Computational Optimization and Applications, 21, 263–275, 2002

The Dual Active Set Algorithm and Its Application to Linear Programming

WILLIAM W. HAGER
Department of Mathematics, University of Florida, Gainesville, FL 32611, USA

Received August 15, 2001; Revised September 25, 2001

Abstract. The Dual Active Set Algorithm (DASA), presented in Hager, Advances in Optimization and Parallel Computing, P.M. Pardalos (Ed.), North Holland: Amsterdam, 1992, pp. 137–142, for strictly convex optimization problems, is extended to handle linear programming problems. Line search versions of both the DASA and the LPDASA are given.

Keywords: dual active set algorithm, linear programming, active set, line search, nonlinear programming

1. Introduction

In [6, 8, 9] we present the Dual Active Set Algorithm (DASA) and prove its convergence when a strict convexity assumption holds. In this approach, a series of subproblems are solved in which some of the inequality constraints are treated as equalities while other inequality constraints are ignored. Using the multipliers associated with the subproblem, the active and inactive sets are updated and the iteration repeats. The first paper [9] gives a local quadratic convergence result for a "full step" version of this algorithm applied to control problems in which the cost function is defined in terms of an integral. Also, in the very recent paper [1], convergence is established for the full step version in quadratic optimization problems where the matrix in the cost function has a diagonal dominance property. For general optimization problems, however, this full step version of the DASA may not converge.

The paper [6] gives an adjusted way for updating the active set which yields convergence in a finite number of steps when the cost function is strictly convex. The adjustment involves taking a partial step in the direction of the multipliers gotten from the subproblem, decreasing the active set size, and resolving the subproblem using this smaller active set. After a finite number of partial steps, each one decreasing the size of the active set, it is possible to take a full step; then the iteration repeats, terminating after a finite number of full steps.

The paper [8] focuses on problems with quadratic cost and linear equality and inequality constraints. In this case, the subproblems are equivalent to a linear system of equations, and changes in the active set amount to small rank changes in the matrix for the quadratic term. Changes in the Cholesky factors of a sparse matrix after these small rank changes can be computed efficiently using the techniques developed in [3, 4]. In [8] we apply the

DASA to quadratic network optimization, and compare it to other approaches. For linear programming, the strict convexity assumption used in [6, 8, 9] is not satisfied. Here we give a modification of the algorithm to handle linear programming problems, and we prove convergence in a finite number of steps. We also give line search versions of both the DASA and LPDASA.

We briefly compare the LP dual active set approach to both simplex and interior point approaches for the following linear programming problem:

$$\min \mathbf{c}^\mathsf{T}\mathbf{x} \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}, \ \ \mathbf{x} \geq \mathbf{0}. \tag{1}$$

In the dual active set approach, we start with a guess for the dual multiplier associated with the constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$, and we ascend the dual function, eventually obtaining a subset of the columns of $\mathbf{A}$ which contains $\mathbf{b}$, adjusted by the nonbasic variables, in its span. Either a linear combination of these columns of $\mathbf{A}$ yields an optimal solution to (1) (that satisfies the constraint $\mathbf{x} \geq \mathbf{0}$) and the iteration stops, or one or more of these columns are discarded, and the iteration repeats. Since the algorithm constantly ascends the dual function, the collection of columns obtained in each iteration does not repeat, and convergence is obtained in a finite number of steps. This finite convergence property is similar to that of the simplex method, where the iterates travel along edges of the feasible set, descending the cost function in a finite number of steps. Unlike the simplex method, neither rank nor nondegeneracy assumptions are either invoked or facilitate the analysis. In essence, one is able to prove finite convergence without any assumptions. In the simplex method, as presented in text books, typically one constraint is activated and one constraint is deactivated in each iteration. There has been some work on multiple pivots in a simplex context; for example, [5] considers a more general descent step that might be intermingled with simplex steps. With the LPDASA, constraints typically come and go in groups.

