Consensus Planning with Primal, Dual, and Proximal Agents

Alvaro Maggiar
Amazon Supply Chain Optimization Technologies
maggiara@amazon.com

Lee Dicker
Amazon Supply Chain Optimization Technologies
leehd@amazon.com

Michael W. Mahoney
Amazon Supply Chain Optimization Technologies
zmahmich@amazon.com

August 29, 2024

Abstract

Consensus planning is a method for coordinating decision making across complex systems and organizations, including complex supply chain optimization pipelines. It arises when large interdependent distributed agents (systems) share common resources and must act in order to achieve a joint goal. In this paper, we introduce a generic Consensus Planning Protocol (CPP) to solve such problems. Our protocol allows for different agents to interact with the coordinating algorithm in different ways (e.g., as a primal or dual or proximal agent). In prior consensus planning work, all agents have been assumed to have the same interaction pattern (e.g., all dual agents or all primal agents or all proximal agents), most commonly using the Alternating Direction Method of Multipliers (ADMM) as proximal agents. However, this is often not a valid assumption in practice, where agents consist of large complex systems, and where we might not have the luxury of modifying these large complex systems at will. Our generic CPP allows for any mix of agents by combining ADMM-like updates for the proximal agents, dual ascent updates for the dual agents, and linearized ADMM updates for the primal agents. We prove convergence results for the generic CPP, namely a sublinear $O(1/k)$ convergence rate under mild assumptions, and two-step linear convergence under stronger assumptions. We also discuss enhancements to the basic method and provide illustrative empirical results.

1 Introduction

Consensus Planning. Consensus planning refers to a coordination mechanism to align different distributed agents (or systems) who share common resources and who must act in order to achieve a joint goal. For example, in the context of a large retailer, such a distribution of tasks is necessary, given the scale of the supply chain problem, with separate systems focusing on (say) buying, removals, placement, capacity control, fulfillment, or transfers, to name just a few. These individual components act semi-independently, but some form of communication is required, given their interdependence, to ensure that they work in unison. This alignment is often times built through ad-hoc solutions, or proxies of connected systems; and thus there is a need for generic Consensus Planning Protocols (CPPs) to solve such problems. A CPP starts with the agents, and it introduces a coordinator. Information is exchanged between the agents through the coordinator in a structured manner; and the coordinator reconciles the various favored plans of the individual agents in a principled way to eventually reach a consensus. Associated with such mechanisms are agent-level plans, costs that characterize their favored plans, and costs of deviating from them.

Problem. Consensus planning is often viewed as a variant of distributed optimization, where there are relatively few optimization agents and querying information from each agent is comparatively costly. A

CPP relies on the iterative exchange of information between agents and a coordinator. The mechanism through which the coordinator updates and aligns the plans and prices hinges upon the nature of the agents. A theoretically-popular approach centers around the use of *proximal* interfaces, which are motivated by the theoretically-attractive properties of the Alternative Direction Method of Multipliers (ADMM) [BPC+11]. In this proximal framework, the agents and the coordinator exchange two types of information: 1) a favored plan (primal variable); and 2) a price (or cost of deviation) (dual variable). These two pieces of information are iteratively exchanged and updated by the agents and coordinator until convergence. Implicit in this approach of using ADMM as a coordination mechanism is the assumption that all agents are amenable to interacting with the coordinator via a proximal interface.

In many practical industrial settings, however, this assumption is not satisfied. The reason is that large complex systems already in place cannot be readily modified to fit this framework without incurring large engineering costs and development time. One example of such a hurdle would be in the presence of systems designed as large linear or mixed integer programs that would not allow for the addition of a proximal (nonlinear) term in their objective function. Another example is when a system that knows its utility function and can output the value and the gradient of that function (given a proposed plan) interacts with an agent that takes prices as input and outputs a plan but can not output the value of its utility function.

More generally, we can categorize agents into three broad classes, depending on the interface that they support: primal, dual, and proximal. Primal agents can be probed given a tentative plan, and they return a corresponding cost of deviation; dual agents can be called with a price, and they return a corresponding favored plan; and proximal agent can consume both a tentative plan and tentative cost, and they return updated values thereof. Most CPP algorithms are built with one type of agents in mind, requiring that all agents belong to the same class. The practical difficulty that we address theoretically in this paper is to devise methodologies that allow for a "mix-and-match" of agents, that are compatible with all types of agent interfaces, so as not to require the preliminary burden of modifying them to fit the same type.

Main contributions. The purpose of this paper is to propose a generic CPP algorithm that is compatible with all three type of agents, without needing that any agent be modified. Our main algorithm combines ADMM-like updates for the proximal agents, dual ascent updates for the dual agents, and linearized ADMM updates for the primal agents. We prove several convergence results, including sublinear $O\left(\frac{1}{k}\right)$ convergence rate under mild conditions, and two-step linear convergence under stronger conditions; and we provide numerical examples that illustrate the behavior of the algorithm under different combinations of agent types, as well as the impact of acceleration.

2 Background

2.1 Consensus Problem

The consensus problem is one that involves a set of agents $\mathcal{M}$ that each have their own cost function $g_i(x_i), i \in \mathcal{M}$. The goal is to optimize the sum of these functions for a common plan z: $\min_z \sum_{i\in\mathcal{M}} g_i(z)$. In order to leverage the separability of the agents, the problem is re-written to endow each agent with their own plan variable, subject to the constraint that they must all be equal:

$$\begin{aligned} &\min_{x_i,z} \sum_{i\in\mathcal{M}} g_i(x_i) \\ &s.t.\ x_i = z,\ \forall i \in \mathcal{M}. \end{aligned}$$

Note that each agent's objective function could itself be the result of an optimization problem of the form $g_i(x) := \min_y g_i(x, y)$, where y corresponds to a set of variables that are "private" to the agent. This formulation can be interpreted as the agents working towards agreeing on a common plan z, and optimizing their own objective function, given the consensus plan. Our objective is to define a generic Consensus Planning Protocol (CPP) to solve this general consensus problem. What algorithm we use depends in part on the interface the agents offer, in other words, what information they can consume, and what information they return. In practical systems, there are many ways agents can output information and interact with other agents and/or a coordinating algorithm. We detail in the

next section the three main types of agents considered in our consensus planning problem: primal, dual, and proximal agents.

2.2 Agents

In our consensus planning problem, there exist three types of agent interfaces, primal, dual, and proximal, which are the natural interfaces for the three main approaches to the problem, gradient descent, dual ascent, and the augmented Lagrangian method of multipliers, respectively.

Primal Agents: A primal interface is the one that would be required in a first-order primal method such as gradient descent, and corresponds to the basic formulation of the consensus problem as:

$$\min_z \sum_{i \in \mathcal{M}} g_i(z).$$

In a simple gradient descent scheme, we would in each iteration k query the agents with the current tentative plan z^k, and they would return the gradients $\nabla g_i(z^k)$ at that point. We would then update the plan as $z^{k+1} = z^k - \rho \sum_{i \in \mathcal{M}} \nabla g_i(z^k)$, for some step size $\rho > 0$.

Dual Agents: A dual interface arises from a dualization of the constraint, and then solving of the dual problem as:

$$\max_{\lambda_i} \left\{ \min_{x_i} \sum_{i \in \mathcal{M}} g_i(x_i) - \lambda_i^T (z - x_i) \right\}.$$

We then require from the agents that they be able to consume the dual variables λ_i and return the solutions to the subproblems in bracket as $x_i = \arg\min_x g_i(x) - \lambda_i^T (z - x)$.

Proximal Agents: A proximal interface results from the augmented Lagrangian formulation of the consensus problem, as:

$$\max_{\lambda_i} \left\{ \min_{x_i} \sum_{i \in \mathcal{M}} g_i(x_i) - \lambda_i^T (z - x_i) + \frac{\rho}{2} \|z - x_i\|^2 \right\}.$$

In this case, the agents need to be able to consume three pieces of information, prices λ_i, as in the dual agent, but also a consensus plan z, and a regularizing parameter ρ. They then return the solution to the subproblem: $x_i = \arg\min_x g_i(x) - \lambda_i^T (z - x) + \frac{\rho}{2} \|z - x\|^2$.

One of the motivations of the CPP is the ADMM algorithm and its favorable properties, which make the proximal interface a popular one. However, as a practical matter, in many realistic systems consisting of large complex agents, those agents cannot be readily "turned into" proximal agents. Depending on their implementation, agents may naturally exist as primal or dual agents. This requires that generic CPP algorithms be able to work across interface types.

2.3 Previous Work

Consensus planning is often viewed as a variant of distributed optimization, where there are relatively few optimization agents and querying information from each agent is comparatively costly. This variant is in contrast with much of the distributed optimization work where the central concern is that of scale across many agents [RY22, ch.11]. Most of the distributed optimization algorithms assume the same interface for all agents: for example, a primal interface in the case of distributed (proximal) gradient descent; a dual interface in the case of dual decomposition; and a proximal interface in the case of distributed ADMM. By contrast, this paper allows for any mix of primal, dual, and proximal interfaces.

2.4 Motivating Examples

We present in Table 1 several applications where consensus planning is relevant for a large online retailer. Note that, in many cases, the interface of a given agent is fixed by the existing implementation and design thereof, and thus must be accommodated by a generic CPP. These examples involve a few agents with varying types of interfaces, often times with mixed interfaces. Part of the motivation of this work is to offer a single, unifying CPP algorithm to handle any such use case, in which different agents have different interfaces.

Application	Description
Fullness Optimization	*Objective*: coordinate inventory buying with physical network capacity. *Agents*: buying agent and fullness (capacity) agent. *Interfaces*: dual for the buying agent; primal for the capacity agent.
Throughput coordination	*Objective*: coordinate inventory flows throughout different regions (e.g., inbound, transfer, outbound), given network labor constraints. *Agents*: each region. *Interfaces*: proximal interface for all agents.
Transportation optimization	*Objective*: coordinate transportation capacity across delivery stations and third party carriers. *Agents*: different stations, which cover overlapping geographical areas. *Interfaces*: proximal interface for all agents.
Arrivals and throughput coordination	*Objective*: Like in throughput coordination, coordinate inventory flows, but this differs because there is only one region and only one inventory flow (inbound arrivals) is considered. *Agents*: buying agent (similar to fullness optimization problem) and throughput agent (analogous to throughput coordination problem). *Interfaces*: buying agent has a dual interface; throughput agent may use a primal or dual interface.

Table 1: Examples of consensus problems at a large online retailer.

3 Algorithm

3.1 Derivation

We consider a consensus optimization problem involving a mixed set of agents $i \in \mathcal{M}$ that can be either primal, dual, or proximal. We let $\mathcal{P}$ be the set of primal agents, $\mathcal{D}$ be the set of dual agents, and $\mathcal{X}$ be the set of proximal agents. Each agent's objective function is given by g_i, where i denotes their index, and we assume that the functions g_i are convex, and in particular μ_i-strictly convex for $i \in \mathcal{D}$, and have β_i-Lipschitz continuous gradients for $i \in \mathcal{X}$. The consensus problem to be solved is:

$$\begin{aligned} \min_{\mathbf{x},z} \quad & \sum_{i\in\mathcal{M}} g_i(x_i) \\ \text{s.t.} \quad & Az = \mathbf{x} \end{aligned} \tag{1}$$

where:

$$A = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}.$$

We first make the different treatments of the agents explicit by separating agents based on their type. Applying partial duality (e.g., [LY21, ch. 14]) to the dual and proximal agents, and augmenting the latter with the augmented Lagrangian proximal term associated with their corresponding constraint, yields the following equivalent formulation:

$$\begin{aligned} \max_{\lambda_i, i\in\mathcal{D}\cup\mathcal{X}} \min_{x_i,z} \quad & \sum_{i\in\mathcal{P}} g_i(x_i) + \sum_{i\in\mathcal{D}} g_i(x_i) + \lambda_i^T(z - x_i) + \sum_{i\in\mathcal{X}} g_i(x_i) + \lambda_i^T(z - x_i) + \frac{\rho_i}{2}\|z - x_i\|^2, \\ \text{s.t.} \quad & x_i = z \quad \forall i \in \mathcal{P} \end{aligned} \tag{2}$$

where we explicitly separate the three types of agents, and ρ_i is the augmented Lagrangian parameter associated with agent $i \in \mathcal{X}$.

Formulation (2) already suggests a direction for the solution of the problem, owing in particular to the similarity in structure between ADMM [BPC+11] and traditional dual ascent methods [LY21, ch. 14] used to solve dualized consensus problems. Both of these approaches involve iterations comprised of the same 3 steps:

1. x_i update: update of the agents' primal variables,
2. z update: update of the consensus plan through averaging,
3. λ_i update: update of the dual variables.

The similarity between dual and proximal agents can also be explained by the fact that one interpretation of the augmented Lagrangian relaxation is that it is simply the regular Lagrangian relaxation applied to a penalized version of the objective function (penalized by the addition of the quadratic term). That these two type of updates can be performed jointly is still to be proved, but it provides us with a framework within which to operate. We then need to further incorporate the primal agents, which are queried with a tentative plan, x_i, and return gradient information at that point. We detail in the next paragraph how we treat primal agents as approximate proximal agents, and we integrate them in the framework outlined above.

