AN AUGMENTED LAGRANGIAN BASED ALGORITHM FOR DISTRIBUTED NON-CONVEX OPTIMIZATION

BORIS HOUSKA[1,2] , JANICK FRASCH[3] , AND MORITZ DIEHL[4]

Abstract. This paper is about distributed derivative-based algorithms for solving optimization problems with a separable (potentially nonconvex) objective function and coupled affine constraints. A parallelizable method is proposed that combines ideas from the fields of sequential quadratic programming and augmented Lagrangian algorithms. The method negotiates shared dual variables that may be interpreted as prices, a concept employed in dual decomposition methods and the alternating direction method of multipliers (ADMM). Here, each agent solves its own small-scale nonlinear programming problem and communicates with other agents by solving coupled quadratic programming problems. These coupled quadratic programming problems have equality constraints for which parallelizable methods are available. The use of techniques associated with standard sequential quadratic programming (SQP) methods gives a method with superlinear or quadratic convergence rate under suitable conditions. This is in contrast to existing decomposition methods, such as ADMM, which have a linear convergence rate. It is shown how the proposed algorithm may be extended using globalization techniques that guarantee convergence to a local minimizer from any initial starting point.

1. Introduction. Large scale nonlinear optimization problems arise in a variety of applications ranging from economic optimization under shared resources via statistical learning algorithms for networks and smart grids to distributed nonlinear optimal control for ordinary and partial differential equations. Fortunately, although the optimization problems arising from these fields may be large scale, they are often structured and have a separable objective such that the optimization problem can be written in the form

$$\min_x \sum_{i=1}^N f_i(x_i) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^N A_i x_i = b \\ h_i(x_i) \leq 0 \ , \ \ i \in \{1,\ldots,N\} \ . \end{array} \right. \tag{1.1}$$

This paper concerns distributed local optimization algorithms that can solve problems of the form (1.1) for non-convex functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1,\ldots,N\}$ and $h_i : \mathbb{R}^n \to \mathbb{R}^{n_h}$, and for a potentially large integer N. Here, the matrices $A_1,\ldots,A_N \in \mathbb{R}^{m\times n}$ and the vector $b \in \mathbb{R}^m$ are assumed to be given. In practice, the matrices A_i and the vector b often model dependencies between sub-systems. Examples are resource constraints, couplings between sub-systems in chemical production processes (batches), time-dependencies in dynamic optimization problems, and localization dependencies in distributed sensor networks.

Existing distributed algorithms for problems of the form (1.1) often assume that the functions f_i and h_i are convex. For example, one of the oldest and most basic distributed convex optimization algorithm is the dual decomposition method, which has originally been proposed in [21]. Here, the main idea is to solve the dual ascent problem that is associated with Problem (1.1) by employing a gradient method, while the dual objective function itself is evaluated in a distributed way. More recent articles and literature reviews of the dual decomposition method for convex optimization problems can be found in [5, 48]. In recent years, dual decomposition is employed frequently in distributed optimal control and model predictive control algorithms. A wide variety of algorithms use gradient information-based ascent techniques to obtain a fully distributed algorithm [32, 33, 49, 58, 59]. Other dual decomposition methods employ an interior-point framework with a smoothed dual function [51, 70]. These methods perform well if initialized far from a minimizer, but often require a large number of iterations to achieve medium accuracy in the solution. Yet another class of dual decomposition methods [24, 25, 42] employs a semi-smooth Newton method, similar to the one proposed by [45], for solving the dual optimization problems. While a certain amount of

[1]Corresponding Author (borish@shanghaitech.edu.cn).
[2]School of Information Science and Technology (SIST), ShanghaiTech University, 319 Yueyang Road, Shanghai 200031, China.
[3]Faculty of Mathematics, University of Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany.
[4]Department of Microsystems Engineering (IMTEK) and Department of Mathematics, University of Freiburg, Georges-Koehler-Allee 102, 79110 Freiburg, Germany.

communication is required for these methods, they are still highly parallelizable and often lead to a significant reduction in the number of iterations. For optimal control problems, a numerical implementation of the dual decomposition semi-smooth Newton strategy is the open-source code `qpDunes` [26].

Another important and powerful class of distributed optimization algorithms is based on the Uzawa method [72, 73] or the alternating direction method of multipliers (ADMM), which has originally been introduced in [28, 34]. ADMM methods have been analyzed by many authors [12, 19, 20, 35] in the past. The review article [9] includes a self-contained convergence proof of ADMM for convex optimization problems. One advantage of ADMM in comparison to the standard dual decomposition approach is that it converges more reliably, if the functions f_i are convex but not necessarily strictly convex [9]. Recently, ADMM has also been applied in the field of optimal control and we refer to [53] for an overview.

Unfortunately, if the functions f_i and h_i are non-convex, far fewer approaches exist. Dual decomposition methods are not applicable in this case, since we may have a duality gap. Similarly, despite the successful developments and fortunate properties of ADMM for convex optimization problems outlined above, ADMM is in general not applicable if the functions f_i are non-convex. In Section 2 a non-convex optimization problem is presented, for which ADMM is divergent. In summary, existing distributed optimization methods from the field of convex optimization cannot be applied to solve non-convex problems.

One way to construct parallelizable non-convex optimization algorithms is to start with a standard nonlinear programming method and try to parallelize or even distribute most of its operations. Problem (1.1) can be solved by employing a sequential quadratic programming method [6, 56, 57, 75], where the partially separable structure of the objective function can be exploited for computing Hessian approximations [69]. In this case, the evaluation of the functions f_i and h_i as well as the evaluation of their first or even second order derivatives can trivially be parallelized. The convex quadratic programming (QP) sub-problems can be solved with any of the distributed algorithms from the field of convex optimization outlined above. This or very similar parallelization strategies are for example analyzed in [50] in a more general sequential convex programming (SCP) framework. Other variants deal with the potentially large number of inequality constraints by using external active set strategies [14, 61]. An example for an implementation of an SQP method for medium- to large-scale problems is the code `SNOPT` [29]. This way of applying the sequential quadratic- or a more general sequential convex programming approach can lead to an unnecessarily large number of potentially expensive communication steps. For example, if we have

$$\sum_{i=1}^{N} A_i x_i^* = b \qquad \text{for} \qquad x^* \in \underset{x}{\operatorname{argmin}} \ \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad h_i(x_i) \leq 0 \ , \ i \in \{1, \ldots, N\} \ ,$$

the above outlined sequential quadratic programming method solves a potentially large number of coupled convex optimization problems. Solving these convex problems requires communication, although the original non-convex problem is decoupled. The optimization problems of interest are not decoupled, but in many practical problems only a weak coupling is present. This situation occurs frequently in applications for which the coupling constraints are introduced in order to model a refinement rather than a crucial feature. For example, if every agent in a sensor network can measure its position, all agents in this network can estimate their position independently by solving decoupled (potentially non-convex) maximum likelihood estimation problems. If an additional measurement of the distance between the agents is available, they can improve their position estimates by cooperating and solving jointly a coupled optimization problem. In this case, the coupling constraint is introduced in order to refine the solution of the original decoupled optimization problems. The solution of the coupled optimization problem can be expected to be close to solution of the original decoupled optimization problems. In such a situation, distributed SQP or more general SCP methods are not the best choice, as these methods do not solve the

decoupled NLPs as part of their iteration. Moreover, if we use parallelized variants of standard nonlinear programming methods, globalization routines, such as a line search, may require us to exchange additional information between the agents [71].

An alternative to SQP based methods are augmented Lagrangian methods [1, 39, 67, 55]. These methods have been analyzed exhaustively in the context of large scale non-convex optimization and are implemented in the nonlinear programming software library `GALAHAD` [17, 36]. Recent developments of nonlinear optimization methods also include the exploitation of primal-dual augmented Lagrangians [30], which can be used to construct variants of regularized SQP methods [31]. Early approaches towards the application of augmented Lagrangian methods in the context of distributed nonlinear optimization have been proposed in [64] and [74]. In these articles, the authors develop strategies to approximate the coupled terms in the augmented Lagrangian by first order approximations or minima of separable functions. Another framework for non-convex distributed optimization based on augmented Lagrangians has been proposed by Bertsekas [2]. In this article the local convexity of augmented Lagrangians functions for sufficiently large penalty parameters is exploited in order to apply the concept of dual decomposition to non-convex problems. Variants of Bertsekas' method can be found in [66, 68]. Cohen [15] suggests solving the original coupled problem by constructing a sequence of auxiliary problems. A review of augmented Lagrangian based decomposition methods for convex and non-convex optimization algorithms can be found in [37].

Outline of the Paper and Contributions

Section 2 starts with a review of ADMM, arguably a state-of-the-art algorithm for distributed convex optimization. It is explained why ADMM is not directly applicable to solve non-convex optimization problems without further precaution by discussing an example for which ADMM is divergent. The main contribution of this paper is outlined in Section 3, where a novel "Augmented Lagrangian based Alternating Direction Inexact Newton" method (ALADIN) is introduced. This new method is designed to solve potentially non-convex optimization problems of the form (1.1). The similarities and differences to existing large scale nonlinear optimization methods such as inexact sequential quadratic programming as well as distributed augmented Lagrangian methods are discussed in Section 4. Section 5 discusses cases in which ALADIN has the same iteration cost as existing ADMM methods. It is outlined why ALADIN may lead to a significantly smaller number of overall iterations and thus has the potential to out-perform state-of-the-art distributed optimization methods. The global and local convergence properties of ALADIN are discussed in Section 6 and 7, respectively. Section 8 provides a numerical example, and Section 9 concludes the paper.

Notation
Apart from mathematical standard notation, the dual variables are written immediately after the constraint. For example, the syntax

$$\min_x \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^N A_i x_i = b \;\Big|\; \lambda \\ h_i(x_i) \leq 0 \,|\, \kappa_i \end{array} \right. , \; i \in \{1, \ldots, N\} .$$

means that the multiplier of the affine equality constraints is denoted by λ and the multipliers of the ith decoupled inequality constraint by κ_i. Throughout this paper, a KKT point (x, λ, κ) is called regular if the linear independence constraint qualification (LICQ), strict complementarity conditions (SCC), as well as the second order sufficient condition (SOSC) are satisfied [52].

2. Review of the Alternating Direction Method of Multipliers. This section reviews ADMM for solving Problem (1.1). ADMM has turned out to be a successful distributed algorithm for convex optimization problems [43]. This section illustrates that ADMM is in general divergent and—at least without further modifications—not applicable to non-convex optimization problems.

For the purposes of this section, the structured optimization problem (1.1) is written in scaled consensus form. Let $I_0 : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ denote the indicator function,

$$I_0(r) = \begin{cases} 0 & \text{if } r = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Extended objective functions are given by

$$\forall \in \{1, \ldots, N\}, \qquad g_i(y_i) = \begin{cases} f_i(y_i) & \text{if } h_i(y_i) \leq 0 \\ \infty & \text{otherwise.} \end{cases}$$

The structured optimization problem (1.1) can be written in the equivalent form

$$\min_{x,y} \; \sum_{i=1}^{N} g_i(y_i) + I_0\left(\sum_{i=1}^{N} A_i x_i - b\right) \quad \text{s.t.} \quad A_i(y_i - x_i) = 0\,, \quad i \in \{1, \ldots, N\}\,. \tag{2.1}$$

This equivalence is due to the fact that the variables x_i enter the coupled constraints via the terms $A_i x_i$ only. Thus, instead of requiring $x_i = y_i$, it is sufficient to enforce the constraints $A_i(y_i - x_i) = 0$, although this formulation leads to redundant optimization variables, if the matrices A_i do not have full column rank. Recall that the matrices $A_i : \mathbb{R}^{m \times n}$ and the vectors $b \in \mathbb{R}^m$ are assumed to be given. The main idea of ADMM is to construct an augmented Lagrangian function of the form

$$L_\rho(x, y, \lambda) \;=\; I_0\left(\sum_{i=1}^{N} A_i x_i - b\right) + \sum_{i=1}^{N} \left\{ g_i(y_i) + \lambda_i^\mathsf{T} A_i(y_i - x_i) + \frac{\rho}{2} \left\| A_i(y_i - x_i) \right\|^2 \right\}\,, \tag{2.2}$$

where $\rho > 0$ is a penalty parameter. Starting with an initial guess x for the primal optimization variable and an initial guess λ for the dual vector that is associated with the consensus constraints, the method has the following steps.[1]

1. Solve the optimization problem $y \in \operatorname{argmin}_y L_\rho(x, y, \lambda)$.
2. Terminate, if $\left\| \sum_{i=1}^{N} A_i y_i - b \right\|_1$ is sufficiently small.[2]
3. Compute the dual variable updates $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.
4. Solve the optimization problem $x^+ \in \operatorname{argmin}_{x^+} L_\rho(x^+, y, \lambda^+)$.
5. Set $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$ in order to continue with Step 1.