In interior point approaches, the iterates move through the relative interior of the feasible region. In the LPDASA, the iterates are infeasible, and the algorithm often terminates at a basic feasible solution for the linear programming problem. Each iteration of the interior point algorithm involves a scaled projection of the cost vector into the null space of $\mathbf{A}$. The LPDASA projects the constraint violation vector into the space orthogonal to the "free columns" of $\mathbf{A}$.

2. Dual active set algorithm

We begin with a statement of a new line-search version of the DASA followed by its convergence proof. Consider a problem of the form:

$$\max_{\lambda} \min_{\mathbf{x} \geq \mathbf{0}} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}), \tag{2}$$

where $\mathcal{L} : \mathbf{R}^m \times \mathbf{R}^n \to \mathbf{R}$. The inequality $\mathbf{x} \geq \mathbf{0}$ could be replaced by $\mathbf{1} \leq \mathbf{x} \leq \mathbf{u}$ with essentially no change in the analysis (see [6, 8]). We assume that $\mathcal{L}$ is continuously differentiable, that $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})$ is concave in $\boldsymbol{\lambda}$ for each fixed $\mathbf{x} \in \mathbf{R}^n$, and uniformly strongly convex in $\mathbf{x}$ for each fixed $\boldsymbol{\lambda} \in \mathbf{R}^m$. That is, there exists a constant $\alpha > 0$ such that

$$(\nabla_x \mathcal{L}(\boldsymbol{\lambda}, \mathbf{y}) - \nabla_x \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}))(\mathbf{y} - \mathbf{x}) \geq \alpha \|\mathbf{y} - \mathbf{x}\|^2,$$

where α is independent of $\boldsymbol{\lambda}$, $\mathbf{x}$, and $\mathbf{y}$.

If $B \subset \{1, 2, \ldots, n\}$, let $\mathbf{x}_B$ be the subvector of $\mathbf{x}$ consisting of those components x_i associated with $i \in B$. Two different functions enter into the statement of the DASA:

$$\mathcal{L}_B(\boldsymbol{\lambda}) = \min_{\mathbf{x}_B \geq 0} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}) \quad \text{and} \quad \mathcal{L}_B^0(\boldsymbol{\lambda}) = \min_{\mathbf{x}_B = 0} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}). \tag{3}$$

In carrying out the minimizations in (3), the components of $\mathbf{x}$ corresponding to indices in the complement of B are unconstrained. By the strong convexity of $\mathcal{L}(\boldsymbol{\lambda}, \cdot)$, there exists a unique minimizer $\mathbf{x}(\boldsymbol{\lambda}, B)$ over the set $\mathbf{x}_B \geq \mathbf{0}$, for each choice of $\boldsymbol{\lambda}$ and B. Since $\mathcal{L}$ is continuously differentiable and $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})$ is strongly convex in $\mathbf{x}$, $\mathbf{x}(\boldsymbol{\lambda}, B)$ depends continuously on $\boldsymbol{\lambda}$. To see this (well-known) property, suppose the $\mathbf{x}_i$ is the unique minimizer of $\mathcal{L}_B(\boldsymbol{\lambda}_i, \cdot)$ for $i = 1, 2$. We add together the first-order optimality conditions

$$\nabla_x \mathcal{L}_B(\boldsymbol{\lambda}_1, \mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) \geq 0 \quad \text{and} \quad \nabla_x \mathcal{L}_B(\boldsymbol{\lambda}_2, \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_1) \geq 0,$$

and rearrange to obtain

$$\alpha \|\mathbf{x}_1 - \mathbf{x}_2\| \leq \|\nabla_x \mathcal{L}_B(\boldsymbol{\lambda}_2, \mathbf{x}_2) - \nabla_x \mathcal{L}_B(\boldsymbol{\lambda}_1, \mathbf{x}_2)\|.$$

The continuity of $\mathbf{x}(\boldsymbol{\lambda}, B)$ with respect to $\boldsymbol{\lambda}$ follows directly.

