ON THE RELATIONSHIP BETWEEN BILEVEL DECOMPOSITION ALGORITHMS AND DIRECT INTERIOR-POINT METHODS

ANGEL–VICTOR DEMIGUEL* AND FRANCISCO J. NOGALES†

Abstract. Engineers have been using *bilevel decomposition algorithms* to solve certain non-convex large-scale optimization problems arising in engineering design projects. These algorithms transform the large-scale problem into a bilevel program with one upper-level problem (the master problem) and several lower-level problems (the subproblems). Unfortunately, there is analytical and numerical evidence that some of these commonly used bilevel decomposition algorithms may fail to converge even when the starting point is very close to the minimizer. In this paper, we establish a relationship between a particular bilevel decomposition algorithm, which only performs one iteration of an interior-point method when solving the subproblems, and a direct interior-point method, which solves the problem in its original (integrated) form. Using this relationship, we formally prove that the bilevel decomposition algorithm converges locally at a superlinear rate. The relevance of our analysis is that it bridges the gap between the incipient local convergence theory of bilevel decomposition algorithms and the mature theory of direct interior-point methods.

Key words. Bilevel programming, decomposition algorithms, interior-point methods, local convergence

AMS subject classifications. 49M27, 90C30, 90C51

1. Introduction. Many optimization problems integrate the objective and constraint functions corresponding to a set of weakly connected systems. One type of connectivity occurs when only a few of the variables, known as *global variables*, are relevant to all systems, while the remainder are local to a single component. Mathematically, these problems may be stated as follows:

$$\begin{array}{cll} \min\limits_{x,y_1,y_2,\cdots,y_N} & F_1(x,y_1)+F_2(x,y_2)+\cdots+F_N(x,y_N) & \\ s.t. & c_1(x,y_1) \geq 0, & \\ & c_2(x,y_2) \geq 0, & \\ & \vdots & \\ & c_N(x,y_N) \geq 0, & \end{array} \tag{1.1}$$

where $x \in \mathcal{R}^n$ are the *global* variables, $y_i \in \mathcal{R}^{n_i}$ are the ith system *local* variables, $c_i(x,y_i) : \mathcal{R}^{n+n_i} \to \mathcal{R}^{m_i}$ are the ith system constraints, and $F_i(x,y_i) : \mathcal{R}^{n+n_i} \to \mathcal{R}$ is the objective function term corresponding to the ith system. Note that while global variables appear in all of the objective function terms and constraints, local variables appear only in the objective function term and constraints corresponding to one of the systems. Decomposition algorithms exploit the structure of problem (1.1) by breaking it into a set of smaller independent subproblems, one per system. Then, they use a so-called *master problem* to coordinate the subproblem solutions and find the overall problem minimizer.

A class of large-scale optimization problems that has been extensively analyzed and may be formulated as problem (1.1) is the *stochastic programming* (SP) problem, see [25, 5]. These problems arise, for instance, when a discrete number of *scenarios* is used to model the uncertainty of some of the problem parameters. Two main types

*London Business School, UK (avmiguel@london.edu).

†Department of Statistics, Universidad Carlos III de Madrid, Spain (FcoJavier.Nogales@uc3m.es).

of *decomposition* approaches have been proposed for the SP problem: cutting-plane methods and augmented Lagrangian methods. Cutting-plane methods use convex duality theory to build a linear approximation to the master problem [3, 36, 22, 31].[1] Augmented Lagrangian approaches, on the other hand, use an estimate of the Lagrange multipliers to decompose the stochastic program into a set of subproblems. Then, the subproblem minimizers are used to update the current estimate of the Lagrange multipliers [11, 30, 32].

In addition to decomposition approaches, a number of *specialized direct methods* have been proposed to solve the SP problem (see [5, Section 5.6] and [4] for two-stage problems, [6, 7] for multistage problems, and [34] for optimal control problems). Rather than breaking the problem into a master problem and a set of independent subproblems, direct methods solve the stochastic program in its original (integrated) form. But they usually exploit the structure of the problem to decompose the linear algebra operations required to compute the search direction for the direct method. Thus, specialized direct methods decompose the problem at the algebra level, rather than at the optimization level like decomposition methods. This usually leads to highly efficient methods, but allows for a lower level of decentralization in the solution than the decomposition approaches mentioned before.

But in this paper we focus on a different class of large-scale optimization problems that may be stated as problem (1.1) and that has remained largely unexplored by the operations research community: the *Multidisciplinary Design Optimization* (MDO) problem, see [1, 12]. MDO problems arise in engineering design projects that require the collaboration of several departments within a company. Each department is usually in charge of the design of one of the systems that compose the overall project. Moreover, the different departments often rely on sophisticated software codes (known as legacy codes) that have been under development for many years and whose method of use is subject to constant modification. Integrating all these codes into a single platform is judged to be impractical.

As a result, a priority when choosing a method to solve MDOs, as opposed to SPs, is that the method must allow for a high degree of decentralization in the solution. This decentralization allows the different departments collaborating on the project to find the overall optimal design while working as independently from each other as possible. This precludes the use of specialized direct methods to solve MDOs, because these methods do not allow for the same degree of modularity that decomposition algorithms provide.

At first sight, it may seem reasonable to apply the cutting-plane or augmented Lagrangian decomposition methods available for SP to decompose the MDO problem. Unfortunately, this may not be a good idea. While most real-world stochastic programs are linear, or, at least, convex, most real-world MDO problems are *nonconvex* nonlinear problems. This precludes the use of cutting-plane methods, which rely heavily on convex duality theory, to solve MDO problems. One may feel tempted to use augmented Lagrangian methods to decompose the MDO problem. Unfortunately, the convergence theory available for them applies only to convex problems (see [30]). Moreover, it is well-known that augmented Lagrangian methods may converge slowly in practice (see [24, 10, 16]).

Pressed by the need to solve MDO problems in a truly decentralized manner, engineers have turned to *bilevel decomposition algorithms* [35, 8, 16]. Once decomposed

[1] Another decomposition approach related to cutting-plane methods is bundle based decomposition [28, 29].

into a master problem and a set of subproblems, the MDO problem becomes a particular type of a *bilevel program* [26, 33, 17]. Just as bilevel programming methods, bilevel decomposition algorithms apply nonlinear optimization techniques to solve both the master problem and the subproblems. At each iteration of the algorithm solving the master problem, each of the subproblems is solved, and their minimizers used to compute the master problem derivatives and their associated Newton direction.

The local convergence properties of bilevel decomposition algorithms are very important because, although engineers can usually find good starting points for MDO problems, it is crucial that, once in a neighborhood of the minimizer, the iterates generated by the decomposition algorithm converge quickly. Unfortunately, there is analytical and numerical evidence that certain commonly used bilevel decomposition algorithms may fail to converge even when the starting point is very close to the minimizer [2, 14]. Moreover, although there are some local convergence proofs for certain bilevel decomposition algorithms that solve the subproblems exactly [16], it is safe to say that the local convergence theory of bilevel decomposition algorithms is not nearly as satisfactory as that of direct interior-point methods. In this paper, we establish a relationship between a particular bilevel decomposition algorithm, which only takes one iteration of an interior-point method to solve the subproblems, and a direct interior-point method, which solves the problem in its original (integrated) form. We use this relationship to derive a Gauss-Seidel iteration that ensures the decomposition algorithm achieves superlinear convergence.

Our contribution is twofold. Firstly, our local convergence analysis bridges the gap between the incipient local convergence theory of bilevel decomposition algorithms [2, 16] and the mature local convergence theory of direct interior-point methods [27, 18, 39, 23]. Secondly, we show that bilevel decomposition algorithms that do not solve the subproblems exactly (or only take one step on the subproblems) are viable at least from a local convergence point of view. As a result, we hope our work will encourage researchers and practitioners alike to design and apply other bilevel decomposition approaches based on *inaccurate* subproblem solutions.

The paper is organized as follows. Section 2 describes how a direct interior-point method can be used to solve the MDO problem in its original (integrated) form. In Section 3, we describe a particular bilevel decomposition algorithm that only takes one iteration on the subproblems and analyze its relationship to direct interior-point methods. We use this relationship in Section 4 to show how a Gauss-Seidel iteration can be used to ensure the decomposition algorithm converges locally at a superlinear rate. Section 5 presents some numerical experiments and finally, Section 6 states some conclusions.