Primal agents as approximate proximal agents. A natural idea is to recast the primal agent as a proximal agent, where its function g_i is replaced by some linearized approximation thereof using the gradient information returned by the primal agent. Fortunately, a substantive body of work has demonstrated that the ADMM updates do not need to be exact for the method to converge [HLHY02, EB92, YZ11]. In particular, the terms of the ADMM updates can be linearized, be it the augmented quadratic term [LLS11, YY13], or the agent function [OHTG13, Suz13], or both through a more general use of a Bregman divergence term to replace the quadratic penalty, and an additional Bregman divergence term [WB14]. Leveraging these latter results, a simple Bregman-ADMM formulation in the context of the consensus problem has updates of the form:

$$x_i^{k+1} = \arg\min_x g_i(x) - {\lambda_i^k}^T x + \frac{\rho_i}{2}\|z - x\|^2 + D_{\phi_i}(x, x_i^k),$$

where D_{ϕ_i} is the Bregman distance associated with the convex function ϕ_i (see Appendix B.1). Letting ϕ_i be defined as:

$$\phi_i(x) = \frac{L_i}{2}\|x\|^2 - g_i(x),$$

where $L_i \geq \beta_i$ is a constant greater than the Lipschitz constant of ∇g_i, the update reads:

$$x_i^{k+1} = \arg\min_x g_i(x_i^k) + \nabla g_i(x_i^k)^T x + \frac{L_i}{2}\|x - x_i^k\|^2 - {\lambda_i^k}^T x + \frac{\rho_i}{2}\|z^k - x\|^2. \tag{3}$$

We can observe that the function g_i has been linearized and replaced by a quadratic upper bound. The solution to this subproblem is readily obtained as:

$$x_i^{k+1} = \frac{L_i x_i^k + \rho_i z^k}{L_i + \rho_i} - \frac{1}{L_i + \rho_i}\left(\nabla g_i(x_i^k) - \lambda_i^k\right).$$

We will see in Section 3.3 that when all agents are primal, the application of this linearized ADMM yields updates that bear a strong similarity to the ones performed by distributed gradient descent [NO09].

3.2 Formulation

We detail in this section the steps of the algorithm following the high level derivation presented in Section 3.1. The agents can be either primal ($i \in \mathcal{P}$), dual ($i \in \mathcal{D}$), or proximal ($i \in \mathcal{X}$). We recall that we consider μ_i-strongly convex functions $g_i, i \in \mathcal{D}$ for dual agents, and β_i-Lipschitz continuous gradients for $g_i, i \in \mathcal{P}$. We allow for $\mu_i = 0$ and $L_i = +\infty$ for those other functions that are not strongly convex or do not have Lipschitz continuous gradients, for notational simplicity. Each agent is allowed to have their own regularizing parameter/learning rate ρ_i, although in practice we often use the same one for agents of the same type. We will return to a discussion on the choice of these hyperparameters in Section 5, but since the dual agents essentially perform dual ascent, and as we will confirm in the proof of the convergence results, we impose that $\rho_i < \mu_i, \in \mathcal{D}$. On the other hand, we recall that we require $L_i > \beta_i$ for $i \in \mathcal{P}$, and place no restriction on $\rho_i > 0$ for $i \in \mathcal{X}$. We further assume throughout the paper that the initial prices λ_i^0 are such that $\sum_{i \in \mathcal{M}} \lambda_i = 0$.

In each iteration of the algorithm, we alternate between agent updates, consensus update, and price updates as follows:

Agent Updates: Update the agents $i \in \mathcal{M}$ in parallel, with the following updates depending on the agent type.

Primal Agents ($i \in \mathcal{P}$):

$$\begin{aligned} x_i^{k+1} &= \arg\min_x g_i(x_i^k) + \nabla g_i(x_i^k)^T x + \frac{L_i}{2}\|x - x_i^k\|^2 - {\lambda_i^k}^T x + \frac{\rho_i}{2}\|z^k - x\|^2, \\ &= \frac{L_i x_i^k + \rho z^k}{L_i + \rho_i} - \frac{1}{L_i + \rho_i}\left(\nabla g_i(x_i^k) - \lambda_i^k\right). \end{aligned} \tag{4}$$

Dual Agents ($i \in \mathcal{D}$):

$$x_i^{k+1} = \arg\min_x g_i(x) - {\lambda_i^k}^T x. \tag{5}$$

Proximal Agents ($i \in \mathcal{X}$):

$$x_i^{k+1} = \arg\min_x g_i(x) - {\lambda_i^k}^T x + \frac{\rho_i}{2}\|z^k - x\|^2. \tag{6}$$

Consensus Update:

$$z^{k+1} = \arg\min_z \sum_{i \in \mathcal{P}\cup\mathcal{D}\cup\mathcal{X}} {\lambda_i^k}^T z + \sum_{i \in \mathcal{P}\cup\mathcal{D}\cup\mathcal{X}} \frac{\rho_i}{2}\|z - x_i^{k+1}\|^2.$$

The update takes the form of a weighted average of the agents' plans (using the fact that the sum of the price variables is null):

$$z^{k+1} = \frac{1}{\sum_{i\in\mathcal{M}} \rho_i} \sum_{i \in \mathcal{M}} \rho_i x_i^{k+1}. \tag{7}$$

Price Updates:

$$\lambda_i^{k+1} = \lambda_i^k + \rho_i\left(z^{k+1} - x_i^{k+1}\right) \qquad i \in \mathcal{P}\cup\mathcal{D}\cup\mathcal{X}. \tag{8}$$

Remark 3.1. *All the agents' problems can be expressed through the same equation as:*

$$x_i^{k+1} = \arg\min_x g_i(x) - {\lambda_i^k}^T x + \frac{\tilde{\rho}_i}{2}\|z^k - x\|^2 + D_{\phi_i}(x, x_i^k), \tag{9}$$

by letting:

$$\phi_i(x) = \begin{cases} 0, & i \in \mathcal{X}\cup\mathcal{D} \\ \frac{L_i}{2}\|x\|^2 - g_i(x), & i \in \mathcal{P} \end{cases}, \qquad \tilde{\rho}_i = \begin{cases} \rho_i, & i \in \mathcal{P}\cup\mathcal{X} \\ 0, & i \in \mathcal{D}. \end{cases}$$

Note that, technically, the update of the primal agents is performed by the coordinator. The primal agents are queried with the tentative plan x_i^k and return $\nabla g_i(x_i^k)$, which is then used by the coordinator to yield the updated primal agent plan through (4). Additionally, the consensus update should have a term comprised of a multiple of the sum of the prices, but summing up the price updates, (8) we obtain the optimality condition of the consensus update problem, meaning that the sum of the prices is null after the first iteration (see e.g. [BPC+11]). Making the additional assumption that the sum of the initial prices is also null further simplifies the notation. The algorithm is summarized in Algorithm 1, which we name 3-Agent Consensus Planning (3ACP).

3.3 Comments

We can glean some insight into the generic CPP algorithm by considering its behavior when all agents are of the same type.

Algorithm 1 Vanilla 3-Agent CP (3ACP)

Let $\rho_i > 0,\ \forall i \in \mathcal{M}$, and λ_i^0 such that $\sum_{i\in\mathcal{M}} \lambda_i^0 = 0$.
while convergence criterion not met **do**
  Update the primal, dual, and proximal agents' plans x_i^{k+1} using (4), (5), and (6), respectively.
  Update the consensus plan z^{k+1} using (7).
  Update the agents' prices λ_i^{k+1} using (8).
end while

Primal agents only. Consider the case when all the agents are primal agents, and assume for simplicity that they use common values $L = L_i, \forall i$ and $\rho = \rho_i, \forall i$. The updates take the form of Bregman ADMM [WB14]:

$$x_i^{k+1} = \frac{Lx_i^k + \rho z^k}{L+\rho} - \frac{1}{L+\rho}\left(\nabla g_i(x_i^k) - \lambda_i^k\right),$$

yielding consensus updates of the form:

$$\begin{aligned} z^{k+1} &= \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} x_i^{k+1} \\ &= z^k - \frac{1}{L+\rho}\sum_{i\in\mathcal{P}} \nabla g_i(x_i^k), \end{aligned}$$

by using the fact that the sum of the λ_i is null.

These updates bear strong similarities with a gradient descent type of update. In fact, letting:

$$\begin{aligned} \mathbf{x}^k &= \left[x_1^k, x_2^k, \ldots, x_M^k\right]^T, \\ \nabla\mathbf{g}(x^k) &= \left[\nabla g_1(x_1^k), \nabla g_2(x_2^k), \ldots, \nabla g_M(x_M^k)\right]^T, \\ \boldsymbol{\lambda}^k &= \left[\lambda_1^k, \lambda_2^k, \ldots, \lambda_M^k\right]^T \end{aligned}$$

and

$$W = \frac{1}{L+\rho}\begin{bmatrix} L+\frac{\rho}{M} & \frac{\rho}{M} & \cdots & \frac{\rho}{M} \\ \frac{\rho}{M} & L+\frac{\rho}{M} & \cdots & \frac{\rho}{M} \\ \vdots & & \ddots & \vdots \\ \frac{\rho}{M} & \frac{\rho}{M} & \cdots & L+\frac{\rho}{M} \end{bmatrix},$$

the updates in the case of all primal agents can be rewritten as:

$$\mathbf{x}^{k+1} = W\mathbf{x}^k - \alpha\left(\nabla\mathbf{g}(x^k) - \boldsymbol{\lambda}^k\right),$$

with $\alpha = \frac{1}{L+\rho}$.

Given that the matrix W is symmetric and doubly stochastic, these updates are almost those of the *decentralized gradient descent* [NO09], except for the presence of dual variables λ^k in the update. Decentralized gradient descent requires that the step sizes be decreasing in order to converge to the optimal, otherwise it only converges to within a neighborhood of the optimal [NO09, YLY16]. When the functions are strongly convex, the consensus plan converges linearly to the optimal until it reaches said neighborhood [YLY16]. By working on both primal and dual variables through the coordinator, the linearized (Bregman) ADMM overcomes the need for decreasing step sizes and converges linearly to the optimal solution.

Dual agents only. When all agents are dual and use common values $L = L_i, \forall i$, and $\rho = \rho_i, \forall i$, the algorithm yields updates of the form:

$$x_i^{k+1} = \arg\min_{x_i} g(x_i) - {\lambda_i^k}^T x_i,$$

$$z^{k+1} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} x_i^{k+1},$$
$$\lambda_i^{k+1} = \lambda_i^k + \rho(z^{k+1} - x_i^{k+1}),$$

which are simply the updates resulting from a dual ascent procedure.

Proximal agents only. When all the agents are proximal, the algorithm reduces to the traditional consensus ADMM [BPC+11].

4 Convergence

4.1 Overview of Results

We prove in this section different forms of convergence of Algorithm 1 (3ACP). We saw in Section 3.3 that when the agents are exclusively primal, dual, or proximal, the algorithm reduces to Bregman ADMM, dual ascent, and regular ADMM, respectively. Individually, each of these algorithms is known to converge sublinearly when the functions are merely assumed to be generally convex, and strongly convex in the case of dual ascent. Under stricter conditions, namely strong-convexity and Lipschitz continuous gradients, they reach linear convergence rates (see [DY16, HL17] for ADMM and [WB14, LLF22] for Bregman ADMM).

Our main convergence result in Section 4.4 below shows that when all three types of agents are combined, and with similar general convexity assumptions, strong convexity for the dual agents, and Lipschitz-continuous gradients for primal agents, we preserve the sublinear convergence rate. Furthermore, the additional strong convexity and Lipschitz continuity of the gradients for all agents allow us to establish two-step linear convergence, which we prove in Section 4.5.

We divide the convergence proofs in three parts: we first establish the plain convergence of the theorem in Section 4.3; we prove its sublinear convergence rate in Section 4.4; and we consider linear convergence in Section 4.5.

4.2 Preliminaries and Assumptions

We detail here the main assumptions and some additional notation used in the proofs.

Assumption 4.1. *Let $\mathcal{M} = \mathcal{P} \cup \mathcal{D} \cup \mathcal{X}$ be a set of primal ($\mathcal{P}$), dual ($\mathcal{D}$), and proximal agents ($\mathcal{X}$), with objective functions g_i, $i \in \mathcal{M}$. For all $i \in \mathcal{M}$, the functions g_i are convex, and in particular μ_i-strongly convex for $i \in \mathcal{D}$, and with β_i-Lipschitz continuous gradients for $i \in \mathcal{P}$.*

These assumptions are the weakest we can require, and the ones we will use to prove the $O\left(\frac{1}{k}\right)$ convergence of the algorithm. The strong convexity of the dual agents is necessary for the convergence of dual ascent, which is a special case of 3ACP, while the Lipschitz-continuity of the primal agents is necessary in order to bound them above by a quadratic function.

The second assumption has to do with the existence of a solution to the problem in the form of a saddle point for its (regular) Lagrangian, defined as:

$$\mathcal{L}(\mathbf{x}, z, \boldsymbol{\lambda}) := \sum_{i \in \mathcal{M}} g_i(x_i) + \lambda_i^T (z - x_i).$$

Assumption 4.2. *The (regular) Lagrangian associated with the consensus problem (1) has a saddle point $(\mathbf{x}^*, z^*, \boldsymbol{\lambda}^*)$:*

$$\mathcal{L}(\mathbf{x}^*, z^*, \boldsymbol{\lambda}) \leq \mathcal{L}(\mathbf{x}^*, z^*, \boldsymbol{\lambda}^*) \leq \mathcal{L}(\mathbf{x}, z, \boldsymbol{\lambda}^*), \qquad \forall\ \mathbf{x}, z, \boldsymbol{\lambda}. \tag{10}$$

The final assumption concerns the learning rate of the dual agents, which we require to be less than the strong convexity parameter μ_i, and is the usual assumption for dual ascent.