The unique minimizer of (2) corresponding to $B = \{1, 2, \ldots, n\}$ in (3) is denoted $\mathbf{x}(\boldsymbol{\lambda})$. We let $\mathcal{L}(\boldsymbol{\lambda})$ denote the dual function $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}(\boldsymbol{\lambda}))$:

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}(\boldsymbol{\lambda})) = \min_{\mathbf{x} \geq 0} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}).$$

In the DASA, we start from an arbitrary $\boldsymbol{\lambda}_0$ and generate a finite sequence of iterates. If $\boldsymbol{\lambda}_k$ denotes the current iterate (initially $k = 0$), then either $\boldsymbol{\lambda}_k$ maximizes the dual function and we stop, or we move to the next iterate $\boldsymbol{\lambda}_{k+1}$ using a finite sequence of subiterates $\boldsymbol{\nu}_0 = \boldsymbol{\lambda}_k, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots$ The algorithmic steps are the following:

Dual Active Set Algorithm (with line search)

- *Convergence test*: If $\boldsymbol{\lambda}_k$ maximizes the dual function, then stop.
- *Dual initialization*: Set $j = 0, \boldsymbol{\nu}_0 = \boldsymbol{\lambda}_k, \quad B_0 = \{i : x_i(\boldsymbol{\lambda}_k) = 0\}$.
- *Dual subiteration*:

$$\boldsymbol{\mu}_j \in \arg\max_{\lambda} \mathcal{L}_{B_j}^0(\boldsymbol{\lambda}) \quad \text{and} \quad \boldsymbol{\nu}_{j+1} \in \arg\max_{\lambda \in [\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]} \mathcal{L}_{B_j}(\boldsymbol{\lambda}).$$

If there are multiple maxima on the line segment $[\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]$ connecting $\boldsymbol{\nu}_j$ and $\boldsymbol{\mu}_j$, then $\boldsymbol{\nu}_{j+1}$ should be the point closest to $\boldsymbol{\nu}_j$.

- *Constraint deletion*:

$$B_{j+1} = \{i \in B_j : y_i = 0,\ (\nabla_x \mathcal{L}(\boldsymbol{\nu}_{j+1}, \mathbf{y}))_i > 0\}$$

where $\mathbf{y} = \mathbf{x}(\boldsymbol{\nu}_{j+1}, B_j)$.
- *Stopping criterion*: If $\mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}^0_{B_j}(\boldsymbol{\mu}_j)$, then increment k, set $\boldsymbol{\lambda}_k = \boldsymbol{\nu}_{j+1}$, and go to convergence test. Otherwise, increment j and continue the dual subiteration.

Remark 1. Since $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})$ is concave in $\boldsymbol{\lambda}$, the dual function $\mathcal{L}(\boldsymbol{\lambda})$ is concave in $\boldsymbol{\lambda}$ and differentiable by [2, Theorem 2.1]. Hence, the dual function is maximized if and only if its derivative vanishes.

Remark 2. Implicitly, we assume in the dual subiteration that the maximum of $\mathcal{L}^0_B$ exists. For the LPDASA in the next section, this existence follows naturally. In the nonlinear case, the original $\mathcal{L}$ could be modified by a proximal term to ensure existence. See [8] for further discussion of existence.

Remark 3. In [6, 8] we took $\boldsymbol{\nu}_{j+1}$ to be the last point $\boldsymbol{\lambda}$ on the line segment $[\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]$ for which the following equality held:

$$\mathcal{L}^0_{B_j}(\boldsymbol{\lambda}) = \mathcal{L}_{Bj}(\boldsymbol{\lambda}) \quad \text{for all } \boldsymbol{\lambda} \in [\boldsymbol{\nu}_j, \boldsymbol{\nu}_{j+1}].$$

In this case B_{j+1} is often one element smaller than B_j. However, in practice it is often more efficient to perform a line search along the line segment $[\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]$, as is done in the version of the DASA given above.

Remark 4. In the full step version of the DASA mentioned in the introduction, we take $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\mu}_0$. The resulting algorithm is convergent in certain situation, as shown in [1, 9], but not in general.

