is square summable.
but not summable. This decrease property of the stepsize is essential in the vast
majority of works within the relevant stochastic approximation literature; see, e.g.,
Kushner and Yin (2003). Finally, SADAL introduces the additional auxiliary vari-
ables $y_{i}^{k+1} = x_{i}^{k} + \frac{1}{q}(\hat{x}_{i}^{k} - x_{i}^{k})$ that are updated locally at each agent $i \in \mathcal{I}$, cf. (3.91).

Note that the difference between the $y_{i}^{k+1}$ and the $x_{i}^{k+1}$ updates lies in the stepsize
choice; we always use $\frac{1}{q}$ for the $y_{i}^{k+1}$ and $\tau_{k}$ for the $x_{i}^{k+1}$. The $y_{i}^{k+1}$ variables are then
used for the dual update step of SADAL, cf. (3.92).

In what follows, we elaborate on the specific noise terms that appear during the
iterative execution of SADAL (3.89)-(3.92), such that we provide a specific definition
for the notion of uncertainty and noise corruption in our particular setting. We
assume that there is a probability space $(\Omega, \mathcal{F}, P)$, where the set $\Omega$ is arbitrary, $\mathcal{F}$
is a $\sigma$-algebra of subsets of $\Omega$, and $P$ is a probability measure defined on $\mathcal{F}$. All
$\sigma$-algebras will be sub-$\sigma$-algebras of $\mathcal{F}$, and all random variables will be defined on
this space.

**Noise in the message exchanges for the formation of the local aug-
mented Lagrangians:** At each iteration $k$, agent $i$ receives, via communication,
the noise corrupted primal variables $A_{j}\tilde{x}_{ij}^{k}$ from agent $j$, and also the noise corrupted
dual variables $\tilde{\lambda}^{k}_{i}$ according to

\begin{align*}
A_{j}\tilde{x}_{ij}^{k} &= A_{j}x_{j}^{k} + v_{ij}^{k} \quad (3.93) \\
\tilde{\lambda}^{k}_{i} &= \lambda^{k} + w_{i}^{k} \quad (3.94)
\end{align*}

where $v_{ij}^{k} : \Omega \rightarrow \mathbb{R}^{m}$, $w_{i}^{k} : \Omega \rightarrow \mathbb{R}^{m}$ are random vectors of appropriate size whose en-
tries are assumed to be i.i.d. random variables with zero mean and bounded variance.
Essentially, $v_{ij}^{k}$ represents the noise corruption in the communication of the actual
primal variables $A_{j}x_{j}^{k}$ of agent $j$ towards agent $i$. Similarly, $w_{i}^{k}$ denotes the noise
corruption on the dual variables $\lambda^{k}$ as perceived by agent $i$ after the corresponding
message exchanges.
Note that we formulate the message exchanges with respect to the products $A_jx^k_j$; despite the fact that the $x^k_j$ are the actual variables and the matrices $A_j$ are essentially problem parameters. This is because each agent $i$ does not need to know the matrices $A_j$ of the other agents; it is only interested in the products $A_jx^k_j$. In fact, it only needs those entries of the vector $A_jx^k_j$ which correspond to their common coupling constraints.

**Noise in the local computations:** After receiving the communicated variables, each agent $i$ determines the minimizers $\hat{x}^k_i$ of its local augmented Lagrangian $\Lambda^i_\rho(x_i, \tilde{x}^k_i, \tilde{\lambda}^k_i, \xi^k_i)$ according to (3.89). Each $\Lambda^i_\rho(x_i, \tilde{x}^k_i, \tilde{\lambda}^k_i, \xi^k_i)$ contains the function $F_i : \mathbb{R}^{n_i} \times \Omega \to \mathbb{R}$, where $\xi$ is an element of the probability space $(\Omega, \mathcal{F}, P)$. We assume that each $F_i(x_i, \cdot)$ is a convex function of $x_i$ for each $\xi_i \in \Omega$, and that $F_i(x_i, \cdot)$ is an integrable function of $\xi_i$ for each $x_i \in \mathbb{R}^{n_i}$, i.e., we assume that $E[|F_i(x_i, \xi_i)|] < \infty$ for each $x_i \in \mathbb{R}^{n_i}$. We also assume that the functions $F_i$ satisfy

$$f_i(x_i) = E[F_i(x_i, \xi_i)].$$

The above relation implies that the $f_i(x_i)$ are convex and that the following relation also holds (see, e.g., Bertsekas (1973))

$$\partial f_i(x_i) = E[\partial F_i(x_i, \xi_i)],$$

where $\partial f_i(x_i)$ and $\partial x_i F_i(x_i, \xi_i)$ denote the convex subdifferentials of the convex functions $f_i(x_i)$ and $F_i(x_i, \xi_i)$, respectively, at a point $x_i$. If we let $s_{f_i} \in \partial f_i(x_i)$ denote a subgradient of $f_i$ at the point $x_i$, and $s_{F_i} \in \partial F_i(x_i, \xi_i)$ be a subgradient of $F_i$ with respect to $x_i$, then $s_{f_i} = E[s_{F_i}]$ also holds, which can be equivalently expressed as

$$s_{f_i} = s_{F_i} + e^{k}_i,$$  (3.95)

where the noise vector $e^{k}_i : \Omega \to \mathbb{R}^{n_i}$ must satisfy $E[e^{k}_i] = \mathbf{0}$ for all iterations $k$. Since the functions $F_i$ appear only in the local computation steps (3.89), the
above arguments reveal that, for our particular method, the requirement \( f_i(x_i) = \mathbb{E}[F_i(x_i, \xi_i)] \) is equivalent to \( s_{f_i} = s_{F_i} + e_i \).

Essentially, the aforementioned formulation for noise in the local computations of SADAL enables us to model cases where: i) at each iteration \( k \) the agents have access only to noisy observations \( F_i(x_i, \xi_i^k) \) of their true objective functions \( f_i \), or ii) cases where the agents want to optimize the expected values of the \( F_i \)'s, but have access only to sample values \( F_i(x_i, \xi_i^k) \), or even iii) cases where the subgradients of \( F_i \) can only be computed with an error \( e_i^k \) at each iteration \( k \).

**Noise in the message exchanges for the dual updates:** We assume that there exists a set \( \mathcal{M} \subseteq \{1, \ldots, m\} \) of agents that perform the dual updates. After the local minimization and primal update steps have been performed, each agent \( i \in \mathcal{I} \) communicates the updated \( A_i y_i^{k+1} \) variables to the agents responsible for the dual updates. This message exchange is also corrupted by noise, such that the received messages \( A_i \tilde{y}_i^{k+1} \) take the form

\[
A_i \tilde{y}_i^{k+1} = A_i y_i^{k+1} + u_i^{k+1}
\]

where \( u_i^{k+1} : \Omega \rightarrow \mathbb{R}^m \) is a random vector whose entries are assumed to be i.i.d. random variables with zero mean and bounded variance. Here, the entry \([u_i^{k+1}]_j\) corresponds to the noise corruption on the respective actual variable \([A_i]_j y_i^{k+1}\) as received by agent \( j \in \mathcal{M} \) after the corresponding message exchanges.

Note that we formulate the message exchanges (3.96) with respect to the products \( A_i \tilde{y}_i \), following the same reasoning as discussed above regarding the message exchanges for the formation of the local ALs (3.93)-(3.94). Furthermore, note that in practice if a row \( j \) of \( A_i \) is zero, then the corresponding \( j \)-th entry of \( u_i \) should also be identically zero (i.e., it has zero mean and variance), since agent \( i \) does not need to communicate anything for the update of the dual variable of constraint \( j \).
3.5.2 Almost Sure Convergence of Stochastic ADAL

In this section we establish the almost sure convergence of SADAL to the optimal solution of (3.1). We need the following assumptions:

(A1) For every $i \in \mathcal{I}$ consider the function $F_i(x_i, \xi_i)$, where $F_i : \mathbb{R}^{n_i} \times \Omega \rightarrow \mathbb{R}$. We assume that $F_i(\cdot, \xi_i)$ is a convex function of $x_i$ for each $\xi_i \in \Omega$, and that $F_i(x_i, \cdot)$ is an integrable function of $\xi_i$ for each $x_i \in \mathbb{R}^{n_i}$, i.e., $\mathbb{E}[|F_i(x_i, \xi_i)|] < \infty$ for each $x_i \in \mathbb{R}^{n_i}$. Moreover, each $F_i$ satisfies the relation $f_i(x_i) = \mathbb{E}[F_i(x_i, \xi_i)]$, where $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$. It follows that $f_i$ is also convex; see, e.g., Bertsekas (1973). We also assume that the sets $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ are nonempty closed, convex sets for all $i \in \mathcal{I}$.

(A2) The Lagrange function $L(x, \lambda)$ has a saddle point $(x^*, \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$:

$$L(x^*, \lambda) \leq L(x, \lambda^*) \leq L(x, \lambda^*),$$

for every $x \in \mathcal{X}$, and $\lambda \in \mathbb{R}^m$.

(A3) All subproblems (3.89) are solvable at any iteration $k = 1, 2, \ldots$.

(A4) The stepsize sequence $\tau_k$ satisfies

$$\tau_k \in (0, \frac{1}{q}) \quad , \quad \sum_{k=1}^{\infty} \tau_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \tau_k^2 < \infty. \quad (3.97)$$

(A5) The penalty parameter $\rho$ is strictly positive.

(A6) The noise corruption vectors $e^k_i, v^k_{ij}, w^k_i, u^k_i$, for every $i, j \in \mathcal{I}$, have entries which are i.i.d random variables with zero mean. Moreover, the entries of the noise terms $v^k_{ij}, w^k_i, u^k_i$ have bounded variance.
The noise terms

\[ \epsilon_i^k = w_i^k + \rho \sum_{j \neq i} v_{ij}^k \]  

satisfy a.s.

\[ \sum_{k=1}^{\infty} \tau_k \mathbb{E}_k[\|\epsilon_i^k\|^2] < \infty, \]  

where \( \mathbb{E}_k \) denotes conditional expectation with respect to the \( \sigma \)-algebra pertaining to iteration \( k \).

The noise terms in the local computation steps satisfy

\[ \epsilon_i^k \xrightarrow{a.s.} 0. \]

Assumptions (A2) and (A3) are the same as in the deterministic ADAL method. For the sake of clarity, we recall that assumption (A3) is satisfied if for every \( i = 1, \ldots, N \), either the set \( \mathcal{X}_i \) is compact, or the function \( F_i(x_i, \tilde{\xi}) + \frac{\rho}{2}\|A_ix_i - \tilde{b}\|^2 \) is inf-compact for any given \( \tilde{\xi} \) and \( \tilde{b} \). The latter condition, means that the level sets of the function are compact sets, implying that the set \( \{x_i \in \mathcal{X}_i : F_i(x_i, \tilde{\xi}) + \frac{\rho}{2}\|A_ix_i - \tilde{b}\|^2 \leq \alpha \} \) is compact for any \( \alpha \in \mathbb{R} \).

Assumption (A4) includes the stepsize condition from ADAL, i.e., the fact that \( \tau_k < \frac{1}{\eta} \). Moreover, the conditions that the stepsize sequence should be square-summable, but not summable, are typical in relevant stochastic approximation literature; see, e.g., Kushner and Yin (2003) for a comprehensive overview. Assumption (A5) is the typical assumption that \( \rho > 0 \), which is necessary in all augmented Lagrangian methods.

Assumptions (A6)-(A8) are necessary to prove the a.s. convergence of SADAL. The zero mean assumption is a common assumption ensuring that the presence of noise does not introduce bias in the computations in the long run, while the bounded variance assumption is a mild technical condition that is needed to guarantee the
convergence of the iterative procedure. Assumption (A7) is necessary to establish the a.s. convergence of a supermartingale sequence that we construct to show the a.s. convergence of SADAL; similar assumptions have been used in the existing literature, e.g., Nedic (2011); Wang and Spall (1999, 2003). Assumption (A8) is used to guarantee that SADAL indeed converges to the optimal set of the original problem (3.1); see, e.g., Singh et al. (2007); Salimans and Knowles (2014); Johnson and Zhang (2013) for a range of applications where this assumption can be valid.

Essentially, assumptions (A6)-(A8) require that the noise corruption terms appearing in the local AL computations vanish in the limit; relation (23) also requires that the noise terms $\epsilon^k$ vanish “quickly enough”. Note that we do not impose any decrease conditions on the noise terms $u^k$ that appear in the dual update step of SADAL, cf. (3.92) and (3.96). An intuitive explanation about the different assumptions on the noise terms can be reached if we recall that in the AL framework we perform gradient ascent in the dual domain, where the gradient of the dual function at each iteration is given by the residual of the constraints. For instance, the AL method (cf. Alg. 1) can be viewed as the proximal point method applied to the dual problem Bertsekas and Tsitsiklis (1997). In classical stochastic gradient descent methods it is not essential that the gradient noise terms vanish in the limit, just that they are unbiased, cf. Robbins and Monro (1951); Kushner and Yin (2003). This is exactly the case here; for the noise terms $u^k$ that directly affect the residual calculation (the gradient of the dual function) we only require that they are unbiased, cf. assumption (A6). However, we cannot guarantee the same unbiased behavior for the noise terms $v_{ij}^k$, $w_i^k$, and $e_i^k$ that appear in the local AL computations, since the effects of noise corruption can propagate and affect the dual gradient in more complicated ways that may cause bias. While this work constitutes a first effort to address the presence of noise within ADAL, it is certainly an interesting topic to characterize the error bounds, in terms of optimality and feasibility, for scenarios
where the aforementioned noise terms do not necessarily vanish in the limit, or even propose alternative algorithms that allow us to relax the decrease conditions.

To avoid cluttering the notation, in what follows we will use the simplified notation $\sum_i$ to denote summation over all $i \in I$, i.e. $\sum_i = \sum_{i=1}^N$, unless explicitly noted otherwise. We use $\mathcal{N}_\mathcal{X}(x)$ to denote the normal cone to the set $\mathcal{X}$ at point $x$ Ruszczyński (1995), i.e.,

$$\mathcal{N}_\mathcal{X}(x) = \{ h \in \mathbb{R}^n : \langle h, y - x \rangle \leq 0, \ \forall y \in \mathcal{X} \}.$$  

We also define the auxiliary variables:

$$\hat{\lambda}^k = \lambda^k + \rho r(\hat{x}^k), \quad (3.100)$$

available at iteration $k$.

Note that, in the following analysis, the various primal and dual variables at each iteration $k$ are essentially functions of the entire history of the generated random process up to that iteration and, hence, are random variables. With that in mind, recall that $(\Omega, \mathcal{F}, P)$ is our probability space, and let $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots$ denote an increasing sequence of sub-$\sigma$-algebras of $\mathcal{F}$, with $\mathcal{F}_k$ denoting the sub-$\sigma$-algebra pertaining to iteration $k$, i.e., for $1 \leq s < k$

$$\mathcal{F}_k = \{ x^s_i, y^s_i, \lambda^s, \tilde{y}^s_i, \tilde{x}^s_i, \bar{x}^s, e^s_i, e_i^s, u_i^s : i, j \in I, 1 \leq s < k \}.$$  

(3.101)

The main idea behind the convergence proof is to show that the sequence $\{\phi(x^k, \lambda^k)\}$, defined as

$$\phi(x^k, \lambda^k) = \rho \| A_i(x^k_i - x_i^*) \|^2 + \frac{1}{\rho} \| \lambda^k + \rho(1 - \frac{1}{q}) r(x^k) - \lambda^* \|^2, \quad (3.102)$$

converges a.s. to some finite random variable $\bar{\phi}$. Note that (3.102) is almost the same as (3.9) from the deterministic case discussed in section 3.2, with the only difference
being that we have now replaced $\tau$ with its upper bound $\frac{1}{q}$ in the term involving the dual variables. To establish the a.s. convergence of $\{\phi(x^k, \lambda^k)\}$ in the stochastic setting, we will make use of the following theorem from Robbins and Siegmund (1971), which is called the non-negative almost-supermartingale convergence theorem.

**Theorem 33.** Let $(\Omega, F, P)$ be a probability space and $F_1 \subset F_2 \subset \ldots$ be a sequence of $\sigma$-subfields of $F$. For each $k = 1, 2, \ldots$, let $\zeta_k, \chi_k, \psi_k, \eta_k$ be nonnegative, $F_k$-measurable random variables such that

$$E(\zeta_{k+1}|F_k) \leq (1 + \eta_k)\zeta_k + \chi_k - \psi_k.$$  

(3.103)

If $\sum_1^\infty \eta_k < \infty$ and $\sum_1^\infty \chi_k < \infty$ hold, then $\lim_{k \to \infty} \zeta_k$ exists and is finite almost surely, and $\sum_1^\infty \psi_k < \infty$.

Essentially, in what follows we prove that $\{\phi(x^k, \lambda^k)\}$ is such a non-negative almost-supermartingale. Then, we use this convergence result to infer that the sequence of dual variables $\{\lambda^k(\omega)\}$ converges a.s. to an optimal solution of problem (2.28), and that any sequence of primal variables $\{x^k\}$ generated by SADAL has an accumulation point and any such point is a.s. an optimal solution of problem (3.1). In the following lemma, we utilize the first order optimality conditions for each local subproblem (3.89) to obtain a first result towards proving the a.s. convergence of the sequence $\{\phi(x^k, \lambda^k)\}$.  

**Lemma 34.** Assume (A1)-(A3). Then, the following inequality holds:

$$\frac{1}{\rho} \left\langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda}^k \right\rangle \geq \rho \sum_i \left\langle A_i(\hat{x}_i^k - x_i^*), \sum_{j \neq i} A_j(x_j^k - \hat{x}_j^k) \right\rangle$$  

(3.104)

$$+ \sum_i \left[ \left\langle A_i(\hat{x}_i^k - x_i^*), \epsilon_i^k \right\rangle + f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \zeta_i^k) - F_i(x_i^*, \zeta_i^k) \right],$$

where $(x^*, \lambda^*)$ is a saddle point of the Lagrangian $L$ and $\hat{\lambda}^k, \hat{x}_i^k$ are calculated at iteration $k$.  

96
Proof. The first order optimality conditions for problem (3.89) imply the following inclusion for the minimizer $\hat{x}_i^k$

$$0 \in \partial x_i F_i(\hat{x}_i^k, \xi_i^k) + A_i^T \hat{\lambda}_i^k + \rho A_i^T \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j \hat{x}_j^k - b \right) + N_{\chi_i}(\hat{x}_i^k).$$ (3.105)

We infer that subgradients $s_{F_i}^k \in \partial x_i F_i(\hat{x}_i^k, \xi_i^k)$ and normal elements $z_i^k \in N_{\chi_i}(\hat{x}_i^k)$ exist such that we can express (3.105) as follows:

$$0 = s_{F_i}^k + A_i^T \hat{\lambda}_i^k + \rho A_i^T \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j \hat{x}_j^k - b \right) + z_i^k.$$

Taking inner product with $x_i^* - \hat{x}_i^k$ on both sides of this equation and using the definition of a normal cone, we obtain

$$\langle s_{F_i}^k + A_i^T \hat{\lambda}_i^k + \rho A_i^T \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j \hat{x}_j^k - b \right), x_i^* - \hat{x}_i^k \rangle = -z_i^k, x_i^* - \hat{x}_i^k \rangle \geq 0.$$

(3.106)

Using the variables $\hat{\lambda}_i^k$ defined in (3.100) and also substituting $\hat{x}_j^k, \hat{\lambda}_i^k$ from (3.93)-(3.94) in (3.106), we obtain

$$0 \leq \langle s_{F_i}^k + A_i^T \left( \hat{\lambda}_i^k - \rho \left( \sum_j A_j \hat{x}_j^k \right) + w_i^k + \rho \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j \hat{x}_j^k \right) \right), x_i^* - \hat{x}_i^k \rangle$$

$$= \langle s_{F_i}^k + A_i^T (w_i^k + \rho \sum_{j \neq i} v_{ij}^k) + A_i^T \left( \hat{\lambda}_i^k + \rho \left( \sum_{j \neq i} A_j x_j^k - \sum_{j \neq i} A_j \hat{x}_j^k \right) \right), x_i^* - \hat{x}_i^k \rangle.$$  

From the convexity of $F_i$, we have that $F_i(x_i^*, \xi_i^k) - F_i(\hat{x}_i^k, \xi_i^k) \geq s_{F_i}(\hat{x}_i^k)^T (x_i^* - \hat{x}_i^k)$, so the above inequality can be expressed as

$$F_i(x_i^*, \xi_i^k) - F_i(\hat{x}_i^k, \xi_i^k)$$

$$+ \langle w_i^k + \rho \sum_{j \neq i} v_{ij}^k + \hat{\lambda}_i^k + \rho \left( \sum_{j \neq i} A_j x_j^k - \sum_{j \neq i} A_j \hat{x}_j^k \right), A_i (x_i^* - \hat{x}_i^k) \rangle \geq 0.$$ (3.107)

The assumptions (A1) and (A2) entail that the following optimality conditions are satisfied at the point $(x^*, \lambda^*)$:

$$0 \in \partial f_i(x_i^*) + A_i^T \lambda^* + N_{\chi_i}(x_i^*), \quad \forall i = 1, \ldots, N.$$ (3.108)
Inclusion (3.108) implies that subgradients \( s_{f_i}^* \in \partial f_i(x_i^*) \) and normal vectors \( z_i^* \in N_{X_i}(x_i^*) \) exist, such that we can express (3.108) as:

\[
0 = s_{f_i}^* + A_i^\top \lambda^* + z_i^*
\]

Taking inner product with \( \hat{x}_i^k - x_i^* \) on both sides of this equation and using the definition of a normal cone, we infer

\[
\left\langle s_{f_i}^* + A_i^\top \lambda^*, \hat{x}_i^k - x_i^* \right\rangle \geq 0, \quad \forall \ i = 1, \ldots, N,
\]

or, using the convexity of \( f_i \),

\[
f_i(\hat{x}_i^k) - f_i(x_i^*) + \left\langle A_i^\top \lambda^*, \hat{x}_i^k - x_i^* \right\rangle \geq 0,
\]

(3.109)

for all \( i = 1, \ldots, N \). Denoting \( \epsilon_i^k = w_i^k + \rho \sum_{j\neq i} v_{ij}^k \), cf. (3.98), and combining (3.107) and (3.109), we obtain the following inequalities for all \( i = 1, \ldots, N \):

\[
\left\langle \lambda^* - \hat{\lambda}^k - \epsilon_i^k - \rho \left( \sum_{j\neq i} A_j x_j^k - \sum_{j\neq i} A_j \hat{x}_j^k \right), A_i (\hat{x}_i^k - x_i^*) \right\rangle \\
\geq f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \xi_i^k) - F_i(x_i^*, \xi_i^k).
\]

Adding the inequalities for all \( i = 1, \ldots, N \) and rearranging terms, we get:

\[
\left\langle \lambda^* - \hat{\lambda}^k, \sum_i A_i (\hat{x}_i^k - x_i^*) \right\rangle \\
\geq \rho \sum_i \left\langle A_i \hat{x}_i^k - A_i x_i^*, \sum_{j\neq i} A_j (x_j^k - \hat{x}_j^k) \right\rangle \\
+ \sum_i \left[ \left\langle A_i (\hat{x}_i^k - x_i^*), \epsilon_i^k \right\rangle + f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \xi_i^k) - F_i(x_i^*, \xi_i^k) \right].
\]

Substituting \( \sum_{i=1}^N A_i x^* = b \) and \( \sum_{i=1}^N A_i \hat{x}_i^k - b = \frac{1}{\rho} (\hat{\lambda}^k - \lambda^k) \) in the left hand side of (3.110), we conclude that

\[
\frac{1}{\rho} \left\langle \hat{\lambda}^k - \lambda^*, \lambda^k - \hat{\lambda}^k \right\rangle \\
\geq \rho \sum_i \left\langle A_i (\hat{x}_i^k - x_i^*), \sum_{j\neq i} A_j (x_j^k - \hat{x}_j^k) \right\rangle \\
+ \sum_i \left[ \left\langle A_i (\hat{x}_i^k - x_i^*), \epsilon_i^k \right\rangle + f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \xi_i^k) - F_i(x_i^*, \xi_i^k) \right],
\]

as required. \( \square \)
In the next lemma, we further manipulate the result from Lemma 34 to bring us one step closer to proving the a.s. convergence of the sequence \( \{ \phi(x^k, \lambda^k) \} \). To avoid cluttering the notation, in what follows we denote the rightmost term in (3.104) as

\[
\alpha^k = \sum_{i} \left[ \langle A_i(\hat{x}_i^k - x_i^*), \epsilon_i^k \rangle + f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \zeta_i^k) - F_i(x_i^*, \zeta_i^k) \right].
\]

(3.111)

Lemma 35. Under assumptions (A1)-(A3), the following estimate holds:

\[
\sum_i \rho \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \frac{1}{\rho} \langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda}^k \rangle \\
\geq \sum_i \rho \| A_i(x_i^k - \hat{x}_i^k) \|^2 + \frac{1}{\rho} \| \lambda^k - \lambda^k \|^2 + \langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \rangle + \alpha^k.
\]

(3.112)

Proof. Add the term \( \rho \sum_i \langle A_i(\hat{x}_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle \) to both sides of inequality (3.104) from Lemma 34, to get

\[
\rho \sum_i \langle A_i(\hat{x}_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \frac{1}{\rho} \langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda}^k \rangle \\
\geq \rho \sum_i \langle A_i(\hat{x}_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \rho \sum_{\substack{j \neq i}} \langle A_i(\hat{x}_i^k - x_i^*), A_j(x_j^k - \hat{x}_j^k) \rangle + \alpha^k.
\]

Grouping the terms in the right-hand side of the above inequality by their common factor, we transform it as follows:

\[
\rho \sum_i \langle A_i(\hat{x}_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \frac{1}{\rho} \langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda}^k \rangle \\
\geq \rho \sum_i \langle A_i(\hat{x}_i^k - x_i^*), \sum_i A_i(x_i^k - \hat{x}_i^k) \rangle + \alpha^k.
\]

(3.113)

Recall that \( \sum_i A_i(x_i^k - \hat{x}_i^k) = r(x^k) - r(\hat{x}^k) \). This term does not depend on the summation over \( i \) in the right hand side of (3.113). Moreover, \( \sum_i A_i x_i^* = b \). Substituting
these terms at the right-hand side of (3.113), yields
\[
\rho \sum_i \left\langle A_i (\hat{x}_i - x_i^*), A_i (x_i^* - \hat{x}_i^*) \right\rangle + \frac{1}{\rho} \left\langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda} \right\rangle \\
\geq \rho \left\langle \sum_i A_i (x_i^k - x_i^*), r(x^k) - r(\hat{x}^k) \right\rangle + \alpha^k \\
= \rho \left\langle \sum_i A_i \hat{x}_i^k - \mathbf{b}, r(x^k) - r(\hat{x}^k) \right\rangle + \alpha^k \\
= \left\langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \right\rangle + \alpha^k. 
\] (3.114)

In a last step, we represent
\[
(A_i \hat{x}_i^k - A_i x_i^*) = (A_i x_i^k - A_i x_i^*) + (A_i \hat{x}_i^k - A_i x_i^*)
\]
and \(\hat{\lambda}^k - \lambda^* = (\lambda^k - \lambda^*) + (\hat{\lambda} - \lambda^k)\)
in the left-hand side of (3.114). We obtain
\[
\sum_i \rho \left\langle A_i (x_i^k - x_i^*), A_i (x_i^k - \hat{x}_i^k) \right\rangle + \frac{1}{\rho} \left\langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda} \right\rangle \\
\geq \sum_i \rho \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{\rho} \|\lambda^k - \lambda^k\|^2 + \left\langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \right\rangle + \alpha^k,
\]
as required. \(\square\)

To avoid cluttering the notation, in what follows we use the following variable
\[
\hat{\lambda}^k = \lambda^k + \rho (1 - \frac{1}{q}) r(x^k). 
\] (3.115)

Note that \(\hat{\lambda}^k\) appears in the term containing the dual variables of our stochastic Lyapunov/Merit function \(\phi(x^k, \lambda^k)\), cf. (3.102).