Assumption 4.3. *For $i \in \mathcal{D}$, the learning rate ρ_i used in Algorithm 1 is less than the strong-convexity bound of g_i: $\rho_i \leq \mu_i$.*

To prove linear convergence results, as well as to tighten the convergence bounds even in the sublinear case, we will assume that all agents have strongly convex objective functions with Lipschitz continuous gradients.

Assumption 4.4. *Let $\mathcal{M}$ be a set of primal, dual, and proximal agents, with objective functions g_i, $i \in \mathcal{M}$. For all $i \in \mathcal{M}$, the functions g_i are μ_i-strongly convex with β_i-Lipschitz continuous gradients.*

We additionally define some terms and functions that will be useful in the proofs:

$$V^k := \sum_{i\in\mathcal{M}} \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^*\|^2 + \sum_{i\in\mathcal{P}\cup\mathcal{X}} \frac{\rho_i}{2}\|z^k - z^*\|^2 + \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^*, x_i^k) + \sum_{i\in\mathcal{D}} D_{-g_i^*}(\lambda_i^k, \lambda_i^*),$$
$$r^k := \sum_{i\in\mathcal{D}} \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^{k-1}\|^2 - \sum_{i\in\mathcal{D}} D_{-g_i^*}(\lambda_i^k, \lambda_i^{k-1}) + \sum_{i\in\mathcal{P}\cup\mathcal{X}} \frac{\rho_i}{2}\|z^{k-1} - x_i^k\|^2 + \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^k, x_i^{k-1}),$$

where g_i^* stands for the convex conjugate of g_i (see Appendix C). The function V^k measures in some way the distance of the variables, both primal and dual, to their optimal values. Enforcing Assumption 4.3 makes r^k non-negative, and it is similar to the notion of residual in ADMM-related proofs, since r^k being null would make the optimality conditions satisfied.

Remark 4.5.

- *When $\mathcal{D} = \emptyset$, r^k can also be expressed as:*

$$r^k = \sum_{i\in\mathcal{M}} \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^{k-1}\|^2 + \sum_{i\in\mathcal{M}} \frac{\rho_i}{2}\|z^k - z^{k-1}\|^2 + \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^k, x_i^{k-1}),$$

  owing to the fact that $\sum_{i\in\mathcal{M}}\|z^{k-1} - x^k\|^2 = \sum_{i\in\mathcal{M}}\|z^{k-1} - z^k + z^k - x^k\|^2 = \sum_{i\in\mathcal{M}}\|z^k - z^{k-1}\|^2 + \frac{1}{\rho_i^2}\|\lambda^k - \lambda^{k-1}\|^2 + \frac{2}{\rho_i}(z^{k-1} - z^k)^T(\lambda^k - \lambda^{k-1}) = \sum_{i\in\mathcal{M}}\|z^k - z^{k-1}\|^2 + \frac{1}{\rho_i^2}\|\lambda^k - \lambda^{k-1}\|^2$, where we used $\lambda_i^k - \lambda_i^{k-1} = \rho_i(z^k - x_i^k)$, and $\sum_{i\in\mathcal{M}} \lambda_i^k = 0$.

- *When $\mathcal{M} = \mathcal{D}$, we have:*

$$V^k = \sum_{i\in\mathcal{D}} D_{\psi_i}(\lambda_i^k, \lambda_i^*),$$
$$r^k = \sum_{i\in\mathcal{D}} D_{\psi_i}(\lambda_i^k, \lambda_i^{k-1}),$$

  where $\psi_i(\lambda) := -g_i^*(\lambda) + \frac{1}{2\rho_i}\|\lambda\|^2$.

4.3 Plain Convergence

We consider in this section the plain convergence of the algorithm. The first step is to leverage the existence of a saddle point $(\mathbf{x}^*, z^*, \boldsymbol{\lambda}^*)$ and the optimality conditions of the agents' subproblems to establish inequalities that will serve as the basis for the various convergence proofs.

Proposition 4.6. *Consider the consensus problem (1) and suppose Assumptions 4.1 and 4.2 hold. Then, the iterates generated by Algorithm 1 satisfy the following inequality for all $k \geq 0$:*

$$0 \leq \sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - g_i(x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}) \leq V^k - V^{k+1} - r^{k+1}, \tag{11}$$

leading to:

$$\sum_{i\in\mathcal{M}} \frac{\mu_i}{2}\|x_i^{k+1} - x_i^*\|^2 \leq V^k - V^{k+1} - r^{k+1}, \text{ and } \sum_{i\in\mathcal{D}} \frac{\mu_i}{2L_i^2}\|\lambda_i^k - \lambda_i^*\|^2 \leq V^k - V^{k+1} - r^{k+1}. \tag{12}$$

Proof. See Appendix A.1. □

The following proposition justifies the need for Assumption 4.3, as it corresponds to the condition under which r^{k+1} is non-negative.

Proposition 4.7. *Under Assumption 4.3, we have:*

$$r^{k+1} \geq 0, \ \forall k \geq 0.$$

Proof. See Appendix A.2 □

Remark 4.8. *An important observation in Proposition 4.6 is that under Assumption 4.3 we have in particular $V^{k+1} \leq V^k$, and thus that the (non-negative) sequence $\{V^k\}$ is non-increasing and thus bounded, and can serve as a* Lyapunov *function.*

The next theorem concerns the convergence of the algorithm, its asymptotic primal feasibility, as well as convergence of the dual variables for the dual agents. When the set of dual agents is not null, Algorithm 1 also implies convergence of the primal and dual variables of the dual agents, as well as convergence of the consensus variable.

Theorem 4.9 (Convergence of Algorithm 1)**.** *Consider the consensus problem (1) and suppose Assumptions 4.1, 4.2 , and 4.3 hold. Then, for the iterates generated by Algorithm 1 we have:*

$$\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) \to 0,$$
$$z^{k+1} - x_i^{k+1} \to 0,\ \forall i \in \mathcal{M}.$$

Additionally, if $\mathcal{D} \neq \emptyset$, we have:

$$\lambda_i^k \to \lambda_i^*,\ \forall i \in \mathcal{D},$$
$$x_i^k \to x_i^*,\ \forall i \in \mathcal{M},$$
$$z^k \to z^*.$$

Proof. See Appendix A.3. □

4.4 Sublinear Convergence

We prove in this section the $O\left(\frac{1}{k}\right)$ ergodic convergence rate of the vanilla 3ACP algorithm (1) under Assumptions 4.1, 4.2, and 4.3. By ergodic, we mean the convergence of the running average of the iterates. While Theorem 4.9 showed the plain convergence of Algorithm 1, we show here that at any iteration K, both the distance between the value of the the running average up to K, as well as its violation of the feasibility constraint decreases at a rate of $1/K$.

Theorem 4.10 (Ergodic Sublinear Convergence)**.** *Consider the consensus problem (1) and let Assumptions 4.1, 4.2, and 4.3 be satisfied. Let:*

$$C := 2V^0, \qquad \bar{\rho} := \max_i\{\rho_i\}, \qquad \underline{\rho} := \min_i\{\rho_i\},$$

and define the following running average iterates:

$$\hat{\mathbf{x}}^{K+1} = \frac{1}{(K+1)}\sum_{k=0}^{K}\mathbf{x}^{k+1}, \qquad \hat{z}^{K+1} = \frac{1}{(K+1)}\sum_{k=0}^{K} z^{k+1}.$$

Then, after K iterations of Algorithm 1, we have:

$$\left|\sum_{i\in\mathcal{M}} g_i(\hat{x}_i^{K+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*)\right| \leq \frac{C}{2(K+1)} + \frac{2\sqrt{\bar{\rho}C}\|\boldsymbol{\lambda}^*\|}{\underline{\rho}(K+1)},$$
$$\left\|A\hat{z}^{k+1} - \hat{\mathbf{x}}^{k+1}\right\| \leq \frac{2\sqrt{\bar{\rho}C}}{\underline{\rho}(K+1)}.$$

Proof. See Appendix A.4. □

4.5 Linear Convergence

We prove in this section the two-step linear convergence of the 3ACP Algorithm under the stronger assumptions that all agents' functions be strongly convex with Lipschitz continuous gradients (Assumption 4.4) rather than the milder assumptions that only the dual agents be strongly convex, and only the primal agents have Lipschitz continuous gradients. We first define a few terms to simplify the notation, letting $\mathcal{S} \subset \mathcal{M}$ be any subset of agents:

$$\underline{\rho}_{\mathcal{S}} := \min_{i\in\mathcal{S}}\{\rho_i\}, \qquad \alpha_{\mathcal{S}} := \max_{i\in\mathcal{S}}\{\beta_i + L_{\phi_i} + \rho_i\}, \qquad L_{\phi_i} := \beta_i - \mu_i,$$
$$\bar{\rho}_{\mathcal{S}} := \max_{i\in\mathcal{S}}\{\rho_i\}, \qquad \underline{\mu}_{\mathcal{S}} := \min_{i\in\mathcal{S}}\{\mu_i\}, \qquad \overline{L_\phi} := \max_{i\in\mathcal{P}}\{L_{\phi_i}\}.$$

Theorem 4.11 (Linear Convergence of 3ACP)**.** *Consider the consensus problem (1) and let Assumptions 4.4, 4.2, and 4.3 be satisfied. Then, the iterates generated by Algorithm 1 satisfy the following two-step linear convergence rate:*

$$V^{k+2} \leq \left(1 + \frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}}, \frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}, \frac{\underline{\mu}_{\mathcal{P}\cup\mathcal{X}}}{\bar{\rho}_{\mathcal{P}\cup\mathcal{X}}}, \frac{\underline{\mu}_{\mathcal{P}}}{\overline{L_\phi}}\right\}\right)^{-1} V^k.$$

Proof. See Appendix A.5. □

Note that if all the agents are of the same type, we recover known results about the (one-step) linear convergence of the algorithm, corresponding to the linear convergence of ADMM in the case of all proximal agents (see [LLF22, Thm 3.4]), of Bregman ADMM in the case of all primal agents (see [LLF22, Thm 3.8]), and of dual ascent in the case of all dual agents. These results are summarized below.

Corollary 4.12 (Linear Convergence of 3ACP in the case of a single interface)**.** *Consider the consensus problem (1) and let Assumptions 4.4, 4.2, and 4.3 be satisfied. Then, the iterates generated by Algorithm 1 satisfy the following linear convergence rates when the agents are all of the same type:*

All Primal Agents:

$$V^{k+1} \leq \left(1 + \frac{1}{3}\min\left\{\frac{\underline{\rho}_{\mathcal{P}}}{\alpha_{\mathcal{P}}}, \frac{\underline{\mu}_{\mathcal{P}}}{\bar{\rho}_{\mathcal{P}}}, \frac{\underline{\mu}_{\mathcal{P}}}{\overline{L_\phi}}\right\}\right)^{-1} V^k.$$

All Dual Agents:

$$V^{k+1} \leq \left(1 + \frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}, \frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}\right\}\right)^{-1} V^k.$$

All Proximal Agents:

$$V^{k+1} \leq \left(1 + \frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{X}}}{\alpha_{\mathcal{X}}}, , \frac{\underline{\mu}_{\mathcal{X}}}{\bar{\rho}_{\mathcal{X}}}, \right\}\right)^{-1} V^k.$$

Proof. See Appendix A.6. □

5 Practical Considerations

Algorithm 1 presented the most basic version of the algorithm, and it can be improved upon in a number of different ways, either by relaxing some assumptions, or by incorporating algorithmic improvements. We consider in this section a few such extensions and practical considerations, some of which could represent future research directions.

5.1 Acceleration

The algorithms considered so far are essentially first-order methods, only making use of gradient information in some form or another. Such algorithms can often be accelerated through schemes that use previous iterates in the updates. One important such acceleration scheme is Nesterov's accelerated gradient descent [Nes83], which has been successfully ported to many first order algorithms, including ADMM. This is especially interesting in our context since the primal and dual agents are handled as approximations of proximal agents, and are thus likely to lose some performance with respect to the latter.

Given that ADMM works both on the primal (individual plans and consensus) and dual (prices) variables, acceleration techniques may target either of those sets of variables, or both. For example, [GOSB14] presents an accelerated ADMM where the second set of primal variable and the prices are accelerated, while [KCSB15] only modifies the dual variable but requires an additional update of the primal variables. [OHTG13] considers accelerating the linearized ADMM algorithm used for the primal agents and is thus particularly relevant to us. Additionally, accelerated methods are known to not be monotone in the objective value, and they usually display rippling effects. Adaptive restart strategies to alleviate these issue were proposed in [OC15], and they can be adapted to the ADMM acceleration schemes, as was the case in [GOSB14].

One difficulty is combining these different schemes. We can readily apply the accelerated scheme of [GOSB14] when the agents are dual and/or proximal, while we can also directly apply the algorithm of [OHTG13] when the agents are all primal, and some results resulting from those implementations are presented in Section 6. Combining these in order to allow for the acceleration of the algorithm for any combination of agents will be an interesting research topic.