2. A Direct Interior-Point Method. In this section, we describe how a direct primal-dual interior-point method [9, 18, 20, 21, 37] can be applied to solve problem (1.1) in its original (integrated) form. In doing so, we introduce notation that will help to understand the decomposition algorithm discussed in Section 3. To facilitate the exposition and without loss of generality, herein we consider the following simplified problem composed of only one system:

$$\begin{array}{ll} \underset{x,y,r}{\text{minimize}} & F(x,y), \\ \text{subject to} & c(x,y) - r = 0, \\ & r \geq 0, \end{array} \tag{2.1}$$

where $x \in \mathcal{R}^{n_x}$ are the *global* variables, $y \in \mathcal{R}^{n_y}$ are the *local* variables, $r \in \mathcal{R}^m$ are the *slack* variables, and $c(x,y) : \mathcal{R}^{n_x+n_y} \to \mathcal{R}^m$ and $F(x,y) : \mathcal{R}^{n_x+n_y} \to \mathcal{R}$ are

smooth functions. Note that, in addition to considering only one system, we have introduced slack variables so that only equality constraints and nonnegativity bounds are present.

2.1. The perturbed KKT conditions. The perturbed KKT conditions for problem (2.1) are derived from the classical logarithmic barrier problem. We follow the barrier approach because it is useful in the development of the proposed decomposition algorithm (for other interpretations see [18, 38]). The logarithmic barrier problem is obtained from problem (2.1) by introducing barrier terms in order to remove the nonnegativity bounds. The result is the following *barrier problem*:

$$\begin{array}{ll} \underset{x,y,r}{\text{minimize}} & F(x,y) - \mu \sum_{i=1}^{m} \log(r_i) \\ \text{subject to} & c(x,y) - r = 0, \end{array} \tag{2.2}$$

where $r > 0$ and μ is the barrier parameter.

Given a suitable constraint qualification holds, a minimizer to problem (2.2) must satisfy the *perturbed KKT conditions*:

$$g(\mu) \equiv \begin{pmatrix} \nabla_x F(x,y) - \nabla_x c(x,y)^T \lambda \\ \nabla_y F(x,y) - \nabla_y c(x,y)^T \lambda \\ -\sigma + \lambda \\ -c(x,y) + r \\ -R\sigma + \mu\, e_m \end{pmatrix} = 0, \tag{2.3}$$

where $R = \operatorname{diag} r$, $\lambda \in \mathcal{R}^m$ are the Lagrange multipliers, $\sigma \in \mathcal{R}^m$ are the dual variables, $e_m \in \mathcal{R}$ is the vector whose components are all ones, and the variables r, λ, σ are strictly positive.

2.2. The Newton search direction. In essence, a primal-dual interior-point method consists of the application of a modified Newton's method to find a solution to the nonlinear system (2.3). At each iteration, the Newton search direction is computed by solving a linearization of system (2.3). Then, a step size is chosen such that all nonnegative variables remain strictly positive.

Before we state the Newton linear system, it is useful to realize that the problem variables can be split into two different components: the global component x and the local component $\hat{y} = (y, r, \lambda, \sigma)$. Likewise, $g(\mu)$[2] can also be split into two different components

$$g(\mu) = \begin{pmatrix} g_1 \\ g_2(\mu) \end{pmatrix} = 0, \tag{2.4}$$

where

$$g_1 = \nabla_x F(x,y) - \nabla_x c(x,y)^T \lambda, \tag{2.5}$$

and

$$g_2(\mu) = \begin{pmatrix} \nabla_y F(x,y) - \nabla_y c(x,y)^T \lambda \\ -\sigma + \lambda \\ -c(x,y) + r \\ -R\sigma + \mu\, e_m \end{pmatrix}. \tag{2.6}$$

[2]Note that, to simplify notation, we have omitted the dependence of g on the variables and multipliers.

Let $w_k = (x_k, \hat{y}_k)$ be the current estimate of the global and local components. Then, the *Newton search direction*, $\Delta w_k^N = (\Delta x_k^N, \Delta \hat{y}_k^N)$, is the solution to the following system of linear equations:

$$\begin{pmatrix} W_k & -\widehat{A}_k^T \\ -\widehat{A}_k & M_k \end{pmatrix} \begin{pmatrix} \Delta x_k^N \\ \Delta \widehat{y}_k^N \end{pmatrix} = - \begin{pmatrix} g_{1,k} \\ g_{2,k}(\mu_k) \end{pmatrix}, \tag{2.7}$$

where $g_{1,k}$ and $g_{2,k}$ denote the functions g_1 and g_2 evaluated at w_k, $W_k = \nabla_x g_{1,k}$, $\widehat{A}_k = -\nabla_x g_{2,k}(\mu_k) = -(\nabla_{\hat{y}} g_{1,k})^T$, and

$$M_k = \nabla_{\hat{y}} g_{2,k}(\mu_k). \tag{2.8}$$

For convenience, we rewrite the Newton system (2.7) as

$$K_k^N \ \Delta w_k^N = -g_k(\mu_k). \tag{2.9}$$

2.3. The step size. As mentioned above, in addition to computing the Newton step, interior-point methods choose a step size such that all nonnegative variables remain strictly positive. In our case, r, λ and σ must remain positive. To ensure this, we assume that the step sizes are chosen such as those in [39]. Therefore, at iteration k,

$$\alpha_{r,k} = \min\big\{1, \gamma_k \ \min\{-\frac{r_{ki}}{\Delta r_{k,i}^N}\} \ \text{ s.t. } \ \Delta r_{k,i}^N < 0\big\} \tag{2.10}$$

$$\alpha_{\lambda,k} = \min\big\{1, \gamma_k \ \min\{-\frac{\lambda_{ki}}{\Delta \lambda_{k,i}^N}\} \ \text{ s.t. } \ \Delta \lambda_{k,i}^N < 0\big\}, \tag{2.11}$$

$$\alpha_{\sigma,k} = \min\big\{1, \gamma_k \ \min\{-\frac{\sigma_{ki}}{\Delta \sigma_{k,i}^N}\} \ \text{ s.t. } \ \Delta \sigma_{k,i}^N < 0\big\}, \tag{2.12}$$

where $\gamma_k \in (0, 1)$. Because the global and local variables are not required to be nonnegative, we can set

$$\alpha_{x,k} = \alpha_{y,k} = 1. \tag{2.13}$$

If we define the matrix Λ_k as

$$\Lambda_k = \begin{bmatrix} \alpha_{x,k} I & 0 & 0 & 0 & 0 \\ 0 & \alpha_{y,k} I & 0 & 0 & 0 \\ 0 & 0 & \alpha_{r,k} I & 0 & 0 \\ 0 & 0 & 0 & \alpha_{\lambda,k} I & 0 \\ 0 & 0 & 0 & 0 & \alpha_{\sigma,k} I \end{bmatrix},$$

the kth iteration of a primal-dual algorithm has the following form:

$$w_{k+1} = w_k + \Lambda_k \ \Delta w_k^N. \tag{2.14}$$

2.4. Solution of the Newton system via the Schur complement. Assuming M_k is invertible, the Schur complement of W_k is the matrix

$$S_k = W_k - \widehat{A}_k^T \ M_k^{-1} \ \widehat{A}_k. \tag{2.15}$$

If S_k is invertible, the global component of the Newton search direction Δx_k^N can be computed as:

$$S_k \ \Delta x_k^D = -\big(g_{1,k} + \widehat{A}_k^T \ M_k^{-1} \ g_{2,k}(\mu_k)\big). \tag{2.16}$$

Then the local component $\Delta\widehat{y}_k^N$ is

$$M_k \ \Delta\widehat{y}_k^N = -\big(g_{2,k} - \widehat{A}_k \Delta x_k^D\big). \tag{2.17}$$

Note that, for the general problem (1.1) with N systems, M_k is a block diagonal matrix composed of N blocks. Thus, the Schur complement allows one to decompose the linear system (2.17) into N smaller independent linear systems. This is the basis for many specialized direct methods (see, [5, Section 5.6]). Though these methods are very efficient, they allow for a lower degree of decentralization than decomposition algorithms. As discussed before, the main priority when choosing a methodology to solve an MDO problem, is that it must allow for a high degree of decentralization. For this reason, the remainder of this paper focuses on decomposition algorithms.