Remark 5. For quadratic programming problems, the iterates in successive dual subiterations are the solutions of linear systems whose matrices differ by a rank k matrix, where k is the size of the set $B_{j+1} \backslash B_j$. The methods developed in [3, 4] show how the Cholesky factorization of a sparse symmetric, positive definite matrix changes after either a rank one [3] or a multiple rank [4] change in the matrix.

Remark 6. All the indices contained in the final B_j of the dual subiteration will be included in the initial B_0 of the next iteration. Hence, the dual initialization step adds indices to the current B set, while the constraint deletion step removes indices.

We now give a convergence proof for the line-search version of the DASA given above.

Theorem 1. *Assume that $\mathcal{L}$ is continuously differentiable with $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})$ uniformly strongly convex in $\mathbf{x}$ for each fixed $\boldsymbol{\lambda} \in \mathbf{R}^m$, and concave in $\boldsymbol{\lambda}$ for each fixed $\mathbf{x} \in \mathbf{R}^n$. If in each step of the DASA, a maximizer $\boldsymbol{\mu}_j$ in the dual subiteration exists, then the DASA generates a solution of* (2) *in a finite number of iterations.*

Proof: We first show that B_{j+1} is strictly contained in B_j for each j when the stopping criterion is not satisfied. The proof is by contradiction. If $B_{j+1} = B_j$, then $\mathcal{L}^0_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1})$. Since $\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})$ is concave in $\boldsymbol{\lambda}$, it follows that $\mathcal{L}^0_{B_j}(\boldsymbol{\lambda})$ is concave in $\boldsymbol{\lambda}$. Hence, if $\mathcal{L}^0_{B_j}(\boldsymbol{\mu}_j) = \mathcal{L}^0_{B_j}(\boldsymbol{\nu}_j)$, then since $\boldsymbol{\mu}_j$ maximizes $\mathcal{L}^0_{B_j}$, we have $\mathcal{L}^0_{B_j}(\boldsymbol{\lambda}) = \mathcal{L}^0_{B_j}(\boldsymbol{\mu}_j)$ for each $\boldsymbol{\lambda} \in [\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]$. In particular, $\mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}^0_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}^0_{B_j}(\boldsymbol{\mu}_j)$, and the stopping criterion is satisfied. Now suppose that $\mathcal{L}^0_{B_j}(\boldsymbol{\mu}_j) > \mathcal{L}^0_{B_j}(\boldsymbol{\nu}_{j+1})$. Again, since $\mathcal{L}^0_{B_j}(\boldsymbol{\lambda})$ is concave in $\boldsymbol{\lambda}$,

$$\mathcal{L}^0_{B_j}(\boldsymbol{\lambda}) > \mathcal{L}^0_{B_j}(\boldsymbol{\nu}_{j+1}) \tag{4}$$

when $\boldsymbol{\lambda} \in (\boldsymbol{\nu}_{j+1}, \boldsymbol{\mu}_j]$. Since $\mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}^0_{B_j}(\boldsymbol{\nu}_{j+1})$, $(\nabla_x \mathcal{L}(\boldsymbol{\nu}_{j+1}, y))_i > 0$ for each $i \in B_{j+1}$ where $\mathbf{y} = \mathbf{x}(\boldsymbol{\nu}_{j+1}, B_j)$, and $\mathbf{x}(\boldsymbol{\lambda}, B_j)$ depends continuously on $\boldsymbol{\lambda}$, it follows that $x_i(\boldsymbol{\lambda}, B_j) = 0$ for all $i \in B_j$ and $\boldsymbol{\lambda}$ near $\boldsymbol{\nu}_{j+1}$, and hence,

$$\mathcal{L}_{B_j}(\boldsymbol{\lambda}) = \mathcal{L}^0_{B_j}(\boldsymbol{\lambda}) \tag{5}$$

for $\boldsymbol{\lambda}$ near $\boldsymbol{\nu}_{j+1}$. Together, (4) and (5) contradict the fact the $\boldsymbol{\nu}_{j+1}$ maximizes $\mathcal{L}_{B_j}$ over the line segment $[\boldsymbol{\nu}_j, \boldsymbol{\mu}_j]$. Hence, B_{j+1} is strictly contained in B_j when the stopping criterion is not satisfied, and the dual subiteration will eventually stop.