**Lemma 36.** Under the assumptions (A1)-(A3), the following estimate holds
\[
\sum_i \rho \left\langle A_i (x_i^k - x_i^*), A_i (x_i^k - \hat{x}_i^k) \right\rangle + \frac{1}{\rho} \left\langle \lambda^k - \lambda^*, \lambda^k - \hat{\lambda} \right\rangle \\
\geq \sum_i \frac{\rho}{2} \|A_i (x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{2q\rho} \|\lambda^k - \hat{\lambda}^k\|^2 + \alpha^k,
\]
100
where the $\hat{\lambda}^k$ are defined in (3.115).

Proof. Add the term $\rho(1 - \frac{1}{q})\langle r(x^k), \frac{1}{p}(\lambda^k - \hat{\lambda}^k) \rangle = \rho(1 - \frac{1}{q})\langle r(x^k), -r(\hat{x}^k) \rangle$ to inequality (3.112) to get:

$$\sum_i \rho\langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \frac{1}{\rho}\langle \hat{\lambda}^k - \lambda^*, \lambda^k - \hat{\lambda}^k \rangle$$

$$\geq \sum_i \rho\|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2 + \alpha^k$$

(3.116)

$$+ \langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \rangle - \rho(1 - \frac{1}{q})\langle r(x^k), r(\hat{x}^k) \rangle.$$ 

Isolate the term $\langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \rangle - \rho(1 - \frac{1}{q})\langle r(x^k), r(\hat{x}^k) \rangle$ at the right hand side for a bit. We manipulate it to yield:

$$\langle \hat{\lambda}^k - \lambda^k, r(x^k) - r(\hat{x}^k) \rangle - \rho(1 - \frac{1}{q})\langle r(x^k), r(\hat{x}^k) \rangle$$

$$= \rho\langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - \rho(1 - \frac{1}{q})\langle r(x^k), r(\hat{x}^k) \rangle$$

$$= \rho\langle r(\hat{x}^k), r(x^k) - r(\hat{x}^k) \rangle - \rho(1 - \frac{1}{q})\langle r(\hat{x}^k) - r(x^k) + r(\hat{x}^k), r(\hat{x}^k) \rangle$$

$$= \frac{1}{q}\rho\langle r(\hat{x}^k), r(\hat{x}^k) - r(x^k) \rangle - (1 - \frac{1}{q})\rho\|r(x^k)\|^2$$

$$= \frac{1}{q}\langle \hat{\lambda}^k - \lambda^k, \sum_i A_i(x_i^k - \hat{x}_i^k) \rangle - (1 - \frac{1}{q})\frac{1}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2.$$ 

Then, (3.116) becomes:

$$\sum_i \rho\langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \frac{1}{\rho}\langle \hat{\lambda}^k - \lambda^*, \lambda^k - \hat{\lambda}^k \rangle$$

(3.117)

$$\geq \sum_i \rho\|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{q\rho}\|\hat{\lambda}^k - \lambda^k\|^2 + \frac{1}{q}\langle \hat{\lambda}^k - \lambda^k, \sum_i A_i(x_i^k - \hat{x}_i^k) \rangle + \alpha^k.$$ 

The terms $\frac{1}{q}\langle \hat{\lambda}^k - \lambda^k, A_i(x_i^k - \hat{x}_i^k) \rangle$ can be bounded below by considering

$$\frac{1}{q}\langle \hat{\lambda}^k - \lambda^k, A_i(x_i^k - \hat{x}_i^k) \rangle \geq \frac{1}{2}\left( \rho\|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{q^2\rho}\|\hat{\lambda}^k - \lambda^k\|^2 \right).$$
Summing the inequality over all $i$, we observe that the quantity $|\hat{\lambda}_k - \lambda_j|^2$, where $\lambda_j$ indicates the Lagrange multiplier of the $j$-th constraint, appears at most $q$ times. This is because

$$\sum_{i=1}^N \frac{1}{q} \left\langle \hat{\lambda}^k - \lambda^k, A_i(x_i^k - \hat{x}_i^k) \right\rangle = \frac{1}{q} \sum_{i=1}^N \sum_{j=1}^m (\hat{\lambda}_j^k - \lambda_j^k) [A_i(x_i^k - \hat{x}_i^k)]_j$$

$$= \frac{1}{q} \sum_{j=1}^m (\hat{\lambda}_j^k - \lambda_j^k) \sum_{i=1}^N [A_i(x_i^k - \hat{x}_i^k)]_j.$$  

Thus, recalling that $q$ denotes the maximum number of non-zero blocks $[A_i]_j$ over all $j$, we can conclude that each term $\|\hat{\lambda}_j^k - \lambda_j^k\|^2$, $j = 1, \ldots, m$ appears at most $q$ times in the summation. This observation leads us to

$$\sum_{i=1}^N \frac{1}{q} \left\langle \hat{\lambda}^k - \lambda^k, A_i(x_i^k - \hat{x}_i^k) \right\rangle \geq -\frac{1}{2} \left( \sum_i \rho \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{q\rho} \|\hat{\lambda}^k - \lambda^k\|^2 \right).$$  

(3.118)

Finally, substituting (3.118) into (3.117) we get

$$\sum_i \rho \left( A_i(x_i^k - x_i^*) , A_i(x_i^k - \hat{x}_i^k) \right) + \frac{1}{\rho} \left\langle \hat{\lambda}^k - \lambda^k, x_i^k - \hat{x}_i^k \right\rangle$$

$$\geq \sum_i \frac{\rho}{2} \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{1}{2q\rho} \|\lambda^k - \hat{\lambda}^k\|^2 + \alpha^k,$$

which completes the proof. $\square$

Now we are ready to prove the a.s. convergence of our Lyapunov/Merit function $\phi(x^k, \lambda^k)$ by utilizing the almost-supermartingale convergence result from Theorem 33.

**Lemma 37.** Assume (A1)-(A7). Then, the sequence

$$\phi(x^k, \lambda^k) = \sum_{i=1}^N \rho \|A_i(x_i^k - x_i^*)\|^2 + \frac{1}{\rho} \|\lambda^k - \lambda^*\|^2$$  

(3.119)
generated by SADAL converges a.s. to some finite random variable $\bar{\phi}$. Moreover, for all $i = 1, \ldots, N$ we have

$$r(\bar{x}^k) \xrightarrow{L^2} 0$$

and $$A_i \bar{x}^k \xrightarrow{L^2} A_i x^k,$$

where $\xrightarrow{L^2}$ denotes mean-square convergence.

**Proof.** First, we show that the dual update step (3.6) in the SADAL method results in the following update rule for the variables $\bar{\lambda}^k$, which are defined in (3.115):

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \tau_k \rho r(\bar{x}^k) + \tau_k \sum_i u^{k+1}_i$$

Indeed,

$$\lambda^{k+1} = \lambda^k + \tau_k \rho r(\bar{y}^{k+1})$$

$$= \lambda^k + \tau_k \rho [r(\bar{y}^{k+1}) + \sum_i u^{k+1}_i]$$

$$= \lambda^k + \tau_k \rho [(1 - \frac{1}{q})r(x^k) + \frac{1}{q} r(\bar{x}^k) + \sum_i u^{k+1}_i]$$

$$= \lambda^k + \tau_k [- (1 - \frac{1}{q}) \rho (r(\bar{x}^k) - r(x^k)) + \rho r(\bar{x}^k) + \sum_i u^{k+1}_i]$$

$$= \lambda^k - (1 - \frac{1}{q}) \rho \tau_k (r(\bar{x}^k) - r(x^k)) + \tau_k \rho r(\bar{x}^k) + \tau_k \sum_i u^{k+1}_i,$$

where the third equality follows from the definition of the $\bar{y}^{k+1}$ variables in the primal update step of SADAL, cf. (3.90). Adding $(1 - \frac{1}{q}) \rho r(x^k)$ on both sides of the above relation and rearranging terms, we obtain

$$\lambda^{k+1} + (1 - \frac{1}{q}) \rho r(x^k) + \tau_k (r(\bar{x}^k) - r(x^k))$$

$$= \lambda^k + (1 - \frac{1}{q}) \rho r(x^k) + \tau_k \rho r(\bar{x}^k) + \tau_k \sum_i u^{k+1}_i.$$
The left hand side of the above is equal to $\hat{\lambda}^{k+1}$ by definition (3.115) and the fact that $r(x^{k+1}) = r(x^k) + \tau_k (r(\hat{x}^k) - r(x^k))$. Hence, we arrive at

$$
\hat{\lambda}^{k+1} = \lambda^k + (1 - \frac{1}{q})\rho r(x^k) + \tau_k \rho r(\hat{x}^k) + \tau_k \sum_i u_i^{k+1}
$$

as required. Using (3.120), we can now evaluate $\phi(x^{k+1}, \hat{\lambda}^{k+1})$ as

$$
\phi(x^{k+1}, \hat{\lambda}^{k+1}) = \sum_{i=1}^{N} \rho ||A_i(x_i^k - x_i^*)||^2 + \frac{1}{\rho} ||\hat{\lambda}^{k+1} - \lambda^*||^2
$$

After expanding the right hand side of the above relation, we get

$$
\phi(x^{k+1}, \hat{\lambda}^{k+1}) = \sum_{i=1}^{N} \rho ||A_i(x_i^k - x_i^*)||^2 + \frac{1}{\rho} ||\lambda^k - \lambda^*||^2
$$

After expanding the very last term on the above and recalling the definition $\hat{\lambda}^k = \lambda^k + \rho r(\hat{x}^k)$, cf. (3.100), we arrive at

$$
\phi(x^{k+1}, \hat{\lambda}^{k+1}) = \phi(x^k, \hat{\lambda}^k)
$$

$$
- 2\tau_k \left[ \rho \sum_{i=1}^{N} \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle - \frac{1}{\rho} \langle \hat{\lambda}^k - \lambda^*, \rho r(\hat{x}^k) + r(u^{k+1}) \rangle \right]
$$

$$
+ \tau_k^2 \left[ \sum_{i=1}^{N} \rho ||A_i(\hat{x}_i^k - x_i^k)||^2 + \frac{1}{\rho} ||\rho r(\hat{x}^k) + r(u^{k+1})||^2 \right].
$$
We use Lemma 36 to substitute the term \(-2\tau_k\rho\sum_{i=1}^N \langle A_i(x_i^k - x_i^\ast), A_i(x_i^k - \hat{x}_i^k) \rangle\) with its lower bound in the above relation, to arrive at

\[
\phi(x^{k+1}, \lambda^{k+1}) \leq \phi(x^k, \lambda^k) - \sum_i \rho(\tau_k - \tau_k^2) \|A_i(x_i^k - \hat{x}_i^k)\|^2 - \left(\frac{\tau_k}{q} - \tau_k^2\right) \frac{1}{\rho} \|\lambda^k - \hat{\lambda}^k\|^2 + \frac{2\tau_k}{\rho} \langle \hat{\lambda}^k - \lambda^*, r(u^{k+1}) \rangle + \frac{2\tau_k^2}{\rho} \langle \hat{\lambda}^k - \lambda^k, r(u^{k+1}) \rangle + \frac{\tau_k}{\rho} \|r(u^{k+1})\|^2 - 2\tau_k \alpha^k.
\]

(3.121)

Now, take the conditional expectation \(E_k\) (with respect to \(F_k\)) on the above relation. First, by the definition of \(F_k\) in (3.101), we get that

\[
E_k \langle \hat{\lambda}^k - \lambda^*, r(u^{k+1}) \rangle = \langle \hat{\lambda}^k - \lambda^*, E_k r(u^{k+1}) \rangle = 0,
\]

(3.122)

where in the last equality we used the zero-mean assumption (A6) for \(u^{k+1}\). Moreover, it is true that

\[
E_k \langle \hat{\lambda}^k - \lambda^k, r(u^{k+1}) \rangle = E_k r(\hat{x}^k)^T E_k r(u^{k+1}) = 0.
\]

(3.123)

This follows from the following facts. The terms \(r(\hat{x}^k)\), and \(\sum_i u_i^{k+1}\) are conditionally independent (recall from (3.96) that \(u^{k+1}\) denotes the noise in the message exchanges for the dual updates). Moreover, due to assumption (A3) the minimizers \(\hat{x}_i^k\) belong to compact sets for all \(i = 1, \ldots, N\), hence we have that \(E_k r(\hat{x}^k) < \infty\). Finally, assumption (A6) implies that \(E_k \sum_i u_i^{k+1} = 0\).

Thus, after taking the conditional expectation of \(\phi(x^{k+1}, \lambda^{k+1})\) with respect to \(F_k\), and using (3.121)-(3.123), we get that

\[
E_k \phi(x^{k+1}, \lambda^{k+1}) \leq \phi(x^k, \lambda^k) - E_k \left[ \sum_i \rho(\tau_k - \tau_k^2) \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \left(\frac{\tau_k}{q} - \tau_k^2\right) \frac{1}{\rho} \|\lambda^k - \hat{\lambda}^k\|^2 - \tau_k \rho \|r(u^{k+1})\|^2 + 2\tau_k \alpha^k \right].
\]

105
Consider the term $-2\tau_k \mathbb{E}_k \alpha^k$, and recall from (3.111) that
\[
\alpha^k = \sum_i \left[ \langle A_i (\hat{x}_i^k - x_i^k), \epsilon_i^k \rangle + f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \xi_i) - F_i(x_i^*, \xi_i) \right].
\]
By the definition of the functions $F_i$ and $f_i$, cf. assumption (A1), we have that
\[
\mathbb{E}_k \left[ f_i(x_i^*) - f_i(\hat{x}_i^k) + F_i(\hat{x}_i^k, \xi_i) - F_i(x_i^*, \xi_i) \right] = 0,
\]
for all $i = 1, \ldots, N$. Note that, according to the definition of the sub-$\sigma$-algebra $\mathcal{F}_k$ in (3.101), the conditional expectation in (3.124) is taken with respect to both the random variables $\hat{x}_i^k$ and $\xi_i^k$. Hence, to see why (3.124) is true, we need to consider the tower property of conditional expectation, which states that for some random variable $X$ and some sub-$\sigma$-algebras $\mathcal{S}_1 \subset \mathcal{S}_2 \subset \mathcal{F}$ we have $\mathbb{E}(X \mid \mathcal{S}_1) = \mathbb{E}(\mathbb{E}(X \mid \mathcal{S}_2) \mid \mathcal{S}_1)$. Now, recall that assumption (A1) essentially says that $\mathbb{E}(F_i(x_i, \xi_i) - f_i(x_i) \mid x_i) = 0$. Then, (3.124) holds true from the tower property for $\mathcal{S}_1 = \mathcal{F}_k$ and $\mathcal{S}_2 = \mathcal{F}_k \cup \sigma(\hat{x}_i^k)$.

Hence, we have that
\[
-2\tau_k \mathbb{E}_k \alpha^k = -2\tau_k \mathbb{E}_k \sum_i \left[ \langle A_i (\hat{x}_i^k - x_i^k), \epsilon_i^k \rangle \right]
\]
Now, by assumption (A6) we have that $\mathbb{E}_k (A_i x_i^*)^T \epsilon_i^k = 0 = \mathbb{E}_k (A_i x_i^k)^T \epsilon_i^k$, since $\mathbb{E}_k \epsilon_i^k = 0$ and the fact that $\epsilon_i^k$ and $x_i^k$ are conditionally independent given the definition of $\mathcal{F}_k$ in (3.101). Thus, we can substitute $A_i x_i^*$ with $A_i x_i^k$ in the term involving $\epsilon_i^k$ in the above relation, and then use the fact that $-2 \langle A_i (\hat{x}_i^k - x_i^k), \epsilon_i^k \rangle \leq \frac{1}{C} \| A_i (\hat{x}_i^k - x_i^k) \|^2 + C \| \epsilon_i^k \|^2$, for any $C < \infty$, to get
\[
\mathbb{E}_k \phi (x^{k+1}, \lambda^{k+1}) \leq \phi (x^k, \lambda^k) + \mathbb{E}_k \left[ \tau_k \sum_{i=1}^N \left( \frac{1}{C} \| A_i (\hat{x}_i^k - x_i^k) \|^2 + C \| \epsilon_i^k \|^2 \right) \right]
- \sum_i \rho (\tau_k - \tau_k^2) \| A_i (x_i^k - \hat{x}_i^k) \|^2 - \left( \frac{\tau_k}{\rho} - \tau_k^2 \right) \| \lambda^k - \hat{\lambda}^k \|^2 + \tau_k^2 \rho \| r(u^{k+1}) \|^2 \right],
\]
106
which, after rearranging terms, can be expressed as
\[
\mathbb{E}_k \phi(x^{k+1}, \lambda^{k+1}) \leq \phi(x^k, \lambda^k) + \frac{1}{\rho} (\tau_k^2 - \frac{\tau_k}{q}) \mathbb{E}_k \|\lambda^k - \hat{\lambda}^k\|^2 + \left[\rho \tau_k^2 - (\rho - \frac{1}{C}) \tau_k\right] \mathbb{E}_k \sum_{i=1}^N \|A_i(x_i^k - \hat{x}_i^k)\|^2 + C \sum_{i=1}^N \tau_k \mathbb{E}_k \|e_i^k\|^2 + \rho \tau_k^2 \frac{\mathbb{E}_k \|r(u^{k+1})\|^2}{\rho}.
\] (3.125)

We can now recall Theorem 33 and observe that relation (3.125) is of the form (3.103), with
\[
\eta_k = 0,
\]
\[
\chi_k = C \sum_{i=1}^N \tau_k \mathbb{E}_k \|e_i^k\|^2 + \rho \tau_k^2 \frac{\mathbb{E}_k \|r(u^{k+1})\|^2}{\rho} + \rho \tau_k^2 \mathbb{E}_k \sum_{i=1}^N \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{\tau_k^2}{\rho} \mathbb{E}_k \|\lambda^k - \hat{\lambda}^k\|^2,
\]
\[
\psi_k = (\rho - \frac{1}{C}) \tau_k \mathbb{E}_k \sum_{i=1}^N \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \frac{\tau_k}{q \rho} \mathbb{E}_k \|\lambda^k - \hat{\lambda}^k\|^2.
\]

By assumption (A5), we have that \((\rho - \frac{1}{C}) > 0\), and by assumption (A4), we have that \(\frac{\tau_k}{q} > 0\). Hence, the variable \(\psi_k\) is nonnegative at all times, as required for the application of Theorem 33. Moreover, by assumption (A6), we have that \(\mathbb{E}_k \|r(u^{k+1})\|^2 \leq M_1 < \infty\) for all iterations \(k = 1, 2, \ldots\), and by assumption (A4) we have that \(\sum_{k=1}^\infty \tau_k^2 < \infty\), from which we infer that \(\sum_{k=1}^\infty \rho \tau_k^2 \mathbb{E}_k \|r(u^{k+1})\|^2 < \infty\).

Furthermore, note that the random variables \(\|A_i(x_i^k - \hat{x}_i^k)\|^2\) for all \(i \in \mathcal{I}\), and \(\|\lambda^k - \hat{\lambda}^k\|^2 = \|\rho r(\hat{x}_i^k)\|^2\) are bounded for every \(k\). This is because the iterates \(\hat{x}_i^k\) belong to compact sets for every \(i \in \mathcal{I}\) and all \(k\), due to assumption (A3). From the fact that \(x_i^{k+1}\) is a convex combination between \(\hat{x}_i^k\) and \(x_i^k\), cf. (3.90), and given that the initial value \(x_i^1\) is bounded, it is straightforward to show by induction that the sequences \(x_i^k\) remain bounded for every \(i \in \mathcal{I}\). Hence, we have that \(\mathbb{E}_k \sum_{i=1}^N \|A_i(x_i^k - \hat{x}_i^k)\|^2 \leq M_2 < \infty\), and \(\mathbb{E}_k \|\lambda^k - \hat{\lambda}^k\|^2 \leq M_3 < \infty\) for all iterations \(k = 1, 2, \ldots\). We infer
that $\sum_{k=1}^{\infty} \left[ \rho \sum_{i=1}^{N} ||A_i(x_i^k - \hat{x}_i^k)||^2 + \frac{\tau^2}{\rho} \sum_{k=1}^{\infty} ||\lambda^k - \hat{\lambda}^k||^2 \right] < \infty$. In addition, by assumption (A7) we have that $C \sum_{i=1}^{N} \sum_{k=1}^{\infty} \tau_k E_k ||\epsilon_i^k||^2 < \infty$. These facts combined lead to $\sum_{k=1}^{\infty} \chi_k < \infty$.

Thus, the conditions of Theorem 33 are satisfied. We conclude that the sequence $\{\phi(x^k, \lambda^k)\}$ converges almost surely to some finite random variable $\bar{\phi}$. By Theorem 33, we also have that

$$\sum_{k=1}^{\infty} \left[ (\rho - \frac{1}{C}) \tau_k E_k \sum_{i=1}^{N} ||A_i(x_i^k - \hat{x}_i^k)||^2 + \frac{\tau_k}{q\rho} E_k ||\lambda^k - \hat{\lambda}^k||^2 \right] < \infty.$$ 

The random variables $||A_i(x_i^k - \hat{x}_i^k)||^2$ for all $i \in I$, and $||\lambda^k - \hat{\lambda}^k||^2 = ||\rho r(x^k)||^2$ are integrable, due to the aforementioned arguments about their boundedness. Hence, we can use the law of iterated expectation on the above relation which, combined with assumption (A4) that states $\sum_{k=1}^{\infty} \tau_k = \infty$, finally gives us that $r(x^k) \overset{L^2}{\longrightarrow} 0$ and $A_i \hat{x}_i^k \overset{L^2}{\longrightarrow} A_i x_i^k$ for all $i = 1, \ldots, N$.