5.2 Tighter Quadratic Bounds for Primal Agents

Our approach to primal agents involved the use of a quadratic approximation of the objective functions g_i of the form $x \mapsto g_i(x_i^k) + \nabla g_i(x_i^k)^T(x - x_i^k) + \frac{L_i}{2}\|x\|^2$. This nonetheless restricts the type of linearization to spherical quadratic functions. We can tighten the bounds by considering not just spherical quadratic functions but more general quadratic functions so long as they dominate g_i. Letting H_i be a symmetric positive definite matrix such that $H_i \succeq \nabla^2 g_i(x)$, $\forall x$, we may then use the following function ϕ_i in the definition of the Bregman divergence D_{ϕ_i}: $\phi_i(x) = \frac{1}{2}x^T H_i x - g_i(x) = \frac{1}{2}\|x\|_{H_i}^2 - g_i(x)$, instead of $\frac{L_i}{2}x^T x - g_i(x)$.

5.3 Second Order Information

Related to Section 5.2 above, we might actually have access to second order information. This is especially useful for the primal and dual agents, since in the former case we could make use of a (close to) second-order linearization, while in the latter case, the price update could result from a Newton (or quasi-Newton) update as opposed to a simple first-order dual ascent.

For example, for primal agents, we could consider Bregman divergence D_{ϕ_i} letting $\phi_i(x) = \frac{1}{2}x^T(\nabla^2 g_i(x) + (\beta_i - \|\nabla^2 g_i(x)\|)I)x - g_i(x)$ or $\phi_i(x) = \frac{1}{2}x^T(\nabla^2 g_i(x) + (\beta_i - \mu_i)I)x - g_i(x)$.

For dual agents, we could consider a price update of the form $\lambda_i^{k+1} = \lambda_i^k + \rho_i \nabla^2 g_i(x_i^{k+1})(z^{k+1} - x_i^{k+1})$, or alternatively $\lambda_i^{k+1} = \lambda_i^k + \rho_i(\nabla^2 g_i(x_i^{k+1}) + \epsilon I)(z^{k+1} - x_i^{k+1})$, for some $\epsilon > 0$, where ρ_i and ϵ are appropriately chosen to guarantee convergence. In that case, however, the consensus update would need to be modified to:

$$z^{k+1} = \left(\sum_{i\in\mathcal{M}} H_i^{k+1}\right)^{-1}\left(\sum_{i\in\mathcal{M}} H_i^{k+1}x_i^{k+1}\right),$$

where:

$$H_i^{k+1} := \begin{cases} \rho_i I, & \text{for } i \in \mathcal{P}\cup\mathcal{X}, \\ \rho_i \nabla^2 g_i(x_i^{k+1}), & \text{for } i \in \mathcal{D} \end{cases}.$$

5.4 Choice of Hyperparameters

One critical aspect in the implementation of Algorithm 1, similarly to ADMM algorithms, is the choice of the hyperparameters $\rho_i, i \in \mathcal{M}$. In ADMM, it is traditional to use a single parameter across agents, although there is a benefit to allowing for individual values, especially because different agents might have different scales of variables for which a single regularizing parameter might not be appropriate.

Convergence bounds and in particular bounds characterizing the linear convergence of ADMM are often optimized for choices of ρ_i given by the geometric mean of the strong convexity and gradient Lipschitz continuity: $\rho_i = \sqrt{\mu_i \beta_i}$, although this requires both that all the functions be strongly convex with Lipschitz continuous gradients, and that we have (approximate) knowledge thereof. On the other hand, dual ascent, which is essentially what the dual agents are performing, is optimized for $\rho_i = \mu_i$. As a result, even if we aim at using as few parameters as possible, it might be beneficial to set different values for dual agents than for primal/proximal ones.

In the absence of knowledge of μ_i and/or β_i, we can conceive of evaluating them "on the fly" based on accumulated information, or alternatively, dynamically adjusting them, as suggested by [HYW00]. In particular, the self-adaptive parameter tuning can be applied at an agent level by considering their respective primal and and dual residuals.

We also note that the consensus update (7) is performed as a weighted sum of the agents' preferred plans, where the weights are precisely given by the learning rates ρ_i. Differences in the learning rates across agents imply differences in the relative importance of each agent's plan in computing the consensus plan. The interaction between progress made by the individual agents' plans and their weight in the consensus update is not obvious.

5.5 Regularized Dual Updates

One scheme we considered, but discarded for now as it didn't seem to have much effect on numerical examples, was to update the dual agents through a two step procedure, as follows:

$$\tilde{x}_i^{k+1} = \arg\min_x g_i(x) - {\lambda_i^k}^T x,$$
$$x_i^{k+1} = \frac{L_i \tilde{x}_i^{k+1} + \rho_i z^k}{L_i + \rho_i},$$

for some $L_i > 0$, although we do not require that $L_i \geq \beta_i$ unlike in the primal case. x_i^{k+1} is thus written as a weighted average of $\tilde{x}_i^{k+1}$ (which would have been the regular update in dual ascent) and z_k. The plan is thus regularized towards the consensus. This nonetheless put restrictions on the admissible values of (ρ_i, L_i). This procedure can be motivated through several observations:

- The optimality condition for the problem yielding $\tilde{x}_i^{k+1}$ is $\nabla g_i(\tilde{x}_i^{k+1}) = \lambda_i^k$. As a result, x_i^{k+1} can be expressed as:

$$x_i^{k+1} = \arg\min_x g_i(\tilde{x}_i^{k+1}) + \nabla g_i(\tilde{x}_i^{k+1})^T x + \frac{L_i}{2}\|x - \tilde{x}_i^{k+1}\|^2 - {\lambda_i^k}^T x + \frac{\rho_i}{2}\|z^k - x\|^2.$$

  This is an update of the same form as the linearized ADMM update (3) for the primal agent, where the linearization is here performed at $\tilde{x}_i^{k+1}$, but because we do not require that $L_i \geq \beta_i$, the linearization does not necessarily yield an upper bound of g_i.

- Suppose that all the agents are dual agents and we use ρ as a learning rate for the price update and a common value $L = L_i, \forall i$. Performing dual ascent using this two step approach yields the following steps:

$$\tilde{x}_i^{k+1} = \arg\min_{x_i} g(x_i) - {\lambda_i^k}^T x_i,$$
$$x_i^{k+1} = \frac{L \tilde{x}_i^{k+1} + \rho_i z^k}{L + \rho_i},$$
$$z^{k+1} = \frac{1}{|\mathcal{D}|}\sum_i x_i^{k+1} = \frac{L \tilde{z}^{k+1} + \rho_i z^k}{L + \rho},$$

$$\lambda_i^{k+1} = \lambda_i^k + \rho_i(z^{k+1} - x_i^{k+1}) = \lambda_i^k + \frac{\rho_i L}{\rho_i + L}\left(\tilde{z}^{k+1} - \tilde{x}_i^{k+1}\right).$$

We observe that this would result in updates that are the ones of a dual ascent procedure with rate $\alpha = \frac{\rho L}{\rho+L}$, which also imposes the condition $\rho(L-\mu) \leq \mu L$ since the dual ascent update requires that $\alpha \leq \mu$.

- Suppose the function g_i is quadratic of the form $g_i(x) = \frac{1}{2}Lx^Tx + b^Tx$. Solving the problem as a proximal agent problem yields $x_i^{k+1} = \frac{\lambda_i^k - b + \rho z^k}{L+\rho}$. On the other hand, using the above two step procedure leads to $\tilde{x}_i^{k+1} = \frac{\lambda_i^k - b}{L}$, and then $x_i^{k+1} = \frac{L\tilde{x}_i^{k+1} + \rho z^k}{L+\rho} = \frac{\lambda_i^k - b + \rho z^k}{L+\rho}$, which is the same update as the one for the proximal agent. As a result, in the case of a spherical quadratic function, any dual agent can be turned into a proximal agent through this regularization. Solving the dual agent problem and then regularizing is equivalent to directly solving the proximal agent problem.

6 Example: Mixed Quadratic Agents

To illustrate the basic properties of our generic CPP, we consider in this section a synthetic example involving mixed quadratic agents. The advantage of such a setting, albeit simple, is that quadratic functions yield closed-form solutions that allow one to get more insight into how different agents are handled.

6.1 Agents

Consider agents whose objective functions are given by the following, where Q_i are symmetric definite positive matrices:

$$g_i(x) = \frac{1}{2}x^TQ_ix + b_i^Tx.$$

Primal Agents: When $i \in \mathcal{P}$, the agent receives x_i^k and returns the gradient $\nabla g_i(x_i^k) = Q_ix_i^k + b_i$. Using (4), we obtain an update of the form:

$$x_i^{k+1} = \frac{(L_iI_n - Q_i)x_i^k + \rho_i z^k}{L_i + \rho_i} - \frac{b_i - \lambda_i^k}{L_i + \rho_i}.$$

Dual Agents: When $i \in \mathcal{D}$, the agent receives λ_i^k and returns x_i^k, the solution to the conjugate dual problem of g_i at λ_i^k:

$$x_i^{k+1} = Q_i^{-1}(\lambda_i^k - b_i).$$

Proximal Agents: When $i \in \mathcal{X}$, the agent receives both λ_i^k and z^k and returns the solution to (6):

$$x_i^{k+1} = (Q_i + \rho_iI_n)^{-1}(\rho_iz^k + \lambda_i^k - b_i).$$

We rewrite these updates in a way that allows for a more direct comparison of their respective mechanisms. Let

$$\hat{x}_i^{k+1} := \arg\min_x g_i(x) - {\lambda_i^k}^Tx_i^k = Q_i^{-1}(\lambda_i^k - b_i)$$

be the solution to the dual problem, which in the case of a dual agent is simply the definition of x_i^{k+1}, but which we extend to the other types of agents using a different notation to avoid confusion. We then have:

$$\begin{aligned}
\text{primal update} \quad x_i^{k+1} = \quad & (L_iI_n + \rho_iI_n)^{-1}\rho_iz^k + \quad (L_iI_n + \rho_iI_n)^{-1}L_i\hat{x}_i^{k+1} + \frac{(L_iI_n - Q_i)}{L_i + \rho_i}(x_i^k - \hat{x}_i^{k+1}) \\
\text{dual update} \quad x_i^{k+1} = \quad & \hat{x}_i^{k+1} \\
\text{prox. update} \quad x_i^{k+1} = \quad & (Q_i + \rho_iI_n)^{-1}\rho_iz^k + \quad (Q_i + \rho_iI_n)^{-1}Q_i\hat{x}_i^{k+1}.
\end{aligned}$$

Formulated in this manner, the update equations yield a number of comments:

- All the updates share a common structure and interpretation as resulting from a weighted average of the current consensus plan z^k and the tentative plan that would have resulted from the current price λ_i^k in the absence of regularization, i.e., the solution resulting from the direct dualization of the constraint.

- In the case of a spherical quadratic function of the form $\frac{L_i}{2}x^T x + b_i^T x$, the updates of the primal and proximal agents would be exactly the same and would have the form:

$$x_i^{k+1} = \frac{\rho_i z^k + L_i \hat{x}_i^{k+1}}{L_i + \rho_i}.$$

  This would also be the exact same form of the updates for dual agents, were we to use the regularized dual update suggested in Section 5.5.

- The expressions highlight the fact that $L_i I_n$ is a substitute for the Hessian of the agents' functions. This also emphasizes the potential benefits of having tight quadratic bounds, as suggested in Section 5.2.

- Compared to the proximal update, the primal update directly replaces the Hessian by $L_i I_n$. As a result, the weighing between the current consensus z^k and $\hat{x}_i^{k+1}$ is the same in all directions, while in the proximal update, the effect of the Hessian Q_i is to weigh different components differently towards one or the other vector.

- The dual update doesn't just take a weighted average of z^k and $\hat{x}_i^{k+1}$, it also contains an additional term that drags it towards the previous plan x_i^k. The larger the gap between the upper quadratic bound and the function, the larger this effect.

- The interpretation of the update as a weighted average also provides some guidance as to how to set ρ. It is reasonable that we would want the weights to be of the same order of magnitude; and, as a result, we should set ρ to have the same order of magnitude as the eigenvalues of the Q_i, for example, as the geometric mean of the smallest and largest eigenvalues $\sqrt{\mu\beta}$.

6.2 Data

We generate 30 random quadratic functions by generating symmetric definite matrices Q_i as $Q_i = \alpha I_n + A_i^T A_i$, where $\alpha > 0$ guarantees that the matrix is definite, and $A_i = r_{1i}(2U_i - 1)$, where $r_{1i} > 0$ and U_i is a $n \times n$ matrix of random uniform numbers over [0,1]. We use $\alpha = 1$ and $r_1 = \tilde{U}_i$, which yields matrices with condition numbers ranging from a little over 1 to 60. We then generate 30 random vectors b_i as $b_i = r_2 u_i$, where u_i is a random uniform vector over [0,1]. We used $r_2 = 1e4$.

The quadratic functions were then assigned to be: 1) all primal; 2) all dual; 3) all proximal; 4) one third primal, one third dual, and one third proximal; 5) one half primal and one half dual; 6) one half primal and one half proximal; 7) one half dual and one half proximal.

6.3 Algorithm

We solved all the configurations above using a version of the algorithm that allows for different regularizing parameters ρ_p, ρ_d, ρ_x, and also applied acceleration with adaptive restart in the configurations that allowed it (all primal, all dual, all proximal, and part dual/proximal).

6.4 Results

The results of the algorithm on the different configurations are presented below for different levels of the regularizing parameters. We tried the following settings:

- $\rho_p = \rho_d = \rho_x = 0.1$ (Figure 1 top left),

- $\rho_p = \rho_d = \rho_x = 1$ (Figure 1 top right),

- $\rho_p = \rho_x = 10$ and $\rho_d = 1$ (Figure 1 bottom left),

- $\rho_p = \rho_x = 50$ and $\rho_d = 1$ (Figure 1 bottom right).