The first optimization problem for the variable y is decoupled and can be solved in parallel. The second optimization problem for the variable x^+ can be solved explicitly since this amounts to solving a quadratic program with linear equality constraints. Algorithm 1 summarizes ADMM.

If the functions f_i and h_i are convex, convergence of Algorithm 1 can be established under mild assumptions. This result is independent of how far the initial (x, λ) is away from the optimal solution and independent of how the penalty parameter $\rho > 0$ is chosen [9]. In this context, we also mention that Algorithm 1 presents only one way to solve Problem (1.1) based on ADMM, and there are many other variants possible as discussed in [9]. Algorithm 1 has the advantage that it is parallelizable, but it requires us to solve a coupled equality constrained quadratic programming problem in each loop. In large part the linear algebra operations required for solving this QP can be performed in an initialization step. As the constraint matrices A_i, the constraint right-hand b, as well as the Hessian matrix $\rho A_i^\mathsf{T} A_i$ of this QP are constant, linear algebra decompositions of

[1] We could also start the loop with Step 4 assuming that we have an initial guess for y and λ^+.

[2] The termination criterion is not scale invariant. We use one-norms to measure constraint violations throughout, although it would be possible to use other norms.

Algorithm 1: Alternating Direction Method of Multipliers (Consensus Form)

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda_i \in \mathbb{R}^m$, a penalty parameter $\rho > 0$, and a tolerance $\epsilon > 0$.

Repeat:

1. Solve for all $i \in \{1, \ldots, N\}$ the decoupled NLPs

$$\min_{y_i} \; f_i(y_i) + \lambda_i^\mathsf{T} A_i y_i + \frac{\rho}{2} \|A_i(y_i - x_i)\|_2^2 \quad \text{s.t.} \quad h_i(y_i) \leq 0 \; . \tag{2.3}$$

2. If $\left\| \sum_{i=1}^N A_i y_i - b \right\|_1 \leq \epsilon$, terminate and return $x^* = y$ as a numerical solution.
3. Implement the dual gradient steps $\lambda_i^+ = \lambda_i + \rho A_i(y_i - x_i)$.
4. Solve the coupled equality constrained quadratic programming problem

$$\min_{x^+} \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| A_i(y_i - x_i^+) \right\|_2^2 - (\lambda_i^+)^\mathsf{T} A_i x_i^+ \right\} \quad \text{s.t.} \quad \sum_{i=1}^N A_i x_i^+ \; = \; b \; . \tag{2.4}$$

5. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$ and continue with Step 1.

all matrices in this QP can be pre-computed. Notice that only the objective gradient of the QP changes in each iteration.

The convergence properties of ADMM have been analyzed in the context of maximal monotone operators [19, 20] and by using convex analysis [9]. The question under which assumptions ADMM is also applicable for non-convex functions f_i and h_i is an open problem. In general, ADMM does not converge for nonconvex problems. In order to illustrate this, it is assumed that we have no inequality constraints and that the functions f_i are twice continuously differentiable. In this case, the Hessian matrix of the decoupled sub-problems in Step 1 of Algorithm 1 is given by

$$H_i(y_i, \rho) \; = \; \nabla^2 f_i(y_i) + \rho A_i^\mathsf{T} A_i \; .$$

Thus, even if the functions f_i are non-convex, the decoupled sub-problems satisfy the SOSC condition at a local minimizer y_i, as long as a sufficiently large augmented Lagrangian parameter ρ with $H_i(y_i, \rho) \succ 0$ exists. One might conjecture that Algorithm 1 converges to a local minimizer if the functions f_i are non-convex, as long as we choose the initialization in a small neighborhood of this minimizer and if the sub-problems are strictly convex. However, unfortunately, this conjecture is false. In order to illustrate this, a counter-example is provided.

EXAMPLE 2.1. *This example is about the case $N = 1$ and $n = 2$ with $f_1(x) = x_1 \cdot x_2$, $A = (1, -1)$, and $b = 0$ (but no inequalities) such that Problem (1.1) takes the form*

$$\min_x \; x_1 \cdot x_2 \qquad \text{s.t.} \qquad x_1 - x_2 = 0 \; .$$

Clearly, this optimization problem has a unique and regular minimizer at $x_1^* = x_2^* = \lambda^* = 0$ at which the linear independence constraint qualification as well as the second order sufficient KKT condition is satisfied. Moreover, for $\rho = \frac{3}{4}$, the sub-problems in Step 1 of Algorithm 1 are strictly convex, since the matrix

$$H_1\left(0, \frac{3}{4}\right) \; = \; \frac{1}{4} \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

is positive definite. The iteration is started with $x = 0$ but $\lambda \neq 0$. In the first step, the decoupled NLP has the form

$$\min_y \; \frac{1}{2} y^\mathsf{T} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} y + \lambda(y_1 - y_2) + \frac{3}{8} \|y_1 - y_2\|_2^2 \; .$$

It can be solved explicitly finding the optimal solution

$$y \;=\; \begin{pmatrix} -2 \\ 2 \end{pmatrix} \lambda \,.$$

Next, the quadratic programming problem from Step 4 of Algorithm 1 has a trivial solution at $x^+ = 0$, since x^+ enters the optimization problem only via the term Ax^+ that is however enforced to be equal to zero at the optimal solution. Thus, the only non-trivial iterate is the variable λ, which satisfies

$$\lambda^+ \;=\; \lambda + \frac{3}{4}(y_1 - y_2) \;=\; -2\lambda \,.$$

This iteration is divergent. ◇

There exist several variants of ADMM depending on how the constraints are formulated and at which point the dual variable updates are implemented. The above example is a counter-example for the convergence of one particular variant of ADMM for a particular choice of ρ. The aim of this paper is to fix the convergence problem that can be observed in Example 2.1.

3. Distributed Nonlinear Optimization Algorithm. This section proposes a novel algorithm for solving structured optimization problems of the form

$$\min_x \; \sum_{i=1}^{N} f_i(x_i) \qquad \text{s.t.} \qquad \begin{cases} \sum_{i=1}^{N} A_i x_i \;=\; b \\ h_i(x_i) \;\leq\; 0 \,, \;\; i \in \{1, \ldots, N\} \,. \end{cases} \tag{3.1}$$

The functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}^{n_h}$ are assumed to be twice continuously differentiable for all $i \in \{1, \ldots, N\}$, but not necessarily convex. It is also assumed that Problem (3.1) is feasible and that all local minimizers are regular KKT points. The aim is to find a local minimizer numerically. As in the previous sections, the matrices $A_i \in \mathbb{R}^{m \times n}$ and the vector $b \in \mathbb{R}^m$ are assumed to be given.

In the following, the dual variables of the inequality constraints $h_i(x_i)$ are denoted by $\kappa_i \in \mathbb{R}^{n_h}_+$. The multipliers of the affine coupling constraints are denoted by λ as in the previous section. The following sections are about Algorithm 2.

Notice that this "high-level interface" presentation of the main algorithmic idea relies on the assumption that we have already low-level tools for solving the coupled and potentially distributed equality constrained quadratic programming problems of the form (3.3) as well as a centralized NLP solver for solving problems of the form (3.2). In a practical implementation the accuracy of these lower level QP and NLP solvers can be adjusted by auxiliary routines. Nevertheless, the analysis and discussion in this paper is based on the assumption that the QP and NLPs are solved with high-precision. An analysis of variants, which solve the NLPs and QP inexactly are beyond the scope of this paper. The decoupled optimization problems (3.2) are feasible whenever the original problem (3.1) is feasible. The penalty parameter $\rho \geq 0$ is in principle redundant in the sense that the symmetric and positive definite scaling matrices $\Sigma_i \succeq 0$ introduced in Algorithm 2 may be adjusted during the iterations. The QP subproblems (3.3) are always feasible, as the point

$$(\Delta y, s) \;=\; \left(0, \sum_{i=1}^{N} A_i y_i - b \right)$$

is a feasible point of Problem (3.3). In Step 3 of Algorithm 2 the modified gradient $g_i = \nabla f_i(y_i) + (C_i^* - C_i)^\mathsf{T} \kappa_i$ is constructed. This is in analogy to inexact SQP methods, which are based on modified gradients that correct errors in the step direction due inexact constraint Jacobian approximations as explained in [18]. The first order stationarity conditions of Problem (3.2) are given by

$$\nabla f_i(y_i) + A_i^\mathsf{T} \lambda + (C_i^*)^\mathsf{T} \kappa - \rho \Sigma_i (y_i - x_i) \;=\; 0 \,.$$

Algorithm 2: Augmented Lagrangian based Alternating Direction Inexact Newton Method

Input: Initial guesses $x_i \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$ and a numerical tolerance $\epsilon > 0$.

Repeat:

1. Choose a sufficiently large penalty parameter $\rho \geq 0$ and positive semi-definite scaling matrices $\Sigma_i \in \mathbb{S}_+^{n_x}$ and solve for all $i \in \{1, \ldots, N\}$ the decoupled optimization problems

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \|y_i - x_i\|_{\Sigma_i}^2 \quad \text{s.t.} \quad h_i(y_i) \leq 0 \;\; | \;\; \kappa_i \tag{3.2}$$

   to either local or global optimality.

2. If $\left\| \sum_{i=1}^N A_i y_i - b \right\|_1 \leq \epsilon$ and $\rho \, \|\Sigma_i (y_i - x_i)\|_1 \leq \epsilon$, terminate with $x^* = y$ as a numerical solution.

3. Choose constraint Jacobian approximations $C_i \approx C_i^*$ of the matrices C_i^* defined by

$$C_{i,j}^* \; = \; \begin{cases} \frac{\partial}{\partial x} (h_i(x))_j \Big|_{x=y_i} & \text{if } \; (h_i(y_i))_j = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for} \quad i \in \{1, \ldots, N\}, \, \forall j \in \{1, \ldots, n_h\} \; .$$

   Compute the modified gradient $g_i = \nabla f_i(y_i) + (C_i^* - C_i)^\mathsf{T} \kappa_i$ and choose symmetric Hessian approximations $H_i \approx \nabla^2 \left\{ f_i(y_i) + \kappa_i^\mathsf{T} h_i(y_i) \right\}$.