We are now ready to prove the main result of this section. The central idea behind the following proof is to show that there exists a subsequence over which $\{\phi(x^k, \lambda^k)\}$ converges almost surely to zero. Then, we use the result of lemma 37, which states that $\{\phi(x^k, \lambda^k)\}$ converges a.s. over all $k$ to a finite limit, to infer that the generated sequences of primal and dual variables converge to their respective optimal sets almost surely over all $k$.

**Theorem 38.** Assume (A1)-(A8). Then, the SADAL method generates sequences of dual variables $\{\lambda^k(\omega)\}$ that converge to an optimal solution of problem (2.28) for almost all $\omega \in \Omega$. Moreover, any sequence $\{x^k(\omega)\}$ generated by SADAL has an accumulation point and any such point is an optimal solution of problem (3.1) for almost all $\omega \in \Omega$. 

108
Proof. In Lemma 37, we proved that \( r(\hat{x}^k) \xrightarrow{L^2} 0 \) and \( A_i \hat{x}_i^k \xrightarrow{L^2} A_i x_i^k \) for all \( i = 1, \ldots, N \). It is known that the mean square convergence of a sequence of random variables implies convergence in probability, which in turn implies that there exists a subsequence such that a.s. convergence holds. Hence, there exists a subsequence \( \mathcal{K}_1 \subset \mathcal{K} \) such that \( A_i \hat{x}_i^k \xrightarrow{a.s.} A_i x_i^k \) for \( k \in \mathcal{K}_1 \) holds. Similarly, \( r(\hat{x}^k) \xrightarrow{L^2} 0 \) over \( \mathcal{K}_1 \), which in turn means that there exists a sub-subsequence \( \mathcal{K}_2 \subset \mathcal{K}_1 \) such that \( r(\hat{x}^k) \xrightarrow{a.s.} 0 \) for \( k \in \mathcal{K}_2 \). Hence, we have that \( \{r(\hat{x}^k(\omega))\}_{k \in \mathcal{K}_2} \) and \( \{A_i \hat{x}_i^k(\omega) - A_i x_i^k(\omega)\}_{k \in \mathcal{K}_2} \) converge to zero for almost all \( \omega \). Combining these two results, we infer that \( r(x^k) \xrightarrow{a.s.} 0 \) over \( \mathcal{K}_2 \), also.

Recall from (3.92) that the update law for the dual sequence is \( \lambda^{k+1} = \lambda^k + \rho \tau_k \left( \sum_{i=1}^{N} A_i y_i^{k+1} - b \right) \), where \( A_i y_i^{k+1} = A_i y_i^{k+1} + u_i^{k+1} \), cf. (3.96). Combining these two, we have that

\[
\lambda^{k+1} = \lambda^k + \rho \tau_k r(y^{k+1}) + \rho \tau_k \sum_{i=1}^{N} u_i^{k+1}. \tag{3.126}
\]

By definition (3.91), it holds that \( A_i y_i^{k+1} = A_i x_i^k + \frac{1}{q}(A_i \hat{x}_i^k - A_i x_i^k) \). Thus, using the results that \( r(x^k) \) and \( r(\hat{x}^k) \) converge a.s. to zero over \( \mathcal{K}_2 \), we infer that \( r(y^k) \xrightarrow{a.s.} 0 \) over \( \mathcal{K}_2 \). Moreover, from Chebyshev’s inequality we have that for every \( \delta > 0 \) the following holds

\[
P(\rho \tau_k u_i^{k+1} \geq \delta) \leq \frac{\rho^2 \tau_k^2 E\|u_i^{k+1}\|^2}{\delta^2}, \tag{3.127}
\]

From assumption (A6) we have that \( E\|u_i^{k+1}\|^2 \leq M_4 < \infty \) for all \( k \) which, combined with assumption (A4), gives us that \( \sum_{k=1}^{\infty} \rho^2 \tau_k^2 E\|u_i^{k+1}\|^2 < \infty \). This implies that \( \sum_{k=1}^{\infty} P(\rho \tau_k u_i^{k+1} \geq \delta) < \infty \), and by the Borel-Cantelli lemma this means that \( \rho \tau_k u_i^{k} \xrightarrow{a.s.} 0 \). This, combined with the previous result that \( r(y^k) \xrightarrow{a.s.} 0 \) over \( \mathcal{K}_2 \) and the a.s. convergence of \( \{\phi(x^k, \lambda^k)\} \) from Lemma 37, gives us that the dual sequence \( \{\lambda^k\}_{k \in \mathcal{K}_2} \) (3.126) converges a.s. to some finite limit \( \mu \), i.e., \( \lambda^k \xrightarrow{a.s.} \mu \) over \( \mathcal{K}_2 \).
From assumption (A3), all sequences \( \{ \hat{x}_i^k(\omega) \}, i = 1, \ldots, N \), are bounded. This, combined with the fact that the \( x_{i+1}^k \) is a convex combination between \( \hat{x}_i^k \) and \( x_i^k \), cf. (3.90), and given that the initial value \( x^1 \) is bounded, means that the sequences \( \{ x_i^k(\omega) \}, i = 1, \ldots, N \), are bounded also. This in turn implies that the sequences \( \{ \hat{x}_i^k(\omega) \} \) have accumulation points \( \bar{x}_i(\omega) \), which are also accumulation points of \( \{ x_i^k(\omega) \} \) due to the update step (3.90) of SADAL. We can choose a subsequence \( K_3 \subset K_2 \) so that \( \{ x_i^k(\omega) \}_{k \in K_3} \) and \( \{ \hat{x}_i^k(\omega) \}_{k \in K_3} \) converge to \( \bar{x}_i(\omega) \) for all \( i = 1, \ldots, N \). Denoting \( \bar{x}(\omega) = [\bar{x}_1(\omega), \ldots, \bar{x}_N(\omega)]^T \), we observe that the point \( \bar{x}(\omega) \) is feasible due to the closedness of the sets \( X_i \) and the continuity of \( r(\cdot) \).

For any \( i = 1, \ldots, N \), consider the sequence \( \{ s_{F_i}^k(\omega) \}_{k \in K_3} \). The subdifferential mapping \( x \mapsto \partial f(x) \) of any finite-valued convex function defined on \( \mathbb{R}^n \) is upper semi-continuous and has compact images. Therefore, the sequences \( \{ s_i^k(\omega) \}_{k \in K_3} \) have convergent subsequences due to a fundamental result that goes back to Berge (1959). We can choose \( K_4 \subset K_3 \) such that \( \{ s_i^k(\omega) \}_{k \in K_4} \) converge to some \( \bar{s}_i(\omega) \subset \partial f_i(\bar{x}_i(\omega)) \) for all \( i = 1, \ldots, N \) and almost all \( \omega \).

We recall that the optimality conditions for each subproblem \( i = 1, \ldots, N \), cf. (3.105), take the form

\[
0 = s_{F_i}^k + A_i^\top \bar{x}_i^k + \rho A_i^\top \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j \bar{x}_j^k - b \right) + z_i^k. \tag{3.128}
\]

Gathering all the noise terms, the above equation can be equivalently expressed as

\[
0 = s_{F_i}^k + A_i^\top \bar{x}_i^k + \rho A_i^\top \left( A_i \hat{x}_i^k + \sum_{j \neq i} A_j x_j^k - b \right) + e_i^k + w_i^k + \rho \sum_{j \neq i} v_{ij}^k + z_i^k. \tag{3.128}
\]

From assumptions (A7) and (A8), we have that the noise terms \( e_i^k + w_i^k + \rho \sum_{j \neq i} v_{ij}^k \) converge to \( 0 \) a.s. as \( k \to \infty \). Hence, passing to the limit over \( K_4 \) in (3.128), we infer that each sequence \( \{ z_i^k(\omega) \}_{k \in K_4} \) converges to a point \( z_i(\omega) \), for all \( i = 1, \ldots, N \).
and almost all $\omega$. The mapping $x_i \mapsto \mathcal{N}_{x_i}(x_i)$ has closed graph and, hence, $z_i(\omega) \in \mathcal{N}_{x_i}(\bar{x}_i(\omega))$ (Ruszczynski, 2006, Lemma 2.42). After the limit pass in (3.128) over $k \in \mathcal{K}_4$, we conclude that

$$
0 = \bar{s}_i(\omega) + A_i^T \bar{\mu}(\omega) + \bar{z}_i(\omega), \quad \forall \ i = 1 \ldots, N.
$$

for almost all $\omega$. These relations are exactly the optimality conditions of each $i \in \mathcal{I}$ for the saddle point of the original problem (3.1), cf. (3.108). This result, together with the feasibility of $\bar{x}(\omega)$, implies that $\bar{x}(\omega)$ is a solution of the primal problem (3.1) and $\bar{\mu}(\omega)$ is a solution of the dual problem (2.28).

Due to the continuity of $\phi(\cdot)$, it follows that $\{\phi(x^k, \lambda^k)\}$ converges a.s. to zero over $\mathcal{K}_4$. Combining this with the result that $\{\phi(x^k, \lambda^k)\}$ converges a.s. to some finite limit for all $k = 1, 2, \ldots$ from Lemma 37, we infer that $\{\phi(x^k, \lambda^k)\}$ converges a.s. to zero for all $k = 1, 2, \ldots$. This further implies that the terms $\|A_i x^k_i - A_i x^*_i\|^2$ for all $i \in \mathcal{I}$ and $\|\lambda^k - \lambda^*\|^2$ converge a.s. to zero for all $k = 1, 2, \ldots$, due to the nonnegativity of all these terms in $\phi(\cdot)$. Hence, we infer that $A_i x^k_i - A_i x^*_i \xrightarrow{a.s.} 0$ for all $i \in \mathcal{I}$, and $\lambda^k \xrightarrow{a.s.} \lambda^*$. Combining the result that $A_i x^k_i - A_i x^*_i \xrightarrow{a.s.} 0$ for all $i \in \mathcal{I}$ with the definition $r(x^k) = \sum_i A_i x^k_i - b$ and the fact that the optimal solution is feasible $\sum_i A_i x^*_i = b$, we get that $r(x^k) \xrightarrow{a.s.} 0$. Hence, after recalling that $\lambda^k = \lambda^k + \rho(1 - \frac{1}{q})r(x^k)$ by the definition (3.115), we can use the results $\lambda^k \xrightarrow{a.s.} \lambda^*$ and $r(x^k) \xrightarrow{a.s.} 0$ to infer that $\lambda^k$ converges a.s. to $\lambda^*$ for all $k = 1, 2, \ldots$, as required.
In this chapter we discuss some applications of the ADAL method. We also present numerical results and compare the performance of ADAL with that of the current state-of-the-art methods in distributed optimization, the ADMM and DQA (see sections 2.4.2 and 2.4.3, respectively, for a brief overview of these methods). We examine problems in convex, non-convex, and stochastic settings in an effort to validate the novelty and effectiveness of the proposed method in all the areas of the aforementioned theoretical contributions that were presented in Chapter 3.

4.1 Network Utility Maximization Problems

First, we consider Network Utility Maximization (NUM) problems. Consider an undirected graph \( G = (N, A) \) with a set of nodes \( N \) and a set of arcs \( A \). The set of nodes is consisted of two subsets \( N = \{S, D\} \), where \( S \) is the set of source nodes and \( D \) the set of destination nodes. Let \( s_i \) denote the rate of resource production at node \( i \in S \) and also let \( t_{ij} \) denote the rate of a commodity flowing through arc \((i, j)\). Each arc \((i, j)\) has a feasible range of flows \( a_{ij} \leq t_{ij} \leq b_{ij} \), where \( a_{ij}, b_{ij} \) are given numbers. Denote the neighborhood of node \( i \) as \( C_i = \{j : (i, j) \in A\} \). At this point, note that
\[ q = \max_i |C_i| \] and, according to the convergence analysis, the stepsize in the ADAL algorithm must be \( \tau < \frac{1}{q} \). Nevertheless, the numerical experiments indicate that significant acceleration is achieved if the stepsizes for the primal and dual updates are relaxed to \( \tau = \frac{\beta_p}{q} \) and \( \tau = \frac{\beta_d}{q} \), respectively, where \( \beta_p \in [1, 2.5) \) and \( \beta_d \in [1, q) \).

The NUM problem entails solving

\[
\text{(NUM)} \quad \begin{align*}
\max \quad & U(s) = \sum_{i \in S} U_i(s_i) \\
\text{subject to} \quad & \sum_{j \in C_i} t_{ij} - \sum_{j \in C_j} t_{ji} = s_i, \quad \forall \ i \in S \\
& a_{ij} \leq t_{ij} \leq b_{ij}, \quad \forall \ (i, j) \in A
\end{align*}
\]

The NUM problem maximizes the amount of resources produced at the source nodes and route the resources to the destination nodes. The constraints \( \sum_{j \in C_i} t_{ij} - \sum_{j \in C_j} t_{ji} = s_i \) express the conservation of commodity flow at each source node. Note that, in our consideration, the destination nodes are modeled as sinks and can absorb any amount of incoming rates. We consider normalized rates \( s_i, t_{ij} \in [0, 1] \), without any loss of generality.

The requirement on the utility functions \( U_i(s_i) \) is that they are monotonically non-decreasing expressing preference to larger transmission rates, i.e. an increase in the rate of one node does not decrease the value of the total utility function to be maximized. In our simulations, we choose \( U(s) = \prod_{i \in S}(s_i) \) in order to maximize the product of rates, which can be recast as the sum of logarithms \( U(s) = \sum_{i \in S} \log(s_i) \). This choice is typical in NUM problems Shakkottai and Srikant (2007) and aims to produce a fairer resource allocation among all source nodes in the network. Note that the choice \( U(s) = \sum_{i \in S} s_i \), would result in a problem where the maximum rates are rewarded. In our case, this would lead to a trivial problem, in which the nodes in communication range of the destinations are rewarded with the maximum rate 1 and the rest with 0.
Fig. 4.1 shows a network consisting of 50 sources and 2 sinks that is returned after applying the ADAL algorithm on (NUM). All subsequent simulation results involve networks of this form, unless otherwise noted. Fig. 4.2 shows the evolution of the individual and total rates, $s_i$ and $\sum_{i \in S} s_i$, respectively, corresponding to maximization of the utility $U(s) = \sum_{i \in S} \log(s_i)$ for fair allocation. We observe that the utility converges sufficiently to its optimal value in only about 25 iterations. Next, we plot in Fig. 4.3 the evolution of the maximum residual during the execution of ADAL. The figure contains results for networks of different sizes. An encouraging observation is that network size does not appear to affect speed of convergence dramatically, at least for the NUM problem considered here. Repeated simulations have shown that convergence speed remains at this level of magnitude. Note that, in all cases, the ratio of sources-to-sinks has been maintained the same at 25/1, in an effort to keep the randomly generated networks as similar as possible.

As already mentioned, ADAL can be viewed as a truncated form of the DQA algorithm. In Fig. 4.4, we explore the connections between the two methods. Fig.4.4(a)
Figure 4.2: Evolution of the sum of rates $\sum_{i \in S} s_i$ during implementation of ADAL. The horizontal line depicts the value obtained by solving the centralized problem. We observe that the distributed utility function converges to the optimal solution very fast. Also included is the subfigure illustrating the evolution of individual rates for every source.

comparis results of truncating the inner loop of DQA at a predefined number of iterations $M$. For $M = 1$, we obtain the ADAL method. We observe that truncating the inner loop yields accelerated convergence for DQA, with the fastest case being the ADAL algorithm. Note that no theoretical proof for the convergence of DQA for intermediate values of $M$ exists. Moreover, since the performance of both algo-

Figure 4.3: Constraint violation convergence of ADAL for different network sizes of 25, 50, 100 and 400 source nodes. The ratio of sources-to-destinations is kept at $25/1$ and the maximum degree $q = 6$ for all cases.
Figure 4.4: Constraint violation convergence for: a) Different exit criteria from the inner loop of DQA. The line labeled 'open' accounts for repeating the DQA inner loop until \( \| A_i \hat{x}_i^k - A_i x_i^k \|_\infty \leq 10^{-2} \) for every \( i = 1, \ldots, N \). In the other instances we force exit if either the aforementioned criterion is satisfied or if the indicated amount of iterations \( M \) has been surpassed. For \( M = 1 \) we obtain the ADAL method. The results correspond to a network of 50 sources and 2 sinks with \( q = 6 \). b) Different network densities for the ADAL and DQA methods. The results correspond to networks of 50 sources and 2 sinks with \( q = 5, 9, 14 \), respectively. In both figures, the horizontal axis depicts the inner loop iterations for the case of DQA. Also, note that the step shape of the DQA graphs in the figures is caused by the dual updates at each outer loop iteration.

Figure 4.5: Comparison between the ASM, DQA and ADAL methods, for a network of 50 sources, 2 sinks and \( q = 7 \).
rithms appears to depend on the maximum degree $q$ we have conducted simulations for different values of $q$. The results are illustrated in Fig. 4.4(b). Surprisingly and in contrast to DQA, it appears that ADAL is not greatly affected by the value of $q$, at least for the NUM problems considered here.

In Fig. 4.5, we compare the performance of the three methods. The ASM was implemented for $\sigma = 1.9$, the maximum value of $\sigma$ that did not compromise convergence, while returning the best results. Also, in all three methods the penalty parameter and initialization points were the same, in order to preserve the homogeneity of results. We notice that ADAL performs significantly better than both ASM and DQA.

Finally, we examine how sensitive ADAL is to the choice of the user-defined penalty coefficient $\rho$ for the NUM problem under consideration here. Fig. 4.6 depicts simulation results for a problem with 50 source nodes and 2 sinks. We let $\rho$ take the values 1, 5, 10, 20, 40, and 100. We observe that, for this particular problem, ADAL exhibits a very robust convergence behavior that is not significantly affected by the choice of $\rho$. We can safely choose $\rho$ from a very wide range of values without affecting
either the objective function or the constraint violation convergence patterns, apart from the smallest value case $\rho = 1$ which leads to slower convergence overall.

4.2 Network Flow Problems

Closely related to the NUM problem is the optimal network flow problem. Here, we examine the Linear Network Flow (LNF) case, where the arc costs are linear. LNF is a classical problem that has been studied extensively. The assignment, max-flow and shortest path problems are special cases of LNF Bertsekas and Tsitsiklis (1997).

Consider a directed graph $G = (N, A)$, with a set of nodes $N$ and a set of arcs $A$. Each arc $(i, j)$ has associated with it a scalar $c_{ij}$ referred to as the cost coefficient of $(i, j)$. Let $t_{ij}$ denote the flow of arc $(i, j)$ and consider the problem

\[
\begin{aligned}
\text{(LNF)} & \quad \min \sum_{(i,j) \in A} c_{ij} t_{ij} \\
& \text{subject to} \quad \sum_{\{j|(i,j) \in A\}} t_{ij} - \sum_{\{j|(j,i) \in A\}} t_{ji} = s_i, \quad \forall \ i \in N \\
& \quad a_{ij} \leq t_{ij} \leq b_{ij}, \quad \forall \ (i, j) \in A
\end{aligned}
\]

Essentially, the difference here is that we do not seek to maximize the $s_i$ production rates as in the NUM, but rather set some desired levels of $s_i$ and seek to find the flows that keep the problem feasible while minimizing the total cost. Moreover, the objective function is linear. For the set of $S$ source nodes, we have $s_i > 0$, $\forall \ i \in S$, while for the set of $D$ destination nodes we have $s_i < 0$, $\forall \ i \in D$. The conservation of flow in the network requires that $\sum_{i \in N} s_i = 0$. In the examples shown below, we also set a set of $R$ nodes to be relays, that is $s_i = 0$, $\forall \ i \in R$. In addition, we set the cost coefficients $c_{ij} = 1$ and the arc flow bounds $0 \leq t_{ij} \leq 1$ for simplicity, without any loss of generality.

Fig. 4.7 depicts the two typical, 50 node networks that were considered here. Also shown are the corresponding flows as solved by the ADAL method. In Fig.
Figure 4.7: Two typical LNF cases considered, with $N = 50$ nodes. Blue dots denote source nodes, while green dots correspond to sinks and red dots to relays. The flow $t_{ij}$ through arc $(i, j)$ defines the thickness of the corresponding drawn line. Thin lines indicate weaker links. a) For this case, $S = D = 5$, $R = 40$ and the source nodes are positioned to be as far away from the destinations as possible. b) For this case, $S = D = 7$, $R = 36$ and the source nodes are only half the network diameter apart from the destinations.

4.7(a) we consider a case with 5 sources and 5 destinations, which are set a network diameter apart. On the other hand, Fig. 4.7(b) depicts a case with 7 sources and 7 destinations, which are set half the network diameter apart. In Fig. 4.8, we plot the evolution of the objective functions after applying DQA, ASM and ADAL on the two networks depicted in Fig. 4.7. The evolutions of the respective maximum residuals are shown in Fig. 4.9.

Next, we consider a larger network of 400 nodes with 20 source-destination pairs, which are set a network diameter apart, and maximum degree $q = 8$; see Fig. 4.10(a). In Fig. 4.10(b) we plot the evolution of the sequence $\phi(t^k, \lambda^k)$ to verify the correctness of our proof. The sequence is strictly monotonically decreasing at each iteration, as required. Fig. 4.11 presents the corresponding convergence results for ADAL.
Figure 4.8: Evolution of the sum of flows $\sum_{(i,j) \in A} t_{ij}$ after implementation of ADAL, ASM and DQA on: a) the case depicted in Fig. 4.7(a) and b) the case depicted in Fig. 4.7(b). The horizontal lines depict the objective function values obtained after solving the corresponding centralized problem. Note how the ASM oscillates between positive and negative values of the total flow, which normally should not be the case (since $t_{ij} \geq 0$). This is caused by the fact that we have used stepsize $\sigma = 1.9$ (cf. (2.41)) in our simulations. This choice of $\sigma$ returned the fastest convergence for ASM, even though it lead to this “counter-intuitive” behavior.

Figure 4.9: Evolution of the maximum residual after implementation of ADAL, ASM and DQA on: a) the case depicted in Fig. 4.7(a) and b) the case depicted in Fig. 4.7(b).
Figure 4.10: a) A network with $N = 400$ and $S = D = 20$. b) Evolution of $\phi(t^k, \lambda^k)$ for ADAL applied on the aforementioned network.

Figure 4.11: Convergence results for ADAL, ASM and DQA applied on the 400 node network depicted in Fig. 4.10(a): a) Objective function convergence, and b) Constraint violation convergence.
ASM and DQA. We observe that ADAL is still significantly faster than both the ASM and DQA,

Next, we examine how sensitive ADAL is to the choice of the user-defined penalty coefficient $\rho$ for the problem under consideration here. Fig. 4.12 depicts simulation results for a 100 node problem with 10 sources and 10 destinations, which are placed a full network diameter apart (similar to the configuration shown in Fig. 4.10(a)). We let $\rho$ take the values 0.1, 0.3, 0.5, 1, 3, and 10. We observe that, for this particular
Figure 4.13: Simulation results of ADMM for different values of $\rho$, on a 100 node problem with 10 sources and 10 destinations. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence.

problem, the choice of $\rho$ has a significant impact on the convergence behavior of ADAL. Small values of $\rho$, e.g. 0.1 and 0.3, will cause a very slow convergence of the constraint violation. On the other hand, larger values of $\rho$, e.g. 3 and 10, will accelerate the overall convergence, however, the problem becomes ill-conditioned and the objective function converges to a suboptimal value. For $\rho$ taking the values 0.5 and 1, we obtain the desired behavior, with the value 1 appearing to be the optimal choice here. For comparison purposes, in Fig. 4.13 we present the equivalent simulation results for the ADMM algorithm. We observe that the sensitivity pattern
in terms of the $\rho$ choice is similar to ADAL, albeit ADMM converges significantly slower in all cases.