Here, ρ_p is the learning rate used for primal agents, ρ_d the learning rate used for dual agents, and ρ_x the learning rate used for proximal agents.

The plots show the relative error of the objective function $f(z) = \sum_{i \in \mathcal{M}} g_i(z)$ for the consensus plan iterates z_k: $\frac{f(z^k) - f(z^*)}{|f(z^*)|}$.

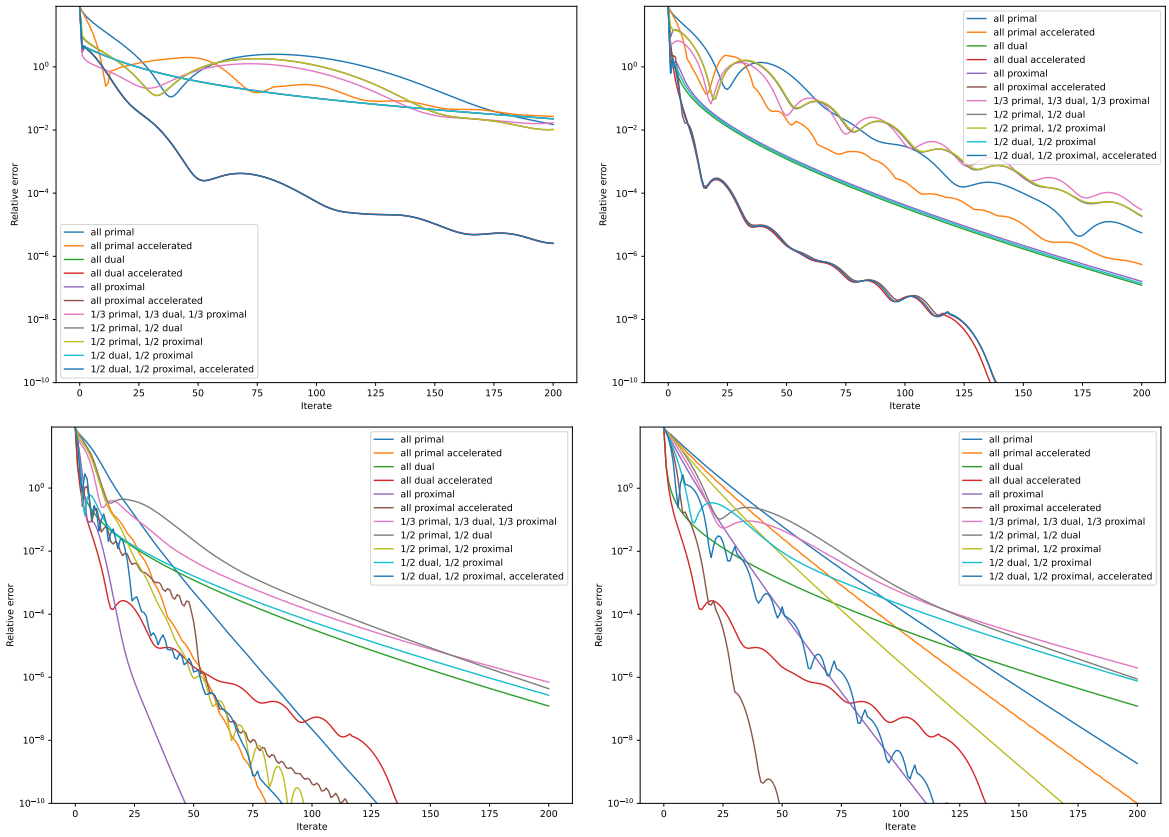

Figure 1: Relative error of the consensus plan iterates for different configurations of the mix of agents and learning rates: $\rho_p = \rho_d = \rho_x = 0.1$ (top left), $\rho_p = \rho_d = \rho_x = 1$ (top right), $\rho_p = \rho_x = 10$, $\rho_d = 1$ (bottom left), $\rho_p = \rho_x = 50$, $\rho_d = 1$ (bottom right).

The results in Figure 1 highlight several points:

- Regardless of the mix of agents (which in many cases is constrained by the application and is not able to be adjusted), the general CPP achieves good convergence. In each case, the convergence speed of the algorithm is impacted by the choice of the learning rate, which should thus be addressed in any practical implementation.
- Overall, there is (if possible) a preference for proximal agents because of their superior theoretical properties and guarantees is reflected in the results.
- The choice between dual and primal agents (when such a choice is possible) is not straightforward. Dual agents require knowledge or a good guess of the strong convexity constant, while proximal agents require the use of a Lipschitz constant. Nonetheless, dual agents require a single value to be set (the learning rate), while primal agents call for two (learning rate and upper bound on the Lipschitz constant), which could make the dual interface an easier one to implement.
- Acceleration usually yields improvements, but also requires that some parameters be set.

7 Conclusion

We presented in this paper a general Consensus Planning Protocol that doesn't place any restriction on agent type, and thus allows for any combination of primal, dual, and proximal agents. We proved

the convergence of the algorithm, and its sublinear $O(\frac{1}{k})$ convergence rate, as well as two-step linear convergence under strong convexity and Lipschitz continuous gradients for all functions. Numerical examples illustrate the behavior of the algorithm under different combinations of agent types, and the impact of acceleration. While we touched on some practical considerations, several of them will be explored in future work, in particular as relates to acceleration and asynchronous updates.

References

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[DY16] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66:889–916, 2016.

[EB92] Jonathan Eckstein and Dimitri P Bertsekas. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical programming*, 55:293–318, 1992.

[GOSB14] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[HL17] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

[HLHY02] Bingsheng He, Li-Zhi Liao, Deren Han, and Hai Yang. A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92:103–118, 2002.

[HYW00] Bing-Sheng He, Hai Yang, and SL Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106:337–356, 2000.

[KCSB15] Mojtaba Kadkhodaie, Konstantina Christakopoulou, Maziar Sanjabi, and Arindam Banerjee. Accelerated alternating direction method of multipliers. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 497–506, 2015.

[LLF22] Zhouchen Lin, Huan Li, and Cong Fang. *Alternating Direction Method of Multipliers for Machine Learning*. Springer, 2022.

[LLS11] Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *Advances in neural information processing systems*, 24, 2011.

[LY21] David G Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. International series in operations research & management science. Springer Nature, Cham, Switzerland, 5 edition, November 2021.

[Nes83] Yurii Evgen'evich Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Doklady Akademii Nauk*, volume 269, pages 543–547. Russian Academy of Sciences, 1983.

[NO09] Angelia Nedic and Asuman Ozdaglar. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[OC15] Brendan O'donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15:715–732, 2015.

[OHTG13] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *International conference on machine learning*, pages 80–88. PMLR, 2013.

[RY22] Ernest K Ryu and Wotao Yin. *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators*. Cambridge University Press, 2022.

[Suz13] Taiji Suzuki. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *International Conference on Machine Learning*, pages 392–400. PMLR, 2013.

[WB14] Huahua Wang and Arindam Banerjee. Bregman Alternating Direction Method of Multipliers. *Advances in Neural Information Processing Systems*, 27, 2014.

[YLY16] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[YY13] Junfeng Yang and Xiaoming Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 82(281):301–329, 2013.

[YZ11] Junfeng Yang and Yin Zhang. Alternating direction algorithms for l_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278, 2011.

A Proofs

A.1 Proof of Proposition 4.6

The proof of the inequality makes use of the optimality conditions of both the agents' subproblems, and the optimality condition of the main problem through the saddle point inequality (10). We thus start by stating the subproblems' optimality conditions, which we will subsequently use to transform the saddle point inequality and express it through relations that involve distances of the iterates to their optimal values. We will also use the following definition:

$$\tilde{V}^k := \sum_{i\in\mathcal{M}} \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^*\|^2 + \sum_{i\in\mathcal{P}\cup\mathcal{X}} \frac{\rho_i}{2}\|z^k - z^*\|^2 + \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^*, x_i^k).$$

Subproblems optimality conditions: Algorithm 1 carries out two main types of subproblems, the agent updates, and the consensus update. The agents' subproblems correspond to Equations (4), (5), and (6). Their optimality conditions read:

$$\nabla g_i(x_i^{k+1}) = \lambda_i^k + \rho_i(z^k - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right),\ i\in\mathcal{P} \tag{13}$$

$$\nabla g_i(x_i^{k+1}) = \lambda_i^k + \rho_i(z^k - x_i^{k+1}),\ i\in\mathcal{X}, \tag{14}$$

$$\nabla g_i(x_i^{k+1}) = \lambda_i^k,\ i\in\mathcal{D}. \tag{15}$$

Saddle-point inequality: We now apply the right-hand side of the saddle-point inequality (10) to the iterate $(\mathbf{x}^{k+1}, z^{k+1}, \boldsymbol{\lambda}^{k+1})$, thus yielding:

$$\begin{aligned} 0 &\le \mathcal{L}(\mathbf{x}^{k+1}, z^{k+1}, \boldsymbol{\lambda}^*) - \mathcal{L}(\mathbf{x}^*, z^*, \boldsymbol{\lambda}^*) \\ &= \sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - g_i(x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}), \\ &\le \sum_{i\in\mathcal{M}} \nabla g_i(x_i^{k+1})^T(x_i^{k+1} - x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}), \end{aligned} \tag{16}$$

using the convexity of the functions g_i. We then apply the optimality conditions (13), (14), and (15) to replace $\nabla g_i(x_i^{k+1})$ in the inequality above. For ease of exposition, we consider each term in turn.

- For the primal agents, we have:

$$\sum_{i\in\mathcal{P}} \nabla g_i(x_i^{k+1})^T(x_i^{k+1} - x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1})$$

$$
\begin{aligned}
&= \sum_{i\in\mathcal{P}} \left(\lambda_i^k + \rho_i(z^k - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&\quad + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}) \\
&= \sum_{i\in\mathcal{P}} \left(\lambda_i^k + \rho_i(z^k - z^{k+1}) + \rho_i(z^{k+1} - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&\quad + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}) \\
&= \sum_{i\in\mathcal{P}} \lambda_i^{k+1T}(x_i^{k+1} - z^{k+1}) + \lambda_i^{k+1T}(z^{k+1} - z^*) + \rho_i(z^k - z^{k+1})^T(x_i^{k+1} - x_i^*) \\
&\quad + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&= \sum_{i\in\mathcal{P}} -(\lambda_i^{k+1} - \lambda_i^*)^T(x_i^{k+1} - z^{k+1}) + \rho_i(z^k - z^{k+1})^T(x_i^{k+1} - x_i^*) \\
&\quad - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) + \lambda_i^{k+1T}(z^{k+1} - z^*) \\
&= \sum_{i\in\mathcal{X}} -\frac{1}{\rho_i}(\lambda_i^{k+1} - \lambda_i^*)^T(\lambda_i^{k+1} - \lambda_i^k) + \rho_i(z^k - z^{k+1})^T(x_i^{k+1} - z^{k+1}) \\
&\quad + \rho_i(z^k - z^{k+1})^T(z^{k+1} - z^*) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&\quad + \lambda_i^{k+1T}(z^{k+1} - z^*) \\
&= \sum_{i\in\mathcal{P}} -\frac{1}{\rho_i}(\lambda_i^{k+1} - \lambda_i^*)^T(\lambda_i^{k+1} - \lambda_i^k) - \rho_i(z^{k+1} - z^k)^T(z^{k+1} - z^*) \\
&\quad + (z^{k+1} - z^k)^T(\lambda_i^{k+1} - \lambda_i^k) \\
&\quad - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) + \lambda_i^{k+1T}(z^{k+1} - z^*) \\
&= \sum_{i\in\mathcal{P}} -\frac{1}{2\rho_i}\|\lambda_i^{k+1} - \lambda_i^*\|^2 + \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^*\|^2 - \frac{1}{2\rho_i}\|\lambda_i^{k+1} - \lambda_i^k\|^2 \\
&\quad - \frac{\rho_i}{2}\|z^{k+1} - z^*\|^2 + \frac{\rho_i}{2}\|z^k - z^*\|^2 - \frac{\rho_i}{2}\|z^{k+1} - z^k\|^2 \\
&\quad + \frac{1}{2\rho_i}\|\lambda_i^{k+1} - \lambda_i^k\|^2 + \frac{\rho_i}{2}\|z^{k+1} - z^k\|^2 - \frac{\rho_i}{2}\|z^k - x_i^{k+1}\|^2 \\
&\quad - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) + \lambda_i^{k+1T}(z^{k+1} - z^*) \\
&= \sum_{i\in\mathcal{P}} -\frac{1}{2\rho_i}\|\lambda_i^{k+1} - \lambda_i^*\|^2 + \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^*\|^2 - \frac{\rho_i}{2}\|z^{k+1} - z^*\|^2 + \frac{\rho_i}{2}\|z^k - z^*\|^2 \\
&\quad - \frac{\rho_i}{2}\|z^k - x_i^{k+1}\|^2 \\
&\quad - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) + \lambda_i^{k+1T}(z^{k+1} - z^*)
\end{aligned}
$$