4. Choose a sufficiently large penalty parameter $\mu > 0$ and solve the coupled QP

$$\begin{aligned} &\min_{\Delta y, s} \quad \textstyle\sum_{i=1}^N \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} + \lambda^\mathsf{T} s + \frac{\mu}{2} \|s\|_2^2 \\ &\text{s.t.} \qquad \begin{cases} \sum_{i=1}^N A_i \left(y_i + \Delta y_i \right) \; = \; b + s \;\; \Big| \;\; \lambda_{\text{QP}} \\ C_i \Delta y_i \; = \; 0 \; , \quad i \in \{1, \ldots, N\} \; . \end{cases} \end{aligned} \tag{3.3}$$

5. Run Algorithm 3 in order to find step-sizes $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}_+$ or, alternatively, set $\alpha_1 = \alpha_2 = \alpha_3 = 1$ in order to run ALADIN without global convergence guarantees. Define

$$x^+ = x + \alpha_1 (y - x) + \alpha_2 \Delta y \quad \text{and} \quad \lambda^+ = \lambda + \alpha_3 \left(\lambda_{\text{QP}} - \lambda \right) \; .$$

6. Update the iterates $x \leftarrow x^+$ and $\lambda \leftarrow \lambda^+$ and continue with Step 1.

The linear independence constraint qualification for the lower level inequality constraints is assumed to be satisfied. Thus, if Algorithm 2 terminates by satisfying the termination criterion from Step 2, the estimate $\rho \, \|\Sigma_i (y_i - x_i)\|_1 \leq \epsilon$ holds. Consequently,

$$\left\| \nabla f_i(y_i) + A_i^\mathsf{T} \lambda + (C_i^*)^\mathsf{T} \kappa \right\|_1 \; \leq \; \epsilon \; ;$$

that is, the point $x^* = y$ with associated dual solution $\lambda^* = \lambda$ and $\kappa^* = \kappa$ satisfies the first order stationarity condition for Problem (3.1) up to the user-specified numerical tolerance $\epsilon > 0$. Moreover, if the termination criterion in Step 2 is satisfied y also satisfies the primal feasibility condition $\left\| \sum_{i=1}^N A_i y_i - b \right\|_1 \leq \epsilon$. It can be checked easily that $(x^*, \lambda^*, \kappa^*)$ is a primal-dual KKT point of Problem (3.1)—up to the user specified numerical accuracy ϵ. In particular, y is always feasible with respect to the inequality constraints. As mentioned earlier, the termination criterion is not scale invariant. The one-norm could also be replaced by other norms.

In the following, Algorithm 2 is called "Augmented Lagrangian based Alternating Direction Inexact Newton Method" (ALADIN). The connection of this algorithm to augmented Lagrangian, ADMM, and generic inexact Newton methods will be discussed in the following sections, which focus on explaining similarities but also differences and advantages of the proposed algorithm to existing large scale and distributed optimization methods. A mathematical analysis of the convergence properties of ALADIN follows in Sections 6 and 7.

4. Similarities and Differences Compared to (Inexact-) SQP and Augmented Lagrangian Methods. The above outlined Algorithm 2 has a certain similarity with both SQP as well as augmented Lagrangian methods. One way to highlight this similarity is by considering the

Powell-Hestenes augmented Lagrangian function with respect to the coupled constraints

$$\mathcal{K}_\mu(y,\lambda) = \sum_{i=0}^{N} f_i(y_i) + \lambda^\mathsf{T}\left(\sum_{i=1}^{N} A_i y_i - b\right) + \frac{\mu}{2}\left\|\sum_{i=1}^{N} A_i y_i - b\right\|_2^2 .$$

One variant of the augmented Lagrangian method defines an approximate minimizer of the optimization problem

$$\min_{\Delta y}\ \mathcal{K}_\mu(y+\Delta y,\lambda) \qquad \text{s.t.} \qquad h_i(y+\Delta y) \le 0 \tag{4.1}$$

and updates the primal and dual variables as

$$y^+ = y + \Delta y \quad \text{and} \quad \lambda^+ = \lambda + \mu\left(\sum_{i=1}^{N} A_i y_i^+ - b\right) . \tag{4.2}$$

Clearly, one way to approximately solve problem (4.1) is by performing exactly one SQP step. Using the same notation as in Algorithm 2, this leads to a QP subproblem of the form

$$\begin{aligned} \min_{\Delta y} \quad & \textstyle\sum_{i=1}^{N}\left\{\frac{1}{2}\Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T}\Delta y_i\right\} + \lambda^\mathsf{T}\left(\sum_{i=1}^{N} A_i\left(y_i+\Delta y_i\right) - b\right) \\ & + \tfrac{\mu}{2}\left\|\sum_{i=1}^{N} A_i\left(y_i+\Delta y_i\right) - b\right\|_2^2 \\ \text{s.t.} \quad & h_i(y_i) + C_i\Delta y_i \ \le\ 0\ , \quad i\in\{1,\dots,N\}\ . \end{aligned} \tag{4.3}$$

Here, the choice $C_i = \nabla h_i(y_i)^\mathsf{T}$ with $g_i = \nabla f(y_i)$ corresponds to an exact SQP method. Otherwise, if C_i is only an approximation of the exact constraint Jacobian and $g_i = \nabla f(y_i) + (C_i^* - C_i)^\mathsf{T}\kappa_i$ the modified gradient, this corresponds to an inexact SQP method [18, 40]. For numerical reasons, when μ is large, it is better to solve the equivalent QP[3]

$$\begin{aligned} \min_{\Delta y, s} \quad & \textstyle\sum_{i=1}^{N}\left\{\frac{1}{2}\Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T}\Delta y_i\right\} + \lambda^\mathsf{T} s + \frac{\mu}{2}\|s\|_2^2 \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^{N} A_i\left(y_i+\Delta y_i\right) = b + s \ \Big|\ \lambda^+ \\ h_i(y_i) + C_i\Delta y_i \ \le\ 0\ , \quad i\in\{1,\dots,N\}\ , \end{cases} \end{aligned} \tag{4.4}$$

by introducing the slack variable s. Here, the updated multiplier λ^+ from Equation (4.2) can be identified with the multiplier that belongs to the equality constraint in the above QP[4]. Interestingly, this QP basically coincides with the quadratic programming problem (3.3) that is solved in Step 4 of Algorithm 2. Here, the only difference between the two QPs is that in (3.3) the inequalities in the current working set are enforced as equalities, while QP (4.3) keeps all linearized inequalities without relying on any guess of the active set. From this perspective, the proposed ALADIN algorithm is a mixture of an augmented Lagrangian and an SQP method. However, as explained in the introduction, both SQP and augmented Lagrangian methods are centralized optimization algorithms that do not solve decoupled optimization problems of the form (3.2). In this respect, ALADIN differs from augmented Lagrangian based SQP methods.

Solving large-scale quadratic programming problems with inequality constraints, such as Problem (4.3), can be expensive. Clearly, solving the equality constrained optimization problem (3.3) is still expensive, as this requires communication between the distributed agents, but at least this equality constrained QP can be solved by a suitable sparse or distributed linear algebra

[3] QP (4.4) is still ill-conditioned if μ is large, but the advantage of the re-formulation with the slack variable s is that the large entries in the associated Hessian matrix are all on the diagonal such that the problem can easily be re-scaled by employing a suitable pre-conditioner.

[4] The stationarity condition for the QP (4.4) with respect to s has the form $\lambda + \mu s - \lambda^+ = 0$, which yields the update rule (4.2), $\lambda^+ = \lambda + \mu s = \lambda + \mu\left(\sum_{i=1}^{N} A_i y_i^+ - b\right)$.

solver [10, 13, 65]. This concept is in a similar form adopted by external active set methods [14, 61] (also known as "Outer Active Set Methods"), which are well-known tools for solving nonlinear optimization problems with a large number of constraints. However, a major difference of Algorithm 2 compared to external active set methods is the way how the active set is determined. Here again, the main difference is the introduction of Step 1 in combination with the approximation $C_i \approx C_i^*$, where the set of indices of the non-zero rows of C_i can be interpreted as our current guess for the active set. For the special choice $C_i = C_i^*$ the working set corresponds to the active constraint indices at the optimal solution of the decoupled nonlinear programming problems (3.2). This is in contrast to conventional active-set methods, which typically determine a new working set by adding a constraint to maintain feasibility, or removing a constraint based on the sign of its Lagrange multiplier. A conventional active-set method does not solve an intermediate non-trivial decoupled NLP.

REMARK 4.1. *The above variant of the augmented Lagrangian method has similarities but is not equivalent to Linearly Constrained Lagrangian (LCL) method [27].*

REMARK 4.2. *The method in Algorithm 2 is equivalent to dual decomposition if $\rho = 0$ is enforced. This result is established in Appendix A.*

5. Similarity to ADMM for $H_i = \rho A_i^\mathsf{T} A_i$, $\Sigma_i = A_i^\mathsf{T} A_i$, $C_i = 0$, and $\mu \to \infty$. Algorithm 2 is inspired by ADMM. For $\lambda_i = \lambda$ and $\Sigma_i = A_i^\mathsf{T} A_i$, Step 1 of Algorithm 1 coincides with Step 1 of Algorithm 2. Algorithm 2 is an "alternating direction method" in the sense that it alternates—similar to Algorithm 1—between solving small scale decoupled nonlinear programming problems and large scale coupled equality constrained quadratic programming problems. The following discussion assumes that we use the constraint Jacobian approximation $C_i = 0$, $i \in \{1, \dots, N\}$. If no inequality constraints are active, this approximation is exact. For $\mu \to \infty$, i.e., $s = 0$ in Problem (3.3), the quadratic programming problem (3.3) can be written in the equivalent form

$$\begin{array}{ll} \min_{\Delta y} & \sum_{i=1}^N \left\{ \frac{1}{2} \Delta y_i^\mathsf{T} H_i \Delta y_i + g_i^\mathsf{T} \Delta y_i \right\} \\ \text{s.t.} & \sum_{i=1}^N A_i \left(y_i + \Delta y_i \right) \;=\; b \;\Big|\; \lambda_{\text{QP}} \; . \end{array} \tag{5.1}$$

For $C_i = 0$, the modified gradient is given by

$$g_i \;=\; \nabla f_i(y_i) + (C_i^* - 0)^\mathsf{T} \kappa \;=\; A_i^\mathsf{T} \hat{\lambda}_i^{\text{ADMM}} \qquad \text{with} \qquad \lambda_i^{\text{ADMM}} \;=\; \lambda + \rho(A_i y_i - v_i) \; .$$

In the latter equality the stationarity condition has been substituted. Substituting this equation for the gradient and assuming $H_i = \rho A_i^\mathsf{T} A_i$ the above QP can equivalently be written in the form

$$\begin{array}{ll} \min_{\Delta y} & \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| A_i \Delta y_i \right\|_2^2 + \left(\lambda_i^{\text{ADMM}} \right)^\mathsf{T} A_i \Delta y_i \right\} \\ \text{s.t.} & \sum_{i=1}^N A_i \left(y_i + \Delta y_i \right) \;=\; b \;\Big|\; \lambda_{\text{QP}} \; . \end{array} \tag{5.2}$$

The choice $\alpha_1 = \alpha_2 = 1$ implies $\Delta y_i = y_i - x_i^+$ revealing that the above QP is coinciding with the quadratic programming problem (2.4) as long as λ_i^{ADMM} is associated with the updated dual variable "λ_i^+" in Step 3 of Algorithm 1. Thus, for $H_i = \rho A_i^\mathsf{T} A_i$, $\Sigma_i = A_i^\mathsf{T} A_i$, and $C_i = 0$ the cost of one ALADIN iteration is exactly the same as the cost of one ADMM iteration, since the only difference between these two algorithms is that ALADIN maintains only one dual variable $\lambda \in \mathbb{R}^m$ that is updated in a slightly different manner than the dual variables $\lambda_1, \dots, \lambda_N \in \mathbb{R}^m$ from Algorithm 1. However, given the fact that the ADMM equivalent choice $C_i = 0$ and $H_i = \frac{\rho}{2} A_i^\mathsf{T} A_i$ corresponds in general to a rather poor approximation of the constraint Jacobian and Hessian matrix, gives a strong indication that ALADIN has the potential to out-perform ADMM in many practical applications. This is motivated by the fact that it is possible to construct more accurate approximations C_i and H_i of the constraint Jacobian and Hessian matrix—in some special cases, e.g., if the number m of coupling constraints is small, it may even be tractable to compute the exact matrices C_i^* and H_i^*.