A possible modification of DQA, ASM and ADAL is to implement these methods in a “Gauss-Seidel” fashion, where the corresponding minimization steps of each method are performed in a sequential fashion for every $i = 1, \ldots, N$. In this case, the dual updates are performed after all the local minimization steps have been completed, or in the case of DQA after the inner loop has terminated. The convergence proofs of all methods are only valid for the “Jacobi” type implementation, where the minimization steps are executed in parallel, however, it is interesting to compare the relative performances between these two approaches. Towards this goal, Fig. 4.14 depicts the convergence results after applying the Gauss-Seidel counterparts of DQA, ASM and ADAL on the network depicted in Fig. 4.7(b). Note that the respective results for the “normal” Jacobi versions of these methods are depicted in Fig. 4.8(b) and Fig. 4.9(b). We observe that the Gauss-Seidel versions converge faster, in terms of the number of iterations, than the Jacobi ones for all algorithms. Nevertheless, the sequential nature of the Gauss-Seidel means that in applications where paral-
lel computation is available, the Jacobi type implementations will converge faster in terms of real-time computation, with the difference increasing for increasing problem sizes. Moreover, the sequential execution of the Gauss-Seidel counterparts requires some form of global coordination to maintain the pre-defined ordering for the local minimizations and perform the dual updates at the completion of a full Gauss-Seidel iteration, which hinders the distributed applicability of such approaches in practical settings.

4.3 Two-Stage Stochastic Capacity Expansion problems

In this section we are concerned with a class of stochastic multicommodity network flow problems, the so called capacity expansion planning problems. We consider a two-stage stochastic optimization formulation that incorporates uncertainty in the problem parameters. To address the computational complexity of these stochastic models, we propose a decomposition method to divide the original problem into smaller, tractable subproblems that are solved in parallel at the network nodes. Unlike relevant techniques in existing literature that decompose the problem with respect to the possible realizations of the random parameters, our approach can be applied to networked systems that lack a central processing unit and instead perform all decision making at the network nodes.

The literature on multicommodity flow problems (MFP) is vast and dates back to the early 1960s Ford and Fulkerson (1962); Hu (1963); Assad (1978); Ouorou et al. (2000). Problems of this form emerge in cases where several different commodities need to travel from a number of origins to a number of specific destinations along the arcs of an underlying network, subject to supply restrictions, arc capacity restrictions, and flow conservation conditions. Typical areas of MFP applications involve transportation, communication and data networks, and product distribution and logistics systems.
MFPs can be solved using general optimization methods. However, their special structure also allows for the application of decomposition methods that are particularly attractive as they divide the original problem into smaller subproblems that are more efficient to solve. Various decomposition techniques have been applied to linear MFPs, such as the dual subgradient method Gondran and Minoux (1979), the Dantzig-Wolfe decomposition algorithm Jones et al. (1993), and other partitioning methods Farvolden et al. (1993). Some popular algorithms for nonlinear, convex MFPs are the Flow Deviation Fratta et al. (1973), the Projected Newton Bertsekas and Gafni (1983), the Analytic Center Cutting Plane Goffin et al. (1997), and the Proximal Decomposition Mahey et al. (1998) methods.

Classical, deterministic MFP formulations, such as the above, cannot account for uncertainties in the problem parameters. This may lead to inferior, or even inaccurate, solutions in most real world applications, where the flow demands and the arc travel attributes may be random, evolve over time, or get affected by disturbances. Hence, it is desirable to develop stochastic formulations in which decisions are evaluated against a variety of future scenarios that represent alternative outcomes of the MFP’s parameters. The most common approach involves representing the uncertain quantities as random variables, and formulating the MFP as a multi-stage stochastic optimization problem, wherein the uncertain problem parameters are allowed to evolve over multiple time periods (stages) Shapiro et al. (2009); Ruszczynski and Shapiro (2003). At each stage, decisions need to be made prior to observing specific realizations of these random variables, based only on the information that has been revealed thus far.

In this section, we focus on a two-stage stochastic model for the nonlinear network capacity expansion problem Chang and Gavish (1993); Mollaghasemi et al. (1998); Li and Ierapetritou (2012); Chen et al. (2002), which is a popular, specialized class within the family of MFPs. Such problems arise in many and diverse applications
such as communication networks Chang and Gavish (1993); Mollaghanemi et al. (1998), heavy process industries Sahinidis and Grossman (1992), electric utilities Murphy and Weiss (1990), product production scheduling Chen et al. (2002), service industries Berman and Ganz (1994), chemical process planning Li and Ierapetritou (2012), electronics and semiconductor industries Rajagopalan (1994) and many more. We specifically consider the two-stage model since it is the most basic formulation for a stochastic optimization problem. Also, compared to multi-stage schemes, two-stage models result in simpler stochastic optimization problems of smaller size as they assume that all uncertain parameters are revealed at the same time, following the selection of first-stage decisions Shapiro et al. (2009); Ruszczynski and Shapiro (2003). For our particular network capacity expansion problem, at the first stage, decisions to allocate capacities on the given set of arcs must be made, before observing specific realizations of any random parameters. Once the first-stage capacity allocation decisions are made, the values of all uncertain quantities are revealed and the flow routing decisions of the second-stage must be made. These second-stage flow variables depend not only on the first-stage capacity decisions, but also on the realization of the uncertain parameters.

There is considerable literature on methods to solve stochastic nonlinear network capacity expansion problems, that dates back to approaches which rely on stochastic control theory Manne (1961); Bean et al. (1992). Nevertheless, even the simplest stochastic formulations are significantly more complex than their deterministic counterparts. Their computational complexity grows multiplicatively with the size of the network, the number of decision stages, and the number of realizations of the uncertain parameters. This motivates the use of decomposition methods as the only effective alternative to address the increased complexity. We refer here to some of the the most popular decomposition methods which have found application on the capacity expansion problem. These include the L-shaped decomposition Van Slyke and
Wets (1969); Birge (1985); Laporte and Louveaux (1993), the Diagonal Quadratic Approximation (DQA) Mulvey and Ruskčzyński (1995), the Auxiliary Problem Principle Cohen (1980), the Progressive Hedging Rockafellar and Wets (1991) and the Dantzig-Wolfe decomposition Singh et al. (2009). Some representative literature on algorithms that can handle integer variables can be found in Laporte and Louveaux (1993); Singh et al. (2009); Li and Ierapetritou (2012); Ahmed et al. (2003); Ahmed and Sahinidis (2003); Huang and Ahmed (2009).

The aforementioned decomposition methods render the problem tractable typically by decomposing it over the set of realizations of the random variables, also called scenarios. Instead, in this section we propose a distributed optimization technique that decomposes the computation of the network variables over the set of nodes, so that every node computes only the capacity and flow variables that are local to itself. The advantage of our method is that it can be widely applied to address problems in distributed communications, sensor, and robotic networks, where the nodes need to operate independently and autonomously in the absence of a central processing unit. Our method retains the benefit of dividing the large centralized problem into tractable, smaller subproblems solved in parallel at the node locations.

4.3.1 Two-Stage Stochastic Optimization Problems

We begin this section with a brief introduction of two stage stochastic optimization problems. The basic idea is that a decision must be made ‘here-and-now’ for the first stage decision variables, before a realization of the corresponding random variables becomes known, so that they optimize the expectation of the optimal value of the second-stage problem. In turn, at the second stage, the optimal value for each specific realization of the random parameters is determined, after solving the corresponding deterministic optimization problem. Note that the second stage decisions depend not only on the realization of the random components, but also on the first stage
decisions. The classical two-stage stochastic linear program can be written as follows

$$\min_x c^T x + \mathbb{E} [Q(x, \xi(\omega))]$$

s.t. $Ax = b, x \geq 0,$ (4.3)

where $Q(x, \xi)$ is the optimal value of the second stage problem

$$\min_y q^T y$$

s.t. $Tx + Wy = h, y \geq 0.$

Here, $x$ and $y$ are the vectors of first and second stage decision variables, respectively; the quantities $c, q, b, h,$ and $A, T, W$ are vectors and matrices, respectively, of appropriate dimensions, and $\mathbb{E}(\cdot)$ denotes the expectation operator. The second stage problem depends on the vector of data $\xi$ that consists of the elements of $q, h, T,$ and $W,$ some or all of which may be random. Therefore, $\xi = \xi(\omega)$ is a random vector, with its random elements collectively denoted as $\omega$. The expectation in the first stage problem is taken with respect to the probability distribution of $\xi(\omega)$ which is supposed to be known, or can be estimated. In general, $\omega$ is an element of a sample space $\Omega$ equipped with a sigma algebra $\mathcal{F}$. Practically, it is often assumed that the random vector $\xi(\omega)$ has a discrete and finite support. That is, $\xi(\omega)$ has a finite number $S$ of realizations, called scenarios, and each such scenario has its respective probability $p_s$ of occurring. Then, if we denote the set of all possible scenarios by $\mathcal{S},$ we have that $\mathbb{E} [Q(x, \xi(\omega)))] = \sum_{s \in \mathcal{S}} p_s Q(x, \xi(\omega_s)),$ where $\xi(\omega_s)$ denotes the specific realization of the random vector $\xi$ for scenario $s \in \mathcal{S}$. It follows that problem (4.3) can be formulated as one large linear optimization problem. Note that, in cases where the support is infinite or finite but impractically large, sample-average approximation methods can be used to approximate the original problem, by considering only randomly selected subsets of the set of all possible scenarios Shapiro (1993); Linderoth et al. (2002).
4.3.2 The Two-Stage Capacity Expansion Problem

In this paper, we specifically consider the two-stage stochastic nonlinear network capacity expansion problem described in (Ruszczyński and Shapiro, 2003, Example 4, p.14). According to this formulation, during the first stage of the problem, capacities are allocated to the arcs of the network, followed by the observation of random demands for network traffic. Then, at the second stage, a flow plan is determined utilizing the available network capacity so that the observed demand is satisfied. The problem objective is to plan the optimal flow routes and also allocate capacities to arcs in a cost efficient manner.

Specifically, consider a directed graph \( G = (\mathcal{N}, \mathcal{A}) \), with node set \( \mathcal{N} \) and arc set \( \mathcal{A} \). Denote the directed arc from node \( i \) to node \( j \) by \((i, j)\). The capacity of each arc \((i, j) \in \mathcal{A}\) is a first-stage decision variable designated by \( x_{ij} \). The cost of installing capacity on arc \((i, j)\) is denoted by \( V_{ij}(x_{ij}) \). For each pair of nodes \((m, n) \in \mathcal{N} \times \mathcal{N}\), we require a random commodity flow \( d^{mn} \), which is defined as the traffic that must be sent from node \( m \) to node \( n \). We denote the flow from \( m \) to \( n \) sent through arc \((i, j)\) by \( y_{ij}^{mn} \), which is a part of the second stage decisions. The cost of routing a flow \( y_{ij}^{mn} \) through arc \((i, j)\) is denoted by \( Q_{ij}(y_{ij}^{mn}) \). Our objective is to assign capacities to the arcs of the network in such a way that the combined cost of capacity allocations and the expected cost of all possible optimal flow plans is minimized. In particular, let \( \mathcal{A}_i^- \subseteq \mathcal{A} \) and \( \mathcal{A}_i^+ \subseteq \mathcal{A} \) denote the sets of incoming and outgoing arcs for node \( i \in \mathcal{N} \), respectively. Then, the second stage problem takes the form of the following
multicommodity network flow problem

\[
\begin{align*}
\min_{y_{ij}} & \quad \sum_{m,n \in \mathcal{N}} \sum_{(i,j) \in \mathcal{A}} Q_{ij}(y_{ij}^{mn}) \\
\text{s.t.} & \quad \sum_{m,n \in \mathcal{N}} y_{ij}^{mn} \leq x_{ij}, \quad \forall (i,j) \in \mathcal{A}, \quad (4.4a) \\
& \quad y_{ij}^{mn} \geq 0, \quad \forall (i,j) \in \mathcal{A}, \; m, n \in \mathcal{N}, \quad (4.4b) \\
& \quad \sum_{\{j:(i,j) \in \mathcal{A}^\dagger\}} y_{ij}^{mn} - \sum_{\{j:(j,i) \in \mathcal{A}_{i}^\dagger\}} y_{ji}^{mn} = \begin{cases} 
   d_{mn}, & \text{if } i = m, \\
   -d_{mn}, & \text{if } i = n, \\
   0, & \text{otherwise}, 
\end{cases} \quad \forall (i,j) \in \mathcal{A}, \; m, n \in \mathcal{N}, \quad (4.4c)
\end{align*}
\]

where we require that the sum of all commodity flows on an arc does not exceed its allocated capacity (4.4a), impose nonnegativity constraints on the flows (and thus capacities also) for each arc (4.4b), and include commodity flow balance constraints for each node (4.4c).

Denote the optimal value of (4.4) as \( Q(x, d) \), where \( x \in \mathbb{R}^{[A]} \) and \( d \in \mathbb{R}^{N(N-1)} \) are the vectors of all capacity allocations \( x_{ij} \) and demands \( d_{mn} \), respectively, and \( N \) is the cardinality of the node set \( \mathcal{N} \), i.e. \( |\mathcal{N}| = N \). Then, the first stage problem has the form

\[
\min_{x \geq 0} \sum_{(i,j) \in \mathcal{A}} V_{ij}(x_{ij}) + \mathbb{E}[Q(x, d)] . \quad (4.5)
\]

We assume that all matrices and cost vectors are deterministic, except for the flow demand entries \( d_{mn} \) in the second stage equality constraints (4.4c), which are random. As discussed before, the exact knowledge of the probability distribution of the random vector \( d \) is not necessary. Instead, it is sufficient to know a finite number of the possible realizations of the uncertain demands \( d_{mn} \). Then, we can create scenarios by taking all possible combinations of these realizations. For instance, if there are \( C \) independent nonzero commodity demands and each one of them has \( R \) possible independent realizations, then the number of resulting scenarios will be
This approach, however, entails the, not so realistic, assumption that all demands are independent and also leads easily to an exceedingly large number of scenarios. Alternatively, we can assume that there is a multivariate distribution for the demand vector $d$ and generate $S$ scenarios by taking $S$ samples from this joint distribution. This is a more realistic approach as historical data may be available; for example data of demands for all commodities at some given time instances.

Assuming that $S$ scenarios are available, the two-stage stochastic nonlinear network capacity expansion problem (4.4)-(4.5) can be formulated as a large deterministic optimization problem

$$
\min \limits_{x_{ij},y_{ij}} \left[ \sum \limits_{\{i,j:(i,j) \in A\}} V_{ij}(x_{ij}) + \sum \limits_{s \in S} p_s \left( \sum \limits_{m,n \in N} \sum \limits_{\{i,j:(i,j) \in A\}} Q_{ij}(y_{ij}^{s,mn}) \right) \right]
$$

subject to

$$
\sum \limits_{m,n \in N} y_{ij}^{s,mn} \leq x_{ij}, \quad \forall \ (i, j) \in A, \ s \in S
$$

$y_{ij}^{s,mn} \geq 0, \quad \forall \ (i, j) \in A, \ m, n \in N, \ s \in S,$

$$
\sum \limits_{\{j:(i,j) \in A^*\}} y_{ij}^{s,mn} - \sum \limits_{\{j:(j,i) \in A^*\}} y_{ji}^{s,mn} = \begin{cases} 
    d^{s,mn}, & \text{if } i = m, \\
    -d^{s,mn}, & \text{if } i = n, \\
    0, & \text{otherwise,}
\end{cases}
$$

$\forall \ i, m, n \in N, \ s \in S,$

where the superscript $s$ now denotes the scenario index. Clearly, even if only few of the commodity demands are nonzero and they have discrete distributions, the size of the emerging optimization problem (4.6) is enormous for most real world applications.

4.3.3 Capacity Expansion Decomposition with respect to the Network Nodes

Existing techniques reduce the complexity of two-stage stochastic capacity expansion problems by decomposing the original centralized problem (4.6) with respect to scenario bundles Rockafellar and Wets (1991); Mulvey and Ruszczyński (1995); Nielsen
and Zenios (1996); Singh et al. (2009); Ruszczynski and Shapiro (2003). This approach still requires the presence of a central unit that will gather information from the network, solve the individual subproblems for the scenario bundles, and then send back the decision variables to the corresponding nodes. Instead, in this paper we present a new method to decompose (4.6) into \( N \) subproblems that are solved in parallel at the node locations, using the recently proposed ADAL method Chatzipanagiotis et al. (2014a). The necessary coordination among subproblems is achieved using only information that is available within a local communication neighborhood of the corresponding nodes. The advantage of our proposal is that it retains the feature of reducing the problem’s size, while at the same time it eliminates the need for a central coordinating and processing unit.

In order to develop the proposed distributed algorithm to solve the capacity expansion problem, we first express problem (4.6) in an alternative, equivalent form that can be decomposed with respect to the network nodes. To this end, let \( C \) be the set of commodities for a given problem, with cardinality \( |C| = C \), and, also, recall that \( \mathcal{N} \) and \( \mathcal{S} \) denote the node and scenario sets with cardinalities \( |\mathcal{N}| = N \) and \( |\mathcal{S}| = S \), respectively. Moreover, define \( \mathbf{d}^{s,c} \in \mathbb{R}^N \) to be the vector of flow demands for commodity \( c \in C \) at scenario \( s \in \mathcal{S} \). Note that \( \mathbf{d}^{s,c} \) has only two nonzero entries, corresponding to the nodes that constitute the source-sink pair for commodity \( c \). Also, let \( \mathbf{y}^{s,c}_i = [y^{s,c}_{i1}, y^{s,c}_{i2}, \ldots, y^{s,c}_{iN}]^T \in \mathbb{R}^N \) be the vector of flows of commodity \( c \) that node \( i \) routes to all other nodes at scenario \( s \), where \( y^{s,k}_{ij} \) denotes the flow of commodity \( c \) that node \( i \) routes towards node \( j \) when scenario \( s \) is considered. Define the routing matrix \( \mathbf{A}_i \in \mathbb{R}^{N \times N} \) of node \( i \) as

\[
\mathbf{A}_i = \begin{bmatrix}
-a_{i1} & 0 & \cdots & 0 \\
0 & -a_{i2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -a_{iN}
\end{bmatrix},
\]
where \( a_{ij} = 1 \) if the arc \((i, j) \in \mathcal{A}\) and is zero otherwise. Then, (4.6) can be equivalently written as

\[
\min \left[ \sum_{i \in \mathcal{N}} \sum_{(j, i) \in \mathcal{A}_i^+} V_{ij}(x_{ij}) + \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{N}} \sum_{(j, i) \in \mathcal{A}_i^+} Q_{ij}(y_{ij}^{s,c}) \right]
\]

subject to

\[
\sum_{i \in \mathcal{N}} \mathbf{A}_i y_i^{s,c} = \mathbf{d}^{s,c}, \quad \forall \ s \in \mathcal{S}, \ c \in \mathcal{C},
\]

\[
\sum_{c \in \mathcal{C}} y_{ij}^{s,c} \leq x_{ij}, \quad \forall \ (i, j) \in \mathcal{A}, \ s \in \mathcal{S},
\]

(4.7)

\[
y_{ij}^{s,c} \geq 0, \quad \forall \ (i, j) \in \mathcal{A}, \ s \in \mathcal{S}, \ c \in \mathcal{C}.
\]

We observe that (4.7) consists of: (i) a separable objective function with respect to the nodes, (ii) constraints that are local to every node \(i\), whose feasible set we denote by

\[
\mathcal{Z}_i = \{ \mathbf{x}_i \in \mathbb{R}^{\left| \mathcal{A}_i^+ \right|}, \mathbf{y}_i \in \mathbb{R}^{\left| \mathcal{A}_i^+ \right|S\mathcal{C}} : \sum_{k \in \mathcal{K}} y_{ij}^{s,c} \leq x_{ij},
\]

\[
y_{ij}^{s,c} \geq 0, \quad \forall \ j : (i, j) \in \mathcal{A}_i^+, \ s \in \mathcal{S}, \ c \in \mathcal{C} \},
\]

(4.8)

where \( \mathbf{x}_i, \mathbf{y}_i \) are the vectors stacking the capacity and flow decision variables for node \(i\), respectively, and (iii) \(SC\) affine, flow balance constraints \( \sum_{i \in \mathcal{N}} \mathbf{A}_i y_i^{s,c} = \mathbf{d}^{s,c} \). The latter constraint set introduces coupling between the flow decision variables of the different nodes and requires decomposition techniques to obtain efficient distributed solutions.

4.3.4 Distributed Solution using the ADAL method

To apply the ADAL method to the problem under consideration we assume that:

(A1) The functions \( V_{ij}, Q_{ij} \) are convex and the flow \( x_{ij} \) and capacity \( y_{ij}^{s,c} \) variables are constrained to lie in convex, compact sets.

(A2) The Lagrange function for problem (4.7) has a saddle point.
Let \( k \) denote the iteration index and define \( \mathbf{\lambda}^k = [\mathbf{\lambda}^{k,1,1}, \ldots, \mathbf{\lambda}^{k,1,C}, \mathbf{\lambda}^{k,2,1}, \ldots, \mathbf{\lambda}^{k,S,C}]^T \in \mathbb{R}^{NSC} \) to be the vector of Lagrange multipliers at iteration \( k \), associated with the coupling flow balance constraints in (4.7), where \( \mathbf{\lambda}^{k,s,c} = [\lambda_{1}^{k,s,c}, \ldots, \lambda_{N_{s}}^{k,s,c}]^T \in \mathbb{R}^{N} \) for every scenario \( s \in \mathcal{S} \) and commodity \( c \in \mathcal{C} \). Let also

\[
\Lambda_i(x, y^i, y^k, \mathbf{\lambda}^k) = \sum_{j: (i,j) \in \mathcal{A}^+} V_{ij}(x_{ij}) + \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}} p_s \sum_{u} Q_{ij}(y^k_{ij}^{s,c})
\]

\[
+ \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}} (\mathbf{\lambda}^{k,s,c})^T A_i y^{s,c}_i + \sum_{s \in \mathcal{S}} \sum_{c \in \mathcal{C}} \frac{\rho}{2} \|A_i y^{s,c}_i + \sum_{j \neq i} A_j y^{k,s,c}_j - d^{s,c}\|^2
\]

denote the local AL of node \( i \), where \( \rho \in \mathbb{R}_+ \) is a properly defined penalty coefficient, and \( y^k_j \) denotes the vector of all flow decision variables of node \( j \) at iteration \( k \) that is communicated to node \( i \), and is considered by node \( i \) as a vector of fixed parameters. Then, at every iteration \( k \), ADAL begins with every node \( i \) minimizing its local AL (4.9), subject to the local constraints (4.8), i.e.,

\[
\tilde{y}_i^k = \arg \min_{x, y^i} \Lambda_i(x, y^i, \{y^k_j\}_{j \in \mathcal{T}_i}, \{\mathbf{\lambda}^k_l\}_{l \in \mathcal{H}_i \cup \{i\}})
\]

Observe that this optimization step requires only that node \( i \) has access to the, locally available, variables associated with its own flow balance constraint and those of its neighbors. More formally, if we define the neighborhood \( \mathcal{H}_i \) of node \( i \) as the set of all nodes \( j \) that share an arc with \( i \), i.e. \( \mathcal{H}_i = \{ j \in \mathcal{N} : (i, j) \cup (j, i) \in \mathcal{A} \} \), then, it is not hard to verify that node \( i \) only needs access to the Lagrange multipliers \( \mathbf{\lambda}^k_l \) of its neighbors’ flow constraints \( l \in \mathcal{H}_i \), and the flow variables \( y^k_j \) from the set \( \mathcal{T}_i = \{ j \in \mathcal{N} : \mathcal{H}_i \cap \mathcal{H}_j \neq \emptyset \} \) of its two-hop neighbors. To see this, consider the ordinary Lagrangian terms first

\[
(\mathbf{\lambda}^{k,s,c})^T A_i y^{s,c}_i = \sum_{j \in \mathcal{N}} \lambda_{j}^{k,s,c} [A_i]_{j} y^{s,c}_i
\]
Algorithm 10 Accelerated Distributed AL (ADAL)

Require: Set $k = 1$ and define initial Lagrange multipliers $\lambda^1$ and primal variables $y^1_i$ for every $i \in \mathcal{N}$.

1: For fixed $\lambda^k$, $y^k_i$ calculate for all nodes $i \in \mathcal{N}$ the $\hat{y}^k_i$ as the solution of

\[
\hat{y}^k_i = \arg \min_{x_i, y_i} A_i \left( x_i, y_i; \{y^k_j\}_{j \in \mathcal{T}_i}, \{\lambda^k_l\}_{l \in \mathcal{H}_i \cup \{i\}} \right)
\]

s.t. $x_i, y_i \in Z_i$

2: If $\sum_{i \in \mathcal{N}} A_i y^s,c_i = d^s,c$ for all $s \in \mathcal{S}$ and $c \in \mathcal{C}$, then stop (optimal solution found). Otherwise, for every $i \in \mathcal{N}$ set

\[
y^{k+1}_i = y^k_i + \tau (\hat{y}^k_i - y^k_i)
\]

and communicate $y^{k+1}_i$ to every $j \in \mathcal{T}_i$.