2.5. Convergence. The local convergence theory of this class of algorithms is developed in the papers by [27, 18, 39] and recently in [23]. These papers establish conditions on parameters μ_k and γ_k under which the iteration (2.14) converges superlinearly or quadratically to a solution of (2.1) (and under standard assumptions made in the analysis of interior-point methods). As in this paper our analysis focuses on the local convergence properties of the algorithms, no procedures are given to ensure global convergence, though the techniques in [37, 21, 20, 9] could be adapted.

3. The Interior-Point Decomposition Algorithm. In this section, we first explain how bilevel decomposition algorithms work in general. Then, we describe a particular bilevel decomposition algorithm that only takes one Newton iteration when solving the subproblems. Finally, we analyze the relationship between the search directions provided by this decomposition algorithm and the direct interior-point method described in the previous section.

3.1. Bilevel decomposition algorithms. Bilevel decomposition algorithms divide the job of finding a minimizer to problem (2.1) into two different tasks: (i) finding an optimal value of the local variables $y^*(x)$ for a given value of the global variables x, and (ii) finding an overall optimal value of the global variables x^*. The first task is performed by solving a subproblem. Then the subproblem solution is used to define a master problem whose solution accomplishes the second task.

A general bilevel programming decomposition algorithm for problem (2.1) may be described as follows. Solve the following *master problem*:

$$\underset{x}{\text{minimize}} \ \ F^*(x). \tag{3.1}$$

where $F^*(x) = F(x, y^*(x))$ is the subproblem optimal value function

$$\begin{array}{rll} F^*(x) = & \underset{y,r}{\text{minimum}} & F(x,y) - \mu \sum_{i=1}^m \log(r_i) \\ & \text{subject to} & c(x,y) - r = 0. \end{array} \tag{3.2}$$

Note that the above master problem depends only on the global variables. The local variables are kept within the subproblem. In the general case where there are more than one system, the above formulation allows the different systems to be dealt with almost independently, and only a limited amount of information regarding the global variables is exchanged between the master problem and the subproblems. This makes bilevel decomposition approaches suitable for MDO problems where, as mentioned before, it is crucial to allow the different groups participating in a project to work as independently from each other as possible.

After breaking the original problem into a master problem and a subproblem, a bilevel decomposition algorithm applies a nonlinear optimization method to solve the master problem. At each iteration, a new estimate of the global variables x_k is generated and the subproblem is solved *exactly* using x_k as a parameter. Then, sensitivity analysis formulae [19] are used to compute the master problem objective and its derivatives from the *exact* subproblem minimizer. Using this information, a new estimate of the global variables x_{k+1} is computed. This procedure is repeated until a master problem minimizer is found.

Unfortunately, there is analytical and numerical evidence that certain commonly used bilevel decomposition algorithms may fail to converge even when the starting point is very close to the minimizer [2, 14]. Moreover, although there are some local convergence proofs for certain bilevel decomposition algorithms that solve the subproblems exactly [16], it is safe to say that the local convergence theory of bilevel decomposition algorithms is not nearly as satisfactory as that of direct interior-point methods.

In the remainder of this section, after stating our assumptions, we state a bilevel decomposition algorithm that only takes one iteration of an interior-point method to solve the subproblem. A difficulty is that by taking only one iteration, we obtain only a rough approximation to the subproblem minimizer and thus it is not straightforward to use sensitivity formulae to compute the master problem derivatives. We overcome this difficulty by showing that the Schur complement iteration (2.16) can be seen as an approximation to the master problem Newton iteration. Finally, we analyze the relationship between the proposed decomposition algorithm and the direct method outlined in Section 2. This relationship will be used in Section 4 to derive a Gauss-Seidel iteration that ensures the decomposition algorithm achieves superlinear convergence.

3.2. Assumptions. We make the following assumptions. We assume there exists a minimizer (x^*, y^*, r^*) to problem (2.1) and a Lagrange multiplier vector (λ^*, σ^*) satisfying the KKT conditions (2.3) with $\mu = 0$. The following conditions are assumed on the problem functions and on the so-called KKT point

$$w^* = (x^*, y^*, r^*, \lambda^*, \sigma^*).$$

A.1 The second derivatives of the functions in problem (2.1) are Lipschitz continuous in an open convex set containing w^*.

A.2 The linear independence constraint qualification is satisfied at w^*, that is, the matrix

$$L = \begin{pmatrix} \nabla_x c(x^*, y^*) & \nabla_y c(x^*, y^*) & -I \\ 0 & 0 & I_N \end{pmatrix} \tag{3.3}$$

has full row rank, where N is the active set $\{i : \ r_i^* = 0\}$ and I_N is the matrix formed by the rows of the identity corresponding to indices in N.

A.3 The strict complementary slackness condition is satisfied at w^*; that is, $\sigma_i^* > 0$ for $i \in N$.

A.4 The second order sufficient conditions for optimality are satisfied at w^*; that is, for all $d \neq 0$ satisfying $Ld = 0$ we have

$$d^T \nabla^2 \mathcal{L}(w^*) d > 0, \tag{3.4}$$

where the Lagrangian function is $\mathcal{L}(w^*) = F(x^*, y^*) - (\lambda^*)^T (c(x^*, y^*) - r^*) - (\sigma^*)^T r^*$, and $\nabla^2 \mathcal{L}(w^*)$ is the Hessian of the Lagrangian function with respect to the primal variables x, y, r.

In addition, the following condition is assumed in order to ensure that the subproblems iterations are well-defined near the solution, w^*.

C.1 The strong linear independence constraint qualification holds at w^*, that is, the matrix

$$L = \begin{pmatrix} \nabla_y c(x^*, y^*) & -I \\ 0 & I_N \end{pmatrix} \tag{3.5}$$

has full row rank.

3.3. The subproblem iteration. The decomposition algorithm takes just one Newton iteration of a primal-dual interior-point method to solve the subproblems. Following the notation introduced in Section 2, the subproblem perturbed KKT conditions can be written in compact form as $g_2(\mu) = 0$, see (2.6). Then, the Newton system for the above perturbed KKT conditions is simply

$$M_k \ \Delta \hat{y}_k^D = -g_{2,k}(\mu_k).^3 \tag{3.6}$$

3.4. The master problem iteration. The decomposition algorithm applies Newton's method to solve the master problem (3.1). The Newton search direction is the solution to

$$\nabla^2_{xx} F^*(x_k) \ \Delta x_k^D = -\nabla_x F^*(x_k). \ ^4 \tag{3.7}$$

Unfortunately, because the algorithm only takes one iteration to solve the subproblem, the exact expressions for $\nabla^2_{xx} F^*(x_k)$ and $\nabla_x F^*(x_k)$ can *not* be computed from standard sensitivity formulae as is customary in bilevel decomposition algorithms. In the remainder of this section, we show how approximations to $\nabla_x F^*(x_k)$ and $\nabla^2_{xx} F^*(x_k)$ can be obtained from the estimate of the subproblem minimizer given by taking only one Newton iteration on the subproblem. In particular, the following two propositions show that the right hand side in equation (2.16) can be seen as an approximation to the master problem gradient $\nabla_x F^*(x_k)$ and that the Schur complement matrix S_k can be interpreted as an approximation to the master problem Hessian $\nabla^2_{xx} F^*(x_k)$.