Numerical Linear Algebra Considerations. If the approximations C_i and H_i are constant, some of the matrix decompositions for solving the quadratic programming problem (3.3) can be computed in advance. The local convergence properties of ALADIN are, however, much better and less scaling dependent if we maintain updates C_i and H_i during the algorithm. Examples for such updates include low-rank updates of C_i and H_i, e.g., computed via BFGS- or even limited memory BFGS updates, or updates of our guess of the optimal active set referring to the decision of which rows of C_i are chosen to be exactly equal to zero. These updates are not necessarily expensive. For example, low-rank updates of C_i and H_i can directly be translated into cheap updates of the matrix decompositions and, similarly, updates of the active set can be realized in analogy to the update techniques that are employed in state-of-the-art active set QP solvers [23]. In practice, it depends on the dimension and sparsity pattern of the matrices A_i, as well as the cost for communication between the distributed agents whether the time investment for maintaining non-constant constraint Jacobian and Hessian approximations as well as updates of the associated matrix decompositions that are required for solving QP (3.3) pays off in terms of the overall number of iterations due to better convergence rate. Notice that this is nothing but the well-known trade-off of cost per iteration and convergence rate that is omni-present in nonlinear optimization algorithms.

6. Global Convergence Analysis. This section analyzes how to enforce global convergence of Algorithm 2 to local minimizers of the original optimization problem (1.1). As in the above discussions, it is assumed that the linear independence constraint qualification holds such that all local minimizers of the decoupled optimization problem (3.2) as well as the original optimization problem (3.1) satisfy the first order KKT conditions. Notice that this is a rather mild assumption, which always holds under the linear independence constraint qualification and which is in a similar form employed in the context of global convergence of other nonlinear programming methods [52]. Moreover, it is assumed that the Hessian approximation matrices H_i are symmetric and positive semi-definite. However, neither the approximations $H_i \approx H_i^*$ of the Hessians nor the approximations $C_i \approx C_i^*$ of the constraint Jacobians have to be accurate. The following convergence analysis is rather general and includes in particular the ADMM-inspired approximation $C_i = 0$ and $H_i = \rho A_i^\mathsf{T} A_i$ as well as low-rank approximations of H_i and C_i as special cases. The scaling matrices Σ_i are assumed to be positive definite. Notice that for the ADMM-inspired choice, $\Sigma_i = A_i^\mathsf{T} A_i$, this assumption is only satisfied if the matrices A_i have full column rank.

Following a standard framework for measuring global progress of an optimization algorithm towards a local minimizer [38, 52], a globalization routine for ALADIN is summarized in the form of Algorithm 3. This algorithm is based on the L1-penalty function

$$\Phi(x) \;=\; \sum_{i=1}^{N} f_i(x_i) + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i x_i - b \right\|_1 + \overline{\kappa} \sum_{i,j} \max\{\, 0,\, (h_i(x_i))_j \,\}\,,$$

where $0 < \overline{\lambda} < \infty$ and $0 < \overline{\kappa} < \infty$ are assumed to be sufficiently large constants such that Φ is an exact penalty function for Problem (1.1). The aim of the following consideration is to show that Algorithm 3 enforces a descent of Φ after a finite number of steps. In order to ensure that this is sufficient to prove convergence, the descent condition

$$\Phi(x) - \Phi(x^+) \;\geq\; \gamma \left(\sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \right\} + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i y_i - b \right\|_1 \right) \tag{6.3}$$

is introduced. Here, $0 < \gamma \ll 1$ is fixed and y denotes the solution of the decoupled optimization problems (3.2). Notice that if the termination criterion from Step 2 of Algorithm 2 is not satisfied for the given tolerance $\epsilon > 0$, the expression on the right-hand side of the above inequality is always bounded below by

$$\gamma \left(\sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \right\} + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i y_i - b \right\|_1 \right) \;\geq\; \gamma \min \left\{ \frac{\sigma \epsilon^2}{2\rho},\, \overline{\lambda} \epsilon \right\}\,.$$

Algorithm 3: Globalization Strategy for ALADIN

Initialization: Choose sufficiently large $0 < \overline{\lambda} < \infty$ and $0 < \overline{\kappa} < \infty$ as well as a fixed $0 < \gamma \ll 1$.

Globalization Steps:

a) Set $\alpha_1 = \alpha_2 = \alpha_3 = 1$. If the iterate x^+ from Step 5 of Algorithm 2 satisfies

$$\Phi(x) - \Phi(x^+) \;\geq\; \gamma \left(\sum_{i=1}^{N} \left\{ \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \right\} + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i y_i - b \right\|_1 \right) \tag{6.1}$$

for the L_1 penalty function

$$\Phi(x) \;=\; \sum_{i=1}^{N} f_i(x_i) + \overline{\lambda} \left\| \sum_{i=1}^{N} A_i x_i - b \right\|_1 + \overline{\kappa} \sum_{i,j} \max\{\, 0,\, (h_i(x_i))_j \,\} \,,$$

return $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

b) If the full step is not accepted, set $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$ such that $x^+ = y$. If Condition (6.1) is satisfied, return $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$.

c) If neither a) nor b) was successful, set $\alpha_1 = \alpha_2 = 0$ such that $x^+ = x$. Next, determine the global maximizer $\alpha_3^* \in (0, 1]$ of the function $V_\rho(x, \lambda + \alpha_3 \left(\lambda_{\mathrm{QP}} - \lambda\right))$ with V_ρ denoting the optimal objective of the decoupled dual problem

$$\begin{aligned} V_\rho(\bar{x}, \lambda) \;=\; \min_{y} \;& \sum_{i=1}^{N} \left\{ f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - \bar{x}_i \right\|_{\Sigma_i}^2 \right\} - \lambda^\mathsf{T} b \\ \text{s.t.} \quad & h_i(y_i) \leq 0 \,, \quad i \in \{1, \ldots, N\} \,. \end{aligned} \tag{6.2}$$

Return $\alpha_1 = \alpha_2 = 0$ and $\alpha_3 = \alpha_3^*$.[a]

Output: Line-Search parameters $\alpha_1, \alpha_2, \alpha_3$ needed in Step 5 of Algorithm 2.

[a] We assume here for simplicity of presentation that the line search is exact. For a practical implementation this strategy should be refined, e.g., by implementing an inexact line-search based on Goldstein or Wolfe conditions [52].

Here, $\sigma > 0$ is a constant that depends—due to the equivalence of norms in finite dimensional Euclidean spaces—on the choice of the norm in the termination criterion only. Let

$$\begin{aligned} V_\rho(\bar{x}, \lambda) \;=\; \min_{y} \;& \sum_{i=1}^{N} \left\{ f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - \bar{x}_i \right\|_{\Sigma_i}^2 \right\} - \lambda^\mathsf{T} b \\ \text{s.t.} \quad & h_i(y_i) \leq 0 \,, \quad i \in \{1, \ldots, N\} \end{aligned} \tag{6.4}$$

denote a dual merit function. Under the assumption that ρ and μ are sufficiently large, Algorithm 3 enforces convergence. Notice that Algorithm 3 is divided into three steps, a), b), and c). The motivation for the Steps a) or b) is rather obvious, since these steps both aim at satisfying the strict descent condition (6.3). However, it is possible to construct cases where neither a) nor b) is successful. For example, if the constraint Jacobian approximation is exact, $C_i = C_i^*$, while at the optimal solution of the decoupled optimizations problem from Step 1 many of the inequality constraints are weakly active, the only way to satisfy the equality constraints $C_i \Delta y_i = 0$ in the equality constrained QP (3.3) might be to choose $\Delta y = 0$. If, in addition, $(y - x)$ is not a strict descent direction—for example, due to a poor initial guess for λ that is far away from the optimal solution, it is not possible to satisfy Condition (6.3). In such situations, Algorithm 3 proceeds with Step c), which sets $\alpha_1 = \alpha_2 = 0$, i.e., no update of the primal variable is implemented. Notice that the evaluation of the auxiliary function V_ρ in Step c) can be distributed. The line search in Step c) requires communication, which is in a similar form needed in dual decomposition methods, but must be interpreted as a disadvantage compared to ADMM methods for convex optimization

problems. In order to understand why in Step c) only the dual variable λ is updated, the auxiliary optimization problem

$$
\begin{aligned}
Z_\rho(x) \;=\; & \min_z \quad \textstyle\sum_{i=1}^N \left\{ f_i(z_i) + \frac{\rho}{2} \left\| z_i - x_i \right\|_{\Sigma_i}^2 \right\} \\
& \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^N A_i z_i - b = 0 \\ h_i(z_i) \leq 0 \;, \quad i \in \{1, \ldots, N\} \end{array} \right.
\end{aligned}
\tag{6.5}
$$

is introduced. Notice that optimization problems of the form (6.5) are often employed in the field of proximal algorithms, which have been studied extensively in the literature [41, 44]. Nowadays, proximal algorithms are standard tools in convex optimization and are widely used in the field of distributed optimization. In particular, the connections with alternating direction methods are elaborated in [54]. In the following, the proximal optimization problem (6.5) is used as an auxiliary tool for proving convergence of ALADIN. Here, the main idea is to exploit the fact that the optimal solution z^* of the above optimization problem is a strict descent step as we have

$$
\begin{aligned}
\Phi(x) \;&\geq\; \sum_{i=1}^N f_i(x_i) + \overline{\lambda} \left\| \sum_{i=1}^N A_i x_i - b \right\|_1 + \overline{\kappa} \sum_{i,j} \max\{\, 0,\, (h_i(x_i))_j \,\} \\
&=\; \sum_{i=1}^N \left\{ f_i(x_i) + \frac{\rho}{2} \left\| x_i - x_i \right\|_{\Sigma_i}^2 \right\} + \overline{\lambda} \left\| \sum_{i=1}^N A_i x_i - b \right\|_1 + \overline{\kappa} \sum_{i,j} \max\{\, 0,\, (h_i(x_i))_j \,\} \\
&\geq\; \sum_{i=1}^N \left\{ f_i(z_i^*) + \frac{\rho}{2} \left\| z_i^* - x_i \right\|_{\Sigma_i}^2 \right\} \;=\; \Phi(z^*) + \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| z_i^* - x_i \right\|_{\Sigma_i}^2 \right\} \;,
\end{aligned}
$$

which implies

$$
\Phi(x) - \Phi(z^*) \;\geq\; \sum_{i=1}^N \left\{ \frac{\rho}{2} \left\| z_i^* - x_i \right\|_{\Sigma_i}^2 \right\} + \overline{\lambda} \underbrace{\left\| \sum_{i=1}^N A_i z_i^* - b \right\|_1}_{=0} \;.
\tag{6.6}
$$

Now, clearly, Step c) of Algorithm 3 ensures that if there are no updates of the primal variable, the dual iterate λ converges for any choice of $\mu > 0$ to a limit point $\lambda^* \in \operatorname{argmax}_\lambda \; V_\rho(\bar{x}, \lambda)$ as long as the maximum exists. Fortunately, for sufficiently large ρ and under certain regularity assumptions, it can be shown that this maximum always exists and, even more importantly, that there is no duality gap. In order to prove this statement, the projection problem

$$
\begin{aligned}
& \min_\zeta \quad \textstyle\sum_{i=1}^N \frac{1}{2} \left\| \zeta_i - x_i \right\|_{\Sigma_i}^2 \\
& \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^N A_i \zeta_i - b = 0 \\ h_i(\zeta_i) \leq 0 \;, \quad i \in \{1, \ldots, N\} \end{array} \right.
\end{aligned}
\tag{6.7}
$$

is introduced. Its optimal solution ζ^* can be interpreted as the projection of the point x_i onto the feasible set of the original optimization problem (3.1). Next, if the objective of the auxiliary optimization (6.5) is scaled with $\frac{1}{\rho}$, it becomes clear that Problem (6.5) and (6.7) are equivalent in the limit for $\rho \to \infty$. More generally, Problem (6.5) (with rescaled objective) can be obtained from Problem (6.7) by perturbing the objective function with a term of order $\frac{1}{\rho}$.

LEMMA 1. *Let the functions f_i and h_i be twice continuously differentiable with the second order derivative of the f_is being bounded on the feasible sets $\mathcal{F}_i = \{\, y_i \,|\, h_i(y_i) \leq 0 \,\}$ and let the matrices Σ_i be positive definite. If the minimizer of Problem* (6.7) *is a regular KKT point, then we have*

$$
\sup_\lambda \; V_\rho(\bar{x}, \lambda) \;=\; Z_\rho(\bar{x}) \;,
$$

for all sufficiently large ρ, i.e., there is no duality gap.