3: For every $i \in \mathcal{N}$, $s \in \mathcal{S}$ and $c \in \mathcal{C}$ set

\[
\lambda^{k+1,s,c}_i = \lambda^{k+1,s,c}_i + \rho \tau \left( \sum_{j: (i,j) \in \mathcal{A}_i^+} y^{k+1,s,c}_{ij} - \sum_{j: (i,j) \in \mathcal{A}_i^-} y^{k+1,s,c}_{ji} - d^s,c \right),
\]

communicate $\lambda^{k+1}_i$ to every $j \in \mathcal{H}_i$, increase $k$ by 1 and go to Step 1.

where the notation $[A_i]_j$ stands for the $j$-th row of matrix $A_i$. Noticing that $[A_i]_j \neq 0$ if and only if $j \in \mathcal{H}_i \cup \{i\}$, we obtain

\[
\sum_{j \in \mathcal{N}} \lambda^{k,s,c}_j [A_i]_j y^{s,c}_i = \sum_{j \in \mathcal{H}_i \cup \{i\}} \lambda^{k,s,c}_j [A_i]_j y^{s,c}_i.
\]

Similarly, for the quadratic penalty term of node $i$ we have

\[
\sum_{j \in \mathcal{N}} \left( [A_i]_j y^{s,c}_i + \sum_{l \neq i} [A_i]_j y^{k,s,c}_l \right)^2 =
\]

\[
\sum_{j \notin \mathcal{H}_i \cup \{i\}} \left( \sum_{l \neq i} [A_i]_j y^{k,s,c}_l \right)^2 + \sum_{j \in \mathcal{H}_i \cup \{i\}} \left( [A_i]_j y^{s,c}_i + \sum_{l \in \mathcal{H}_j \cup \{j\}} [A_i]_j y^{k,s,c}_l \right)^2,
\]

since $[A_i]_j \neq 0$ if and only if $l \in \mathcal{H}_j \cup \{j\}$. The term $\sum_{j \notin \mathcal{H}_i \cup \{i\}} \left( \sum_{l \neq i} [A_i]_j y^{k,s,c}_l \right)^2$, which includes information on the flow constraints of nodes that do not belong in the neighborhood of node $i$, is a constant in the optimization of (4.9) and as such it has no effect and can be excluded from the minimization of the local AL.
After calculating the minimizers $\hat{y}_i^k$ of the local ALs (4.10), ADAL proceeds with every node $i \in \mathcal{N}$ updating its primal variables $y_i^k$ (that will be communicated to its neighbors) according to

$$y_i^{k+1} = y_i^k + \tau (\hat{y}_i^k - y_i^k)$$

where $\tau$ is a stepsize, that plays a critical role in the convergence of the ADAL method. Specifically, convergence of ADAL is guaranteed if $0 \leq \tau \leq a/q$, where $a \in [1, 2)$ is a relaxation factor and $q = \max_{i \in \mathcal{N}} |\mathcal{H}_i|$ Chatzipanagiotis et al. (2014a).

In other words $\tau$ is determined by the density of the network and, in particular, by its maximum node degree.

The final step of ADAL at iteration $k$ involves the update of the dual variables $\lambda_{i}^{k+1,s,c}$, corresponding to the flow balance constraint of commodity $c$ at node $i$ for scenario $s$, according to

$$\lambda_{i}^{k+1,s,c} = \lambda_{i}^{k+1,s,c} + \rho \tau \left( \sum_{j : (i,j) \in A^+_i} y_{ij}^{k+1,s,c} - \sum_{j : (j,i) \in A^-_i} y_{ji}^{k+1,s,c} - d_{i}^{s,c} \right).$$

The dual updates are distributed by structure. Each node can update the dual variables corresponding to its own flow balance constraints after receiving the updated primal (flow decision) variables from its neighbors $\{ j \in \mathcal{N} : i \in \mathcal{H}_j \}$. Essentially, the constraint violation with respect to the flow decision variables serves to define the direction for the dual updates. The proposed method is summarized in Alg. 10.

Note that, additional constraints can be added to (4.6), without destroying the ability to apply the proposed distributed method. Specifically, any kind of local constraint sets $x_{ij} \in \mathcal{X}_{ij}$ and $y_{ij}^{s,mn} \in \mathcal{Y}_{ij}^{s,mn}$, $\forall i, j, m, n \in \mathcal{N}$, $s \in \mathcal{S}$ can be added to (4.6), as long as they satisfy the convexity and compactness conditions. For example, the sets $\mathcal{X}_{ij}$ can refer to capacity bounds $b_{ij}$ imposed on individual arcs, i.e., $\mathcal{X}_{ij} = \{ x_{ij} \in \mathbb{R} : 0 \leq x_{ij} \leq b_{ij} \}$. Moreover, additional coupling constraints can be introduced as long as they take a linear form (or can be equivalently expressed in a linear form). For example, we can impose bounds $B$ on the total available capacity.
Figure 4.15: Typical structure of the networks considered in simulations. The blue, green, and red dots correspond to source, sink and relay nodes, respectively. The blue lines correspond to edges between nodes. A network of 50 nodes and 6 commodities is depicted.

\[ \sum_{(i,j) \in A} x_{ij} \leq B. \] Another possible modification of (4.6) with multiple practical applications can be obtained if the arcs in the network are subject to failures resulting in the loss of random fractions \( \theta_{ij} \) of their capacities. In this case, the capacity constraints take the form 
\[ \sum_{m, n \in N} y_{ij}^{s, mn} \leq (1 - \theta_{ij}^s)x_{ij}. \]

4.3.5 Numerical Analysis

In all simulations, the examined networks were randomly generated with the agents uniformly distributed in rectangle boxes. We imposed the condition that the generated networks are connected and also intentionally positioned the source nodes one network diameter apart from their corresponding sink nodes, in order to prevent trivial problem setups. A typical network for a problem with 50 nodes and 6 commodities is depicted in Fig. 4.15.

Unless otherwise noted, the flow and capacity allocation costs were taken to be
linear functions, i.e., $Q_{ij}(y_{ij}^{s,c}) = q_{ij}y_{ij}^{s,c}$ and $V_{ij}(x_{ij}) = v_{ij}x_{ij}$, respectively. The cost coefficients $q_{ij}$, $v_{ij}$ were randomly generated from the uniform distributions $U[1, d_{ij}]$ and $U[1, 10d_{ij}]$, respectively, where $d_{ij}$ denotes the Euclidean distance between nodes $i$ and $j$. The idea here is that we let edges between nodes that are farther apart to have potentially higher costs. Also, we define relatively larger capacity allocation costs (compared to flow costs), in order to force the nodes to diversify their flow decisions onto multiple neighbors, instead of just routing all their flows through the edge with the minimum flow cost. Moreover, the different scenarios were generated by taking samples from uniformly distributed random commodity demand vectors, whose entries were assumed to be independent.

In all the cases presented, we plot the optimality and constraint feasibility convergence results for both ADAL and ADMM. For optimality convergence, we monitor the evolution in iterations of the cost function, while for constraint feasibility we monitor the evolution of the maximum absolute constraint violation. The initial values of the primal and dual variables were always set to 1, in order to minimize
the influence of initialization on the convergence behavior of the iterative distributed algorithms. Moreover, the penalty parameter $\rho$ is in general user defined in AL methods. In our simulations, we have found that fastest convergence for ADAL is obtained for values $\rho \in [1, 5]$, while at the same time preventing ill-conditioning. After extensive sensitivity analysis, we found that ADMM requires relatively larger values of $\rho \in [5, 20]$, compared to ADAL.

Fig. 4.16 depicts typical convergence results obtained after applying ADAL and ADMM on a problem with $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 5$. Fig. 4.16(a) and 4.16(b) demonstrate the convergence of the objective function value and constraint feasibility, respectively. We observe that ADAL and ADMM perform similarly. Next, we consider how the density of the network affects the performance of both methods. The motivation behind this stems from the fact that the primal stepsize of ADAL is affected by the maximum node degree in the network ($0 \leq \tau \leq 2/q$). Note also, that a similar relation holds for the ADMM, wherein the primal stepsize used by each node is affected by that node’s
Figure 4.18: Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with $N = 60$ nodes, $K = 6$ commodities, $S = 100$ scenarios and maximum node degree $q = 6$: (i) Objective function convergence, and (ii) Maximum constraint violation evolution.

degree. Towards this goal, we define again a problem with $N = 30$, $K = 5$, $S = 50$ as before, but now the maximum and average node degree are 10 and 7.78, respectively. The corresponding results are shown in Fig. 4.17. Both methods appear to slow down for denser networks, but not by a significant margin.

Next, we examine how both methods are affected by the size of the two-stage stochastic capacity expansion problem. For this, we consider cases with $N = 60$ nodes, $K = 6$ commodities, $S = 100$ scenarios and maximum node degree $q = 6$. Representative convergence results for such problems are depicted in Fig. 4.18. Moreover, Fig. 4.19 shows typical results for problems with $N = 90$ nodes, $K = 6$ commodities, $S = 40$ scenarios and maximum node degree $q = 6$. We observe that both methods are not severely affected by the increased problem size and perform similarly, albeit ADAL appears to converge slightly faster for these larger size problems. Note that solving the centralized problem (4.6) directly on our 28GB RAM workstation was not possible for problems of this size.

Finally, we apply ADAL and ADMM on a problem with nonlinear cost functions.
Specifically, we consider the quadratic costs $Q_{ij}(y_{ij}^{s,c}) = q_{ij}(y_{ij}^{s,c})^2$ and $V_{ij}(x_{ij}) = v_{ij}(x_{ij})^2$, where again the cost coefficients $q_{ij}, v_{ij}$ were generated as described above for the linear cost functions case. Fig. 4.20 depicts the corresponding convergence results for a case with $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 5$. Again, both methods are very close in terms of convergence speed.

4.4 Non-convex Problems

In this section we present numerical results of ADAL applied to non-convex optimization problems. The main objectives here are two. First, we verify the correctness of the theoretical analysis developed in Section 3.4.2 by showing that the proposed distributed method converges to a local minimum. We also show that the Lyapunov function defined in (3.61) is indeed strictly decreasing for all iterations, as expected. Second, we examine how sensitive ADAL is to the choice of the user-defined penalty coefficient $\rho$, and also to different initialization points.
Figure 4.20: Comparison of the two different distributed algorithms, ADAL and ADMM, for a problem with quadratic cost functions and $N = 30$ nodes, $K = 5$ commodities, $S = 50$ scenarios and maximum node degree $q = 5$: (i) Objective function convergence (the horizontal line corresponds to the value of the centralized solution), and (ii) Maximum constraint violation evolution.

Since the problems are non-convex, ADAL will converge to some local minimum. To evaluate the quality of this local minimum, we use the solution that is obtained by directly solving the non-convex problems with a commercial nonlinear optimization solver; we refer to that solution as “centralized”, as we do not enforce any decomposition when using this solver. Note that the goal here is not to compare the centralized solution to the solution that is returned by ADAL, but rather to establish that ADAL does not converge to trivial solutions. In comparison, in the convex case we would compare the solution of ADAL to the global optimal solution and show that they are the same. The simulations were carried out in MATLAB, using the `fmincon` command to solve the centralized problem, as well as the non-convex local subproblems (3.54) at each iteration of ADAL. ¹ The results correspond to the "active-set" solver option of `fmincon`, which performed better than all other

¹ We note that, for the problems considered here, the `fmincon` solver of Matlab returned the same solutions as other solvers such as MINOS, LANCELOT, SNOPT, and IPOPT in AMPL for the vast majority of cases. Since the purpose of this paper is not to compare the performance of nonlinear optimization solvers, we have focused just on the `fmincon`.
options, in terms of optimality and computation time.

First, we examine a simple non-convex optimization problem with $N = 2$ agents that control their decision variables $x_1$ and $x_2$, respectively. The problem is:

$$
\min_{x_1, x_2} x_1 \cdot x_2, \quad \text{s.t.} \quad x_1 - x_2 = 0.
$$

This problem is particularly interesting because the straightforward application of the popular ADMM algorithm fails to converge, as discussed in Houska et al. (2015). The problem has an obvious optimal solution at $x_1^* = x_2^* = \lambda^* = 0$. It is shown in Houska et al. (2015) that initializing ADMM at $x_1^1 = x_2^1 = 0$ and $\lambda^1 \neq 0$ for this problem gives iterates of the form $x^{k+1} = 0$ and $\lambda^{k+1} = -2\lambda^k$, and we can see how the latter update produces a diverging dual sequence. On the other hand, the proposed ADAL method is convergent, as can be seen in Fig. 4.21.

Next, we consider a non-convex problem with $N = 6$ agents, where each agent controls a scalar decision variable $x_i$, $i = 1, \ldots, 6$ that is subject to box constraints. Each agent has a different non-convex objective function and all decisions are coupled
The simulation results for this problem are depicted in Fig. 4.22, where we compare the solutions of ADAL and the centralized solver for 50 different initialization instances. For each instance, the initialization points for each $x_i$, $i = 1, \ldots, 6$, are generated by sampling from the uniform distribution with support $[-5, 5]$. We set $\rho = 1$, and terminate ADAL after the maximum residual $\max_j \|r_j(x^k)\|$, i.e., the maximum constraint violation among all constraints $j = 1, \ldots, m$, reached a value of
Figure 4.23: Simulation results of ADAL applied to problem (4.13) for different values of the penalty parameter $\rho = 1, 3, 10, 20$: a) Objective function convergence, and b) Constraint violation convergence.

1e-4. We note that this termination criterion was satisfied at around 100 iterations for practically all instances. Also note that for this case $m = 1$ and $q = 6$, hence, the stepsize is simply a scalar that is set to $\tau = 1/6$. For this problem, we observe an interesting behavior: ADAL converges to the “best” local minimum of the problem in almost all cases, which is not always true for the centralized solver. Both schemes are initialized at the same point at each instance.

Next, we consider a problem with multiple constraints $m = 5$, more agents $N = 8$, and larger box constraint sets

$$
\min_{x} \quad \cos(x_1) + \sin(x_2) + e^{x_3} + 0.1x_4^3 \\
+ 0.1/(1 + e^{-x_5}) + 0.01(x_6^5 - x_6 - x_4^4 + x_6^2) \\
+ \sqrt{x_7 + 15}\sin(x_7/10) + e^{x_8}/(x_8^2 + e^{x_8})
$$

s.t. \hspace{1cm} Ax = b, \hspace{1cm} (4.13)

$$
-10 \leq x_i \leq 10, \quad \forall \ i = 1, \ldots, 8,
$$

where the constraint parameters $A \in \mathbb{R}^{5 \times 8}$ and $b \in \mathbb{R}^5$ are randomly generated with entries sampled from the standard normal distribution (such that the problem is
feasible). When generating $A$, we always ensure that it has full row rank (to prevent trivial constraint sets), and that at least two decision variables are coupled in each constraint.

Fig. 4.23 depicts the convergence results of ADAL applied to problem (4.13), where the generated matrix $A$ is

$$
\begin{pmatrix}
0 & 0 & 1.2634 & 0.9864 & 0 & 0.4970 & -0.2259 & -0.2783 \\
0 & 1.6995 & 0 & 0 & 0 & 1.9616 & 0 & 0 \\
-1.8780 & 0 & 0 & 0 & 0 & -2.5970 & -0.8325 & 0 \\
0 & 0 & -0.3894 & 0 & 0 & 0 & 0.8270 & 0 \\
-0.8666 & 0 & 0 & 0.2461 & -0.1226 & 0 & 0 & 0
\end{pmatrix},
$$

and $b = [-0.0579, -1.6883, 0.8465, 0.1843, 0.6025]^\top$. In this case the stepsizes are set to $T = \text{diag}(1/5, 1/2, 1/3, 1/2, 1/3)$. To examine how sensitive ADAL is to the choice of the user-defined penalty coefficient $\rho$, we present convergence results for four different choices $\rho = 1, 3, 10, 20$. We terminate ADAL after reaching a maximum constraint violation of $3e^{-4}$. Two significant observations can be made based on these results. On one hand, choosing larger values of $\rho$, e.g. 10 or 20, leads to faster convergence, albeit at the cost of converging to a worse local minimum in terms of objective function value. On the other hand, choosing small $\rho$, e.g. 1 or 3, allows ADAL to find a better solution, however, convergence of the constraint violation slows down significantly after reaching accuracy levels of about $1e^{-3}$.

In order to test the sensitivity of ADAL to initialization for problem (4.13), we test it for 50 different initialization instances. The results are depicted in Fig. 4.24(a), where we also include the solutions obtained from the centralized scheme for the same initializations as ADAL. We observe that, for this problem, the choice of initialization point plays a more significant role in determining which local minimum ADAL will converge to, as compared to the corresponding results for the previous problem (4.12) where ADAL converged to the same local minimum for the vast majority of initializations. Moreover, in Fig. 4.24(b) we plot the evolution of the Lyapunov function $\phi(x^k, \lambda^k)$, cf. (3.61). We observe that $\phi$ is strictly decreasing at
Figure 4.24: a) Simulation results for ADAL and the centralized solver applied to problem (4.13). The results correspond to 50 different initialization instances. At each instance, the initialization point is the same for both ADAL and the centralized solver. The red and blue squares indicate the objective function value at the point of convergence for the centralized method and ADAL, respectively. A blue dashed line indicates that ADAL converged to a better (or the same) local minimum, while a red dashed line indicates the opposite. b) Evolution of the Lyapunov function $\phi(x^k, \lambda^k)$ each iteration, as expected.

Finally, we test ADAL on problems of the form (4.13) for 50 different instances of the problem parameters $A$ and $b$. The objective of this experiment is to examine the behavior of ADAL with a predefined value of $\rho$ for a wide range of problems, instead of finding the best $\rho$ for a given problem as in Fig. 4.23. This is important for practical applications, where we need to choose a value for $\rho$ without knowing the exact problem parameters. In order to ensure that $\rho$ is sufficiently large for all problem realizations, in this experiment we set $\rho = 5$. We terminate ADAL after reaching a maximum constraint violation of $3e^{-4}$. The results are shown in Fig. 4.25. We observe that overall the performance of ADAL is satisfactory, judging by the fact that it converges to the same local minimum as the centralized solver for most of the cases.
4.5 Problems with Noise and Uncertainties

In this section we present numerical results for SADAL, cf. Section 3.5, applied on a network optimization problem. Consider an undirected graph $G = (N, A)$ with a set of nodes $N$ and a set of arcs $A$. The set of nodes consists of two subsets $N = \{S, D\}$, where $S$ is the set of source nodes and $D$ the set of destination nodes. Let $s_i$ denote the flow generated at source node $i \in S$ and $c_i \geq 0$ denote the reward coefficient for $s_i$. Each $s_i$ is subject to a minimum threshold constraint $s_i \geq s_{i}^{\min}$, where the $s_{i}^{\min}$ is a problem parameter. Also, let $t_{ij}$ denote the flow through arc $(i, j)$. Each arc $(i, j)$ has a feasible range of flows $a_{ij} \leq t_{ij} \leq b_{ij}$, where $a_{ij}, b_{ij}$ are given numbers. Denote the neighborhood of node $i$ as $C_i = \{j : (i, j) \in A\}$. The conservation of flow at each node $i \in S$ is expressed as $\sum_{j \in C_i} t_{ij} - \sum_{j \in C_j} t_{ji} = s_i$. The problem we consider is a network utility maximization (NUM) problem, where we seek to find routing
decisions \( t_{ij} \) that maximize the amount of flow generated at all source nodes, subject to flow conservation, arc capacity, and minimum flow generation constraints. The NUM takes the form

\[
\begin{align*}
\max & \quad \sum_{i \in S} c_i s_i \\
\text{subject to} & \quad \sum_{\{j \in \mathcal{C}_i\}} t_{ij} - \sum_{\{j \in \mathcal{C}_i\}} t_{ji} = s_i, \quad \forall \ i \in S \\
& \quad a_{ij} \leq t_{ij} \leq b_{ij}, \quad \forall \ (i, j) \in A, \\
& \quad s_i \geq s_i^{\min}, \quad \forall \ i \in S.
\end{align*}
\]

In our consideration, the destination nodes are modeled as sinks and can absorb any amount of incoming rates. Hence, no flow conservation constraints are necessary for these nodes. Note that, if some nodes are neither sources or destinations then we set the \( c_i \) and \( s_i^{\min} \) equal to zero. Moreover, observe that \( q = \max_i |\mathcal{C}_i| \) and, according to the convergence analysis, the stepsize \( \tau \) of ADAL and the initial stepsize \( \tau_1 \) of SADAL must be less than \( \frac{1}{q} \).

In what follows, we consider normalized rates \( s_i, t_{ij} \in [0, 1] \), without any loss of generality. The parameters \( c_i \) and \( s_i^{\min} \) are randomly generated by sampling uniformly in the intervals \([0.1, 1]\) and \([0, 0.3]\), respectively. Unless otherwise noted, the penalty parameter is set to \( \rho = 1 \). In all simulations, the objective function value and the maximum residual \( \max_j r_j(x^k) \), i.e., the maximum constraint violation among all constraints \( j = 1, \ldots, m \), were monitored as the criteria of convergence. The examined networks were randomly generated with the agents uniformly distributed in rectangle boxes. A typical network configuration along with the routing decisions after solving the problem with SADAL is depicted in Fig. 4.26.

In Fig. 4.27 we present simulation results for different levels of noise corruption and compare them with the deterministic solution from ADAL. We consider two cases with different variances of the noise terms, labeled “easy” and “hard”; in the
Figure 4.26: A randomly generated network for the NUM problem with 50 sources (red dots) and 4 sinks (green squares): a) All available arcs \( (i, j) \in A \) are depicted as blue lines, b) Flow routing decisions after solving the problem with SADAL. The rate of flow \( t_{ij} \) through arc \( (i, j) \) defines the thickness and color of the corresponding drawn line. Thicker, darker blue lines indicate larger flows.

Figure 4.27: Comparative simulation results for different noise levels. We consider the noise-free case (deterministic) where ADAL is applied, and two noise scenarios, labeled “easy” and “hard” (larger noise), where SADAL is applied. a) Objective function convergence, b) Maximum constraint violation convergence.
hard case all the noise terms have larger variance, compared to the easy case. In both cases the noise terms are modeled as uniform random variables. For the “easy” case, the noise corruption terms are modeled as follows: $v^k_{ij} \sim \frac{1}{\mu^k} U(-0.1, 0.1)$, and $w^k_i \sim \frac{1}{\mu^k} U(-0.1, 0.1)$, where $U(u, v)$ denotes the uniform distribution with support $[u, v]$. Here, $\mu^k$ is a coefficient that we introduce to make the noise terms decreasing; we initially set $\mu^1 = 1$ and then every 5 iterations we increase $\mu$ by one, i.e., $\{\mu\}_{k=1}^\infty = \{1, 1, 1, 1, 2, 2, 2, 2, 2, 3, \ldots \}$. The non-decreasing noise terms are generated as $u^k_i \sim U(-0.03, 0.03)$, and the noise in the linear objective functions is modeled as $\tilde{c}^k_i = c_i p^k_i$, where $p^k_i \sim \frac{1}{\mu^k} U(-0.3, 0.3)$. For the “hard” case the random variables are generated according to $v^k_{ij} \sim \frac{1}{\mu^k} U(-0.2, 0.2)$, $w^k_i \sim \frac{1}{\mu^k} U(-0.2, 0.2)$, $u^k_i \sim U(-0.05, 0.05)$, and $p^k_i \sim \frac{1}{\mu^k} U(-0.7, 0.7)$. In both cases, the stepsize sequence $\tau_k$ is defined as $\tau_k = 1/(\nu^k)$. The sequence $\nu^k$ is generated similar to $\mu^k$, with the difference that we increase it by one every 30 iterations.

Not surprisingly, the results of Fig. 4.27 indicate that the stochastic problem converges slower than the noise-free case. On the other hand, the difference in the noise levels does not appear to affect the convergence speed significantly, at least for the two different noise scenarios studied here. We can also see that the residual (feasibility) convergence of SADAL slows down significantly after reaching accuracy levels of about $10^{-3}$. To put these results into perspective, note that, at iteration $k = 1000$, the noise term $v^k_{ij}$ in the message exchange from agent $i$ to $j$ is distributed according to $v^k_{ij} \sim 10^{-3} U(-0.5, 0.5)$. Now, consider that this is the noise just from one neighbor, however, each agent $i$ has multiple neighbors, which means that the noise corruptions add up; the generated networks typically have a neighborhood size of about 5 or 6 for each node. Thus, there exists a (relatively) substantial amount of noise corruption even at iteration 1000 (recall also that the $u^k_i$ noise terms do not decrease), which should be taken into consideration when evaluating the convergence...
results of Fig. 4.27.