PROPOSITION 3.1. *Let $(x^*, y^*, r^*, \lambda^*, \sigma^*)$ be a KKT point satisfying assumptions A.1–A.4 and condition C.1 for problem* (2.1)*. Then, for x_k close to x^*, the subproblem optimal value function $F^*(x_k)$ and its gradient $\nabla_x F^*(x_k)$ are well defined and*

$$\|\nabla_x F^*(x_k) - (g_{1,k} + \widehat{A}_k^T \ M_k^{-1} \ g_{2,k}(\mu_k))\| = o(\|\hat{y}(x_k) - \hat{y}_k\|),$$

where $\hat{y}(x_k) = (y(x_k), r(x_k), \lambda(x_k), \sigma(x_k))$ is the locally unique once continuously differentiable trajectory of minimizers to subproblem (3.2) *with $\hat{y}(x^*) = (y^*, r^*, \lambda^*, \sigma^*)$ and $\hat{y}_k = (y_k, r_k, \lambda_k, \sigma_k)$.*

Proof. Note that if Condition C.1 holds at $(x^*, y^*, r^*, \lambda^*, \sigma^*)$, then the linear independence constraint qualification (LICQ) holds at $(y^*, r^*, \lambda^*, \sigma^*)$ for subproblem (3.2) with $x = x^*$. Moreover, it is easy to see that if $(x^*, y^*, r^*, \lambda^*, \sigma^*)$ is a KKT point satisfying assumptions A.1–A.4, then $(y^*, r^*, \lambda^*, \sigma^*)$ is a minimizer satisfying the strict complementarity slackness (SCS) and second-order sufficient conditions (SOSC) for

[3] When solving system (3.6), we actually solve the equivalent linear system obtained by first eliminating the dual variables σ from the system. The resulting linear system is smaller and symmetric.

[4] Note that the master problem objective function $F^*(x_k)$ depends also on the barrier parameter μ. However, we do not write μ explicitely to simplify notation.

subproblem (3.2) with $x = x^*$. It follows from [19, Theorem 6] that there exists a locally unique once continuously differentiable trajectory of subproblem minimizers $\hat{y}(x_k) = (y(x_k), r(x_k), \lambda(x_k), \sigma(x_k))$ satisfying LICQ, SCS and SOSC for the subproblem with $x = x_k$. As a result, the subproblem optimal value function $F^*(x_k)$ can be defined as $F^*(x_k) = F(x_k, y(x_k))$ and it is once continuously differentiable. Moreover, its gradient is simply

$$\nabla_x F^*(x_k) = \frac{d[F\big(x_k, y(x_k), r(x_k)\big) - \mu \sum_{i=1}^m \log(r_i(x_k))]}{dx}, \tag{3.8}$$

where d/dx denotes the total derivative. Moreover, because the LICQ holds at the subproblem minimizer for x_k we have,

$$\nabla_x F^*(x_k) = \frac{d[F\big(x_k, y(x_k), r(x_k)\big) - \mu \sum_{i=1}^m \log(r_i(x_k))]}{dx} = \frac{d\mathcal{L}_y\big(x_k, \hat{y}(x_k)\big)}{dx}, \tag{3.9}$$

where $\mathcal{L}_y$ is the subproblem Lagrangian function:

$$\mathcal{L}_y(x, \hat{y}(x)) = F(x, y) - \mu \sum_{i=1}^m \log(r_i) - \lambda^T \big(c(x, y) - r\big). \tag{3.10}$$

Applying the chain rule, we get:

$$\begin{aligned}
\frac{d\mathcal{L}_y(x_k, \hat{y}(x_k))}{dx} &= \nabla_x \mathcal{L}_y(x_k, \hat{y}(x_k)) && (3.11)\\
&+ \nabla_y \mathcal{L}_y(x_k, \hat{y}(x_k))\ y'(x_k) && (3.12)\\
&+ \nabla_r \mathcal{L}_y(x_k, \hat{y}(x_k))\ r'(x_k) && (3.13)\\
&+ \nabla_\lambda \mathcal{L}_y(x_k, \hat{y}(x_k))\ \lambda'(x_k), && (3.14)\\
&+ \nabla_\sigma \mathcal{L}_y(x_k, \hat{y}(x_k))\ \sigma'(x_k), && (3.15)
\end{aligned}$$

where $y'(x_k)$, $r'(x_k)$, $\lambda'(x_k)$, and $\sigma'(x_k)$ denote the Jacobian matrices of y, r, λ, and σ evaluated at x_k, respectively. Note that (3.12) and (3.13) are zero because of the optimality of $\hat{y}(x_k)$, (3.14) is zero by the feasibility and strict complementarity slackness of $\hat{y}(x_k)$, and (3.15) is zero because the Lagrangian function does not depend on σ. Thus, we can write the master problem objective gradient as

$$\nabla_x F^*(x_k) = \ \nabla_x \mathcal{L}_y\big(x_k, \hat{y}(x_k)\big). \tag{3.16}$$

If we knew the subproblem minimizer $\hat{y}(x_k)$, we could easily compute the master problem gradient by evaluating the gradient of the Lagrangian function (3.10) at x_k and $\hat{y}(x_k)$. Unfortunately, after taking only one interior-point iteration on the subproblem, we do not know $\hat{y}(x_k)$ exactly but rather the following approximation

$$\hat{y}(x_k) \simeq \hat{y}_k + \Delta\hat{y}_k^D, \tag{3.17}$$

where $\Delta\hat{y}_k^D$ is the subproblem search direction computed by solving system (3.6).

But by Taylor's Theorem we know that the master problem gradient can be approximated as:

$$\nabla_x F^*(x_k) = \nabla_x \mathcal{L}_y(x_k, \hat{y}_k) + \nabla_{x,\hat{y}} \mathcal{L}_y(x_k, \hat{y}_k)(\hat{y}(x_k) - \hat{y}_k) + O(\|\hat{y}(x_k) - \hat{y}_k\|^2).$$

Moreover, if $\hat{y}_k$ is close enough to $\hat{y}(x_k)$, we know from the local convergence theory of Newton's method that $\|\hat{y}(x_k) - (\hat{y}_k + \Delta\hat{y}_k^D)\| = o(\|\hat{y}(x_k) - \hat{y}_k\|)$ and thus

$$\nabla_x F^*(x_k) = \nabla_x \mathcal{L}_y(x_k, \hat{y}_k) + \nabla_{x,\hat{y}} \mathcal{L}_y(x_k, \hat{y}_k)\Delta\hat{y}_k^D + o(\|\hat{y}(x_k) - \hat{y}_k\|). \tag{3.18}$$

From A.3, A.4 and C.1, we know that the matrix M_k is nonsingular once the iterates are close to the minimizer [19, Theorem 14]. Since $\Delta\hat{y}_k^D = -M_k^{-1}\ g_{2,k}(\mu_k)$ and $\widehat{A}_k^T = -\nabla_{\hat{y}} g_{1,k} = -\nabla_{x,\hat{y}} \mathcal{L}_y(x_k, \hat{y}_k)$, the result follows from (3.18). □

PROPOSITION 3.2. *Let $(x^*, y^*, r^*, \lambda^*, \sigma^*)$ be a KKT point satisfying assumptions A.2–A.4 and condition C.1 for problem* (2.1)*. Moreover, assume all functions in problem* (2.1) *are three times continuously differentiable. Then, for x_k close to x^*, the Hessian of the subproblem optimal value function $\nabla^2_{xx} F^*(x_k)$ is well defined and*

$$\|\nabla^2_{xx} F^*(x_k) - S_k\| = O(\|\hat{y}(x_k) - \hat{y}_k\|),$$

where $\hat{y}(x_k) = (y(x_k), r(x_k), \lambda(x_k), \sigma(x_k))$ is the locally unique twice continuously differentiable trajectory of minimizers to subproblem (3.2) *with $\hat{y}(x^*) = (y^*, r^*, \lambda^*, \sigma^*)$, $\hat{y}_k = (y_k, r_k, \lambda_k, \sigma_k)$, and S_k is the Schur complement matrix $S_k = W_k - \widehat{A}_k^T\ M_k^{-1}\ \widehat{A}_k$.*

Proof. By the same arguments as in Proposition 3.1, and the assumption that all problem functions are three times continuously differentiable, we know that the subproblem optimal value function can be defined as $F^*(x_k) = F(x_k, y(x_k))$ and it is twice continuously differentiable.

Moreover, differentiating expression (3.16), we obtain the following expression for the optimal value function Hessian:

$$\begin{aligned} \nabla_{xx} F^*(x_k) &= \frac{d(\nabla_x \mathcal{L}_y(x_k, \hat{y}(x_k))}{dx} \\ &= \nabla_{x,x} \mathcal{L}_y\big(x_k, \hat{y}(x_k)\big) + \nabla_{x,\hat{y}} \mathcal{L}_y\big(x_k, \hat{y}(x_k)\big)\ \hat{y}'(x_k), \end{aligned} \tag{3.19}$$

where $\hat{y}'(x_k)$ is the Jacobian matrix of the subproblem minimizer with respect to x_k.