Proof. Since the second order derivatives of functions f_i and h_i are assumed to be bounded, it is known that the optimal solutions of Problem (6.5) are contained in a small open neighborhood of the projection point ζ^*, whose width scales with $\frac{1}{\rho}$, i.e., this neighborhood can be made arbitrarily small by choosing a sufficiently large ρ. For the following discussion, the shorthand

$$F_\rho(z) \;=\; \sum_{i=1}^{N} \Big\{ f_i(z_i) + \frac{\rho}{2} \left\| z_i - x_i \right\|_{\Sigma_i}^2 \Big\}$$

is introduced. Next, the assumption that ζ^* is a regular KKT point implies that there exist matrices $Q_1 \in \mathbb{R}^{n \times m}$ and $Q_2 \in \mathbb{R}^{n \times (n-m)}$ with $[A_1, \ldots, A_N] Q_2 = 0$ as well as a twice continuously differentiable function $\xi : \mathbb{R}^{n-m-n_a} \to \mathbb{R}^{n-m}$ such that the optimization problem

$$\begin{aligned} Z_\rho(x) \;=\; & \min\nolimits_{z_1, z_2} \quad && F_\rho(Q_1 z_1 + Q_2 \xi(z_2)) \\ & \text{s.t.} && [A_1, \ldots, A_N] Q_1 z_1 - b = 0 \\ & && \| Q_1 z_1 + Q_2 \xi(z_2) - x \|_2 \leq \tfrac{\tau_1}{\rho} \end{aligned} \tag{6.8}$$

is equivalent to the Problem (6.5) via the variable substitution $z = Q_1 z_1 + Q_2 \xi(z_2)$ and a sufficiently large constant $\tau_1 < \infty$. Notice that the above statement follows immediately from a generalized version of the implicit function theorem as discussed in [7] recalling that we assume that ζ^* is a regular KKT point such that the active inequality constraints can locally be eliminated. Next, it is easy to verify that the optimization problem (6.8) is strictly convex in (z_1, z_2) if ρ is sufficiently large. This follows from the fact that ξ is twice continuously differentiable and that the Hessian matrix of the objective function of Problem (6.8) is given by

$$\nabla^2_{(z_1,z_2)} F_\rho(Q_1 z_1 + Q_2 \xi(z_2)) = \mathcal{M}(z_1, z_2) + \rho \sum_{i=1}^{N} \begin{pmatrix} Q_1^\mathsf{T} \\ \nabla_{z_2} \xi(z_2) Q_2^\mathsf{T} \end{pmatrix}_i \Sigma_i \begin{pmatrix} Q_1^\mathsf{T} \\ \nabla_{z_2} \xi(z_2) Q_2^\mathsf{T} \end{pmatrix}_i .$$

Here, the matrix valued function $\mathcal{M}(z_1, z_2)$ denotes the terms in the second order derivative expression that can be bounded by a constant independent of ρ. In other words, $\mathcal{M}$ satisfies $\| \mathcal{M}(z_1, z_2) \|_2 \leq \tau_2$ for all feasible points (z_1, z_2) of Problem (6.8) and for a sufficiently large constant $\tau_2 < \infty$, which does not depend on ρ. As the minimizer of Problem (6.7) is assumed to be a regular KKT point, $\nabla_{z_2} \xi(z_2)$ is a full-rank matrix for all feasible points z_2 in Problem (6.8), which implies that $\nabla^2_{(z_1,z_2)} F_\rho(Q_1 z_1 + Q_2 \xi(z_2))$ is a positive definite matrix for all feasible points (z_1, z_2) of Problem (6.8) and all sufficiently large ρ. Thus, Problem (6.8) is strictly convex and, consequently, there is no duality gap. Since this optimization problem is equivalent to the original optimization problem (6.5) we have established the statement of the lemma. □

Notice that the above Lemma has been established in very similar versions in [62, 63] in the context of augmented Lagrangian and proximal operator analysis. Here, it should be noted that Lemma 1 is based on the rather strong regularity assumption that the minimizers of Problem (6.7) are regular KKT points. Generalization of the above Lemma are possible by employing standard analysis techniques from the field of parametric optimization [7]. In [62, 63] such "no-duality-gap statements" are established in the context of augmented Lagrangian functions under much weaker regularity assumptions.

THEOREM 2. *Let Problem* (3.1) *be feasible and bounded from below such that a minimum exists. If the assumptions from Lemma 1 are satisfied, if ρ is sufficiently large, and if the line search parameters are adjusted by Algorithm 3, then Algorithm 2 terminates after a finite number of iterations.*

Proof. As Problem (3.1) is assumed to be feasible, all decoupled NLPs as well as the QP are feasible by construction. Consequently, the iterates of Algorithm 2 are well-defined. The proof is in two parts. The first part establishes the fact that Algorithm 2 together with Algorithm 3 applies Step c) of the globalization routine at most for a finite number of steps. The second part establishes that Step a) and b) of Algorithm 3 ensure that Algorithm 2 terminates after a finite number of iterations by using the result from Part 1.

Part 1
Assume that Algorithm 2 together with Algorithm 3 executes Step c) infinitely often, which implies that the primal variable x stops being updated. In this case, the line search in Step c) ensures that the dual iterates λ converge to a local maximum of $V_\rho(x, \cdot)$. Consequently, the considerations from the proof of Lemma 1 imply that the primal solution sequence y converges to the limit point $y^* = z^*$. Thus, the fact that z^* is a strict descent direction (see inequality (6.6)) ensures that for sufficiently large $\rho > 0$ and $\gamma \ll 1$ the strict descent conditions in either Step a) or b) of the above globalization strategy are satisfied. This is a contradiction to the above assumption that the algorithm applies Step c) for an infinite number of iterations.

Part 2
If Algorithm 2 does not terminate after a finite number of steps either Step a) or Step b) of Algorithm 2 are applied infinitely often. This is due to the fact that Part 1 already excludes the case that Step c) is applied infinitely often. Whenever, Step a) or b) is applied the progress difference $\Phi(x) - \Phi(x^+)$ is bounded from below by strictly positive constant. As Φ is bounded from below this is a contradiction. Consequently, Algorithm 2 must terminate after a finite number of steps. □

7. Local Convergence Analysis. This section concerns the local convergence properties of Algorithm 2 under the assumption that the functions f_i are twice continuously differentiable. Here, the main idea is to show that Algorithm 2 inherits the convergence properties from inexact SQP methods if ρ is sufficiently large.

LEMMA 3. *Let the functions f_i, $i \in \{1, \dots, N\}$, be twice continuously differentiable and let the minimizer (x^*, λ^*) of Problem (1.1) be a regular KKT point. Moreover, let $\mathcal{N} \subseteq \mathbb{R}^{N \cdot n} \times \mathbb{R}^m$ be a sufficiently small open set with $0 \in \mathcal{N}$ and let $\rho > 0$ be such that*

$$\forall i \in \{1, \dots, N\}, \qquad \nabla_x^2 \left[f_i(x_i^*) + \kappa_i^\mathsf{T} h_i(x_i^*) \right] + \rho \Sigma_i \succ 0 \, .$$

There exist constants $\chi, \chi_1, \chi_2 < \infty$ such that for every point (x, λ) satisfying the condition $(x - x^*, \lambda - \lambda^*) \in \mathcal{N}$ the decoupled minimization problems

$$\min_{y_i} \; f_i(y_i) + \lambda^\mathsf{T} A_i y_i + \frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2 \qquad s.t. \qquad h_i(y_i) \leq 0$$

have locally unique minimizers $(y_1^\mathsf{T}, \dots, y_N^\mathsf{T})^\mathsf{T} \in \{x^*\} \oplus \chi \mathcal{N}$ that satisfy the inequality

$$\left\| y - x^* \right\| \leq \chi_1 \left\| x - x^* \right\| + \chi_2 \left\| \lambda - \lambda^* \right\| \, .$$

As mentioned earlier, the decomposed optimization problems in Algorithm 2 are very closely related to standard augmented Lagrangian methods and Lemma 3 is in very similar versions well-known in the literature. For example, Bertsekas has analyzed the solutions of the minimizers of augmented Lagrangian functions under small perturbations of the multiplier, see Proposition 4.2.3 in [5]. A very similar analysis can be found in Theorem 17.6 of [52]. However, for the sake completeness we provide a concise proof of Lemma 3:

Proof. The Hessian matrices $\nabla^2 \left[f_i(x_i^*) + \kappa_i^\mathsf{T} h_i(x_i^*) \right] + \rho \Sigma_i$ of the decoupled optimization problems

$$\xi_i(x,\lambda) \;=\; \underset{\xi_i}{\operatorname{argmin}}\; f_i(\xi_i) + \lambda^\mathsf{T} A_i \xi_i + \frac{\rho}{2} \left\| \xi_i - x_i) \right\|_{\Sigma_i}^2 \qquad \text{s.t.} \qquad h_i(\xi) \leq 0 \tag{7.1}$$

are strictly positive for all (x,λ) in a small neighborhood of (x^*,λ^*). Thus, the parametric minimizers $\xi_i(x,\lambda)$ are locally well-defined and continuously differentiable functions in this neighborhood, because (x^*,λ^*) is assumed to be a regular KKT point. Moreover, the equation $\xi_i(x^*,\lambda^*) = x_i^*$ holds. This follows from the first order KKT conditions of Problem (7.1). The statement of the lemma is now an immediate consequence, because the finite constants χ_1 and χ_2 satisfy $\chi_1 > \left\| \frac{\partial \xi_i}{\partial x_i}(x^*,\lambda^*) \right\|$ and $\chi_2 > \left\| \frac{\partial \xi_i}{\partial \lambda}(x^*,\lambda^*) \right\|$, respectively. □

Local convergence rate estimates from the field of standard SQP methods can be applied easily to the full-step variant of Algorithm 2 with $\alpha_1 = \alpha_2 = \alpha_3 = 1$ as long as the requirements of Lemma 3 are satisfied. For example, if the exact Hessian and constraint Jacobian $H_i = H_i^*$, $C_i = C_i^*$ are used and if $\frac{1}{\mu} < \mathbf{O}(\|y - x^*\|)$ when solving the coupled QP problem (3.3), then the inequalities

$$\left\| x^+ - x^* \right\| \;\leq\; \frac{\omega}{2} \left\| y - x^* \right\|^2 \quad \text{and} \quad \left\| \lambda^+ - \lambda^* \right\| \;\leq\; \frac{\omega}{2} \left\| y - x^* \right\|^2$$

hold in a neighborhood of the optimal solution for a constant $\omega < \infty$. This statement follows from the fact that the optimality condition for the QP (3.3) can be written in the form

$$\begin{pmatrix} H^* & A^\mathsf{T} & (C^*)^\mathsf{T} \\ A & -\frac{1}{\mu} I & 0 \\ C^* & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta y \\ \lambda^+ - \lambda \\ \kappa_{\mathrm{QP}} \end{pmatrix} = \begin{pmatrix} \nabla_y \sum_{i=0}^N f_i(y_i) + A^\mathsf{T} \lambda \\ b - Ay \\ 0 \end{pmatrix} .$$

In this form it becomes apparent that solving the QP (3.3) is equivalent to applying an inexact Newton step to the optimality conditions of Problem (3.1) for a fixed active set. Since the Hessian matrix H^* and the constraint Jacobian C^* are exact, the only approximation comes from the term $\frac{1}{\mu} I$ perturbing the central block of the KKT-matrix. As we assume $\frac{1}{\mu} < \mathbf{O}(\|y - x^*\|)$, this term converges to zero when approaching the minimizer x^* such that the locally quadratic convergence rate of Newton's method is preserved [52]. Here, it is assumed that (x^*,λ^*) is a regular KKT point, which, in combination with Lemma 3, yields

$$\chi_1 \left\| x^+ - x^* \right\| + \chi_2 \left\| \lambda^+ - \lambda^* \right\| \;\leq\; \frac{(\chi_1 + \chi_2)\omega}{2} \left(\chi_1 \left\| x^+ - x^* \right\| + \chi_2 \left\| \lambda^+ - \lambda^* \right\| \right)^2 .$$

This is sufficient to prove local quadratic convergence of the algorithm as χ_1, χ_2 are strictly positive constants. Similarly, for the case that the matrices $H_i \to H_i^*$, $\mu \to \infty$, and $C_i \to C_i^*$ converge to the exact Hessians and exact constraint Jacobians, respectively, superlinear convergence can be established. In another variant, if H_i and C_i are sufficiently accurate approximations of H_i^* and C_i^*, then Algorithm 2 converges linearly. Methods and algorithms for adjusting the Levenberg-Marquardt regularization $\frac{1}{\mu}$ in such a way that the above convergence statements hold are discussed exhaustively in [22]. Of course, all these local convergence rate statements are at this point less surprising from an "SQP-perspective" in the sense that they are only a simple consequence of the convergence properties of standard or inexact SQP methods, which have been analyzed exhaustively in the existing literature [18]. However, given the fact that other distributed optimization algorithms such as ADMM typically have a linear convergence rate in the convex case and may even be locally divergent in the non-convex case (as in Example 2.1), the convergence properties of ALADIN established above provide a significant improvement on the existing results for these methods. Moreover, the proposed ALADIN algorithm can be interpreted as a bridge between distributed and centralized optimization algorithms that provides a unifying framework for the convergence analysis of these methods.