In order to test how the choice of stepsize sequences \( \{\tau_k\} \) affects the convergence, we have performed simulations where the \( \nu^k \) is increased every 15, 60, or 100 iterations. In all cases we do not let the stepsize \( \tau_k \) decrease below 0.01, i.e., \( \tau_k = \max\{1/(q\nu^k), 0.01\} \). For completeness, we also test SADAL for a constant stepsize \( \tau = \frac{1}{q} \), although this is not consistent with the assumptions of the algorithm. The results, corresponding to the “hard” formulation, are depicted in Fig. 4.28. We observe that the convergence of SADAL is not significantly affected by the choice of stepsize sequences, albeit stepsizes that decrease faster seem to exhibit a slightly better behavior (keep in mind that we do not let the stepsize decrease below 0.01). Moreover, the constant stepsize choice produces an oscillatory behavior and does not lead to convergence, which is in accordance with the theoretical analysis.

Finally, we examine how sensitive SADAL is to the choice of the user-defined penalty coefficient \( \rho \), at least for the problem under consideration here. Fig. 4.29 depicts simulation results of the “hard” noise scenario for \( \rho \) taking the values 0.3, 1, 3, and 10. The \( \mu^k \) is increased every 5 iterations, and \( \nu^k \) every 30 iterations. We observe that convergence is not significantly affected by the choice of \( \rho \), apart from the smallest value case \( \rho = 0.3 \) which lead to a more “oscillatory” behavior.
Figure 4.28: Simulation results for different choices of the stepsize sequence $\tau_k = 1/(q\nu^k)$. We consider scenarios where $\nu^k$ is increased by one every 15, 60, and 100 iterations, and also a constant stepsize choice $\tau_k = 1/q$. The results correspond to the “hard” noise scenario. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence.
Figure 4.29: Simulation results for different values of $\rho$. a) Objective function convergence, b) Magnified view of the objective function convergence, c) Maximum constraint violation convergence.
Application to Relay Beamforming for Wireless Communications

In this chapter we discuss in detail the application of ADAL on a specific problem in wireless communications. Specifically, we are concerned with cooperative beamforming, a smart signal processing technique that exploits the constructive interference phenomenon in wave propagation to allow for directional signal transmission or reception. The contents of this chapter have been included in the published works Chatzipanagiotis et al. (2014b, 2013).

Cooperative beamforming is a rapidly emerging area of interest in wireless communications, due to its potential to provide extended coverage of the network, throughput enhancement, energy savings, and reliability in long distance transmissions Sendonaris et al. (2003); Gershman et al. (2010); Dong et al. (2008); Li et al. (2011a); Dong et al. (2009); Liu and Petropulu (2010, 2011, 2013); Li et al. (2011b); Chen et al. (2009); Fadel et al. (2012); Ngo and Larsson (2013); Bornhorst et al. (2012); Fazeli-Dehkordy et al. (2009); Cheng and Pesavento (2012); Wang et al. (2011); Kha et al. (2013); Choi (2011); Jing and Jafarkhani (2009); Havary-Nassab et al. (2008); Ding et al. (2008). Traditionally, multihop schemes have been the only
way to relay information over long distances. However, real-time transmissions over multiple hops suffer from large interference, packet collisions and delays, unreliable links, and difficulties in routing Gharavi and Ban (2003); Goldsmith and Wicker (2002); Jain et al. (2003). These issues of multi-hop systems can be avoided by the use of alternative, cooperative, long-distance communication schemes such as beamforming.

In cooperative beamforming, a set of relays form a “virtual antenna array” and re-transmit weighted versions of the source signals (decode-and-forward (DF) relaying), or weighted versions of the received signals (amplify-and-forward (AF) relaying). By exploiting constructive interference effects, the relays focus the transmitted power on the destinations’ locations, thus increasing the directional channel gain. By achieving spatial multiplexing, cooperative beamforming can support the communications of multiple, distinct, single-antenna, source-destination pairs that overlap both in time and frequency. This scenario is also referred to as multiuser peer-to-peer relay networks Li et al. (2011b); Liu and Petropulu (2010, 2011, 2013); Dong et al. (2009); Fazeli-Dehkordy et al. (2009); Chen et al. (2009); Cheng and Pesavento (2012); Wang et al. (2011); Fadel et al. (2012); Bornhorst et al. (2012); Ngo and Larsson (2013).

In general, the per-node throughput capacity of a wireless ad-hoc network reduces rapidly as the network size increases Gupta and Kumar (2000). Therefore, it is often preferable to divide the network nodes into multiple clusters, with each cluster containing nodes which have distinct sub-goals, or are geographically close to each other, e.g., applications involving networks of mobile wireless robots Fax and Murray (2004); Zavlanos et al. (2013).

In this chapter we consider a multi-cluster network, in which multiuser peer-to-peer relay communications take place inside each cluster, while the intra-cluster communications cause inter-cluster interference. In this context, the relay weights are computed based on channel second-order statistics, so that the total relay trans-
mit power is minimized, while meeting certain signal-to-interference-plus-noise-ratio (SINR) constraints at the destinations. First, we show that a computationally efficient approximate solution is attainable by relaxing the original NP-hard non-convex problem employing semidefinite relaxation (SDR) techniques Vandenberghe and Boyd (1996); Huang and Palomar (2010); Hammarwall et al. (2006); Bengtsson and Ottersten (1999). Second, we propose a distributed approach to solve the relaxed problem, which allows for each cluster to compute its optimal beamforming weights based on information exchanges with neighboring clusters only. Such a distributed approach obviates the need for a central processing unit that has access to the channel statistics of all clusters and obtains the relay weights; centralized approaches do not scale well with the number of network nodes, resulting in high complexity and long delays.

We propose two different ways to apply ADAL to the multi-cluster beamforming problem, termed Direct and Indirect, that allow us to model different message exchange patterns (necessary for the iterative execution of ADAL) between the individual clusters. Specifically, the message exchange pattern in the Direct method is determined by the coupling SINR constraints due to inter-cluster interference. On the other hand, the message exchanges in the Indirect method can be defined arbitrarily by the user. Both approaches rely on transforming the SINR coupling constraints to a linear form by introducing appropriate auxiliary variables. We show, via numerical experiments, that the Direct method is generally more efficient than the Indirect. However, the flexibility of the Indirect method in selecting the message exchange pattern between clusters might make it more appropriate for certain applications.

To the best of our knowledge, there is no prior work showing that the multi-cluster relay beamforming problem is amenable to a decentralized solution. The closest scenarios considered in the literature are those of multi-cell downlink beamforming Tolli et al. (2009); Shen et al. (2012), which do not involve relays and thus the
formulation is considerably simpler; the two AF communication stages of the relay problem that we consider in this chapter give rise to several additional interference terms that have to be taken into account. The beamforming weight design in Tolli et al. (2009) and Shen et al. (2012) employs respectively the dual decomposition method and the ADMM. Although one could use similar methods as in Tolli et al. (2009); Shen et al. (2012) to solve our problem, the ADAL method converges faster according to the simulation results presented in Section 5.3.

The rest of this section is organized as follows: In Section 5.1, we first discuss the single cluster relay beamforming scenario. Then, we formulate the multi-cluster problem and propose to pose it as a convex optimization problem using SDR. In Section 5.2, we present two different ways to obtain a decentralized solution to the convex multi-cluster problem by applying ADAL. Finally, in Section 5.3, we present simulation results to verify the validity of our approach.

5.1 Relay Beamforming

To facilitate understanding of the multi-cluster scenario, we first formulate the cooperative beamforming problem for a single cluster. The solution for this problem can be found in Fazeli-Dehkordy et al. (2009). Then, in Section 5.1.2 we formulate and solve the multi-cluster problem.

5.1.1 Single Cluster case

In the single cluster scenario, the goal is to allow communication of multiple single-antenna source-destination pairs, which transmit simultaneously using the same channel. The transmission takes place in two stages, i.e., two consecutive time-slots. In the first stage, all sources transmit, while in the second stage the relays retransmit the signals that they received in an AF fashion. A simple case with two source-destination pairs and three relays is depicted in Fig. 5.1.
Figure 5.1: Simultaneous communication between 2 sources and 2 destinations with the help of 3 relays. The signal transmitted from source 1 (S1) is intended for destination 1 (D1), while the signal transmitted from source 2 (S2) is intended for destination 2 (D2). Signals from S1 and S2 that reach D2 and D1, respectively, are considered interference. Also shown are the channel gains $f_{ij}$ between sources and relays, and $g_{ij}$ between relays and destinations.

Consider a network composed of an index set $\mathcal{M} = \{1, \ldots, M\}$ of sources $S_m, \forall m \in \mathcal{M}$, users (destinations) $U_m, \forall m \in \mathcal{M}$ and a corresponding set $\mathcal{L} = \{1, \ldots, L\}$ of dedicated, single-antenna relay nodes $R_l, \forall l \in \mathcal{L}$. Source $S_m$ wishes to communicate with user $U_m$. During the first communication stage, every source $S_m$ transmits the signal $\sqrt{P_0} s_m$, where $P_0$ is the common power, and $s_m \in \mathbb{C}, m = 1, \ldots, M$ denote the information symbols, which are independent identically distributed (i.i.d.) with unit power. The received signal at every relay $R_l$ is given by

$$x_l = \sqrt{P_0} \sum_{m=1}^{M} f_{ml} s_m + v_l,$$

where $\mathbb{C}$ denotes the set of complex numbers, $f_{ml} \in \mathbb{C}$ is the channel between source $S_m$ and relay $R_l$, and $v_l \in \mathbb{C}$ is the noise at relay $R_l$, assumed to have zero mean and unit variance. The channel coefficients are treated as random, i.i.d., independent between different paths. This assumption is valid when the nodes are sufficiently separated. It is also assumed that the channel coefficients are independent of the
source signals and the noise. Correspondingly, the received signal vector at all relays can be expressed in matrix form as

$$\mathbf{x} = \sqrt{P_0} \mathbf{F} \mathbf{s} + \mathbf{v},$$

where \(\mathbf{s} = [s_1, \ldots, s_M]^T \in \mathbb{R}^M\), \(\mathbf{x} = [x_1, \ldots, x_L]^T \in \mathbb{C}^L\), \(\mathbf{v} = [v_1, \ldots, v_L]^T \in \mathbb{C}^L\), with \((\cdot)^T\) denoting the transposition operation, and

$$\mathbf{F} = \begin{bmatrix} f_{11} & \cdots & f_{M1} \\ \vdots & \ddots & \vdots \\ f_{1L} & \cdots & f_{ML} \end{bmatrix} = \begin{bmatrix} f_1 & \cdots & f_M \end{bmatrix} \in \mathbb{C}^{L \times M}$$

is a channel state matrix, with \(f_m = [f_{m1}, \ldots, f_{mL}]^T \in \mathbb{C}^L\) denoting the channel gain column vector from source \(S_m\) to all relays.

During the second communication stage, the relays retransmit, in an AF fashion, a linear transformation of their received signals. We can express this linear transformation as a multiplication of \(\mathbf{x}\) with a beamforming matrix \(\mathbf{W} \in \mathbb{C}^{L \times L}\). Hence, the vector \(\mathbf{t} \in \mathbb{C}^L\) of relay transmissions can be written as

$$\mathbf{t} = \mathbf{W} \mathbf{x} = \sqrt{P_0} \mathbf{W} \mathbf{F} \mathbf{s} + \mathbf{W} \mathbf{v},$$

Since the relays are physically separated, they do not have access to the signals received at other relays. Each relay operates only on its own received signal, and thus the beamforming matrix is diagonal, i.e. \(\mathbf{W} = \text{diag}\{w_1, \ldots, w_L\}\), where \(w_l\) denotes the complex weight with which relay \(R_l\) multiplies its received signal.

By similar reasoning as above, the received signal vector \(\mathbf{y} \in \mathbb{C}^M\) at the destinations equals

$$\mathbf{y} = \mathbf{G} \mathbf{t} + \mathbf{z} = \sqrt{P_0} \mathbf{G} \mathbf{W} \mathbf{F} \mathbf{s} + \mathbf{G} \mathbf{W} \mathbf{v} + \mathbf{z},$$

where \(\mathbf{z} = [z_1, \ldots, z_M]^T \in \mathbb{C}^M\) denotes the vector stacking i.i.d random noise comp-
ponents with zero mean and unit variance, and
\[
G = \begin{bmatrix}
g_{11} & \cdots & g_{L1} \\
\vdots & \ddots & \vdots \\
g_{1M} & \cdots & g_{LM}
\end{bmatrix} = [g_1 \ldots g_M]^T \in \mathbb{C}^{M \times L},
\]
is a channel state matrix, with \(g_m = [g_{1m}, \ldots, g_{Lm}]^T \in \mathbb{C}^L\) denoting the channel gain column vector from all relays to user \(U_m, m \in \mathcal{M}\). The received signal at user \(U_m\) can be divided in three components that capture: i) the desired signal originating from source \(S_m\), ii) interference due to sources other than \(S_m\) that constitute interference, and iii) noise at the user, i.e.,
\[
y_m = g_m^T t + z_m = g_m^T Wf_m s_m + \sum_{j \neq m} g_m^T Wf_j s_j + g_m^T Wv + z_m, \quad (5.1)
\]
A reasonable optimality criterion for determining the beamforming weights is the minimization of the total relay transmit power, \(P_T\), subject to satisfying certain Quality of Service requirements at all destinations, namely enforcing user-specific SINR bounds \(\gamma_m > 0\). Since the channels are in general random, we will use the average transmit power. Therefore, we can pose the cooperative beamforming problem as the following optimization problem:
\[
\min_W P_T(W) \quad \text{s.t.} \quad \text{SINR}_m(W) \geq \gamma_m, \quad \forall m = 1, \ldots, M. \quad (5.2)
\]
The total average transmit power at the relays equals
\[
P_T(W) = \mathbb{E}\{\|t\|^2\} = \mathbb{E} \left\{ \text{Tr} \left[ \left( \sqrt{P_0} W F s + Wv \right) \left( \sqrt{P_0} W F s + Wv \right)^H \right] \right\} = \text{Tr} \left( P_0 W \mathbb{E}\{FF^H\} W^H \right) + \text{Tr} \left( WW^H \right),
\]
where $\| \cdot \|_F$ denotes the Frobenius norm. The expectation in the first equation is taken over source signals and channels, while in the second and third equations, due to the i.i.d. and unit power assumption on the $s_m$'s, the signal terms have been already averaged out and the expectation refers to the channels. Due to our assumptions, $\mathbb{E}\{F F^H\}$ is a diagonal matrix. Since $W$ is also a diagonal matrix, we can use this property to express the sum transmit power as

$$P_T = w^H R_T w,$$

where $w = [w_1, \ldots, w_L]^T \in \mathbb{C}^L$ is a column vector containing all the diagonal elements of $W$, and

$$R_T = P_0 \text{diag}\left\{ \sum_{m \in M} \mathbb{E}\{|f_{m1}|^2\}, \ldots, \sum_{m \in M} \mathbb{E}\{|f_{mL}|^2\} \right\} + I_L,$$

where $| \cdot |$ denotes the magnitude of a complex number. Based on (5.1), the SINR at every user $U_m$ is defined as

$$\text{SINR}_m \triangleq \frac{\mathbb{E}\left( P_0 |g_m^T W f_m s_m|^2 \right)}{\mathbb{E}\left( P_0 \sum_{j \in \mathcal{M}} |g_m^T W f_j s_j|^2 + |g_m^T W v|^2 + |z_m|^2 \right)},$$

where the term $P_0 \sum_{j \in \mathcal{M}} |g_m^T W f_j s_j|^2$ represents interference at user $U_m$ caused by signals intended for other users, the term $|g_m^T W v|^2$ denotes noise at the relays that was propagated to the user, and $|z_m|^2$ denotes noise at the user level. The expectation in the above equation refers to everything that is random, i.e., signals, channels, noise. Observe that the average SINR is defined as the ratio of the expected values, which is different than the expected value of the ratio. This definition is frequently used in communications textbooks, e.g. Gallager (2008), and in published works related
to the problem considered here Li et al. (2011a); Liu and Petropulu (2010, 2011); Havary-Nassab et al. (2008); Fazeli-Dehkordy et al. (2009).

Similar as before, we can manipulate the SINR expression to write it in a more compact matrix form

\[
\text{SINR}_m = \frac{P_0 w^H R_S^m w_{\text{Desired}}}{P_0 w^H R_I^m w_{\text{Interference}}} + \frac{w^H R_v^m w_{\text{Noise}} + 1}{\text{Noise}}.
\]

The desired signal matrix for user \(U_m\) is Hermitian

\[
R_S^m = \mathbb{E}\{(f_m^T \odot g_m^T) (f_m^T \odot g_m^T)\},
\]

with \(\odot\) denoting the Hadamard (entry-wise) product. The corresponding interference matrix is also Hermitian

\[
R_I^m = \sum_{j \neq m} \mathbb{E}\{(f_j^T \odot g_m^T) (f_j^T \odot g_m^T)\},
\]

and the respective noise matrix is diagonal

\[
R_v^m = \text{diag}\{\mathbb{E}\{|g_{1m}|^2\}, \ldots, \mathbb{E}\{|g_{Lm}|^2\}\}.
\]

Utilizing the above notation, the single-cluster optimization problem (5.2) can be compactly written as

\[
\min_w \quad w^H R_T w \quad \text{(5.3)}
\]

s.t. \(w^H Q^m w \geq 1, \quad \forall m = 1, \ldots, M,\)

where \(Q^m = \frac{P_0}{\gamma_m} R_S^m - P_0 R_I^m - R_v^m.\)
5.1.2 Multiple Clusters case

In this section we consider a multi-cluster network, in which, neighboring clusters’ communications interfere. A simple setup with two clusters is depicted in Fig. 5.2. In this case, the beamforming decisions of the relays of each cluster must also take into account the interference caused to and from the other clusters’ operation. This introduces three new terms in the SINR of each user. First, there is a term quantifying the interference on the relays of each cluster, exerted by the transmissions of other clusters’ sources during the first communication stage, which then propagates to the users of this cluster after the second stage transmission. Second, there is also interference on the users of each cluster exerted by the signals transmitted from the relays of other clusters that are intended for other users. Finally, the noise at the relays of all clusters propagates to the users of each cluster after the second stage transmissions.

Define a set $\mathcal{N} = \{1, \ldots, N\}$ of clusters, where each cluster $C_n, \forall n \in \mathcal{N}$ is now composed of a set $\mathcal{M}_n = \{1, \ldots, M\}$ of single antenna source-destination pairs, and a set $\mathcal{L}_n = \{1, \ldots, L\}$ of dedicated relays. We denote the $m$-th user (destination) of the $n$-th cluster as $U_{nm}$, $\forall n \in \mathcal{N}, m \in \mathcal{M}_n$, the respective source as $S_{nm}$, and the relays as $R_{nl}$, $\forall n \in \mathcal{N}, l \in \mathcal{L}_n$. Note that we assume for simplicity of notation, and without loss of generality, that all clusters contain the same number of source destination pairs $M$ and relays $L$.

In the multi-cluster scenario, the received signal at every relay $R_{nl}$ is a superposition of signals originating from the sources of all clusters

$$x_{nl} = \sqrt{P_0} \sum_{j \in \mathcal{N}} \sum_{m \in \mathcal{M}_j} f_{jm,nt}s_{jm} + v_{nl},$$

where, again, $P_0$ is the common transmit power of all sources and $s_{nm} \in \mathbb{C}$ denotes the, normalized to unit power, information symbol transmitted by source $S_{nm}$. Also,
$v_{nl} \sim \mathcal{CN}(0,1)$ is the noise at relay $R_{nl}$ and $f_{jm,nl}$ denotes the channel gain between source $S_{jm}$ and relay $R_{nl}$. Re-writing in matrix form, the received signal vector at all relays of cluster $C_n$ is

$$x_n = \sum_{j \in N} \sqrt{P_0} F_{jn} s_j + v_n,$$

where $s_j = [s_{j1}, \ldots, s_{jM}]^T \in \mathbb{C}^M$, $x_n = [x_{n1}, \ldots, x_{nL}]^T \in \mathbb{C}^L$, $v_n = [v_{n1}, \ldots, v_{nL}]^T \in \mathbb{C}^L$. The matrix $F_{jn} \in \mathbb{C}^{L \times M}$ is defined as the channel state matrix containing the channels from all sources of cluster $C_j$ to all the relays of cluster $C_n$, i.e.,

$$F_{jn} = \begin{bmatrix} f_{j1,n1} & \cdots & f_{jM,n1} \\ \vdots & \ddots & \vdots \\ f_{j1,nL} & \cdots & f_{jM,nL} \end{bmatrix} = [f_{j1,n} \cdots f_{jM,n}]^T,$$

where $f_{jm,n} = [f_{jm,n1}, \ldots, f_{jm,nL}]^T \in \mathbb{C}^L$ denotes the channel gain vector from source $S_{jm}$ to all relays of cluster $C_n$.

Similar to the single cluster scenario, during the second communication stage the relays of cluster $C_n$ retransmit, in an AF fashion, a linear transformation of their
respective received signals $x_n$, i.e.,

$$ t_n = W_n x_n = \sqrt{P_0} W_n \left( \sum_{j \in N} F_{jn} s_j \right) + W_n v_n, $$

where $t_n \in \mathbb{C}^L$ denotes the forwarded signal vector and $W_n \in \mathbb{C}^{L \times L}$ is the corresponding beamforming matrix of cluster $C_n$. Recall that we consider the case where every relay node carries a single antenna, which translates into the beamforming matrix being diagonal, i.e. $W_n = \text{diag}\{w_n^1, \ldots, w_n^L\} \in \mathbb{C}^{L \times L}$, where $w_n^l$ denotes the complex weight with which relay $R_{nl}$ multiplies its received signal.