By A.3, A.4 and C.1, we know that for x_k close enough to x^*, $\hat{y}(x_k)$ is a minimizer satisfying the LICQ, SCS, and SOSC for the subproblem, and thus it follows from [19, Theorem 6] that:

$$M_k^*\ \hat{y}'(x_k) = \widehat{A}_k^*, \tag{3.20}$$

where M_k^* and $\widehat{A}_k^*$ are the matrices M_k and $\widehat{A}_k$ evaluated at $\hat{y}(x_k)$.

If we knew the subproblem minimizer $\hat{y}(x_k)$ exactly, we could use (3.19) and (3.20) to compute the master problem Hessian. Unfortunately, after taking only one Newton iteration on the subproblems, we do not know $\hat{y}(x_k)$ exactly. But we can approximate $\hat{y}'(x_k)$ as the solution to the following system

$$M_k\ \hat{y}'(x_k) \simeq \widehat{A}_k. \tag{3.21}$$

Note that by A.3, A.4 and C.1, the matrix M_k is nonsingular for $(x_k, \hat{y}_k)$ close to $(x^*, \hat{y}^*)$. Moreover, by the differentiability of all problem functions and Taylor's Theorem we know that

$$\|\hat{y}'(x_k) - M_k^{-1}\widehat{A}_k\| = \|(M_k^*)^{-1}\widehat{A}_k^* - M_k^{-1}\widehat{A}_k\| = O(\|\hat{y}(x_k) - \hat{y}_k\|).$$

The result follows because $W_k = \nabla_x g_{1,k} = \nabla_{x,x} \mathcal{L}_y\big(x_k, \hat{y}_k\big)$ and $\widehat{A}_k^T = -\nabla_{\hat{y}} g_{1,k} = -\nabla_{x,\hat{y}} \mathcal{L}_y(x_k, \hat{y}_k)$. □

Note that Propositions 3.1 and 3.2 show that the Schur complement iteration,

$$S_k\ \Delta x_k^D = -\big(g_{1,k} + \widehat{A}_k^T\ M_k^{-1}\ g_{2,k}(\mu_k)\big). \tag{3.22}$$

described in Section 2.4, provides a suitable approximation to the master problem Newton equation (3.7).

3.5. Decomposition algorithm statement. The decomposition algorithm can be seen as a particular bilevel decomposition algorithm that only takes one iteration to solve the subproblems and uses this iteration to approximate the master problem derivatives as explained in Section 3.4. The interior-point decomposition-algorithm is stated in Figure 3.1.

Initialization: Choose a starting point $w_0^T = (x_0^T, y_0^T, r_0^T, \lambda_0^T, \sigma_0^T)^T$ such that $r_0 > 0$, $\lambda_0 > 0$, $\sigma_0 > 0$. Set $k \leftarrow 0$ and choose the parameters $\mu_0 \geq 0$ and $0 < \hat{\gamma} \leq \gamma_0 < 1$.
Repeat

1. **Solve master problem:** Form the matrix S_k and compute Δx_k^D from system (3.22). Set $x_{k+1} = x_k + \Delta x_k^D$.
2. **Solve subproblem:**
   (a) **Search direction:** Compute $\Delta \hat{y}_k^D$ by solving system (3.6).
   (b) **Line search:** With γ_k, compute the diagonal matrix, Λ_k, from the subproblem step sizes as in (2.10)-(2.13).
   (c) **Update iterate:** Set $\hat{y}_{k+1} = \hat{y}_k + \Lambda_k\ \Delta \hat{y}_k^D$.
3. **Parameter update:** Set $\mu_k \geq 0$, $0 < \hat{\gamma} \leq \gamma_k < 1$, and $k \leftarrow k + 1$.

Until convergence

FIGURE 3.1. *Interior-point Decomposition Algorithm*

3.6. Relationship to the direct method. The following proposition establishes the relationship between the search direction of the proposed decomposition algorithm $(\Delta x_k^D, \Delta y_k^D)$ and the search direction of the direct method $(\Delta x_k^N, \Delta y_k^N)$. In particular, we show that the global variable components of both search directions are identical and we characterize the difference between the local components.

PROPOSITION 3.3. *Under assumptions A.1–A.4 and condition C.1,*

$$\Delta x_k^D = \Delta x_k^N \quad \textit{and} \quad \Delta y_k^D = \Delta y_k^N - M_k^{-1} \widehat{A}_k\ \Delta x_k^N. \tag{3.23}$$

Proof. The first equality follows trivially because we are using a Schur complement iteration to approximate the master problem search direction. The second equality follows from (2.17) and (3.6). □

Note that the difference between the local components of both search directions is not surprising because the global variables are just a parameter to the subproblem solved by the decomposition algorithm. As a result, the local component of the search direction computed by the decomposition algorithm lacks *first-order* information about the global component search direction (in particular, it lacks the following first-order information $-M_k^{-1}\widehat{A}_k\ \Delta x_k^N$). In Section 4, we show how a Gauss-Seidel

strategy can be used to overcome this limitation inherent to bilevel decomposition approaches.

Finally, it is useful to note that the decomposition algorithm search direction is the solution to the following linear system:

$$K_k^D \ \Delta w_k^D = -g_k(\mu_k), \tag{3.24}$$

where

$$K_k^D = \begin{pmatrix} S_k & -\widehat{A}_k^T \\ 0 & M_k \end{pmatrix}. \tag{3.25}$$

Note that the fact that the global variables are a parameter to the subproblems is evident in the structure of K_k^D. In particular, notice that the lower left block in matrix K_k^D is zero instead of $\hat{A}_k$ as in direct method Newton matrix K_k^N. Finally, In Section 4, we give conditions under which the norm of the matrix $(K_k^D)^{-1}$ is uniformly bounded away from zero in the neighborhood of the minimizer and thus the iterates of the proposed decomposition algorithm are well defined.

4. Local convergence analysis. The difference in the local component of the search directions computed by the decomposition algorithm and the direct Newton method precludes any possibility of superlinear convergence for the decomposition algorithm. But in this section we show how one can first compute the global variable component of the search direction, and then use it to update the subproblem derivative information before computing the local variable component. We show that the resulting *Gauss-Seidel iteration* generates a search direction that is equal (up to second-order terms) to the search direction of the direct method. Moreover, we prove that the resulting decomposition algorithm converges locally at a superlinear rate.

4.1. The Gauss-Seidel refinement. The decomposition algorithm defined in Section 3 does not make use of all the information available at each stage. Note that, at each iteration of the decomposition algorithm, we first compute the master problem step as the solution to

$$S_k \ \Delta x_k^G = -\big(g_{1,k} + \widehat{A}_k^T \ M_k^{-1} \ g_{2,k}(\mu_k)\big), \tag{4.1}$$

and update the global variables as $x_{k+1} = x_k + \Delta x_k^G$. At this point, one could use the new value of the global variables x_{k+1} to perform a nonlinear update of the subproblem derivative information and thus, generate a better subproblem step. In particular, after solving for the master problem search direction, we could compute

$$g_{2,k}^+(\mu_k) = \begin{pmatrix} \nabla_y F(x_{k+1}, y_k) - \nabla_y c(x_{k+1}, y_k)^T \lambda_k \\ -\sigma_k + \lambda_k \\ -c(x_{k+1}, y_k) + r_k \\ -R_k \sigma_k + \mu_k e \end{pmatrix}. \tag{4.2}$$

Then, the subproblem search direction would be given as the solution to

$$M_k \ \Delta \hat{y}_k^G = -g_{2,k}^+(\mu_k). \tag{4.3}$$

4.2. Relationship to the direct method. The following proposition shows that the search directions of the proposed Gauss-Seidel decomposition algorithm and the direct method outlined in Section 2 are equal up to second-order terms.