Let C_k denote the final set B_j at iteration k. We now show that if the derivative of the dual function does not vanish at $\boldsymbol{\lambda}_k$, then $C_k \neq C_l$ for all $l < k$. Since there are a finite number of distinct choices for C_k, the DASA reaches a stationary point in a finite number of steps. Since the dual function is concave, this stationary point is a maximizer.

Recall that $\mathbf{x}_k = \mathbf{x}(\boldsymbol{\lambda}_k)$ is the solution to the problem

$$\min_{\mathbf{x} \geq \mathbf{0}} \mathcal{L}(\boldsymbol{\lambda}_k, \mathbf{x}).$$

Since the first-order optimality conditions are both necessary and sufficient for optimality when $\mathcal{L}(\boldsymbol{\lambda}, \cdot)$ is convex (see [10, Chap. 7]), we have

$$\mathcal{L}(\boldsymbol{\lambda}_k) = \mathcal{L}(\boldsymbol{\lambda}_k, \mathbf{x}_k) = \mathcal{L}_{B_0}(\boldsymbol{\lambda}_k) = \mathcal{L}_{B_0}(\boldsymbol{\nu}_0). \tag{6}$$

For the same reason, we have

$$\mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1}) = \mathcal{L}_{B_{j+1}}(\boldsymbol{\nu}_{j+1}) \tag{7}$$

for each $j \geq 0$. Since $\boldsymbol{\nu}_{j+1}$ is obtained from a line search, $\mathcal{L}_{B_j}(\boldsymbol{\nu}_j) \leq \mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1})$. Combining this with (6) and (7) gives

$$\mathcal{L}(\boldsymbol{\lambda}_k) \leq \mathcal{L}_{B_j}(\boldsymbol{\nu}_j) \leq \mathcal{L}_{B_j}(\boldsymbol{\nu}_{j+1}) \leq \mathcal{L}_{B_{j+1}}(\boldsymbol{\nu}_{j+1}) \tag{8}$$

for each $j \geq 0$. This implies that

$$\mathcal{L}(\boldsymbol{\lambda}_k) \leq \mathcal{L}_{C_k}(\boldsymbol{\lambda}_{k+1}) \leq \mathcal{L}(\boldsymbol{\lambda}_{k+1}). \tag{9}$$

The final inequality here is due to the fact that the optimization problem associated with the evaluation of $\mathcal{L}(\boldsymbol{\lambda}_{k+1})$ involves more constraints than the corresponding optimization problem for $\mathcal{L}_{C_k}(\boldsymbol{\lambda}_{k+1})$. We now show that the first inequality in (9) is strict.

If $\boldsymbol{\nu}_0 \neq \boldsymbol{\nu}_1$, then since $\boldsymbol{\nu}_1$ is the first maxima on the interval $[\boldsymbol{\nu}_0, \boldsymbol{\mu}_0]$, it follows that $\mathcal{L}_{B_0}(\boldsymbol{\nu}_1) > \mathcal{L}_{B_0}(\boldsymbol{\nu}_0)$, in which case the first inequality in (9) is strict by (8). Now suppose that $\boldsymbol{\nu}_0 = \boldsymbol{\nu}_1 = \boldsymbol{\lambda}_k$. By [2, Theorem 2.1], $\mathcal{L}'_{B_1}(\boldsymbol{\lambda}_k) = \nabla_\lambda \mathcal{L}(\boldsymbol{\lambda}_k, \mathbf{x}_k) = \mathcal{L}'(\boldsymbol{\lambda}_k)$. Hence, if $\mathcal{L}'(\boldsymbol{\lambda}_k) \neq 0$ and $\boldsymbol{\nu}_0 = \boldsymbol{\nu}_1 = \boldsymbol{\lambda}_k$ we have

$$\mathcal{L}^0_{B_1}(\boldsymbol{\mu}_1) > \mathcal{L}^0_{B_1}(\boldsymbol{\nu}_1) = \mathcal{L}(\boldsymbol{\lambda}_k). \tag{10}$$