Using Lemma B.2, we find:

$$
\begin{aligned}
&\sum_{i\in\mathcal{P}} \nabla g_i(x_i^{k+1})^T(x_i^{k+1} - x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1}) \\
&\leq \sum_{i\in\mathcal{P}} -\frac{1}{2\rho_i}\|\lambda_i^{k+1} - \lambda_i^*\|^2 + \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^*\|^2 - \frac{\rho_i}{2}\|z^{k+1} - z^*\|^2 + \frac{\rho_i}{2}\|z^k - z^*\|^2 \\
&\quad - \frac{\rho_i}{2}\|z^k - x_i^{k+1}\|^2 \\
&\quad - D_{\phi_i}(x_i^*, x_i^{k+1}) + D_{\phi_i}(x_i^*, x_i^k) - D_{\phi_i}(x_i^{k+1}, x_i^k) \\
&\quad + \lambda_i^{k+1T}(z^{k+1} - z^*).
\end{aligned} \tag{17}
$$

- The sum over proximal agents is carried out identically to the primal agents, only with $\phi_i = 0$, yielding:

$$
\sum_{i\in\mathcal{X}} \nabla g_i(x_i^{k+1})^T(x_i^{k+1} - x_i^*) + \lambda_i^{*T}(z^{k+1} - x_i^{k+1})
$$

$$\leq \sum_{i\in\mathcal{X}} -\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^*\|^2+\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^*\|^2-\frac{\rho_i}{2}\|z^{k+1}-z^*\|^2+\frac{\rho_i}{2}\|z^k-z^*\|^2$$
$$-\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2+{\lambda_i^{k+1}}^T(z^{k+1}-z^*). \tag{18}$$

- For the dual agents, we have:

$$\sum_{i\in\mathcal{D}}\nabla g_i(x_i^{k+1})^T(x_i^{k+1}-x_i^*)+{\lambda_i^*}^T(z^{k+1}-x_i^{k+1})$$
$$=\sum_{i\in\mathcal{D}}{\lambda_i^k}^T(x_i^{k+1}-x_i^*)+{\lambda_i^*}^T(z^{k+1}-x_i^{k+1})$$
$$=\sum_{i\in\mathcal{D}}{\lambda_i^k}^T(x_i^{k+1}-z^{k+1})+\lambda_i^k(z^{k+1}-z^*)+{\lambda_i^*}^T(z^{k+1}-x_i^{k+1})$$
$$=\sum_{i\in\mathcal{D}}-\frac{1}{\rho_i}(\lambda_i^k-\lambda_i^*)^T(\lambda_i^{k+1}-\lambda_i^k)+{\lambda_i^k}^T(z^{k+1}-z^*)$$
$$=\sum_{i\in\mathcal{D}}-\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^*\|^2+\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^*\|^2+\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$
$$+{\lambda_i^k}^T(z^{k+1}-z^*) \tag{19}$$

Combining (17), (18), (19) into (16), we find:

$$0\leq\mathcal{L}(\mathbf{x}^{k+1},z^{k+1},\boldsymbol{\lambda}^*)-\mathcal{L}(\mathbf{x}^*,z^*,\boldsymbol{\lambda}^*)$$
$$=\tilde{V}^k-\tilde{V}^{k+1}$$
$$-\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^{k+1},x_i^k)-\sum_{i\in\mathcal{P}\cup\mathcal{X}}\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2+\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$
$$+\sum_{i\in\mathcal{P}\cup\mathcal{X}}{\lambda_i^{k+1}}^T(z^{k+1}-z^*)+\sum_{i\in\mathcal{D}}{\lambda_i^k}^T(z^{k+1}-z^*).$$

We then recall that $\sum_{i\in\mathcal{M}}\lambda_i^k=0$, so that $\sum_{i\in\mathcal{P}\cup\mathcal{X}}\lambda_i^k=-\sum_{i\in\mathcal{D}}\lambda_i^k$, leading to:

$$0\leq\mathcal{L}(\mathbf{x}^{k+1},z^{k+1},\boldsymbol{\lambda}^*)-\mathcal{L}(\mathbf{x}^*,z^*,\boldsymbol{\lambda}^*)$$
$$\leq\tilde{V}^k-\tilde{V}^{k+1}$$
$$-\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^{k+1},x_i^k)-\sum_{i\in\mathcal{P}\cup\mathcal{X}}\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2+\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$
$$-\sum_{i\in\mathcal{D}}\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(z^{k+1}-z^*)$$
$$=\tilde{V}^k-\tilde{V}^{k+1}$$
$$-\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^{k+1},x_i^k)-\sum_{i\in\mathcal{P}\cup\mathcal{X}}\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2+\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$
$$-\sum_{i\in\mathcal{D}}\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(z^{k+1}-x_i^{k+1})-\sum_{i\in\mathcal{D}}\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(x_i^{k+1}-x_i^*)$$
$$=\tilde{V}^k-\tilde{V}^{k+1}$$
$$-\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^{k+1},x_i^k)-\sum_{i\in\mathcal{P}\cup\mathcal{X}}\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2+\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$
$$-\sum_{i\in\mathcal{D}}\frac{1}{\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2-\sum_{i\in\mathcal{D}}\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(x_i^{k+1}-x_i^*)$$
$$=\tilde{V}^k-\tilde{V}^{k+1}$$
$$-\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^{k+1},x_i^k)-\sum_{i\in\mathcal{P}\cup\mathcal{X}}\frac{\rho_i}{2}\|z^k-x_i^{k+1}\|^2-\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^k\|^2$$

$$-\sum_{i\in\mathcal{D}}\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(x_i^{k+1}-x_i^*). \tag{20}$$

We focus on the last term in the sum, and recall that for $i\in\mathcal{D}$, $x_i^{k+1}=-\nabla g_i^*(\lambda_i^k)$ (see Proposition C.2), whence:

$$\begin{aligned}
&\sum_{i\in\mathcal{D}}-\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(x_i^{k+1}-x_i^*)\\
&=\sum_{i\in\mathcal{D}}-\left(\lambda_i^{k+1}-\lambda_i^k\right)^T(\nabla-g_i^*(\lambda_i^k)-\nabla-g_i^*(\lambda_i^*))\\
&=\sum_{i\in\mathcal{D}}D_{-g_i^*}(\lambda_i^k,\lambda_i^*)-D_{-g_i^*}(\lambda_i^{k+1},\lambda_i^*)+D_{-g_i^*}(\lambda_i^{k+1},\lambda_i^k),
\end{aligned}$$

using Lemma B.2. Substituting this last equality in (20), we get:

$$0\leq\mathcal{L}(\mathbf{x}^{k+1},z^{k+1},\boldsymbol{\lambda}^*)-\mathcal{L}(\mathbf{x}^*,z^*,\boldsymbol{\lambda}^*)\leq V^k-V^{k+1}-r^{k+1},$$

which is the first inequality in Proposition 4.6.

To obtain the second inequality, we recall that $\mathcal{L}(\mathbf{x}^{k+1},z^{k+1},\boldsymbol{\lambda}^*)-\mathcal{L}(\mathbf{x}^*,z^*,\boldsymbol{\lambda}^*)=\sum_{i\in\mathcal{M}}g_i(x_i^{k+1})-g_i(x_i^*)+\lambda_i^*(z^{k+1}-x_i^{k+1})$. Using the convexity of the functions g_i, and in particular their strong convexity for $i\in\mathcal{D}$ (letting $\mu_i=0$ for those functions in $\mathcal{P}\cup\mathcal{X}$ that are not strongly convex), we then have:

$$\begin{aligned}
&\sum_{i\in\mathcal{M}}g_i(x_i^{k+1})-g_i(x_i^*)+\lambda_i^*(z^{k+1}-x_i^{k+1})\\
&=\sum_{i\in\mathcal{M}}g_i(x_i^{k+1})-g_i(x_i^*)-\lambda_i^*(x_i^{k+1}-x_i^*)+\lambda_i^*(z^{k+1}-z^*)\\
&\geq\sum_{i\in\mathcal{M}}\frac{\mu_i}{2}\|x_i^{k+1}-x_i^*\|^2+\sum_{i\in\mathcal{M}}\lambda_i^*(z^{k+1}-z^*)\\
&=\sum_{i\in\mathcal{M}}\frac{\mu_i}{2}\|x_i^{k+1}-x_i^*\|^2,
\end{aligned}$$

using the fact that $\sum_{i\in\mathcal{M}}\lambda_i^k=0$. This yields the second inequality in Proposition 4.6.

The third one is then obtained by recalling that $x_i^{k+1}=-\nabla g_i^*(\lambda_i^k)$ for $i\in\mathcal{D}$, and thus that $\|x_i^{k+1}-x_i^*\|^2=\|\nabla g_i^*(\lambda_i^k)-\nabla g_i^*(\lambda_i^k)\|^2$. Additionally, since the functions $g_i, i\in\mathcal{D}$ have L_i-Lipschitz continuous gradients, their convex conjugates g_i^* are $\frac{1}{L_i}$-strongly convex, and so $\|\nabla g_i^*(\lambda_i^k)-\nabla g_i^*(\lambda_i^k)\|^2\geq\frac{1}{L_i^2}\|\lambda_i^k-\lambda_i^*\|^2$, which gives the desired inequality.

A.2 Proof of Proposition 4.7

To prove that $r^{k+1}\geq 0$, we need only prove that $\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^{k-1}\|^2-\sum_{i\in\mathcal{D}}D_{-g_i^*}(\lambda_i^k,\lambda_i^{k-1})\geq 0$. Since the functions g_i are μ_i-strongly convex, the negative of their conjugates are convex with $\frac{1}{\mu_i}$-Lipschitz continuous gradients. Using Lemma B.3, we have:

$$\begin{aligned}
&\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^{k-1}\|^2-\sum_{i\in\mathcal{D}}D_{-g_i^*}(\lambda_i^k,\lambda_i^{k-1})\\
&\geq\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\left(1-\frac{\rho_i}{\mu_i}\right)\|\lambda_i^k-\lambda_i^{k-1}\|^2\\
&\geq 0,
\end{aligned}$$

since by assumption we have $\frac{\rho_i}{\mu_i}\leq 1$.

A.3 Proof of Theorem 4.9

From Proposition 4.6, we have:

$$r^{k+1} \leq V^k - V^{k+1}.$$

Summing this inequality over k from 0 to ∞, and using the telescoping sum, we get:

$$\sum_{k=0}^{\infty} r^{k+1} \leq V^0.$$

Since the sequence $\{r^k\}$ is non-negative and its series sum is bounded above, it implies that $r^k \to 0$. We also noted that the sequence V^k is bounded, whence the sequences $\|\lambda_i^k - \lambda_i^*\|^2$, $\|z^k - z^*\|$, and $D_{\phi_i}(x_i^*, x_i^k)$ are also bounded.

We then consider the cases where $\mathcal{D} = \emptyset$ and $\mathcal{D} \neq \emptyset$ separately.

$\mathcal{D} = \emptyset$: When $\mathcal{D} = \emptyset$, and as shown in Section 4.2, we have:

$$r^k = \sum_{i \in \mathcal{M}} \frac{1}{2\rho_i} \|\lambda_i^k - \lambda_i^{k-1}\|^2 + \sum_{i \in \mathcal{M}} \frac{\rho_i}{2} \|z^k - z^{k-1}\|^2 + \sum_{i \in \mathcal{P}} D_{\phi_i}(x_i^k, x_i^{k-1}).$$

It follows from the convergence of $\{r^k\}$ to 0 that $\|\lambda_i^k - \lambda_i^{k-1}\|^2$, $\|z^k - z^{k-1}\|^2$, and $D_{\phi_i}(x_i^k, x_i^{k-1})$ also converge to 0 for all $i \in \mathcal{M}$. Then, write $\lambda_i^{k+1} - \lambda_i^k = \rho_i(z^{k+1} - x_i^{k+1}) = \rho_i(z^{k+1} - z^*) - \rho_i(x_i^{k+1} - x_i^*)$. We established that $\lambda_i^{k+1} - \lambda_i^k \to 0$, which implies that $z^{k+1} - x_i^{k+1} \to 0$ as well. We also wrote the bounded (and convergent) sequence $\lambda_i^{k+1} - \lambda_i^k$ as the sum of two sequences, one of which is bounded, implying that the other one, $x_i^k - x_i^*$, is bounded as well.

We now turn to the convergence of the objective function, and show that the difference $\sum_{i \in \mathcal{M}} g_i(x_i^{k+1}) - \sum_{i \in \mathcal{M}} g_i(x_i^*)$ can be bounded below and above by sequences that converge to 0.