PROPOSITION 4.1. *Under assumptions A.1–A.4 and condition C.1,*

$$\Delta x_k^G = \Delta x_k^N \;\; \textit{and} \;\; \Delta \hat{y}_k^G = \Delta \hat{y}_k^N + O(\|\Delta x_k^N\|^2). \tag{4.4}$$

Proof. The result for the global components is trivial from (4.1). For the local components, note that the search direction of the resulting Gauss-Seidel decomposition algorithm satisfy

$$\Delta x_k^G = \Delta x_k^D = \Delta x_k^N, \tag{4.5}$$

and

$$\Delta \hat{y}_k^G = \Delta \hat{y}_k^D - M_k^{-1}(g_{2,k}^+(\mu_k) - g_{2,k}(\mu_k)). \tag{4.6}$$

Moreover, from (3.23), we know that

$$\begin{aligned}\Delta \hat{y}_k^G &= \Delta \hat{y}_k^N - M_k^{-1}\widehat{A}_k\ \Delta x_k^N - M_k^{-1}(g_{2,k}^+(\mu_k) - g_{2,k}(\mu_k)) \\ &= \Delta \hat{y}_k^N - M_k^{-1}(g_{2,k}^+(\mu_k) - g_{2,k}(\mu_k) + \widehat{A}_k\ \Delta x_k^N).\end{aligned}$$

The result is obtained by Taylor's Theorem and the fact that $\widehat{A}_k^T = -\nabla_{x,\hat{y}}\mathcal{L}_y(x_k, \hat{y}_k)$. ∎

Proposition 4.1 intuitively implies that the Gauss-Seidel decomposition algorithm converges locally at a superlinear rate. In Section 4.3 we formally show this is the case.

The resulting Gauss-Seidel decomposition algorithm is stated in Figure 4.1. It must be noted that, the only difference between the interior-point decomposition algorithm with the Gauss-Seidel refinement stated in Figure 4.1 and the algorithm stated in Figure 3.1 is that in the Gauss-Seidel version, we introduce a nonlinear update into the derivative information of the subproblem $g_{2,k}(\mu_k)$ using the master problem step Δx_k^G. As a consequence, the refinement requires one more subproblem derivative evaluation per iteration. The advantage is that, as we show in the next section, the Gauss-Seidel refinement guarantees that the proposed algorithm converges at a superlinear rate.

4.3. Convergence of the Gauss-Seidel approach. In this section, we first show that the search direction of the Gauss-Seidel decomposition algorithm is well-defined in the proximity of the minimizer and then, we show that the iterates generated by the Gauss-Seidel decomposition algorithm converge to the minimizer at a superlinear rate.

Note that the search directions of the decomposition algorithms with and without the Gauss-Seidel refinement are related as follows:

$$\Delta w_k^G = \Delta w_k^D - G_k(g_{2,k}^+(\mu_k) - g_{2,k}(\mu_k)), \tag{4.7}$$

where

$$G_k = \begin{pmatrix} 0 \\ M_k^{-1} \end{pmatrix}.$$

Because $\Delta w_k^D = -(K_k^D)^{-1}\, g_k(\mu_k)$, to show that the Gauss-Seidel search direction is well-defined, it suffices to show that $\|(K_k^D)^{-1}\|$ and $\|M_k^{-1}\|$ are uniformly bounded for w_k in a neighborhood of the minimizer w^*.

Initialization: Choose a starting point $w_0^T = (x_0^T, y_0^T, r_0^T, \lambda_0^T, \sigma_0^T)^T$ such that $r_0 > 0$, $\lambda_0 > 0$, $\sigma_0 > 0$. Set $k \leftarrow 0$ and choose the parameters $\mu_0 \geq 0$ and $0 < \hat{\gamma} \leq \gamma_0 < 1$.
Repeat

1. **Solve master problem:** Form the matrix S_k and compute Δx_k^G from system (4.1). Set $x_{k+1} = x_k + \Delta x_k^G$.
2. **Solve subproblem:**
   (a) **Search direction:** Use x_{k+1} to update $g_{2,k}^+(\mu_k)$ and compute $\Delta \hat{y}_k^G$ by solving system (4.3).
   (b) **Line search:** With γ_k, compute the diagonal matrix, Λ_k, from the subproblem step sizes as in (2.10)-(2.13).
   (c) **Update iterate:** Set $\hat{y}_{k+1} = \hat{y}_k + \Lambda_k \ \Delta \hat{y}_k^G$.
3. **Parameter update:** Set $\mu_k \geq 0$, $0 < \hat{\gamma} \leq \gamma_k < 1$, and $k \leftarrow k + 1$.

Until convergence

FIGURE 4.1. *Interior-Point Decomposition Algorithm with Gauss-Seidel refinement*

PROPOSITION 4.2. *Under assumptions A.1–A.4 and condition C.1, $\|(K_k^D)^{-1}\|$ and $\|M_k^{-1}\|$ are uniformly bounded for w_k in a neighborhood of the minimizer w^*.*

Proof. Because

$$(K_k^D)^{-1} = \begin{pmatrix} S_k^{-1} & S_k^{-1} \widehat{A}_k^T M_k^{-1} \\ 0 & M_k^{-1} \end{pmatrix}, \tag{4.8}$$

it is sufficient to prove that $\|S_k^{-1}\|$ and $\|M_k^{-1}\|$ are uniformly bounded. Let M^* be the matrix M_k defined in (2.8) evaluated at w^*. Then, by A.1-A.4 and C.1 we know that M^* is non-singular (see [19, Theorem 14] and [18, Proposition 4.1]). Consequently, $\|M_k^{-1}\|$ is uniformly bounded if w_k is close enough to w^*. Likewise, by A.1-A.4, K^{N*} is non-singular and $\|(K_k^N)^{-1}\|$ is uniformly bounded as well as $\|S_k^{-1}\|$ if condition C.1 holds. □

We now give a result that provides sufficient conditions on the barrier and the step size parameter updates to ensure superlinear convergence of the Gauss-Seidel decomposition algorithm.

THEOREM 4.3. *Suppose that assumptions A.1-A.4 and condition C.1 hold, that the barrier parameter is chosen to satisfy $\mu_k = o\big(\|g_k(0)\|\big)$ and the step size parameter is chosen such that $1 - \gamma_k = o(1)$. If w_0 is close enough to w^*, then the sequence $\{w_k\}$ described in (4.7) is well-defined and converges to w^* at a superlinear rate.*

Proof. As matrices $(K_k^D)^{-1}$ and M_k^{-1} are well-defined by Proposition 4.2, the sequence in (4.7) updates the new point as

$$\begin{aligned} w_{k+1} &= w_k + \Lambda_k \ \Big[\Delta w_k^D - G_k(g_{2,k}^+(\mu_k) - g_{2,k}(\mu_k))\Big] && (4.9) \\ &= w_k - \Lambda_k \ (K_k^D)^{-1} g_k(\mu_k) - \Lambda_k \ G_k \ (g_{2,k}^+(0) - g_{2,k}(0)) \\ &= w_k - \Lambda_k \ (K_k^D)^{-1}(g_k(0) - \bar{\mu}_k) - \Lambda_k \ G_k \ (g_{2,k}^+(0) - g_{2,k}(0)) && (4.10) \end{aligned}$$

where $\bar{\mu}_k = (0, 0, 0, 0, \mu_k e)$. Then,

$$\begin{aligned} w_{k+1} - w^* &= w_k - w^* - \Lambda_k \ (K_k^D)^{-1} g_k(\mu_k) - \Lambda_k \ G_k \ (g_{2,k}^+(0) - g_{2,k}(0)) \\ &= (I - \Lambda_k)(w_k - w^*) \\ &\quad + \Lambda_k \ (K_k^D)^{-1}\big(K_k^D(w_k - w^*) - g_k(0) + \bar{\mu}_k\big) \\ &\quad - \Lambda_k \ G_k \ (g_{2,k}^+(0) - g_{2,k}(0)), \end{aligned} \tag{4.11}$$

which may be rewritten as

$$\begin{aligned} w_{k+1} - w^* = (I - \Lambda_k)(w_k - w^*) &\\ + \Lambda_k \ (K_k^D)^{-1}\bar{\mu}_k &\\ + \Lambda_k \ \big(K_k^D\big)^{-1}(K_k^N(w_k - w^*) - g_k(0)) &\\ + \Lambda_k \ \big(K_k^D\big)^{-1}(K_k^D - K_k^N)(w_k - w^*) &\\ - \Lambda_k \ G_k \ (g_{2,k}^+(0) - g_{2,k}(0))\big) & \end{aligned} \tag{4.12}$$