Finally, it remains to be discussed that the above local convergence analysis is based on the assumption that full-steps are taken in a local neighborhood of the optimal solution. This is the case if Algorithm 3 chooses $\alpha_1 = \alpha_2 = \alpha_3 = 1$. For standard SQP methods a full-step assumption is critical in the sense that it is possible to construct cases for which the favorable local convergence properties of full-step SQP methods are jeopardized by globalization routines that prevent the use of a full-step close to the solution. This phenomenon is known as the Maratos effect [46]. In the context of SQP methods, strategies for avoiding the Maratos effect have been analyzed exhaustively [11, 16, 47]. Notice that there is an important difference between SQP methods and ALADIN, namely, ALADIN solves decoupled NLPs as part of Step 1 of Algorithm 2. Thus, if Algorithm 2 is started at the primal optimal solution x^*, but with a wrong multiplier $\lambda \neq \lambda^*$, the solution y of the decoupled NLPs will in general be different from x^*. As a consequence, Algorithm 2 might choose $x^+ \neq x^*$ if a full step with $\alpha_1 = \alpha_2 = \alpha_3 = 1$ is applied. Of course, this behavior is not desired, as this would mean that the algorithm would choose $x^+ \neq x^*$, although the initialization $x = x^*$ was already optimal. Fortunately, the globalization routine from the previous section prevents this undesirable behavior. In fact, if we start at $x = x^*$ but $\lambda \neq \lambda^*$, the globalization Step c) ensures that only the dual variable λ is updated. In this case, the dual variables converge with the desired rate to the optimal solution. This is guaranteed by the following theorem, which shows that Algorithm 3 chooses $\alpha_3 = 1$ if λ is in a local neighborhood of regular dual solution λ^* as long as H_i and C_i are sufficiently close to H_i^* and C_i^*. Summarizing the above discussion, ALADIN does not necessarily take full steps close to the optimal solution but there are situations in which this behavior is desired. Thus, the goal of the following convergence analysis is to show that the globalization routine never applies Step b) whenever (x, λ) is in a sufficiently small local neighborhood to a regular KKT point (x^*, λ^*). The corresponding result is summarized in the theorem below.

THEOREM 4. *Let (x^*, λ^*) be a regular KKT point of Problem (3.1) as well as $H_i = H_i^*$ and $C_i = C_i^*$ ($H_i \to H_i^*$ and $C_i \to C_i^*$). If the conditions from Lemma (3) are satisfied and if (x, λ) is in a sufficiently small neighborhood of (x^*, λ^*), Algorithm 2 in combination with Algorithm 3 chooses in every step either $\alpha_1 = \alpha_2 = \alpha_3 = 1$ or $\alpha_1 = \alpha_2 = 0$ but $\alpha_3 = 1$.*

Proof. The local convergence analysis of inexact SQP methods [18] ensures that

$$\Phi(x^+) \ \leq \ \Phi(y)$$

for $\alpha_1 = \alpha_2 = \alpha_3$ assuming that (x, λ) is in a sufficiently small neighborhood of a regular minimizer. Thus, whenever the conditions of Step b) of the globalization routine from Section 6 are satisfies the conditions from Step a) are satisfied, too. This implies that the globalization routine never applies Step b) in a local neighborhood of an optimal solution proving the statement of the theorem. □

In order to avoid confusion about this result, notice that for $x \neq x^*$, Algorithm 3 applies the globalization Step c) at most for a finite number of iterates. As the above theorem excludes that Step b) is applied close to the solution, Algorithm 3 must apply a full step after every finite number of iterations. That is, the norm $\|x - x^*\|$ does not necessarily contract quadratically (superlinearly) in every step, but it does after a finite number of iterations.

8. Numerical Case Study. This section illustrates the practical performance of ALADIN versus conventional SQP methods. A circular sensor network localization problem with $N = 25000$ sensors is considered. Let $\chi_i \in \mathbb{R}^2$ denote the unknown position of the ith sensor, $i \in \{1, 2, \ldots, 25000\}$, and $\eta_i \in \mathbb{R}^2$ a measurement of χ_i. The variance of the measurement error $\eta_i - \chi_i$ is assumed to have a Gaussian distribution with given variance $\sigma_i^2 I_{2\times 2}$. Moreover, let $\bar{\eta}_i \in \mathbb{R}$ denote the measurement of the distance between the sensor with index i and the sensor with index $i + 1$. The associated measurement error, given by

$$\|\chi_i - \chi_{i+1}\|_2 - \bar{\eta}_i \ ,$$

is assumed to have a Gaussian distribution with given variance $\bar{\sigma}_i^2$. Here, the notation $\chi_{N+1} = \chi_1$ is introduced, i.e., the distance between the Nth sensor and the first sensor is measured, too. In order to model this problem, the optimization variable

$$x_i = \left(\chi_i^\mathsf{T}\,,\, \zeta_i^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^4$$

is introduced. Here, ζ_i is the ith sensor's estimate of the position of its neighbor. The coupling constraints can be written in the form

$$\forall i \in \{1, \ldots, N\}, \qquad \zeta_i = \chi_{i+1}\ .$$

This is equivalent to enforcing the linear coupling equation $\sum_{i=1}^N A_i x_i = 0$ with

$$A_1 = \begin{pmatrix} 0 & I \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ -I & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -I & 0 \\ 0 & I \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -I & 0 \\ 0 & I \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \ldots\ , A_N = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ -I & 0 \\ 0 & I \end{pmatrix}.$$

In order to find the maximum likelihood estimates of all sensor positions, the objective of sensor i is defined as

$$f_i(x_i) = \frac{1}{4\sigma_i^2}\|\chi_i - \eta_i\|_2^2 + \frac{1}{4\sigma_{i+1}^2}\|\zeta_i - \eta_{i+1}\|_2^2 + \frac{1}{2\bar{\sigma}_i^2}\left(\|\chi_i - \zeta_i\|_2 - \bar{\eta}_i\right)^2$$

for all $i \in \{1, \ldots, N\}$. Here, the definition $\eta_{N+1} = \eta_1$ is used recalling that the first sensor is regarded as a neighbor of the last sensor. Notice that the function f_i is a nonlinear least-squares objective term, which could alternatively be written in the form $f_i(x_i) = \frac{1}{2}\|F_i(x_i)\|_2^2$ for appropriately defined functions F_i. This implies in particular that Gauss-Newton Hessian approximations of the form

$$H_i = \nabla_{x_i} F(x_i) \nabla_{x_i} F(x_i)^\mathsf{T} \approx \nabla_{x_i}^2 f_i(x_i)$$

can be computed easily. In order to make the problem slightly more challenging, additional inequality constraints of the form

$$h_i(x_i) = \left(\|\chi_i - \zeta_i\|_2 - \bar{\eta}_i\right)^2 - \bar{\sigma}_i^2 \leq 0$$

are introduced. These inequalities model additional information about the maximum error of the distance measurements. In this case study, measurements of the form

$$\eta_i = \begin{pmatrix} N\cos\left(\frac{2i\pi}{N}\right) \\ N\sin\left(\frac{2i\pi}{N}\right) \end{pmatrix} + e_i \qquad \text{and} \quad \bar{\eta}_i = 2N\sin\left(\frac{\pi}{N}\right) + d_i$$

are constructed, where e_i and d_i are randomly generated measurement errors using the above mentioned Gaussian probability distributions with $\sigma_i = \bar{\sigma}_i = 10$ for all $i \in \{1, \ldots, N\}$. Figure 1 shows the convergence of ALADIN versus conventional SQP. The SQP method as well as ALADIN have been implemented by using the programming language `JULIA`. Both methods have been started with the same initial values and both methods use the above Gauss-Newton Hessian approximation. ALADIN uses the constraint Jacobian approximations $C_i = 0$ as well as the constant penalty parameter $\rho = 1$. Both the SQP method and ALADIN have a linear convergence rate, since Gauss-Newton Hessian matrix approximations are used. Notice that the optimization problem comprises $4N = 10^5$ primal optimization variables as well as $2N = 7.5 \cdot 10^4$ dual variables. A major advantage of ALADIN compared to conventional SQP with distributed linear algebra

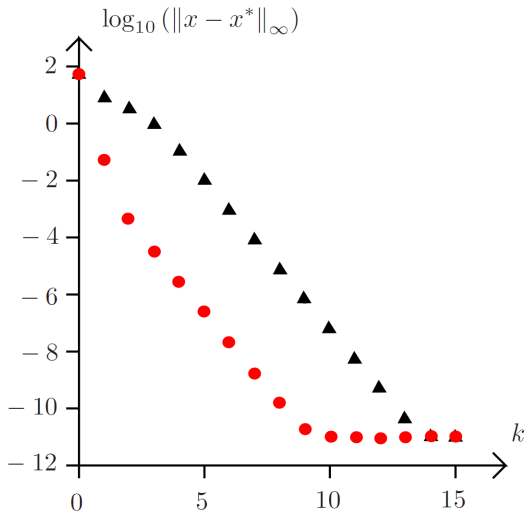

FIG. 1. *The norm $\|x - x^*\|_\infty$ of the difference of the current primal iterate x and the optimal solution x^* versus the iteration number for SQP (black triangles) and ALADIN with $\rho = 1$ (red circles) applied to a sensor network localization problem with $N = 25000$ sensors.*

can be observed during the first three iterates after which ALADIN achieves a primal accuracy of $\|x - x^*\|_\infty < 10^{-3}$, while the corresponding SQP iterate satisfies $\|x - x^*\|_\infty > 1$. The numerical conditioning of SQP and ALADIN is very similar: in this implementation both methods cannot achieve accuracies less than $\|x - x^*\|_\infty \approx 10^{-11}$ as numerical errors cannot be avoided on machines with finite precision arithmetic. Notice that SQP solves a coupled inequality constrained QP during each iteration. This is more expensive than solving the coupled equality constrained QP during the ALADIN iteration. For this particular optimization problem, ALADIN converges faster than conventional SQP in terms of both run-time and number of iterations.

9. Conclusions. This paper introduced an augmented Lagrangian based alternating direction inexact Newton algorithm, named ALADIN, that can solve large-scale and potentially distributed optimization problems of the form (1.1). Here, the main contribution is that this algorithm has been established to converge to local minimizers even if the objective and constraint functions are non-convex. This is in contrast to state-of-the-art distributed optimization algorithms such as dual decomposition methods or ADMM that are in general only applicable to convex optimization problems. Moreover, the proposed ALADIN algorithm has desirable local convergence properties that can in this form not be obtained with standard variants of ADMM. We have also discussed the connections of the proposed algorithms with centralized optimization method such as SQP and augmented Lagrangian methods, which leads to a better understanding of similarities and differences between existing distributed and centralized optimization algorithms. A numerical case study for a sensor network localization problem indicates that ALADIN performs well in practice. The advantages of ALADIN compared to conventional SQP methods have been illustrated numerically.