- We first use the convexity of the functions to bound the difference above:

$$\sum_{i \in \mathcal{M}} g_i(x_i^{k+1}) - \sum_{i \in \mathcal{M}} g_i(x_i^*) \leq \sum_{i \in \mathcal{M}} \nabla g_i(x_i^{k+1})^T (x_i^{k+1} - x_i^*).$$

The expressions for the gradients $\nabla g_i(x_i^{k+1})$ were given in (13), (13), and (15), yielding:

$$\begin{aligned}
&\sum_{i \in \mathcal{M}} g_i(x_i^{k+1}) - \sum_{i \in \mathcal{M}} g_i(x_i^*) \\
&\leq \sum_{i \in \mathcal{M}} \left(\lambda_i^k + \rho_i(z^k - x_i^{k+1})\right)^T (x_i^{k+1} - x_i^*) \\
&\quad + \sum_{i \in \mathcal{P}} \left(\nabla \phi_i(x_i^{k+1}) - \nabla \phi_i(x_i^k)\right)^T (x_i^{k+1} - x_i^*).
\end{aligned}$$

The first sum can be transformed to:

$$\begin{aligned}
&\sum_{i \in \mathcal{M}} \left(\lambda_i^k + \rho_i(z^k - x_i^{k+1})\right)^T (x_i^{k+1} - x_i^*) \\
&= \sum_{i \in \mathcal{M}} \left(\lambda_i^{k+1} + \rho_i(z^k - z^{k+1})\right)^T (x_i^{k+1} - x_i^*) \\
&= \sum_{i \in \mathcal{M}} {\lambda_i^{k+1}}^T (x_i^{k+1} - x_i^*) + \rho_i(z^k - z^{k+1})^T (x_i^{k+1} - x_i^*) \\
&= \sum_{i \in \mathcal{M}} -\frac{1}{\rho_i} {\lambda_i^{k+1}}^T (\lambda_i^{k+1} - \lambda_i^k) + {\lambda_i^{k+1}}^T (z^{k+1} - z^*) + \rho_i(z^k - z^{k+1})^T (x_i^{k+1} - x_i^*) \\
&= \sum_{i \in \mathcal{M}} -\frac{1}{\rho_i} {\lambda_i^{k+1}}^T (\lambda_i^{k+1} - \lambda_i^k) + \rho_i(z^k - z^{k+1})^T (x_i^{k+1} - x_i^*).
\end{aligned}$$

We thus have:

$$\begin{aligned}
&\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) \\
&\leq \sum_{i\in\mathcal{M}} -\frac{1}{\rho_i}{\lambda_i^{k+1}}^T(\lambda_i^{k+1}-\lambda_i^k) + \rho_i(z^k - z^{k+1})^T(x_i^{k+1}-x_i^*) \\
&\quad + \sum_{i\in\mathcal{P}} \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T(x_i^{k+1}-x_i^*) \\
&\leq \sum_{i\in\mathcal{M}} \frac{1}{\rho_i}\|\lambda_i^{k+1}\|\|\lambda_i^{k+1}-\lambda_i^k\| + \rho_i\|z^k - z^{k+1}\|\|x_i^{k+1}-x_i^*\| \\
&\quad + \sum_{i\in\mathcal{P}} \|\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\|\|x_i^{k+1}-x_i^*\|. \qquad (21)
\end{aligned}$$

We then recall that the sequence $\|\lambda_i^k\|$ is bounded while $\|\lambda_i^{k+1}-\lambda_i^k\| \to 0$; that $\|z^{k+1}-z^k\| \to 0$ while $\|x_i^{k+1}-x_i^*\|$ is bounded; and that since $D_{\phi_i}(x_i^{k+1}, x_i^k) \to 0$, we have $x_i^k - x_i^{k+1} \to 0$, and as a result $\|\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\| \to 0$ by continuity. Put all together, this implies that (21) converges to 0.

- On the other hand, the first inequality from Proposition 4.6 gives:

$$\begin{aligned}
\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) &\geq \sum_{i\in\mathcal{M}} -{\lambda_i^*}^T\left(z^{k+1}-x_i^{k+1}\right) \\
&\geq \sum_{i\in\mathcal{M}} -\|\lambda_i^*\|\|z^{k+1}-x_i^{k+1}\| \\
&\to 0,
\end{aligned}$$

since we showed that $z^{k+1} - x_i^{k+1} \to 0$.

$\mathcal{D} \neq \emptyset$: When $\mathcal{D} \neq \emptyset$, we have:

$$\begin{aligned}
r^k := &\sum_{i\in\mathcal{D}} \frac{1}{2\rho_i}\|\lambda_i^k - \lambda_i^{k-1}\|^2 - \sum_{i\in\mathcal{D}} D_{-g_i^*}(\lambda_i^k, \lambda_i^{k-1}) + \sum_{i\in\mathcal{P}\cup\mathcal{X}} \frac{\rho_i}{2}\|z^{k-1} - x_i^k\|^2 \\
&+ \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^k, x_i^{k-1}).
\end{aligned}$$

It follows from the convergence of $\{r^k\}$ to 0 that $\|\lambda_i^k - \lambda_i^{k-1}\|^2$ converges to 0 for all $i \in \mathcal{D}$, as well as $D_{\phi_i}(x_i^{k+1}, x_i^k)$ for all $i \in \mathcal{P}$, and $z^{k-1} - x_i^k \to 0$ for all $i \in \mathcal{P}\cup\mathcal{X}$. Using Proposition 4.6, and summing the second and third inequalities from 0 to ∞ we also have:

$$\begin{aligned}
\sum_{k=0}^{\infty}\sum_{i\in\mathcal{D}} \frac{\mu_i}{2}\|x_i^{k+1} - x_i^*\|^2 \leq V^0, \\
\sum_{k=0}^{\infty}\sum_{i\in\mathcal{D}} \frac{\mu_i}{2L_i^2}\|\lambda_i^k - \lambda_i^*\|^2 \leq V^0,
\end{aligned}$$

whence $\|x_i^{k+1} - x_i^*\|^2 \to 0$ and $\|\lambda_i^k - \lambda_i^*\|^2 \to 0$ for $i \in \mathcal{D}$, i.e. $x_i^k \to x_i^*$, and $\lambda_i^k \to \lambda_i^*$.

Then, since $\rho_i(z^{k+1} - z^*) = \lambda_i^{k+1} - \lambda_i^k + \rho_i(x_i^{k+1} - x_i^*)$ and both $\lambda_i^{k+1} - \lambda_i^k \to 0$ and $x_i^{k+1} - x_i^* \to 0$ for $i \in \mathcal{D}$, we also have $z^{k+1} - z^* \to 0$. It follows immediately from the convergence of x_i^k and z^k to $x_i^* = z^*$ for $i \in \mathcal{D}$ that we also have $z^{k+1} - x_i^{k+1} \to 0$ for all $i \in \mathcal{D}$. For $i \in \mathcal{P}\cup\mathcal{X}$, since we have both that $z^k \to z^*$ and $z^{k-1} - x_i^k \to 0$, it follows that $x_i \to z^* = x_i^*$.

For the proof of the convergence of the objective function, we proceed just as in the case where $\mathcal{D} = \emptyset$ by bounding the difference $\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*)$ by sequences that converge to 0.

- Using the convexity of the functions g_i, we have:

$$\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) \leq \sum_{i\in\mathcal{M}} \nabla g_i(x_i^{k+1})^T(x_i^{k+1} - x_i^*).$$

Using transformations similar to the ones performed above, we find:

$$\begin{aligned}
&\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) \\
&\leq \sum_{i\in\mathcal{P}\cup\mathcal{X}} \rho_i \left(z^k - z^{k+1}\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&\quad + \sum_{i\in\mathcal{P}} \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)^T \left(x_i^{k+1} - x_i^*\right) \\
&\quad + \sum_{i\in\mathcal{D}} \left(\lambda_i^k - \lambda_i^{k+1}\right)^T \left(x_i^{k+1} - x_i^*\right).
\end{aligned}$$

The same arguments using the boundedness of the sequence $\|x_i^{k+1} - x_i^*\|$, and the convergences to 0 of $\|z^{k+1} - z^k\|$, $\|x_i^{k+1} - x_i^k\|$ for $i \in \mathcal{P}$, and $\|\lambda_i^{k+1} - \lambda_i^k\|$ for $i \in \mathcal{D}$.

- The convergence of the lower bound to 0 is identical to the case where $\mathcal{D} = \emptyset$.

A.4 Proof of Theorem 4.10

Consider the first inequality in Proposition 4.6, and average it over $k = 0, \ldots, K$:

$$\frac{1}{K+1}\sum_{k=0}^{K}\left\{\sum_{i\in\mathcal{M}} g_i(x_i^{k+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*)\right\} + \boldsymbol{\lambda}^{*T}(A\hat{z}^{K+1} - \hat{\mathbf{x}}^{K+1}) \leq \frac{C}{2(K+1)}.$$

Using the convexity of the functions g_i, this yields:

$$\sum_{i\in\mathcal{M}} g_i(\hat{x}_i^{K+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*) + \boldsymbol{\lambda}^{*T}(A\hat{z}^{K+1} - \hat{\mathbf{x}}^{K+1}) \leq \frac{C}{2(K+1)},$$

so that

$$\left|\sum_{i\in\mathcal{M}} g_i(\hat{x}_i^{K+1}) - \sum_{i\in\mathcal{M}} g_i(x_i^*)\right| \leq \frac{C}{2(K+1)} + \frac{\|\boldsymbol{\lambda}^*\|}{(K+1)}\left\|\sum_{k=0}^{K}(Az^{k+1} - \mathbf{x}^{k+1})\right\|.$$

Next, letting P be a diagonal matrix of appropriate size with elements equal to the ρ_i:

$$\begin{aligned}
\left\|\sum_{k=0}^{K}(Az^{k+1} - \mathbf{x}^{k+1})\right\| &= \left\|\sum_{k=0}^{K} P^{-1}\left(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\right)\right\|, \\
&= \left\|P^{-1}\left(\boldsymbol{\lambda}^{K+1} - \boldsymbol{\lambda}^0\right)\right\|, \\
&\leq \|P^{-1}\|\left(\left\|\boldsymbol{\lambda}^{K+1} - \boldsymbol{\lambda}^*\right\| + \left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|\right).
\end{aligned}$$

Additionally, as noted in Remark 4.8, Proposition C.2 also implies that $V^k \leq V^0$ for any k, and in particular that:

$$\frac{1}{2}\left\|\sqrt{P}^{-1}(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*)\right\|^2 \leq \frac{C}{2},$$

whence it follows that:

$$\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\right\| \leq \sqrt{\bar{\rho}C},$$

and thus that:

$$\left\|A\hat{z}^{k+1} - \hat{\mathbf{x}}^{k+1}\right\| \leq \frac{2\sqrt{\bar{\rho}C}}{\underline{\rho}(K+1)},$$

which concludes the proof.

A.5 Proof of Theorem 4.11

To prove the result, we will bound each term in V^k by a term involving $V^k - V^{k+1}$. In order to achieve that, we first derive the inequalities resulting from making use of the strong-convexity and Lipschitz-continuity of the gradients of all the functions.

Strong convexity bound: The bound resulting from the application of the functions' μ_i-strong convexity was already established in Equation (12) of Proposition 4.6 as:

$$\sum_{i\in\mathcal{M}} \frac{\mu_i}{2}\|x_i^{k+1} - x_i^*\|^2 \leq V^k - V^{k+1} - r^{k+1},$$

and in particular:

$$\sum_{i\in\mathcal{M}} \frac{\mu_i}{2}\|x_i^{k+1} - x_i^*\|^2 \leq V^k - V^{k+1}, \tag{22}$$

Gradient Lipschitz-continuity bound: The application of gradient Lipschitz continuity to Equation (11) would have added a term $-\frac{1}{2L_i}\|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2$ to the right of Equation (16) in the proof of Proposition 4.6, and yielded:

$$\sum_{i\in\mathcal{M}} \frac{1}{2L_i}\|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2 \leq V^k - V^{k+1} - r^{k+1}. \tag{23}$$

We can carry out a unified treatment of the terms $\frac{1}{2L_i}\|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2$ by recalling that the optimality conditions (13), (15), and (14) can all be written as:

$$\nabla g_i(x_i^{k+1}) = \lambda_i^k + \tilde{\rho}_i(z^k - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right),$$

with:

$$\tilde{\rho}_i = \begin{cases} \rho_i & i \in \mathcal{P}\cup\mathcal{X}, \\ 0 & i \in \mathcal{D} \end{cases}, \qquad\qquad \phi_i(x) = 0,\ i \in \mathcal{D}\cup\mathcal{X}.$$

We then have:

$$\begin{aligned}
&\frac{1}{2L_i}\|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2 \\
&= \frac{1}{2L_i}\left\|\lambda_i^k - \lambda_i^* + \tilde{\rho}_i(z^k - x_i^{k+1}) - \left(\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right)\right\|^2 \\
&\geq \frac{(1-\gamma_i)}{2L_i}\left\|\lambda_i^k - \lambda_i^* + \tilde{\rho}_i(z^k - x_i^{k+1})\right\|^2 - \frac{(\frac{1}{\gamma_i}-1)}{2L_i}\left\|\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right\|^2 \\
&= \frac{1}{2(L_i+L_{\phi_i})}\left\|\lambda_i^k - \lambda_i^* + \tilde{\rho}_i(z^k - x_i^{k+1})\right\|^2 - \frac{1}{2L_\phi}\left\|\nabla\phi_i(x_i^{k+1}) - \nabla\phi_i(x_i^k)\right\|^2 \\
&\geq \frac{1}{2(L_i+L_{\phi_i})}\left\|\lambda_i^k - \lambda_i^* + \tilde{\rho}_i(z^k - x_i^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1}, x_i^k).
\end{aligned}$$

We here used the fact that $\|u+v\|^2 \geq (1-\gamma)\|u\|^2 - \left(\frac{1}{\gamma}-1\right)\|v\|^2$ (see Lemma D.1) and set $\gamma_i = \frac{L_{\phi_i}}{L_i+L_{\phi_i}}$. This value was set so as to cancel the Bregman divergence term between x_i^{k+1} and x_i^k with the one present in r^{k+1}. We then proceed similarly to isolate the term involving $\|\lambda_i^k - \lambda_i^*\|^2$:

$$\begin{aligned}
&\frac{1}{2L_i}\|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2 \\
&\geq \frac{1}{2(L_i+L_{\phi_i})}\left\|\lambda_i^k - \lambda_i^* + \tilde{\rho}_i(z^k - x_i^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1}, x_i^k) \\
&\geq \frac{(1-\nu_i)}{2(L_i+L_{\phi_i})}\left\|\lambda_i^k - \lambda_i^*\right\|^2 - \frac{(\frac{1}{\nu_i}-1)}{2(L_i+L_{\phi_i})}\tilde{\rho}_i^2\left\|(z^k - x_i^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1}, x_i^k)
\end{aligned}$$

$$= \frac{1}{2(L_i + L_{\phi_i} + \rho_i)} \left\|\lambda_i^k - \lambda_i^*\right\|^2 - \frac{\tilde{\rho}_i}{2} \left\|(z^k - x_i^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1}, x_i^k),$$

where we again used Lemma D.1 and set $\nu_i = \frac{\tilde{\rho}_i}{L_i + L_{\phi_i} + \rho_i}$ in order to cancel the $\|z^k - x_i^{k+1}\|^2$ with the one in r^{k+1}. Plugging this back into Equation (23) yields:

$$\begin{aligned}
&\sum_{i \in \mathcal{M}} \frac{1}{2(L_i + L_{\phi_i} + \tilde{\rho}_i)} \left\|\lambda_i^k - \lambda_i^*\right\|^2 - \frac{\tilde{\rho}_i}{2} \left\|(z^k - x_i^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1}, x_i^k) \\
&\leq \sum_{i \in \mathcal{M}} \frac{1}{2L_i} \|\nabla g_i(x_i^{k+1}) - \nabla g_i(x_i^*)\|^2 \\
&\leq V^k - V^{k+1} - r^{k+1}
\end{aligned}$$

which implies

$$\sum_{i \in \mathcal{M}} \frac{1}{2(L_i + L_{\phi_i} + \tilde{\rho}_i)} \left\|\lambda_i^k - \lambda_i^*\right\|^2 \leq V^k - V^{k+1}. \tag{24}$$

We now bound each term in V^{k+1}.