The first term in (4.12) satisfies (see [39])

$$\|(I - \Lambda_k)(w_k - w^*)\| \leq \big((1 - \gamma_k) + O(\|g_k(0)\|) + O(\mu_k)\big) \ \|(w_k - w^*)\|. \tag{4.13}$$

This inequality together with conditions $1 - \gamma_k = o(1)$ and $\mu_k = o\big(\|g_k(0)\|\big)$ imply that

$$\|(I - \Lambda_k)(w_k - w^*)\| = o(\|w_k - w^*\|). \tag{4.14}$$

The second term in (4.12) satisfies

$$\|\Lambda_k \ (K_k^D)^{-1}\bar{\mu}_k\| \leq \|\Lambda_k\| \ \|(K_k^D)^{-1}\| \ \|\bar{\mu}_k\| \leq \beta \ \|\bar{\mu}_k\|, \tag{4.15}$$

which by condition $\mu_k = o\big(\|g_k(0)\|\big)$ imply

$$\|\Lambda_k \ (K_k^D)^{-1}\bar{\mu}_k\| = o(\|w_k - w^*\|). \tag{4.16}$$

By Taylor's Theorem, the third term in (4.12) satisfies

$$\begin{aligned} \|\Lambda_k \ \big(K_k^D\big)^{-1}(K_k^N(w_k - w^*) - g_k(0))\| \leq & \\ \|\Lambda_k\| \ \big\|\big(K_k^D\big)^{-1}\big\| \ \|(K_k^N(w_k - w^*) - g_k(0))\| = o(\|w_k - w^*\|). & \end{aligned} \tag{4.17}$$

Finally, because

$$K_k^D = K_k^N - \begin{bmatrix} W_k - S_k & 0 \\ -\widehat{A}_k & 0 \end{bmatrix}, \tag{4.18}$$

the fourth term in (4.12) is

$$\begin{aligned} \Lambda_k \ \big(K_k^D\big)^{-1}(K_k^D - K_k^N)(w_k - w^*) &= -\Lambda_k \ \big(K_k^D\big)^{-1} \begin{bmatrix} W_k - S_k & 0 \\ -\widehat{A}_k & 0 \end{bmatrix} (w_k - w^*) \\ &= \Lambda_k \ \begin{bmatrix} 0 & 0 \\ M_k^{-1}\widehat{A}_k & 0 \end{bmatrix} (w_k - w^*) \\ &= \Lambda_k \ \begin{bmatrix} 0 \\ M_k^{-1}\widehat{A}_k(x_k - x^*) \end{bmatrix}. \end{aligned} \tag{4.19}$$

Then, adding the fourth and fifth terms in (4.12) and using (4.19) we get

$$\Lambda_k\ (K_k^D)^{-1}(K_k^D - K_k^N)(w_k - w^*) - \Lambda_k\ G_k\ (g_{2,k}^+(0) - g_{2,k}(0)) = \Lambda_k\ \begin{bmatrix} 0 \\ M_k^{-1}\big[\widehat{A}_k(x_k - x^*) - (g_{2,k}^+(0) - g_{2,k}(0))\big] \end{bmatrix}. \quad (4.20)$$

If only the global variable component, x, of equations (4.14), (4.16), (4.17), and (4.20) is considered, then the following relationship is attained:

$$\|x_{k+1} - x^*\| = o(\|w_k - w^*\|). \quad (4.21)$$

Note that this is not a surprising result because we know that the step taken by the Gauss-Seidel decomposition algorithm on the global variables, x, is the same as that of a direct Newton's method.

To finish the proof, it only remains to show that the local variable component, $\hat{y}$, satisfies a similar relationship. The local component of equation (4.20) can be written as

$$\Lambda_{k,y}\ M_k^{-1}\big(\widehat{A}_k(x_k - x^*) - (g_{2,k}^+(0) - g_{2,k}(0))\big) = \Lambda_{k,y}\ M_k^{-1}\big(\widehat{A}_k(x_{k+1} - x^*) - (g_{2,k}^+(0) - g_{2,k}(0)) - \widehat{A}_k(x_{k+1} - x_k)\big), \quad (4.22)$$

which by Taylor's Theorem and the fact that $\widehat{A}_k = -\nabla_x g_{2,k}(\mu_k) = \nabla_x g_{2,k}(0)$ is

$$\Lambda_{k,y}\ M_k^{-1}\big(\widehat{A}_k(x_{k+1} - x^*) - \big(g_{2,k}^+(0) - g_{2,k}(0)\big) - \widehat{A}_k(x_{k+1} - x_k)\big) = \Lambda_{k,y}\ M_k^{-1}\big(\widehat{A}_k(x_{k+1} - x^*) + O(\|x_{k+1} - x_k\|^2)\big) \quad (4.23)$$

Because

$$x_{k+1} - x_k = \Delta x_k^G = -(S_k)^{-1}\ \big(g_{1,k} + \widehat{A}_k^T\ M_k^{-1}\ g_{2,k}(\mu_k)\big), \quad (4.24)$$

we conclude that

$$\|x_{k+1} - x_k\| = O(||g_k(\mu_k)||), \quad (4.25)$$

and thus, the second term in the right hand side of (4.23) is of order $O(\|w_k - w^*\|^2)$.

Moreover, we know by (4.21) that the first term in the right hand side of (4.23) is of order $o(\|w_k - w^*\|)$. This, together with the local variable component in (4.14), (4.16), (4.17), give

$$\|\hat{y}_{k+1} - \hat{y}^*\| = o(\|w_{k+1} - w^*\|). \quad (4.26)$$

Relationships (4.21) and (4.26) prove the result. □

5. Numerical Example. In this section, we illustrate the convergence results given in Section 4 by applying the decomposition algorithm introduced without and with the Gauss-Seidel refinement to solve a simple quadratic program taken from the test problem set proposed by DeMiguel and Murray in [15]. The quadratic program corresponds to an MDO problem with 200 variables and two systems.

Tables 5.1 and 5.2 display the performance of the interior-point decomposition algorithm without and with the Gauss-Seidel iteration, respectively. The first column denotes the iteration number, the second column shows the value of the barrier

parameter at the beginning of the iteration, the third column indicates the relative difference between the global component of the search directions of the decomposition and direct methods, the fourth column gives the relative difference in the local component, the fifth column shows the maximum step size, and the sixth column presents the norm of the KKT conditions at the end of the iteration.

TABLE 5.1
Interior-point decomposition algorithm without Gauss-Seidel refinement.

Iter	μ_k	$\frac{\|\Delta x_k^N - \Delta x_k^D\|}{\|\Delta x_k^N\|}$	$\frac{\|\Delta \hat{y}_k^N - \Delta \hat{y}_k^D\|}{\|\Delta \hat{y}_k^N\|}$	α_k	$\|g_k(0)\|$
1	1.0e-001	1.0e-015	6.8e-001	2.5e-001	2.6e+001
2	1.0e-002	5.8e-015	1.0e-001	3.7e-001	1.4e+001
3	1.0e-003	2.6e-015	1.8e-001	6.5e-001	6.6e+000
4	1.0e-004	6.2e-015	7.5e-002	5.3e-001	3.0e+000
5	1.0e-005	6.4e-015	1.1e-001	8.2e-001	8.4e-001
6	1.0e-006	6.4e-015	1.3e-001	9.8e-001	7.8e-002
7	1.0e-007	1.9e-014	4.9e-002	9.9e-001	4.4e-002
8	1.0e-008	9.5e-014	1.2e-002	1.0e+000	2.3e-003
9	1.0e-009	6.4e-012	3.0e-004	1.0e+000	2.8e-005
10	1.0e-010	8.5e-015	6.0e-002	1.0e+000	9.4e-009
11	8.9e-017	1.1e-014	5.8e-002	1.0e+000	1.2e-009
12	1.4e-018	9.2e-011	2.0e-005	1.0e+000	1.2e-010

TABLE 5.2
Interior-point decomposition algorithm with Gauss-Seidel refinement.