The received signal vector $y_n \in \mathbb{C}^M$ for all users of each cluster $C_n$ is now a superposition of signals from the relays of all clusters, and can be expressed as

$$ y_n = \sum_{j \in N} G_{jn} t_j + z_n = \sum_{j \in N} \left( \sqrt{P_0} G_{jn} W_j \left( \sum_{i \in N} F_{ij} s_i \right) + G_{jn} W_j v_j \right) + z_n, \quad (5.4) $$

where $z_n = [z_{n1}, \ldots, z_{nM}]^T \in \mathbb{C}^M$ denotes the vector of i.i.d random noise components $z_{nm} \sim \mathcal{CN}(0,1)$ at user $U_{nm}$. The matrix $G_{jn} \in \mathbb{C}^{M \times L}$ is defined as the channel state matrix containing the channels from all relays of $C_j$ to all the users of $C_n$, i.e.,

$$ G_{jn} = \begin{bmatrix} g_{j1,n1} & \cdots & g_{jL,n1} \\
\vdots & \ddots & \vdots \\
g_{j1,nM} & \cdots & g_{jL,nM} \end{bmatrix} = [g_{j,n1} \ldots g_{j,nM}]^T, $$

with $g_{j,nm} = [g_{j1,nm}, \ldots, g_{jL,nm}]^T \in \mathbb{C}^L$ denoting the channel gain column vector from all relays of $C_j$ to $U_{nm}$. The received signal at user $U_{nm}$ is given by the $m$-th
entry of the vector $y_n$ in (5.4). The pertinent expression is

$$y_{nm} = \sum_{j \in N} g_{j,nm}^T t_j + z_{nm}$$

\[
\sqrt{P_0} g_{n,nm}^T W_n f_{n,m,n} s_{nm} + g_{n,nm}^T W_n \left( \sum_{i \in M_n} \sqrt{P_0} f_{ni,n} s_{ni} \right)
\]

Desired

Intra-Cluster Interference from same cluster’s sources other than $S_{nm}$

\[
+ g_{n,nm}^T W_n \left( \sum_{j \notin N} \sqrt{P_0} F_{jn} s_j \right) + \sum_{j \notin N} g_{j,nm}^T W_j \left( \sum_{i \in N} \sqrt{P_0} F_{ij} s_i \right).
\]

Inter/Intra-Cluster Interference from out-of-cluster sources

Inter-Cluster Interference

\[
+ \sum_{j \in N} g_{j,nm}^T W_j V_j + z_{nm},
\]

Noise

where we can now see more clearly how the three aforementioned, additional interference terms that arise in the multi-cluster scenario affect the formulation. Specifically, the term labeled “Inter/Intra-Cluster Interference from out-of-cluster sources” denotes the interference on the relays of each cluster, exerted by the transmissions of other clusters’ sources during the first communication stage. This interference propagates to the users in the second stage when the relays of the same cluster transmit, hence the hybrid characterization Inter/Intra-Cluster Interference. Also, the term labeled “Inter-Cluster Interference” denotes the interference on the users of each cluster, exerted by signals transmitted by the relays of other clusters that are intended for other users. Finally, notice that the term labeled Noise now also includes the noise at the relays of all clusters that propagates to the users of each cluster after the second stage transmissions. We assume that the two stages of the AF protocol are synchronized for the whole network, such that interference from source transmissions affecting user receptions directly is not possible.
Subsequently, the average SINR of user $U_{nm}$ is defined as

$$\text{SINR}_{nm} = \mathbb{E} \left( \frac{P_0 |g_{n,nm}^T W_n f_{n,nm} s_{nm}|^2}{\text{Desired}} \right)$$

$$\mathbb{E} \left( P_0 \sum_{i \neq m} |g_{n,ni}^T W_n f_{n,ni} s_{ni}|^2 \right) + \sum_{j \in \mathcal{N}} |g_{j,nm}^T W_j v_j|^2 + \frac{|z_{nm}|^2}{\text{Noise}}$$

$$+ P_0 \sum_{j \neq n} \sum_{k \in \mathcal{M}_j} |g_{n,km}^T W_n f_{n,km} s_{nk}|^2 + P_0 \sum_{j \neq n} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{M}_i} |g_{j,nm}^T W_j f_{jk,js}|^2 \right)$$

Thus, the multi-cluster beamforming problem entails finding $W_n$ that solves the optimization problem

$$\min_{\{W_n, \forall n \in \mathcal{N}\}} \sum_{n \in \mathcal{N}} P^n_T (W_n)$$

$$\text{s.t.} \quad \text{SINR}_{nm}(W_n) \geq \gamma_{nm}, \quad \forall n \in \mathcal{N}, \ m \in \mathcal{M}_n,$$

where the average, total transmitted power at the relays of cluster $C_n$ is calculated as

$$P^n_T = \mathbb{E}\{\|t_n\|^2_F\} = \sum_{j \in \mathcal{N}} \text{Tr}\left( P_0 W_n \mathbb{E}\{F_{jn} F_{jn}^H\} W_n^H \right) + \text{Tr}(W_n W_n^H).$$

To facilitate further exposition, and, also, to accommodate for the manipulations in Section 5.2, in what follows we will express (5.5) in a matrix form. To this end, the total transmitted power at the relays of $C_n$ can be written as

$$P^n_T = w_n^H R^n_T w_n,$$

where $w_n = [w_{n1}, \ldots, w_{nL}]^T \in \mathbb{C}^L$ is a column vector containing all the diagonal elements of $W_n$, and

$$R^n_T = I_L + P_0 \sum_{j \in \mathcal{N}} \sum_{m \in \mathcal{M}_j} \text{diag}\left\{ \mathbb{E}\{|f_{jm,n1}|^2\}, \ldots, \mathbb{E}\{|f_{jm,nL}|^2\} \right\}.$$
with $\mathbf{I}_L$ denoting the identity matrix of size $L$. Doing the same for the SINR expression, we define for every $n \in \mathcal{N}$, $m \in \mathcal{M}_n$ the desired signal matrices as

$$R_{S}^{nm} = \mathbb{E}\{(f_{nm,n}^T \circ g_{n,nm}^T) \mathcal{H}(f_{nm,n}^T \circ g_{n,nm}^T)\},$$

the intra-cluster interference matrices as

$$R_{I}^{nm} = \sum_{i \in \mathcal{M}_n}^{|i \neq m|} \mathbb{E}\{(f_{ni,n}^T \circ g_{n,nm}^T) \mathcal{H}(f_{ni,n}^T \circ g_{n,nm}^T)\},$$

and the inter/intra-cluster interference matrices as

$$R_{II}^{nm} = \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{M}_j}^{|j \neq n|} \mathbb{E}\{(f_{jk,n}^T \circ g_{n,nm}^T) \mathcal{H}(f_{jk,n}^T \circ g_{n,nm}^T)\}.$$

Moreover, for every $n \in \mathcal{N}$, $m \in \mathcal{M}_n$ and $j \in \mathcal{N} \setminus \{n\}$ we define the inter-cluster interference matrices as

$$R_{IC}^{j,nm} = \sum_{i \in \mathcal{N}}^{|i \neq n|} \sum_{k \in \mathcal{M}_i} \mathbb{E}\{(f_{ik,j}^T \circ g_{j,nm}^T) \mathcal{H}(f_{ik,j}^T \circ g_{j,nm}^T)\},$$

and, finally, the noise matrices as

$$R_{v}^{j,nm} = \text{diag}\{\mathbb{E}\{|g_{j1,nm}|^2\}, \ldots, \mathbb{E}\{|g_{jL,nm}|^2\}\}.$$

Note that all the above matrices are Hermitian. Using the above notation, the SINR$_{nm}$ is equivalently expressed as

$$\text{SINR}_{nm} = \left( P_0 w_n^\mathcal{H} R_{S}^{nm} w_n \right) / \left( P_0 w_n^\mathcal{H} R_{I}^{nm} w_n + P_0 w_n^\mathcal{H} R_{II}^{nm} w_n \right)
+ P_0 \sum_{j \neq n} w_j^\mathcal{H} R_{IC}^{j,nm} w_j + \sum_{i \in \mathcal{N}}^{|i \neq n|} w_i^\mathcal{H} R_{v}^{i,nm} w_i + 1 \right).$$

Then, problem (5.5) can be written in matrix form as

$$\min_{\{w_n, \forall n \in \mathcal{N}\}} \sum_{n \in \mathcal{N}} w_n^\mathcal{H} R_{T}^{nm} w_n$$

s.t. $w_n^\mathcal{H} Q_{nnm}^n w_n + \sum_{j \neq n} w_j^\mathcal{H} Q_{jnm}^j w_j \geq 1,$

$$\forall n \in \mathcal{N}, m \in \mathcal{M}_n,$$  

(5.6)
where we have further defined the matrices

\[
Q^{nnm} = \frac{P_0}{\gamma_{nm}} R_{S}^{nm} - P_0 R_{I}^{nm} - P_0 R_{I}^{nm} - R_{v}^{nm}
\]

and

\[
Q^{jnm} = - P_0 R_{jC}^{jnm} \quad \text{if } 
\]

to obtain a more compact notation for each SINR constraint. Note that the term

\[
w_j^H Q^{jnm} w_j
\]

eventually gathers all the terms that depend on the beamforming decisions of cluster \( C_j \) and appear in the SINR constraint of user \( U_{nm} \).

Since the matrices \( Q^{jnm}, \ Q^{nnm} \) will be, in general, indefinite, it follows that the optimization problem (5.6) belongs in the class of nonconvex Quadratically Constrained Quadratic Programming (QCQP) problems, which are NP-hard to solve. Nevertheless, by defining the variables

\[
X_j = w_j w_j^H, \quad \forall j \in \mathcal{N}
\]

and using the fact that

\[
w_j^H Q^{jnm} w_j = \text{Tr}(X_j Q^{jnm})
\]

we can express (5.6) in the equivalent form Vandenberghe and Boyd (1996)

\[
\min_{\{X_n, \forall n \in \mathcal{N}\}} \sum_{n \in \mathcal{N}} \text{Tr}(X_n R_n^T)
\]

\[
\text{s.t.} \quad \text{Tr}(X_n Q^{nnm}) + \sum_{j \notin \mathcal{N}} \text{Tr}(X_j Q^{jnm}) \geq 1,
\]

\[
X_n \in \mathbb{S}_+^L, \quad \forall n \in \mathcal{N}, \ m \in \mathcal{M}_n,
\]

\[
\text{rank}(X_n) = 1, \quad \forall n \in \mathcal{N}.
\]

where \( X_j \in \mathbb{S}_+^L \) imposes the (convex) constraint that matrix \( X_j \) belongs to the cone of symmetric, positive semidefinite matrices of dimension \( L \). Note that, since \( Q^{jnm} \) is Hermitian and \( X_j \) is symmetric, it follows that \( \text{Tr}(X_j Q^{jnm}) = \text{Tr}(X_j \Re(Q^{jnm})) \)

which means that the inequality constraint in (5.7) is well defined, where \( \Re(\cdot) \) returns the real part of a complex number.

Problem (5.7) is equivalent to (5.6) and still nonconvex because of the nonconvex rank constraint. Nevertheless, the rest of the problem is convex, which motivates the
relaxation of the rank constraint in order to obtain a problem that is manageable to solve. The resulting SDR of problem (5.7) becomes

$$\min_{\{x_n, y_n \in N\}} \sum_{\text{ne} N} \text{Tr}(x_n R^n_T)$$

(5.8)

s.t. \(\text{Tr}(x_n Q_{nm}) + \sum_{j \neq n} \text{Tr}(x_j Q^{jnm}) \geq 1, \)

$$x_n \in \mathbb{S}^L_+, \quad \forall n \in N, \ m \in M_n.$$  

Note that, by dropping the rank constraints, we essentially enlarge the feasible set. Hence, in general, the relaxation (5.8) will only yield an approximate solution to (5.7), with an optimal value that provides a lower bound for the original problem. Therefore, the optimizers \(x^*_n, \forall j \in N\) of (5.8) will not be rank-one in general, due to the relaxation. If they are, then they will be the optimal solution to the original problem (5.7). If not, randomization techniques Sidiropoulos et al. (2006) can be employed to obtain a rank one matrix.

Remark 39. Observe that, similar to Li et al. (2011a); Havary-Nassab et al. (2008); Fazeli-Dehkordy et al. (2009); Cheng and Pesavento (2012), the above formulation assumes knowledge of the second order statistics of channel state information (CSI). In a practical setting, this can be obtained based on past observations.

Remark 40. The inequality constraints in (5.8) must be active at the optimal solution (satisfied as equalities), because if they were not, we would be able to decrease the magnitudes of \(x_n\) further, thus invalidating the optimality assumption.

5.2 Distributed Relay Beamforming

Since the beamforming decisions in (5.8) are coupled in the constraint set, a central processing unit would have to be employed to gather the data involving the second order statistics of all channels, compute the optimal solution and then transmit the
optimal beamforming weights, expressed in the form of the beamforming matrices $X_n$, to the corresponding relays. However, this centralized approach would introduce congestion, delays, and would suffer from poor scalability, as the cluster population grows.

In what follows we present two different ways to implement ADAL on our particular multi-cluster beamforming problem, depending on how we express the coupling constraint set of (5.8). We describe these two possible implementations of ADAL, termed Direct and Indirect for reasons to become transparent later, and discuss their practical applications.

5.2.1 Direct method

To apply ADAL we need to reformulate (5.8) so that the coupling constraints are affine. For this, define auxiliary variables $\zeta_{njm}, \forall n, j \in \mathcal{N}, m \in \mathcal{M}_n$ that express the amount of “influence”, namely, either the desired signal power or interference exerted by all the relays of cluster $C_n$ on each user $U_{jm}$ of the system. In particular,

$$
\zeta_{nm} = \text{Tr}(X_n Q^{nm}) - 1 = \text{Tr} \left( X_n \left( \frac{P_0}{\gamma_{nm}} R_S^{nm} - P_0 R_I^{nm} - R_{v,R}^{nm} \right) \right) - 1 \quad (5.9)
$$

denotes the desired signal power for every user $U_{nm}$ belonging in cluster $C_n$, while

$$
\zeta_{njm} = \text{Tr}(X_n Q^{njm}) = \text{Tr} \left( X_n \left( -P_0 R_{v,c}^{nm} - R_{v,\nu}^{nm} \right) \right) \quad (5.10)
$$

denotes the interference exerted by cluster $C_n$ on user $U_{jm}$ that belongs to another cluster $C_j$, that is for every $j \in \mathcal{N} \setminus \{n\}$ and $m \in \mathcal{M}_j$. Furthermore, define the vector $\zeta_n = [\zeta_{n1}, \ldots, \zeta_{nM}]^T \in \mathbb{R}^M$ stacking all the “influences” of $C_n$. Then, problem
(5.8) can be equivalently written as

\[
\begin{align*}
\min_{(x_n, \forall n \in \mathcal{N})} \ & \sum_{n \in \mathcal{N}} \text{Tr}(X_n R^n_T) \\
\text{s.t.} \ & \sum_{n \in \mathcal{N}} \zeta_n = 0 \\
\ & \zeta_{nm} = \text{Tr}(X_n Q^{nm}) - 1, \ \forall n \in \mathcal{N}, \ m \in \mathcal{M}_n \\
\ & \zeta_{njm} = \text{Tr}(X_n Q^{njm}), \ \forall n \in \mathcal{N}, \ j \in \mathcal{N} \setminus \{n\}, \ m \in \mathcal{M}_j \\
\ & X_n \in \mathbb{S}_+^L, \ \forall n \in \mathcal{N},
\end{align*}
\]  

(5.11)

where \(0\) is the zero vector of dimension \(NM\). Note that we have replaced the inequality SINR constraints of (5.8) with the equality constraints \(\sum_{n \in \mathcal{N}} \zeta_n = 0\) in (5.11). This is acceptable since the SINR inequality constraints in (5.8) must be active at the optimal solution, i.e., satisfied as equalities; recall Rem. 40. The idea behind transforming (5.8) into (5.11) is that now the problem involves local constraints for each cluster, except for the coupling \(\sum_{n \in \mathcal{N}} \zeta_n = 0\) which is a simple affine constraint and thus amenable to distributed implementation using the ADAL.

The augmented Lagrangian associated with (5.11) is

\[
\Lambda(X, \zeta, \lambda) = \sum_{n \in \mathcal{N}} \text{Tr}(X_n R^n_T) + \lambda^T \sum_{n \in \mathcal{N}} \zeta_n + \frac{\rho}{2} \left\| \sum_{n \in \mathcal{N}} \zeta_n \right\|^2, \tag{5.12}
\]

where \(\lambda = [\lambda_{11}, \ldots, \lambda_{NM}]^T \in \mathbb{R}^{NM}\) is the vector of Lagrange multipliers (dual variables), \(X = \{X_1, \ldots, X_N\}\) and \(\zeta = \{\zeta_1, \ldots, \zeta_N\}\) denote the collection of all primal and auxiliary variables respectively, and \(\rho \in \mathbb{R}_+\) is a properly defined penalty coefficient. Note that we include only the constraint \(\sum_{n \in \mathcal{N}} \zeta_n = 0\) in (5.12), because the rest of the constraints are local at each cluster \(C_n\). For simplicity of notation, we collectively denote the set of points satisfying these local constraints of each cluster.
We define local augmented Lagrangians for every cluster \( C_n \)

\[
Z_n = \left\{ \zeta_n \in \mathbb{R}^{NM} \, | \, \zeta_{nm} = \text{Tr}(X_n Q^{nm}) - 1, \forall m \in M_n, \right. \\
\zeta_{njm} = \text{Tr}(X_n Q^{njm}), \forall j \in \mathcal{N}\backslash\{n\}, m \in M_j \}.
\]

We define **local augmented Lagrangians** for every cluster \( C_n \)

\[
\Lambda_n(X_n, \zeta_n, \tilde{\zeta}_j, \lambda) = \text{Tr}(X_n R^a) + \lambda^T \zeta_n + \frac{\rho}{2} \| \zeta_n + \sum_{j \neq n} \tilde{\zeta}_j \|^2_2,
\]

(5.13)

where we introduce variables \( \tilde{\zeta}_j \), denoting the primal variables that are controlled by \( C_j \) but communicated to \( C_n \) for optimization of its local Lagrangian \( \Lambda_n \). With respect to \( C_n \), these are considered fixed parameters. At each iteration \( k \) of ADAL, each cluster \( C_n \) begins by finding the minimizers \( \hat{\zeta}_n^k \) of its local augmented Lagrangian, as

\[
\hat{\zeta}_n^k = \arg\min_{X_n \in S^L, \zeta_n \in Z_n} \Lambda_n(X_n, \zeta_n, \tilde{\zeta}_j^k, \lambda).
\]

(5.14)

A key observation here is that each cluster \( C_n \) does not actually need global information to calculate (5.14), as it might appear at first sight by looking at the penalty term of each local AL. Although computing the penalty terms appears to require access to all \( \tilde{\zeta}_j, \forall j \in \mathcal{N}\backslash\{n\} \), one can readily observe that

\[
\| \zeta_n + \sum_{j \neq n} \tilde{\zeta}_j \|^2_2 = \sum_{i \in \mathcal{N}} \sum_{m \in M_i} \left( \zeta_{nim} + \sum_{j \neq n} \tilde{\zeta}_{jim} \right)^2
\]

(5.15)

where we recall that \( \zeta_{jim} \) denotes the “influence” that \( C_j \) exerts on user \( U_{im} \). In practical applications, each \( C_n \) will exert non-negligible interference (above a specified threshold) on a subset \( B_n \subseteq \{U_{11}, \ldots, U_{NM}\} \) of the set of active users and, consequently, we can set to 0 all \( \zeta_{nim}, \forall U_{im} \notin B_n \). Correspondingly, the summation terms in (5.15) for users \( U_{im} \notin B_n \) that do not experience interference from the operation of \( C_n \) are just constant terms in the optimization step (5.14) and can be neglected. In other words, each \( C_n \) only needs information from those clusters that
exert non-negligible “influence” on the users belonging in $\mathcal{B}_n$. Therefore, we can formally define the message-exchange neighborhood of cluster $C_n$ as the set of clusters $\mathcal{C}_n = \{ C_j : j \in \mathcal{N}, \mathcal{B}_j \cap \mathcal{B}_n \neq \emptyset \}$.

After calculating $\hat{\zeta}_n^k$ according to (5.14), each cluster $C_n, \forall n \in \mathcal{N}$ updates its estimates $\tilde{\zeta}_n$ that will be communicated to its neighbors $C_j \in \mathcal{C}_n$ according to

$$\tilde{\zeta}_{n}^{k+1} = \tilde{\zeta}_{n}^{k} + \tau (\hat{\zeta}_{n}^{k} - \tilde{\zeta}_{n}^{k}),$$

(5.16)

where $\tau$ is a stepsize, the determination of which is critical to the convergence properties of the method. Finally, the dual update is performed according to

$$\lambda^{k+1} = \lambda^{k} + \tau \rho \sum_{n \in \mathcal{N}} \tilde{\zeta}_{n}^{k+1}.$$  

(5.17)

The dual updates are distributed by structure. The Lagrange multiplier $\lambda_{nm}$, corresponding to the SINR constraint of user $U_{nm}$, must be updated, at iteration $k$, according to $\lambda_{nm}^{k+1} = \lambda_{nm}^{k} + \tau \rho \sum_{j \in \mathcal{N}} \tilde{\zeta}_{jnm}^{k+1}$. This summation needs to include “influences” only from those clusters that exert non-negligible “influence” on $U_{nm}$, i.e., the set $\mathcal{I}_{nm} = \{ C_j : U_{nm} \in \mathcal{B}_j, \forall j \in \mathcal{N} \}$.

We conclude this section with a remark on the stepsize parameter $\tau$. According to the convergence analysis in section 3.2, the primal stepsize $\tau$ in (5.16) must be determined as $\tau < \frac{2}{\max_{nm} |\mathcal{I}_{nm}|}$, where $|\cdot|$ denotes the cardinality of a set. Essentially, $\tau$ is affected by the number of clusters coupled in the “most populated” constraint in the system, i.e., the constraint corresponding to the user that suffers interference from the largest number of clusters. However here, the sets $\mathcal{I}_{nm}$ are not known a priori since they are determined by the optimal beamforming decisions. In this case, we need to determine conservative estimates for all the sets $\mathcal{I}_{nm}$, e.g., by letting each cluster’s relays send maximum power pilot signals before the execution of the distributed algorithm. An analogous line of reasoning can be used to determine the communication neighborhood $\mathcal{C}_n$ of each cluster $C_n$, which in turn depends on appropriately defining the sets $\mathcal{B}_n$. Last, we note that, based on the analysis in
Algorithm 11 Direct Method
Set $k = 0$ and determine estimates for the sets $C_n$ and $I_{nm}$, $\forall n \in \mathcal{N}$, $m \in \mathcal{M}_n$. Every cluster $C_n$ initializes and transmits its primal $\zeta_n^0$ and dual $\lambda_n^0$ variables.

1. Every cluster $C_n$ minimizes its local AL according to (5.14), after receiving the primal $\tilde{\zeta}_i^k$ and dual $\tilde{\lambda}_j^k$ variables from clusters $C_j \in C_n$.

2. Every cluster $C_n$ updates and transmits its primal variables $\tilde{\zeta}_i^{k+1}$ according to (5.16).

3. Every cluster $C_n$ updates and transmits the dual variables of its users $\lambda_{nm}^{k+1}$, $\forall m \in \mathcal{M}_n$, according to (5.17), after receiving the updated primal variables from clusters $C_j \in I_{nm}$, $\forall m \in \mathcal{M}_n$. Return to Step 1.

section 3.2, the penalty coefficient $\rho$ must remain constant throughout the iterative execution. The Direct method is summarized in Alg. 11.

5.2.2 Indirect Method

The indirect method is also a 3-step primal-dual iterative scheme. Every cluster solves a local convex optimization problem, cf. (5.22), and, in the next two steps, it updates and transmits its primal, cf. (5.23), and dual variables, cf. (5.24), respectively. The main difference with the direct method lies in the way that we formulate the coupling constraints, cf. (5.18), which, in turn, leads to a different message exchange scheme. In particular, the indirect method allows us to manually define the message exchange network, cf. (5.19), without any dependencies on the inter-cluster interference patterns (see the pertinent discussion for the direct method in the end of Section 5.2.1).

As with the direct method, the proposed indirect method also relies on defining appropriate auxiliary variables to introduce affine coupling constraints between the clusters. Consider, again, auxiliary variables $\zeta_{njm}$ exactly as described in (5.9) and (5.10) and $\zeta_n = [\zeta_{n11}, \ldots, \zeta_{nNM}]^T \in \mathbb{R}^{NM}$ and now also define $\zeta = [\zeta_1^T, \ldots, \zeta_N^T]^T \in \mathbb{R}^{N^2M}$ as the vector stacking all “influences” in the system. Furthermore, define local
variables \( \forall n, j \in \mathcal{N} \)

\[
\mathbf{\zeta}^{(n)} = \left[ (\mathbf{\zeta}_{1}^{(n)})^T, \ldots, (\mathbf{\zeta}_{N}^{(n)})^T \right]^T \in \mathbb{R}^{N^2L},
\]

where \( \mathbf{\zeta}_{j}^{(n)} = [\zeta_{j1}^{(n)}, \ldots, \zeta_{jN^{K}}^{(n)}]^T \in \mathbb{R}^{NL} \), so that each \( \mathbf{\zeta}^{(n)} \) acts as an individual estimate of the global vector \( \mathbf{\zeta} \) for every \( C_n \) and \( \mathbf{\zeta}_{j}^{(n)} \) expresses the estimate that \( C_n \) has for the “influences” exerted by \( C_j \) on the system. The key idea behind this approach is to allow each decision maker \( C_n \) to maintain and update its own estimate of the global state of the system. Correspondingly, we need to enforce “consensus” among all these local variables, by imposing coupling, affine constraints of the form

\[
\mathbf{\zeta}^{(1)} = \mathbf{\zeta}^{(2)} = \cdots = \mathbf{\zeta}^{(N)}.
\] (5.18)

There are many ways to express these equality constraints, depending on the message exchange capabilities between different clusters. In fact, let \( G = (V, E) \) denote a directed graph defined on the set of clusters so that \( V = \mathcal{N} \) is the set of vertices and \( E \subseteq V \times V \) is the set of edges so that \( E = \{(i, j) : j \in \mathcal{D}_i, \ i, j \in \mathcal{N}\} \). Here, \( \mathcal{D}_i \) denotes the set of the 1-hop out-neighbors of node \( i \) in the graph \( G \). Then, we can express (5.18) equivalently as

\[
\mathbf{\zeta}^{(1)} = \mathbf{\zeta}^{(j)}, \ \forall j \in \mathcal{D}_1
\] (5.19)

\[
\vdots
\]

\[
\mathbf{\zeta}^{(N)} = \mathbf{\zeta}^{(j)}, \ \forall j \in \mathcal{D}_N
\]

if and only if the graph \( G = (V, E) \) is weakly connected, i.e., if there exists an undirected path between any two nodes in the graph.