- Using (24), we have:

$$\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}} \sum_{i \in \mathcal{M}} \frac{1}{2\rho_i} \|\lambda_i^{k+1} - \lambda_i^*\|^2 \leq \sum_{i \in \mathcal{M}} \frac{1}{2(L_i + L_{\phi_i} + \tilde{\rho}_i)} \left\|\lambda_i^{k+1} - \lambda_i^*\right\|^2 \leq V^{k+1} - V^{k+2}.$$

- Recall that since $-g_i^*$ has $\frac{1}{\mu_i}$-Lipschitz continuous gradients, we have according to Lemma B.3 that $D_{-g_i^*}(\lambda_i^{k+1}, \lambda_i^*) \leq \frac{1}{2\mu_i} \|\lambda_i^{k+1} - \lambda_i^*\|^2$, and so, using (24) once again:

$$\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}} \sum_{i \in \mathcal{D}} D_{-g_i^*}(\lambda_i^{k+1}, \lambda_i^*) \leq V^{k+1} - V^{k+2}.$$

- Using Jensen's inequality and (22), we have:

$$\begin{aligned}
\frac{\underline{\mu}_{\mathcal{P} \cup \mathcal{X}}}{\overline{\rho}_{\mathcal{P} \cup \mathcal{X}}} \sum_{i \in \mathcal{P} \cup \mathcal{X}} \frac{\rho_i}{2} \|z^{k+1} - z^*\|^2 &\leq \frac{\underline{\mu}_{\mathcal{P} \cup \mathcal{X}}}{\overline{\rho}_{\mathcal{P} \cup \mathcal{X}}} \sum_{i \in \mathcal{M}} \frac{\rho_i}{2} \|z^{k+1} - z^*\|^2 \\
&\leq \frac{\underline{\mu}_{\mathcal{P} \cup \mathcal{X}}}{\overline{\rho}_{\mathcal{P} \cup \mathcal{X}}} \sum_{i \in \mathcal{M}} \frac{\rho_i}{2} \|x_i^{k+1} - x_i^*\|^2 \\
&\leq \sum_{i \in \mathcal{M}} \frac{\mu_i}{2} \|x_i^{k+1} - x_i^*\|^2 \\
&\leq V^k - V^{k+1}.
\end{aligned}$$

- Finally, using Lemma B.3, we have $D_{\phi_i}(x_i^*, x_i^{k+1}) \leq \frac{L_{\phi_i}}{2} \|x_i^{k+1} - x_i^*\|^2$, which combined with (22) gives:

$$\begin{aligned}
\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}} \sum_{i \in \mathcal{P}} D_{\phi_i}(x_i^*, x_i^{k+1}) &\leq \frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}} \sum_{i \in \mathcal{P}} \frac{L_{\phi_i}}{2} \|x_i^* - x_i^{k+1}\|^2 \\
&\leq \sum_{i \in \mathcal{P}} \frac{\mu_i}{2} \|x_i^* - x_i^{k+1}\|^2 \\
&\leq \sum_{i \in \mathcal{M}} \frac{\mu_i}{2} \|x_i^* - x_i^{k+1}\|^2 \\
&\leq V^k - V^{k+1}.
\end{aligned}$$

Grouping the four derived inequalities, we have:

$$\begin{aligned}
\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}} \sum_{i\in\mathcal{M}} \frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^*\|^2 &\leq V^{k+1}-V^{k+2},\\
\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}} \sum_{i\in\mathcal{D}} D_{-g_i^*}(\lambda_i^{k+1},\lambda_i^*) &\leq V^{k+1}-V^{k+2}\\
\frac{\underline{\mu}_{\mathcal{P}\cup\mathcal{X}}}{\overline{\rho}_{\mathcal{P}\cup\mathcal{X}}} \sum_{i\in\mathcal{P}\cup\mathcal{X}} \frac{\rho_i}{2}\|z^{k+1}-z^*\|^2 &\leq V^k-V^{k+1}\\
\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}} \sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^*,x_i^{k+1}) &\leq V^k-V^{k+1}.
\end{aligned}$$

Summing these four inequalities yields:

$$\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}},\frac{\underline{\mu}_{\mathcal{P}\cup\mathcal{X}}}{\overline{\rho}_{\mathcal{P}\cup\mathcal{X}}},\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}}\right\}V^{k+1}\leq V^k-V^{k+2}.$$

The monotonicity of the sequence $\{V^k\}$ (see Remark 4.8) further implies that $V^{k+2}\leq V^{k+1}$, and thus that:

$$\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}},\frac{\underline{\mu}_{\mathcal{P}\cup\mathcal{X}}}{\overline{\rho}_{\mathcal{P}\cup\mathcal{X}}},\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}}\right\}V^{k+2}\leq V^k-V^{k+2},$$

whence:

$$V^{k+2}\leq\left(1+\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}},\frac{\underline{\mu}_{\mathcal{P}\cup\mathcal{X}}}{\overline{\rho}_{\mathcal{P}\cup\mathcal{X}}},\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}}\right\}\right)^{-1}V^k.$$

A.6 Proof of Theorem 4.12

All Dual Agents: If all the agents are dual, then $V^k=\sum_{i\in\mathcal{D}}\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^*\|^2+D_{-g_i^*}(\lambda_i^k,\lambda_i^*)$. Summing up only the first two inequalities yields:

$$\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{D}}}{\alpha_{\mathcal{D}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}\right\}V^k\leq V^k-V^{k+1}.$$

The monotonicity of the sequence $\{V^k\}$ (see Remark 4.8) further implies that $V^k\leq V^{k+1}$, and thus that:

$$\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}\right\}V^{k+1}\leq V^k-V^{k+1},$$

whence:

$$V^{k+1}\leq\left(1+\frac{1}{2}\min\left\{\frac{\underline{\rho}_{\mathcal{M}}}{\alpha_{\mathcal{M}}},\frac{\underline{\mu}_{\mathcal{D}}}{\alpha_{\mathcal{D}}}\right\}\right)^{-1}V^k.$$

All Primal Agents: When all the agents are primal (or proximal), we recall from Remark 4.5 that r^k reads:

$$r^k=\sum_{i\in\mathcal{M}}\frac{1}{2\rho_i}\|\lambda_i^k-\lambda_i^{k-1}\|^2+\sum_{i\in\mathcal{M}}\frac{\rho_i}{2}\|z^k-z^{k-1}\|^2+\sum_{i\in\mathcal{P}}D_{\phi_i}(x_i^k,x_i^{k-1}).$$

As a result, we may change the bounds used in the Gradient Lipshitz-continuity bound in proof A.5 as follows:

$$\begin{aligned}
&\frac{1}{2L_i}\|\nabla g_i(x_i^{k+1})-\nabla g_i(x_i^*)\|^2\\
&\geq\frac{1}{2(L_i+L_{\phi_i})}\left\|\lambda_i^{k+1}-\lambda_i^*+\tilde{\rho}_i(z^k-z^{k+1})\right\|^2-D_{\phi_i}(x_i^{k+1},x_i^k)
\end{aligned}$$

$$\geq \frac{(1-\nu_i)}{2(L_i+L_{\phi_i})}\left\|\lambda_i^{k+1}-\lambda_i^*\right\|^2 - \frac{(\frac{1}{\nu_i}-1)}{2(L_i+L_{\phi_i})}\tilde{\rho}_i^2\left\|(z^k-z^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1},x_i^k)$$
$$= \frac{1}{2(L_i+L_{\phi_i}+\rho_i)}\left\|\lambda_i^{k+1}-\lambda_i^*\right\|^2 - \frac{\tilde{\rho}_i}{2}\left\|(z^k-z^{k+1})\right\|^2 - D_{\phi_i}(x_i^{k+1},x_i^k),$$

where we used Lemma D.1 and set $\nu_i = \frac{\tilde{\rho}_i}{L_i+L_{\phi_i}+\rho_i}$ in order to cancel the $\|z^k - z^{k+1}\|^2$ with the one in r^{k+1}. As a result, we may replace (24) with:

$$\sum_{i\in\mathcal{P}} \frac{1}{2(L_i+L_{\phi_i}+\tilde{\rho}_i)}\left\|\lambda_i^{k+1}-\lambda_i^*\right\|^2 \leq V^k - V^{k+1}.$$

Proceeding exactly as in A.5, we obtain the three following inequalities:

$$\frac{\underline{\rho}_{\mathcal{P}}}{\alpha_{\mathcal{P}}}\sum_{i\in\mathcal{P}}\frac{1}{2\rho_i}\|\lambda_i^{k+1}-\lambda_i^*\|^2 \leq V^k - V^{k+1},$$
$$\frac{\underline{\mu}_{\mathcal{P}}}{\bar{\rho}_{\mathcal{P}}}\sum_{i\in\mathcal{P}}\frac{\rho_i}{2}\|z^{k+1}-z^*\|^2 \leq V^k - V^{k+1}$$
$$\frac{\underline{\mu}_{\mathcal{P}}}{\overline{L}_{\phi}}\sum_{i\in\mathcal{P}} D_{\phi_i}(x_i^*,x_i^{k+1}) \leq V^k - V^{k+1}.$$

Summing up these three inequalities yields the desired result.

All Proximal Agents: The proof is identical to the one for all proximal agents, except that there is one fewer inequality due to the absence of the Bregman terms.

B Bregman Divergence

Definition B.1 (Bregman Divergence)**.** *Given a differentiable convex function ϕ, the associated Bregman divergence is defined as:*

$$D_\phi(y,x) = \phi(y) - \phi(x) - \nabla\phi(x)^T(y-x).$$

Lemma B.2. *Let ϕ be a convex function and D_ϕ be the associated Bregman divergence. Then:*

$$\left(\nabla\phi(u) - \nabla\phi(v)\right)^T(w-u) = D_\phi(w,v) - D_\phi(w,u) - D_\phi(u,v), \quad \forall u,v,w.$$

Lemma B.3. *Let ϕ be a convex function with L-Lipschitz continuous gradients, and D_ϕ be the associated Bregman divergence. Then:*

$$D_\phi(x,y) \leq \frac{L}{2}\|x-y\|^2.$$

C Convex Conjugate

Definition C.1 (Convex Conjugate)**.** *Let $f:\mathbb{R}^n\to\mathbb{R}$ be a convex function. Its convex conjugate $f^*:\mathbb{R}^n\to\mathbb{R}$ is defined as:*

$$f^*(y) = \sup_{x\in\mathbb{R}^n}\left\{x^Ty - f(x)\right\}.$$

Proposition C.2. *Let f be a convex and differentiable function, and f^* be its convex conjugate. Then the following statements are equivalent:*

$$y = \nabla f(x) \Leftrightarrow x = \nabla f^*(y) \Leftrightarrow x^Ty = f(x) + f^*(y).$$

D Inequalities

Lemma D.1. *For any two vectors $x, y \in \mathbb{R}^n$, and $\nu > 0$, we have:*

$$\|x - y\|^2 \geq (1 - \nu)\|x\|^2 - \left(\frac{1}{\nu} - 1\right)\|y\|^2.$$

Proof. We have:

$$\begin{aligned}
\|x + y\|^2 &= \|x\|^2 + \|y\|^2 + 2x^T y \\
&= \|x\|^2 + \|y\|^2 + 2(\sqrt{\nu}x)^T \left(\frac{1}{\sqrt{\nu}}\right) y \\
&= \|x\|^2 + \|y\|^2 + \left\|\sqrt{\nu}x + \frac{1}{\sqrt{\nu}}y\right\|^2 - \nu\|x\|^2 - \frac{1}{\nu}\|y\|^2 \\
&\geq (1 - \nu)\|x\|^2 - \left(\frac{1}{\nu} - 1\right)\|y\|^2.
\end{aligned}$$

□