Iter	μ_k	$\frac{\|\Delta x_k^N - \Delta x_k^G\|}{\|\Delta x_k^N\|}$	$\frac{\|\Delta \hat{y}_k^N - \Delta \hat{y}_k^G\|}{\|\Delta \hat{y}_k^N\|}$	α_k	$\|g_k(0)\|$
1	1.0e-001	9.3e-016	1.2e-015	5.4e-001	1.7e+001
2	1.0e-002	1.3e-014	1.4e-015	7.2e-001	3.0e+000
3	1.0e-003	4.5e-015	1.7e-015	9.9e-001	2.0e-001
4	1.0e-004	2.3e-015	8.3e-015	1.0e+000	5.9e-003
5	1.0e-005	1.7e-015	2.1e-013	1.0e+000	7.8e-005
6	6.1e-009	5.5e-015	1.1e-011	1.0e+000	4.8e-008
7	2.3e-015	8.8e-015	1.8e-008	1.0e+000	8.0e-012

The results confirm our convergence analysis of previous sections. In particular, the local components of the direct method and the decomposition algorithm without Gauss-Seidel refinement are different. Moreover, the convergence of the decomposition algorithm without the Gauss-Seidel iteration appears to be only linear or perhaps two-step superlinear. On the other hand, the decomposition algorithm with Gauss-Seidel

refinement converges superlinearly, and both the global and local components of the search direction resemble those of the direct method search direction.

6. Conclusions. In this paper, we establish a relationship between a particular bilevel decomposition algorithm that only takes one iteration to solve the subproblems and a direct interior-point method. Using the insight gained from this relationship, we show how a Gauss-Seidel strategy can be used to ensure that the bilevel decomposition algorithm converges superlinearly.

To the best of our knowledge, this is the first local convergence proof for a bilevel decomposition algorithm that only takes one iteration to solve the subproblems. One may argue, however, that the particular case of bilevel decomposition algorithm analyzed here offers few (if any) practical advantages when compared with the direct Newton method. In particular, the level of decentralization provided by the analyzed decomposition algorithm is very similar to that provided by the direct method. But, in our opinion, our most important contribution is the connection we establish between the bilevel decomposition algorithms used in industry, which do allow for a high degree of decentralization, and direct interior-point methods. We think our work bridges the gap between the incipient local convergence theory of bilevel decomposition algorithms [2, 16] and the mature local convergence theory of direct interior-point methods [27, 18, 39, 23].

Finally, we show that bilevel decomposition algorithms that do not solve the subproblems exactly (or only take a step on the subproblems) are viable at least from a local convergence point of view. We hope our work will encourage researchers and practitioners alike to design and apply other bilevel decomposition approaches based on *inaccurate* subproblem solutions.

Acknowledgments. We would like to acknowledge comments from R.W. Cottle, M.P. Friedlander, F.J. Prieto, D. Ralph, S. Scholtes, and seminar participants at the SIAM Conference on Optimization (Toronto, 2002), INFORMS Annual Meeting (San Jose, 2002), Argonne National Laboratory, and Nortwestern University. This research was partially supported by the Research Development Fund at London Business School.

REFERENCES

[1] N.M. Alexandrov and M.Y. Hussaini, editors. *Multidisciplinary Design Optimization: State of the Art*, Philadelphia, 1997. SIAM.

[2] N.M. Alexandrov and R.M. Lewis. Analytical and computational aspects of collaborative optimization. Technical Report TM-2000-210104, NASA, 2000.

[3] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[4] A. Berkelaar, C .Dert, B. Oldenkamp, and S. Zhang. A primal-dual decomposition-based interior point approach to two-stage stochastic linear programming. *Operations Research*, 50(5):904–915, 2002.

[5] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, New York, 1997.

[6] J. Blomvall and P.O. Lindberg. A riccati-based primal interior point solver for multistage stochastic programming. *European Journal of Operational Research*, 143(2):452–461, 2002.

[7] J. Blomvall and P.O. Lindberg. A riccati-based primal interior point solver for multistage stochastic programming - Extensions. *Optimization methods and software*, 17(3):383–407, 2002.

[8] R.D. Braun and I.M. Kroo. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. In N.M. Alexandrov and M.Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, 1997.

[9] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large–scale nonlinear programming. *SIAM Journal on Optimization*, 9:877–900, 1999.

[10] B.J. Chun and S.M. Robinson. Scenario analysis via bundle decomposition. *Annals of Operations Research*, 56:39–63, 1995.

[11] G. Cohen and B. Miara. Optimization with an auxiliary constraint and decomposition. *SIAM Journal of Control and Optimization*, 28(1):137–157, 1990.

[12] E.J. Cramer, J.E. Dennis, P.D. Frank, R.M. Lewis, and G.R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, November 1994.

[13] A.V. DeMiguel. *Two Decomposition Algorithms for Nonconvex Optimization Problems with Global Variables*. PhD thesis, Stanford University, 2001.

[14] A.V. DeMiguel and W. Murray. An analysis of collaborative optimization methods. In *Eight AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000. AIAA Paper 00-4720.

[15] A.V. DeMiguel and W. Murray. A class of quadratic programming problems with global variables. Technical Report SOL 01-2, Dept. of Management Science and Engineering, Stanford University, 2001.

[16] A.V. DeMiguel and W. Murray. A local convergence analysis of bilevel programming decomposition algorithms. Working paper, London Business School, 2002.

[17] S. Dempe. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, Boston, 2002.

[18] A. S. El–Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of Newton interior–point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89:507–541, June 1996.

[19] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, 1968.

[20] A. Forsgren and P.E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

[21] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. Technical Report 97-4-08, Computing Sciences Research, Bell Laboratories, Murray Hill, NJ, 1997.

[22] A.M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[23] N.I.M. Gould, D. Orban, A. Sartenaer, and P.L. Toint. Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM Journal on Optimization*, 11(4):974–1002, 2001.

[24] T. Helgason and S.W. Wallace. Approximate scenario solutions in the progressive hedging algorithm. *Annals of Operations Research*, 31:425–444, 1991.

[25] G. Infanger. *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. Boyd and Fraser, Danvers, Mass., 1994.

[26] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, 1996.

[27] H. H. Martinez, Z. Parada, and R. A. Tapia. On the characterization of q-superlinear convergence of quasi-Newton interior-point methods for nonlinear programming. *SIAM Boletín de la Sociedad Matemática Mexicana*, 1:1–12, 1995.

[28] D. Medhi. Parallel bundle-based decomposition for large-scale structured mathematical programming problems. *Annals of Operations Research*, 22:101–127, 1990.

[29] S.M. Robinson. Bundle-based decomposition: Description and preliminary results. In A. Prekopa, J. Szelezsan, and B. Strazicky, editors, *System Modelling and Optimization*, Lecture Notes in Control and Information Sciences. Springer-Verlag, 1986.

[30] R.T. Rockafellar and R.J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[31] A. Ruszczynski. A regularized decomposition for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[32] A. Ruszczynski. On convergence of an augmented lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Research*, 20(3):634–656, 1995.

[33] K. Shimizu, Y. Ishizuka, and J.F. Bard. *Nondifferentiable and Two-Level Mathematical Programming*. Kluwer Academic Publishers, Boston, 1997.

[34] M.C. Steinbach. Structured interior point SQP methods in optimal control. *Zeitschrift fur Angewandte Mathematik und Mechanic*, 76(3):59–62, 1996.

[35] K. Tammer. The application of parametric optimization and imbedding for the foundation and realization of a generalized primal decomposition approach. In J. Guddat, H. Jongen, B. Kummer, and F. Nozicka, editors, *Parametric Optimization and Related Topics*,

volume 35 of *Mathematical Research*. Akademie-Verlag, Berlin, 1987.

[36] R. Van Slyke and R.J-B. Wets. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[37] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, Statistics and Operations Research, Princeton University, 1997.

[38] S.J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.

[39] H. Yamashita and H. Yabe. Superlinear and quadratic convergence of some primal–dual interior point methods for constrained optimization. *Mathematical Programming*, 75::377–397, 1996.