Acknowledgements. This research was supported by National Natural Science Foundation China (NSFC), Nr. 61473185, ShanghaiTech University, Grant-Nr. F-0203-14-012, as well as by KU Leuven via OPTEC (Optimization in Engineering Center), the Belgian Federal Science Policy Office via DYSCO (Dynamical systems, control and optimization); and the EU via FP7-ITN-TEMPO (607957), ERC HIGHWIND (259166), and H2020-ITN-AWESCO (642682).

REFERENCES

[1] R. Andreani, E.G. Birgin, J.M. Martinez and M.L. Schuverdt. On Augmented Lagrangian methods with general lower-level constraints. SIAM Journal on Optimization, 18:1286–1309, 2007.

[2] D.P. Bertsekas. Convexification procedures and decomposition methods for non convex optimization problems. Journal of Optimization Theory and Applications, 29:169–197, 1979.

[3] D.P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, 1982.

[4] D.P. Bertsekas, J.N. Tsitsiklis. Parallel and distributed computation: Numerical Methods. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] D.P. Bertsekas. Nonlinear Programming. Athena Scientific, second ed., 1999.

[6] P.T. Boggs, J.W. Tolle. Sequential quadratic programming. Acta Numerica, 4, pp:1–51, 1996.

[7] J.F. Bonnans, A. Shapiro. Perturbation Analysis of Optimization Problems. Springer, Heidelberg, 2000.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3:1–122, 2011.

[10] A. Buttari, J. Langou, J. Kurzak, and J.J. Dongarra. A class of parallel tiled linear algebra algorithms for multicore architectures. Parellel Comput. Syst. Appl., 35:3853, 2009.

[11] R. Chamberlain, C. Lemaréchal, H. C. Pedersen, and M. J. D. Powell. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming*, 16:1–17, 1982.

[12] G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. Mathematical Programming, 64:81–101, 1994.

[13] J. Choi, J.J. Dongarra, L.S. Ostrouchov, A.P. Petitet, D.W. Walker, and R.C. Whaley. Design and implementation of the ScaLAPACK LU, QR, and Cholesky factorization routines. Sci. Program., 5(3):173–184, 1996.

[14] H. Chung, E. Polak, S. Sastry. An External Active-Set Strategy for Solving Optimal Control Problems. IEEE Transactions on Automatic Control, 54(5): 1129– 1133,2009.

[15] G. Cohen. Decomposition et coordination en optimisation deterministe, differentiable et non differentiable. These d'Etat, Univ. Paris Dauphine (France), 1984.

[16] T.F. Coleman and A.R. Conn. Non-linear programming via an exact penalty function: Asymptotic Analysis. *Mathematical Programming*, 24:123–136, 1982.

[17] A.R. Conn, N.I.M. Gould, and P.L. Toint. LANCELOT: a FORTRAN package for large-scale nonlinear optimization (Release A), no. 17 in Springer Series in Computational Mathematics, Springer-Verlag, New York, 1992.

[18] M. Diehl, A. Walther, H. G. Bock, and E. Kostina. An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. Optimization Methods and Software, 25:531–552, 2010.

[19] J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Programming, 55:293–318, 1992.

[20] J. Eckstein and M.C. Ferris. Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control. INFORMS Journal on Computing, 10:218–235, 1998.

[21] H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Operations Research, 11(3):399–417, 1963.

[22] J. Fan, Y.X. Yuan. On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption. Computing, 74(1):23–39, 2005.

[23] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. International Journal of Robust and Nonlinear Control, 18(8):816–830, 2008.

[24] H.J. Ferreau, A. Kozma, and M. Diehl. A Parallel Active-Set Strategy to Solve Sparse Parametric Quadratic Programs arising in MPC. Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, 2012.

[25] J.V. Frasch, S. Sager, M. Diehl. A Parallel Quadratic Programming Method for Dynamic Optimization Problems. Mathematical Programming Computation, DOI: 10.1007/s12532-015-0081-7, 2015.

[26] J.V. Frasch. qpDUNES Website. `http://mathopt.de/qpDUNES`.

[27] M.P. Friedlander, M.A. Saunders. A globally convergent linearly constrained Lagrangian method for nonlinear optimization. SIAM Journal on Optimization, 15(3):863–897, 2005.

[28] D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Computers and Mathematics with Applications, 2:17–40, 1976.

[29] P.E. Gill, W. Murray, M.A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization SIAM Journal on Optimization, 12(4):979–1006, 2002.

[30] P.E. Gill and D.P. Robinson. A primal-dual augmented Lagrangian. Computational Optimization and Applications, 51(1), 1–25, 2010.

[31] P.E. Gill and D.P. Robinson. A globally convergent stabilized SQP method. SIAM Journal on Optimization, 23(4):1983–2010, 2013.

[32] P. Giselsson and A. Rantzer. Distributed Model Predictive Control with suboptimality and stability guarantees. Proceedings of the 49th IEEE Conference on Decision and Control (CDC), 2010.

[33] P. Giselsson, M. Dang Doan, T. Keviczky, B. De Schutter, A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. Automatica 49(3):829-833, 2013.

[34] R. Glowinski, A. Marrocco. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite, d'une classe de problems de Dirichlet non lineares. Revue Francaise d'Automatique, Informatique, et Recherche Operationelle, 9:41–76, 1975.

[35] T. Goldstein, B. O´Donoghue, S. Setzer. Fast Alternating Direction Optimization Methods. Tech. Report, Department of Mathematics, University of California, Los Angeles, USA, 2012.
[36] N.I.M. Gould, D. Orban, and P. Toint. GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. ACM Trans. Math. Software 29(4):353-372, 2004.
[37] A. Hamdi, S.K. Mishra. Decomposition Methods based on Augmented Lagrangian: a Survey. In Topics in Nonconvex Optimization. Mishra, S.K., Chapter 11, 175–204, 2011.
[38] S.P. Han. A globally convergent method for nonlinear programming. Journal of Optimization Theory and Applications, 22:297309, 1977.
[39] M.R. Hestenes. Multiplier and gradient methods. Journal of Optimization Theory and Applications, 4:302–320, 1969.
[40] B. Houska and M. Diehl. A quadratically convergent inexact SQP method for optimal control of differential algebraic equations. Optimal Control Applications & Methods, John Wiley & Sons, 34, pp:396–414, 2012.
[41] A. Iusem. Augmented Lagrangian methods and proximal point methods for convex optimization. Investigación Operativa, 8:11–49, 1999.
[42] A. Kozma, J. Frasch, M. Diehl. A Distributed Method for Convex Quadratic Programming Problems Arising in Optimal Control of Distributed Systems. In Proceedings of the 52nd IEEE Conference on Decision and Control, pp:1526–1531, 2013.
[43] A. Kozma, C. Conte, M. Diehl. Benchmarking Large Scale Distributed Convex Quadratic Programming Algorithms. Optimization Methods and Software, 30(1):191–214, 2015.
[44] B. Lemaire. The proximal algorithm. International Series of Numerical Mathematics, pp. 7387, 1989.
[45] W. Li and J. Swetits. A new algorithm for solving strictly convex quadratic programs. SIAM Journal of Optimization, 7(3):595–619, 1997.
[46] N. Maratos. *Exact penaly function algorithms for finite-dimensional and control optimization problems*. PhD thesis, Imperial College, London, 1978.
[47] D. Q. Mayne and E. Polak. A quadratically convergent algorithm for solving infinite dimensional inequalities. *J. Appl. Math. and Optimization*, 9:25–40, 1982.
[48] I. Necoara, J.A.K. Suykens. Application of a smoothing technique to decomposition in convex optimization. IEEE Transactions on Automatic Control, 53(11):2674–2679, 2008.
[49] I. Necoara, D. Doan, J.A.K. Suykens. Application of the proximal center decomposition method to distributed model predictive control. In Proceedings of the 47th IEEE Conference on Decision and Control, pp:2900-2905, 2008.
[50] I. Necoara, C. Savorgnan, Q. Tran Dinh, J.A.K. Suykens, M. Diehl. Distributed nonlinear optimal control using sequential convex programming and smoothing techniques. In Proceedings of the 48th IEEE Conference on Decision and Control, pp:543–548, 2009.
[51] I. Necoara and J.A.K. Suykens. Interior-point Lagrangian decomposition method for separable convex optimization. J. Optim. Theory and Appl., 2009.
[52] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.
[53] B. O'Donoghue, G. Stathopoulos, S. Boyd. A Splitting Method for Optimal Control. IEEE Transactions on Control Systems Technology, 21(6):2432–2442, 2013.
[54] N. Parikh and S. Boyd. Proximal Algorithms. Foundations and Trends in Optimization, 1(3):123–231, 2013.
[55] M.J.D. Powell. A method for nonlinear constraints in minimization problems. In Optimization, (R. Fletcher, ed.), Academic Press, 1969.
[56] M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In Numerical Analysis Dundee 1977, G.A. Watson, ed., pp:144–157, Springer Verlag, Berlin, 1977.
[57] M.J.D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In Nonlinear Programming 3, Academic Press, pp:27–63, New York and London, 1978.
[58] A. Rantzer. Dynamic dual decomposition for distributed control. In Proceedings of the 2009 American Control Conference, pp:884–888, 2009.
[59] S. Richter, M. Morari, C.N. Jones. Towards computational complexity certification for constrained MPC based on Lagrange Relaxation and the fast gradient method. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011.
[60] R.T. Rockafellar. Convex Analysis. Princeton Mathematics Series, 28, Princeton University Press, 1970.
[61] K. Schittkowski. An active set strategy for solving optimization problems with up to 200,000,000 nonlinear constraints. Applied Numerical Mathematics, 59(12):2999–3007, 2009.
[62] A. Shapiro, J. Sun. Some Properties of the Augmented Lagrangian in Cone Constrained Optimization. Mathematics of Operations Research, 29(3):479–491, 2004.
[63] J. Rückmann, A. Shapiro. Augmented Lagrangians in semi-infinite programming. Mathematical Programming, Series B, 116:499–512, 2009.
[64] G. Stephanopoulos, A.W. Westerberg. The use of Hestenes's method of multipliers to resolve dual gaps in engineering system optimization. Journal of Optimization Theory and Applications, 15:285–309, 1975.
[65] F. Song, A. YarKhan, J. Dongarra. Dynamic task scheduling for linear algebra algorithms on distributed memory multicore systems. In Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, pages 1–11, New York, NY, USA, 2009.
[66] A. Tanikawa, H. Mukai. A new technique for non convex primal-dual decomposition of a large-scale separable optimization problem. IEEE Transactions on Automatic Control, 30:133–143, 1985.
[67] R.A. Tapia. Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods

and a new implementation. In Nonlinear Programming 3, O. Mangasarian, R. Meyer, S. Robinson, eds, Academic Press, pp:125–164, New York, NY, 1978.
[68] P. Tatjewski, B. Engelmann. Two-level primal-dual decomposition technique for large-scale non convex optimization problems with constraints. Journal of Optimization Theory and Applications, 64:183–205, 1990.
[69] P. Toint. On sparse and symmetric matrix updating subject to a linear equation. *Mathemathics of Computation*, 31(140):954–961, 1977.
[70] Q. Tran-Dinh, I. Necoara, C. Savorgnan, and M. Diehl. An Inexact Perturbed Path-Following Method for Lagrangian Decomposition in Large-Scale Separable Convex Optimization. SIAM J. Optimization, 23(1):95–125, 2013.
[71] Q. Tran Dinh, I. Necoara, and M. Diehl. A Dual Decomposition Algorithm for Separable Nonconvex Optimization Using the Penalty Function Framework In Proceedings of the 52nd IEEE Conference on Decision and Control, pp:2372–2377, December 10-13, 2013.
[72] H. Uzawa. Iterative methods for concave programming. In Studies in Linear and Nonlinear Programming, K. Arrow, L. Hurwicz, and H. Uzawa, Eds., pp:154–165, 1958.
[73] H. Uzawa. Market mechanisms and mathematical programming. Econometrica: Journal of the Econometric Society, 28(4):872–881, 1960.
[74] N. Watanabe, Y. Nishimura, M. Matsubara. Decomposition in large system optimization using the method of multipliers. Journal of Optimization Theory and Applications, 25:181–193, 1978.
[75] R.B. Wilson. A simplicial algorithm for concave programming. Ph.D. thesis, Graduate School of Business Administration, Harvard University, 1963.