178
Using the coupling constraints (5.19), problem (5.8) can be transformed into

\[
\min_{\{X_n, \forall n \in N\}} \sum_{n \in N} \text{Tr}(X_n R^n_T) \tag{5.20}
\]

s.t. \[ \zeta^{(n)} = \zeta^{(i)}, \quad \forall n \in \mathcal{N}, \quad i \in \mathcal{D}_n \]

\[ \sum_{j \in \mathcal{N}} \zeta^{(n)}_{jnm} \geq 0, \quad \forall \ j, n \in \mathcal{N}, \ m \in \mathcal{M}_n \]

\[ \zeta^{(n)}_{njm} = \text{Tr}(X_n Q^{nm}) - 1, \quad \forall \ n \in \mathcal{N}, \ m \in \mathcal{M}_n \]

\[ \zeta^{(n)}_{njm} = \text{Tr}(X_n Q^{njm}), \quad \forall \ n \in \mathcal{N}, j \in \mathcal{N}\backslash\{n\}, \ m \in \mathcal{M}_j \]

\[ X_n \in \mathbb{S}^L_+, \quad \forall \ n \in \mathcal{N}. \]

Again, all the constraints in (5.20) are local to each \( C_n \) and the only coupling constraints are the consistency constraints \( \zeta^{(n)} = \zeta^{(i)}, \quad \forall n \in \mathcal{N}, \ i \in \mathcal{D}_n \), which are affine and thus amenable to decomposition by the ADAL algorithm.

Similar to the analysis in Section 5.2.1, the augmented Lagrangian associated with (5.20) is

\[
\Lambda(\{X_n, \zeta^{(n)}\}_{n=1}^N, \lambda) = \sum_{n \in \mathcal{N}} \text{Tr}(X_n R^n_T) + \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{D}_n} (\lambda^{(nl)})^T (\zeta^{(n)} - \zeta^{(l)})
\]

\[
\text{Ordinary Lagrangian}
\]

\[
+ \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{D}_n} \frac{\rho}{2} \|\zeta^{(n)} - \zeta^{(l)}\|^2_2;
\]

\text{Penalty term}

where \( \lambda^{(nl)} \in \mathbb{R}^{N \times M} \) is the vector of Lagrange multipliers corresponding to the constraint \( \zeta^{(n)} = \zeta^{(l)} \), defined for every \( n \in \mathcal{N}, \ l \in \mathcal{D}_n \). As in Section 5.2.1, we define local augmented Lagrangians \( \forall n \in \mathcal{N} \)

\[
\Lambda_n(X_n, \zeta^{(n)}, \tilde{\zeta}^{(j)}, \lambda^{(nl)}) = \text{Tr}(X_n R^n_T) + \sum_{i \in \mathcal{D}_n} (\lambda^{(nl)})^T \zeta^{(n)}
\]

\[
+ \sum_{\{m: m \in \mathcal{D}_m\}} (\lambda^{mn})^T (-\zeta^{(n)}) + \sum_{i \in \mathcal{D}_n} \frac{\rho}{2} \|\zeta^{(n)} - \tilde{\zeta}^{(l)}\|^2_2 + \sum_{\{m: m \in \mathcal{D}_m\}} \frac{\rho}{2} \|\tilde{\zeta}^{(m)} - \zeta^{(n)}\|^2_2;
\]
where, again, the terms $\tilde{\zeta}^{(l)}$ represent the local variables of each neighbor $C_l, l \in D_n$ of $C_n$, that are communicated to $C_n$ and considered constant with respect to the minimization of (5.21) at each respective iteration. Note that the term $\sum_{l \in D_n}(\lambda^{(nl)})^T\zeta^{(n)} + \sum_{\{m:n \in D_n\}}(\lambda^{(mn)})^T(-\zeta^{(n)})$ emerges from the consideration of the set of constraints (5.19).

Then, with every iteration $k$ of the algorithm, every cluster $C_n$ finds the minimizers $\hat{\zeta}^{(n),k}$ of the local problem as

$$
\begin{align*}
\hat{\zeta}^{(n),k} = \arg \min_{\{X_n, \zeta\}} \Lambda_n(X_n, \zeta^{(n)}, \tilde{\zeta}^{(j),k}, \lambda^k) \\
\text{s.t. } X_n \in S_L^n, \zeta^{(n)} \in Z_n,
\end{align*}
$$

where, again, we define

$$
Z_n = \left\{ \zeta^{(n)} \in \mathbb{R}^{N^2M} | \zeta^{(n)}_{nm} = \text{Tr}(X_n Q^{nm}) - 1, \forall m \in M_n, \right\}
$$

as the set of all points that satisfy the local constraints at each cluster. Subsequently, each cluster $C_n$ updates its estimates $\tilde{\zeta}^{(n),k}$ according to

$$
\tilde{\zeta}^{(n),k+1} = \tilde{\zeta}^{(n),k} + \tau(\hat{\zeta}^{(n),k} - \tilde{\zeta}^{(n),k}),
$$

and transmits the results to its in- and out-neighbors in the graph $G$. Finally, each cluster $C_n$ updates its dual variables $\lambda^{(nl),k+1}, \forall l \in D_n$ according to

$$
\lambda^{(nl),k+1} = \lambda^{(nl),k} + \tau \rho(\tilde{\zeta}^{(n),k+1} - \tilde{\zeta}^{(l),k+1}),
$$

and transmits the updated values to its in- and out-neighbors in the graph $G$.

The intuition behind the indirect method is that each $C_n$ tries to control its own variables, i.e., the entries $\zeta^{(n)}_{nm}, \forall j \in N, \ m \in M_j$, based on its current impression of the state of the system as expressed by the rest of the entries in $\zeta^{(n)}$. As the iterations progress, the updated decisions of each cluster diffuse into the system and
Algorithm 12 Indirect Method

Set $k = 0$ and define the consensus graph $G$, cf. (5.19). Every cluster $C_n$ initializes and transmits its primal $\tilde{\zeta}^{(n),0}$ and dual $\lambda^{(nl),0}$ variables.

1. Every cluster $C_n$ minimizes its local AL according to (5.22), after receiving the primal $\zeta^{(l),k}$ and dual $\lambda^{(l),k}$ variables from its in- and out-neighbors in the graph $G$.

2. Every cluster $C_n$ updates and transmits its primal variables $\tilde{\zeta}^{(n),k+1}$ according to (5.23).

3. Every cluster $C_n$ updates and transmits its dual variables $\lambda^{(nl),k+1}$, $\forall l \in D_n$, according to (5.24), after receiving the updated primal variables $\tilde{\zeta}^{(l),k+1}$ from its out-neighbors $l \in D_n$. Return to Step 1.

All clusters are forced to reach a consensus Jadabaie et al. (2005), in parallel with the optimization of the local utilities.

Note that each $\tilde{\zeta}^{(n)}$ does not necessarily need to include global information of the system. Instead, if we are able to estimate which users each cluster $C_n$ will exert a negligible interference on, then we can neglect the corresponding entries of $\tilde{\zeta}^{(n)}$ (and subsequently of $\lambda^{(nl)}$), thus reducing the size of the problem significantly. Moreover, note that according to the convergence analysis of ADAL in section 3.2, the choice of stepsize for the indirect method should be $\tau \leq 1/2$, because, for all possible communication graphs $G$, the coupling constraints will always involve only two decision makers (clusters). This is a direct consequence of the consensus constraints (5.19).

The Indirect method is summarized in Alg. 12.

5.3 Numerical Analysis

In this section, we illustrate the effectiveness of the proposed direct and indirect implementations of the ADAL algorithm for cooperative relay beamforming problems. We conducted simulations to examine the behavior of the proposed algorithms for different spatial configurations of the wireless networks and for various problem sizes. Comparative results with an existing distributed AL method, the Alternating
Direction Method of Multipliers (ADMM) Boyd et al. (2011), are also presented. The ADMM is known to exhibit fast convergence speeds in general, for small though accuracies Boyd et al. (2011).

In all numerical experiments, we followed a channel model encompassing large scale fading effects due to path loss and small scale fading, i.e., we defined the channel between the source $S_{nm}$ and relay $R_{nl}$ as

$$f_{nm, nl} = \alpha_{nm, nl} \cdot c_{nm, nl} \cdot e^{j(2\pi/\lambda)d_{nm, nl}}$$

(5.25)

where $\alpha_{nm, nl}$ captures multipath fading, $\lambda$ denotes the wavelength of carrier waves and $d_{nm, nl}$ denotes the Euclidean distance between the source $S_{nm}$ and relay $R_{nl}$, and $c_{nm, nl} = d_{nm, nl}^{-\mu/2}$, where $\mu = 3.4$ is the path loss exponent and represents the power fall-off rate. Note that for simplicity, we did not include large-scale shadowing effects in (5.25), however, the extension is straightforward. Also, we assumed Rayleigh fading such that the gains $\alpha_{nm, nl}$ are i.i.d circularly symmetric complex Gaussian random variables with zero mean and unit variance, i.e., $\alpha_{nm, nl} \sim \mathcal{CN}(0, 1)$. Correspondingly, for the purpose of simulations we constructed the channel state matrices by sampling realizations of Rayleigh random variables. The signal wavelength was assumed to be $\lambda = c/f = (3 \cdot 10^8)/(2.4 \cdot 10^9) = 0.125$ m which is a reasonable choice for wireless transmissions utilizing ultra high frequency carrier waves (2.4GHz).

In all the cases presented below, we have set the initial values of the primal variables to 0, and randomly sampled the dual variables from a uniform distribution in $[0, 1]$. Note that, different initialization values did not appear to affect the convergence speed significantly. Moreover, the penalty parameter $\rho$ is in general user defined in augmented Lagrangian methods. In our simulations, we have found that fastest convergence is obtained for values $\rho \in [1, 10]$, while at the same time preventing ill-conditioning.

In Fig. 5.4 we compare the two proposed methods, Direct and Indirect, for the
Figure 5.3: Two different spatial configurations of the multi-cluster network beamforming problem with: a) 5 clusters, and b) 5 clusters but larger interference levels compared to case (a), due to denser spatial positioning of the users. The blue and green circles correspond to sources and users, respectively, while the red dots depict the relays.

2 different setups of Fig. 5.3. Fig. 5.3(a) depicts a case with 5 clusters positioned in parallel, while Fig. 5.3(b) presents a case with 5 clusters but denser spatial positioning of the users. In both scenarios, we consider clusters containing 2 source-destination pairs and 5 relays, i.e., $|\mathcal{M}_n| = 2$ and $|\mathcal{L}_n| = 5$, $\forall \ n \in \mathcal{N}$, respectively. The same SINR requirement is set for all users at $\gamma = 10dB$. For the direct method, we assume that there exists at least one user that suffers non-negligible interference from all clusters, such that $\max_{nm} |I_{nm}| = 5$, and hence we set $\tau = \frac{2}{5} = 0.4$; recall the pertinent discussion in section 5.2.1. For the application of the indirect method, we model two cases: i) one which the available communication network between
clusters is a simple line formation, so that we impose the coupling constraints

\[ \zeta^{(i)} = \zeta^{(i+1)}, \quad i = 1, 2, \ldots, N - 1, \]

and ii) a denser network with all-to-all communication between clusters, so that we impose the coupling constraints

\[ \zeta^{(i)} = \zeta^{(j)}, \quad \forall \ i, j \in N. \]

The idea behind considering two different indirect cases is to examine the effect of the communication network on the underlying consensus operations on the decision variables of the clusters. Note that each coupling constraint in the indirect method involves two decision makers, such that \( \tau = \frac{1}{2} \) always; see also the pertinent discussion in section 5.2.2. The simulation results in Fig. 5.4 show that in all cases the distributed ADAL algorithm leads to very fast convergence. Here, we note that the entries of the beamforming matrices obtained by ADAL converge to the respective values of the centralized solution. We also observe that the Indirect method is slower than the Direct one. Moreover, it is true that the Indirect method converges faster for denser communication networks, which is in accordance with the literature on consensus algorithms Jadbabaie et al. (2005).

Fig. 5.4 also demonstrates how different system setups affect the speed of convergence. We observe that problems with a spatial configuration that induces higher interference levels, such as the setup of Fig. 5.3(b) compared to the setup of Fig. 5.3(a), tend to converge slightly slower. This is to be expected, since interference dominated scenarios will have SINR constraints that couple a relatively larger number of clusters, compared to cases with less interference for each user. This increased coupling naturally introduces the need for more coordination between the coupled decision makers (clusters), which leads to the slight increase in the number of iterations needed until convergence. To avoid confusion, we note that here we refer to the coupling between the beamforming decisions of all clusters due to the SINR
constraints, and not the coupling from the consensus constraints (5.19) used in the indirect method.

Next, we compare our proposed distributed algorithm with the ADMM Boyd et al. (2011), which also utilizes augmented Lagrangians. Fig. 5.5 presents the results corresponding to application of the Direct method on both setups of Fig. 5.3.

Finally, in Fig. 5.6 we compare the two algorithms on a larger size problem of 15 clusters, with 3 blocks of the setup depicted in Fig. 5.3(b) positioned in parallel. In this case, we assume that each user suffers non-negligible interference from at most 10 other clusters, i.e., we take a safe estimate $\max_{nm} |I_{nm}| = 10$, and hence we set $\tau = \frac{2}{10} = 0.2$. Fig. 5.6(b) contains the convergence results for the constraint violations $\sum_{n \in \mathcal{N}} \zeta_n = 0$, i.e., how much the sum differs from zero. In all the scenarios considered, we can observe that the ADAL algorithm converges significantly faster than the respective ADMM. Note that, in all cases, we have used the same initialization values for ADAL and ADMM. Moreover, after extensive
Figure 5.5: Comparison of the two different distributed algorithms, ADAL and ADMM. The blue lines correspond to a problem with the setup of Fig. 5.3(a), while the pink lines correspond to a problem with the setup of Fig. 5.3(b). The results correspond to application of the Direct method.

sensitivity analysis in our simulations, we found that ADMM requires relatively larger values of $\rho \in [4, 40]$, compared to ADAL.

5.3.1 Discussion

As was shown above, the direct method converges faster than the indirect approach in general. Nevertheless, depending on the problem setup, it might be necessary that that indirect method is applied. For example, consider the two different setups of Fig. 5.3 and suppose the relays of each cluster are responsible to perform the necessary computations for the execution of ADAL. For the setup of Fig. 5.3(a), we can observe that if one cluster exerts interference to the users of a neighboring cluster, then most likely the corresponding relays are in range to exchange the necessary messages (due to the parallel spatial positioning of the clusters). However, the same does not hold true for setups where the relays of certain clusters may not be in communication.
Figure 5.6: Comparative convergence results for the ADAL and ADMM algorithms on a scenario with 15 clusters and $\gamma = 10$ dB: a) Total transmitted power at the relays and b) Constraint feasibility evolution for the coupling constraints $\sum_{n \in N} \zeta_n = 0$. The results correspond to application of the Direct method.

range to exchange messages directly, even though they exert interference on each other’s users. For instance, that would be the case between the cluster pairs $C_1 - C_4$ and $C_3 - C_5$ in Fig. 5.3(b). In such scenarios, we can employ the indirect method by defining the consensus constraint set (5.19) appropriately, such that there exists a feasible path of communication links between all clusters, e.g. for the setup of Fig. 5.3(b) this could be $\zeta^{(1)} = \zeta^{(2)}$, $\zeta^{(2)} = \zeta^{(3)}$, $\zeta^{(4)} = \zeta^{(2)}$, $\zeta^{(5)} = \zeta^{(2)}$. Alternatively, we can apply the direct method by allowing cluster $C_2$ to act as a “message relay” and convey the necessary message exchanges between the cluster pairs $C_1 - C_4$ and $C_3 - C_5$ at each iteration of ADAL. On the other hand, there might exist cases where there is no feasible communication path between all clusters, i.e., it is not possible to define a weakly connected graph $E$ for the consensus constraints (5.19). In such scenarios, successful application of a distributed algorithm would require the users to act as “message relays” and convey the necessary message exchanges between the clusters.
Conclusions and Future Work

6.1 Summary of Main Results

Motivated by the ever increasing size and complexity of modern day problems, and the ongoing advancements in the parallel processing capabilities of contemporary computers, in this thesis we focused on developing novel distributed optimization algorithms for a certain class of convex constrained optimization problems. In particular, we were concerned with general problems that involve minimizing the sum of local objective functions whose arguments are local variables. The local variables can be constrained to lie in local convex compact sets, and can also be coupled via global linear constraints.

The goal was to come up with methods that possess the following set of desirable characteristics

- fast convergence,
- wide applicability,
- simple implementation,
• low computational costs,

• they can withstand random errors/inaccuracies in the computations,

• they can withstand noise corruption in the message exchanges during the distributed, iterative execution.

Following these guidelines, we proposed the Accelerated Distributed Augmented Lagrangians (ADAL) method and developed a rigorous theoretical framework that addresses the above requirements. ADAL is a distributed iterative scheme, wherein at each iteration all agents update their local decisions based only on local computations and message exchanges with other neighboring agents. ADAL is based on the augmented Lagrangian framework and addresses optimization problems that involve a separable objective function subject to convex local constraints and linear global coupling constraints. We provided formal proofs that the proposed distributed method converges to the optimal solution of the original problem in convex settings, and characterized its worst case convergence rate. Moreover, through extensive simulations, we established that the proposed method outperforms in practice existing state-of-the-art methods.

We also extended the convergence analysis of ADAL to include non-convex settings, where the local objective functions are possibly non-convex. Using assumptions that are common in the analysis of non-convex optimization methods, we established convergence of the distributed method to a local minimum of the problem. To the best of our knowledge, this is the first work that shows convergence of a distributed augmented Lagrangian method applied to non-convex optimization problems.

Moreover, we proposed two alternative ways to select the stepsizes in ADAL. The first way leads to a scalar stepsize that is common for all the distributed agents, but requires knowledge of the global structure of the linear coupling constraint set at
initialization. The second way leads to vector stepsizes, wherein we have \( m \) stepsizes associated with each one of the \( m \) linear coupling constraints; each stepsize must adhere to a condition that requires only local information from the corresponding constraint.

We were also able to extend the deterministic convergence results for convex problems to a stochastic scenario wherein the computations and communications of the decentralized processors are subject to noise and uncertainties. This line of work is directly related to the literature of stochastic approximation. To the best of our knowledge, this is the first attempt to consider this class of problems in the context of distributed stochastic approximation techniques. Moreover, this is the first distributed stochastic approximation method that is based on the augmented Lagrangian framework, which is well-known to be a very efficient approach for optimization in deterministic settings. We established conditions under which our method is guaranteed to converge with probability 1 to the optimal sets in both the primal and dual domains.

Last but not least, we considered the nontrivial problem of cooperative beamforming in relay wireless networks, for scenarios in which multiple clusters of source-destination node pairs, along with their dedicated relays, coexist in space. We provided a novel formulation of the problem in a convex setting and examined two different ways to obtain a distributed solution that allows for autonomous computation of the optimal beamforming decisions by each cluster, while taking into account intra- and inter-cluster interference effects.

6.2 Future Directions

We have identified the following topics as possible future research directions.
6.2.1 Asynchronous Implementation

Distributed optimization algorithms are essentially parallelized methods that separate into several local algorithms operating at different processors. They are iterative procedures that require message exchanges between neighboring processors in order to guarantee convergence to the optimal solution of the original, centralized problem. In practice, the time stamps of the exchanged messages and the times that these exchanges take place play a critical role in the convergence of a distributed algorithm. Synchronous algorithms are characterized by synchronous exchanges (usually at every iteration) of up-to-date messages. Synchronization often requires additional mechanisms that allow processors to detect the end of each iteration. In practical applications, synchronization is a very restrictive condition to impose and extremely hard to satisfy. First, synchronization of clocks in a distributed manner is extremely difficult. Second, synchronization depends on the computation and communication (message exchange) capabilities of the nodes. In fact, even if identical processors are considered (that perform computations at the same speed), the overall iterative execution of the algorithm will still be characterized by inherent asynchronicity, due to the influence that the ambient environment exerts on, e.g., the communications. As a result, synchronization will require that processing nodes in the network need to wait, at every iteration, until the slowest nodes receive all messages from their predetermined neighbors. This, unavoidably, introduces large delays and can be a serious bottleneck for a distributed algorithm.

Our goal in this research thrust is to explore asynchronous implementations of the proposed distributed algorithm, that allow each node to perform computations with possibly outdated information about the system’s current state, without compromising the convergence properties of the overall method. Another, equivalent, way of looking at the problem of outdated information is that the underlying communication
graph of the network has links that suffer from intermittent failures, so that certain messages are received with a delay or are not received at all. Our goal is to study the performance and convergence properties of our algorithm as a function of the delays in the message exchange and update procedure. The conditions for convergence of an asynchronous algorithm might be more stringent than the corresponding conditions for its synchronous counterpart.

6.2.2 More Advanced Stochastic Formulations

Thus far, we have considered optimization settings wherein the problem parameters are assumed to be known a priori and we make rational decisions based on this knowledge. In practice this is rarely the case, since uncertainty is always present. For instance, in network flow problems, uncertainty may materialize in the form of varying channel reliabilities and/or traffic bursts. With that in mind, it is easily understood that deterministic formulations will generally lead to poor performance in practice, because small variations may drastically change the optimality characteristics of a given solution.

Traditionally, network optimization has employed first or second order statistics to characterize the variation of the problem parameters and has significantly relied on the selection of particular utility functions. While the use of first order statistics can provide important insight in the design of a given problem, it is not sufficient to capture, e.g. channel reliability variations and traffic bursts. Moreover, approaches that employ second order statistics produce robust, worst-case solutions that can be overly conservative and do not take into account the particular characteristics of the distribution functions of the randomly varying parameters. More importantly, in practice it is almost impossible to correctly elicit the appropriate utility function of a decision maker explicitly. This problem is aggravated in cases where there is a group of decision makers, e.g. the network nodes in our case, with different utility
functions who have to reach a consensus. For all these reasons, there is a great need for new ways to design networks that can be considered *unanimously stochastically optimal*, i.e. optimal within desired confidence levels and for multiple performance metrics (utility functions) that share only the minimum number of necessary, general characteristics, namely monotonicity (more is better than less) and concavity (risk aversion).

In this objective we propose the use of stochastic dominance theory Wolfstetter (1993); Calafiore and Dabbene (2005); Dentcheva and Ruszczynski (2008, 2004); Dentcheva and Martinez (2012). Optimization with stochastic dominance constraints is a very recent development that has been mainly used in decision theory and economics. It provides us with a way to obtain solutions that essentially introduce a form of ranking among random variables, in a manner that captures all the aforementioned desired characteristics of stochastic optimality and unanimity. The main advantages of this formulation are twofold. First, the optimal decisions are tailored to the specific characteristics of the distribution functions of the randomly varying problem parameters. In contrast, the more common robust optimization formulations compute the decisions based just on the mean and variance of the random varying parameters, which inevitably leads to considerably more conservative solutions. Second, decisions with stochastic dominance constraints are the same for any monotone and concave utility function, meaning that solutions are independent of specific choices for utility functions (that are monotone and concave).
Bibliography


198


201


Shapiro, A. (1993), “Asymptotic behavior of optimal solutions in stochastic pro-

Cone Constrained Optimization,” Mathematics of Operations Research, 29, 479–
491.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2009), Lectures on stochastic pro-
gramming: modeling and theory, Society for Industrial and Applied Mathematics,
Philadelphia, USA.

Robust Multi-Cell Coordinated Beamforming with Imperfect CSI: An ADMM Ap-
proach,” IEEE Trans. on Signal Processing, 60, 29883003.

rection Method for Matrix Separation Based on Low-rank Factorization,” Opti-
mization Methods Software, 29, 239–263.

gence of the ADMM in Decentralized Consensus Optimization,” Signal Process-

Series in Computational Mathematics, Springer.


1286.

based optimal sensor scheduling with application to observer trajectory planning,”
Automatica, 43, 817 – 830.

Spall, J. (1992), “Multivariate stochastic approximation using a simultaneous pertur-
bation gradient approximation,” Automatic Control, IEEE Transactions on, 37,
332–341.

Spall, J. (2000), “Adaptive stochastic approximation by the simultaneous perturba-

Stochastic Optimization,” Selected Topics in Signal Processing, IEEE Journal of,
5, 772–790.

208


Biography

Name: Nikolaos Chatzipanagiotis
Date of birth: 06/30/1982
Place of birth: Athens, Greece

Education

Duke University, Durham, NC, USA 09/2012 – 12/2015
PhD in Mechanical Engineering

Stevens Institute of Technology, Hoboken, NJ, USA 09/2010 – 08/2012
PhD candidate in Mechanical Engineering

National Technical University of Athens, Athens, Greece 09/2006 – 10/2008
MSc in Microsystems and Nanodevices

5 yr. Diploma in Mechanical Engineering

Awards

Gerondelis Foundation Fellowship, 2012
Innovation and Entrepreneurship (I&E) Fellowship, 2010-2012, Stevens IT

Publications

Chatzipanagiotis et al. (2014a); Chatzipanagiotis and Zavlanos (2015a,b); Chatzipanagiotis et al. (2014b, 2013, 2012a,b); Chatzipanagiotis and Zavlanos (2013, 2015c); Kalogerias et al. (2013); Florakis et al. (2009b,a)