Appendix A. Equivalence to Dual Decomposition Methods for $\rho = 0$.

A very important property of Algorithm 2 is that, if we set $\rho = 0$, the iterates of this algorithm are equivalent to a dual decomposition method. This statement might not be obvious immediately, because dual decomposition methods typically apply gradient or inexact Newton updates to the optimality condition for the multiplier λ, also known as price-negotiation steps, which involves communication between the agents. These updates of the dual variable λ are not explicitly highlighted as part of the steps of Algorithm 2. However, a closer look at Steps 4 and 5 of Algorithm 2 reveals that solving the QP (3.3) is nothing but an implicit way to implement an inexact Newton update of the dual variable λ, where the term $\frac{1}{\mu}$ can be interpreted as a Levenberg-Marquardt regularization parameter. In order to elaborate on this aspect, we assume for a moment that the functions f_i are strictly convex and the functions h_i convex. In this case, the objective of the dual optimization problem

$$\max_{\lambda} \; V(\lambda) \qquad \text{with} \qquad V(\lambda) \; = \; \sum_{i=1}^{N} d_i(\lambda) - \lambda^{\mathsf{T}} b \; , \tag{A.1}$$

coincides with the objective value of the original optimization problem (3.1), i.e, we have no duality gap.[5] Here, the functions d_i are for all $i \in \{1, \ldots, N\}$ defined as the optimal values of the decoupled optimization problems

$$d_i(\lambda) \; = \; \min_{y_i} f_i(y_i) + \lambda^{\mathsf{T}} A_i y_i \quad \text{s.t.} \quad h_i(y_i) \; \leq \; 0 \; . \tag{A.2}$$

The functions d_i are once differentiable if the minimizer is unique [4, 60], but they are typically not twice continuously differentiable. Here, the gradient of the function V is given by

$$\nabla V(\lambda) \; = \; \sum_{i=1}^{N} A_i y_i - b \; , \tag{A.3}$$

assuming that y denotes the optimal solution of the decoupled problems (A.2). Thus, one way of implementing dual decomposition methods is by solving the dual optimization problem (A.1) with a semi-smooth Newton method of the form

$$\lambda^{+}_{\text{DD}} \; = \; \lambda - \alpha \left(M - \frac{1}{\mu} I \right)^{-1} \nabla V(\lambda) \; . \tag{A.4}$$

[5] Notice that it is enough to assume convexity of f_i and h_i is sufficient to ensure that there is no duality gap. This is due to the fact that the duality statement is made with respect to the linear equality constraints only; that is, this statement also holds if Slater's constraint qualification for the decoupled inequality constraints is violated [8].

Here, $\alpha \in (0,1]$ is a line-search parameter and $\frac{1}{\mu}$ can be interpreted as a Levenberg-Marquardt regularization parameter, which can be used to ensure that $\lambda_{\mathrm{DD}}^{+}$ leads to a sufficient ascent before updating $\lambda \leftarrow \lambda_{\mathrm{DD}}^{+}$ and continuing with the next iteration. Moreover, M is assumed to be a symmetric and negative semi-definite scaling matrix. For example, the choice $M = -I$ corresponds to a standard gradient (or "steepest ascent") method. However, on the other hand, for all points λ for which the minimizer of Problem (A.2) is a regular KKT point, d_i is twice continuously differentiable and we have

$$\nabla^2 V(\lambda) \;=\; -\sum_{i=1}^{N} \left\{ A_i \left[(H_i^*)^{-1} - (H_i^*)^{-1}(C_i^*)^{\mathsf{T}} \left[C_i^*(H_i^*)^{-1}(C_i^*)^{\mathsf{T}} \right]^{\dagger} C_i (H_i^*)^{-1} \right] A_i^{\mathsf{T}} \right\} \tag{A.5}$$

assuming that the matrices $H_i^* = \nabla^2 \left(f_i(y_i) + \kappa_i^{\mathsf{T}} h_i(y_i) \right)$ denote the exact Hessians and $\left[C_i^*(H_i^*)^{-1}(C_i^*)^{\mathsf{T}} \right]^{\dagger}$ the pseudo inverse of the matrix $C_i^*(H_i^*)^{-1}(C_i^*)^{\mathsf{T}}$. This has for example been established in [25]. Consequently, if we are at such a point where V is twice continuously differentiable, we can set $M = \nabla^2 V(\lambda)$—or at least choose M in such a way that the difference $M - \nabla^2 V(\lambda)$ is small—in order to improve the local convergence rate of the dual updates. Notice that even in the case that we use exact dual Hessians, it is advisable to adjust the Levenberg-Marquardt parameter $\frac{1}{\mu}$ in such a way that the matrix $\nabla^2 V(\lambda) - \frac{1}{\mu} I$ is negative definite and not too ill-conditioned for numerical purposes.[6]

LEMMA 5. *Let the functions f_i and h_i be twice continuously differentiable with the f_is being strictly convex and the h_is being convex. If Algorithm 2 is based on constraint Jacobian approximations $C_i \approx C_i^*$ with full row-rank and positive definite Hessian approximations $H_i \approx H_i^*$ as well $\rho = 0$ and $\alpha_3 = \alpha$, then the iterate λ^+ as computed in Step 5 of Algorithms 2 coincides for all $\mu > 0$ with the iterate $\lambda_{\mathrm{DD}}^{+}$ that is obtained by applying the (inexact) dual Newton step* (A.4) *with*

$$M = -\sum_{i=1}^{N} \left\{ A_i \left[H_i^{-1} - H_i^{-1} C_i^{\mathsf{T}} \left[C_i H_i C_i^{\mathsf{T}} \right]^{\dagger} C_i H_i^{-1} \right] A_i^{\mathsf{T}} \right\} \tag{A.6}$$

for solving the dual optimization (A.1)*, i.e., we have $\lambda_{\mathrm{DD}}^{+} = \lambda^+$.*

Proof. As we assume that the exact Hessians H_i are positive definite, the QP problem (3.3) can be solved via its associated dual optimization problem, given by

$$\begin{aligned} \max_{\lambda_{\mathrm{QP}}} \min_{\Delta y, s} & \sum_{i=1}^{N} \left\{ \frac{\Delta y_i^{\mathsf{T}} H_i \Delta y_i}{2} + g_i^{\mathsf{T}} \Delta y_i + \lambda_{\mathrm{QP}}^{\mathsf{T}} A_i (y_i + \Delta y_i) \right\} + \frac{\mu \left\| s \right\|_2^2}{2} - (\lambda_{\mathrm{QP}} - \lambda)^{\mathsf{T}} s - \lambda_{\mathrm{QP}}^{\mathsf{T}} b \\ \text{s.t.} \;\; & C_i \Delta y_i \;=\; 0 \,, \quad i \in \{1, \ldots, N\} \end{aligned}$$

Clearly, the minimization problem over the slack variable s can be simplified explicitly. Moreover, we directly substitute equation (A.3) and write the above dual QP in the form

$$\begin{aligned} \max_{\lambda_{\mathrm{QP}}} \min_{\Delta y} & \sum_{i=1}^{N} \left\{ \frac{\Delta y_i^{\mathsf{T}} H_i \Delta y_i}{2} + g_i^{\mathsf{T}} \Delta y_i + \lambda_{\mathrm{QP}}^{\mathsf{T}} A_i \Delta y_i \right\} + \lambda_{\mathrm{QP}}^{\mathsf{T}} \nabla V(\lambda) - \frac{1}{2\mu} (\lambda_{\mathrm{QP}} - \lambda)^2 \\ \text{s.t.} \;\; & C_i \Delta y_i \;=\; 0 \,, \quad i \in \{1, \ldots, N\} \,. \end{aligned}$$

Next, due to Step 1 of Algorithm 2 and our assumption $\rho = 0$, we observe that y_i must satisfy the stationarity condition associated with Problem (A.2) given by

$$0 \;=\; \nabla f_i(y_i) + A_i^{\mathsf{T}} \lambda + (C_i^*)^{\mathsf{T}} \kappa_i \;=\; g_i + A_i^{\mathsf{T}} \lambda + C_i^{\mathsf{T}} \kappa_i \,,$$

[6] Recall that the matrix $\nabla^2 V(\lambda)$ is always negative semi-definite as the dual function V is concave. However, $\nabla^2 V(\lambda)$ is not necessarily negative definite.

since $g_i = \nabla f_i(y_i) + (C_i^* - C_i)^\mathsf{T}\kappa_i$. Multiplying this equation by Δy_i^T from the left and exploiting the constraint $C_i \Delta y_i = 0$ allows us to simplify the above QP further, yielding the equivalent form

$$\begin{aligned} \max_{\lambda_{\mathrm{QP}}} \min_{\Delta y} \; & \sum_{i=1}^{N} \left\{ \frac{\Delta y_i^\mathsf{T} H_i \Delta y_i}{2} + (\lambda_{\mathrm{QP}} - \lambda)^\mathsf{T} A_i \Delta y_i \right\} + \lambda_{\mathrm{QP}}^\mathsf{T} \nabla V(\lambda) - \frac{1}{2\mu} \left\| \lambda_{\mathrm{QP}} - \lambda \right\|_2^2 \\ \text{s.t.} \; & C_i \Delta y_i \; = \; 0 \; , \quad i \in \{1, \ldots, N\} \; . \end{aligned}$$

Finally, we solve the remaining minimization problem over Δy explicitly and substitute Equation (A.6), which yields

$$\begin{aligned} \lambda_{\mathrm{QP}} \; &= \; \underset{\lambda_{\mathrm{QP}}}{\operatorname{argmax}} \, \frac{1}{2} (\lambda_{\mathrm{QP}} - \lambda)^\mathsf{T} M (\lambda_{\mathrm{QP}} - \lambda) + \lambda_{\mathrm{QP}}^\mathsf{T} \nabla V(\lambda) - \frac{1}{2\mu} \left\| \lambda_{\mathrm{QP}} - \lambda \right\|_2^2 \\ &= \; \lambda - \left(M - \frac{1}{\mu} I \right)^{-1} \nabla V(\lambda) \; . \end{aligned}$$

Since we define $\lambda^+ = \lambda + \alpha(\lambda_{\mathrm{QP}} - \lambda)$ with $\alpha = \alpha_3$ in Steps 5 and 6 of Algorithm 2, a comparison with Equation (A.4) yields the relation $\lambda^+ = \lambda_{\mathrm{DD}}^+$. □

Notice that, as a consequence of the above lemma, Algorithm 2 is equivalent to dual decomposition if we choose $\rho = 0$, since for this particular choice the decoupled optimization problems from Step 1 of Algorithm 1 coincide with the decoupled optimization problems (A.2). This implies in particular that convergence of Algorithm 2 for $\rho = 0$ and sufficiently small $\alpha_3 = \alpha$ can be established in analogy to dual decomposition methods, as long as we assume that the functions f_i and h_i are strictly convex and convex, respectively. This convergence statement holds independent of how "bad" the approximations H_i and C_i of the Hessian and constraint Jacobian matrices are as long as the matrices H_i are positive definite. The only difference between the proposed ALADIN code and existing dual decomposition methods is the term $\frac{\rho}{2} \left\| y_i - x_i \right\|_{\Sigma_i}^2$ in the objective of the decoupled optimization problems. However, this additional term is important, as it turns out to be the key for transferring the idea of dual decomposition methods to non-convex optimization problems.

Another interesting aspect of Lemma 5 is that the penalty parameter μ, which has previously been interpreted as augmented Lagrangian parameter, can alternatively be interpreted as the inverse of a Levenberg-Marquardt regularization parameter. From this perspective, we may state that a regularization based on augmented Lagrangians corresponds to a dual Levenberg-Marquardt regularization.