Again, by the concavity of $\mathcal{L}^0_B$, (10) implies that

$$\mathcal{L}^0_{B_1}(\boldsymbol{\lambda}) > \mathcal{L}^0_{B_1}(\boldsymbol{\nu}_1) \tag{11}$$

for $\boldsymbol{\lambda} \in (\boldsymbol{\nu}_1, \boldsymbol{\mu}_1]$. Since $\mathbf{x}(B_1, \boldsymbol{\lambda})$ depends continuously on $\boldsymbol{\lambda}$, it follows from the definition of B_1 that

$$\mathcal{L}^0_{B_1}(\boldsymbol{\lambda}) = \mathcal{L}_{B_1}(\boldsymbol{\lambda})$$

for $\boldsymbol{\lambda}$ near $\boldsymbol{\nu}_1$. Combining this with (11), we see that $\mathcal{L}_{B_1}(\boldsymbol{\lambda}) > \mathcal{L}_{B_1}(\boldsymbol{\nu}_1)$, $\boldsymbol{\lambda} \in (\boldsymbol{\nu}_1, \boldsymbol{\mu}_1]$, $\boldsymbol{\lambda}$ near $\boldsymbol{\nu}_1$. As a consequence, $\mathcal{L}_{B_1}(\boldsymbol{\nu}_2) > \mathcal{L}_{B_1}(\boldsymbol{\nu}_1)$. Again, the first inequality in (9) is strict due to (8). In summary, the first inequality in (9) is strict whenever the derivative of the dual function does not vanish at $\boldsymbol{\lambda}_k$. Since $\mathcal{L}^0_{C_k}(\boldsymbol{\lambda}_{k+1}) = \max_\lambda \mathcal{L}^0_{C_k}(\boldsymbol{\lambda})$, it follows from (9) and the stopping criterion that

$$\max_{\boldsymbol{\lambda}} \mathcal{L}^0_{C_l}(\boldsymbol{\lambda}) < \max_{\boldsymbol{\lambda}} \mathcal{L}^0_{C_k}(\boldsymbol{\lambda})$$

for each $l < k$. Hence, $C_l \neq C_k$ for $l < k$, and the proof is complete. □

3. LP dual active set algorithm

For the linear program (1), $\mathcal{L}$ has the following special form:

$$\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}) = \mathbf{c}^\mathsf{T}\mathbf{x} + \boldsymbol{\lambda}^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{x}).$$

For a linear program, unlike the strictly convex setting appearing in the previous section, the dual function achieves the value $-\infty$ when $\boldsymbol{\lambda}$ is "dual infeasible." One approach for dealing with an infeasible starting guess would be a phase one process, similar to what is done in implementations of the simplex method for linear programming. However, in the LPDASA context, another practical approach for dealing with dual infeasibility is to simply introduce large upper bounds $\mathbf{u}$. In other words, impose the constraint $\mathbf{x} \le \mathbf{u}$. For $\mathbf{u}$ sufficiently large, this additional constraint does not effect the optimal value of the linear program, and the choices for $\boldsymbol{\lambda}$ which were previously dual infeasible now yield small (but finite) values for the dual function. Typically, a small number of steps of the LPDASA generates a $\boldsymbol{\lambda}$ that is dual feasible for the orginal LP (1). Hence, in this section we focus on linear programs in the following form:

$$\min \mathbf{c}^{\mathsf{T}}\mathbf{x} \quad \text{subject to } \mathbf{Ax} = \mathbf{b},\ \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \tag{12}$$

where $\mathbf{l} < \mathbf{u}$, with the components of $\mathbf{l}$ and $\mathbf{u}$ finite. If a dual feasible solution of (1) is available as a starting guess in the LPDASA, then we could allow components of $\mathbf{l}$ and $\mathbf{u}$ to take the values $\pm\infty$. These infinite value never enter into any of the steps of the algorithm since the value of the dual function increases.

In the context of (12), the dual function becomes

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_{\mathbf{l} \le \mathbf{x} \le \mathbf{u}} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}). \tag{13}$$

By the usual LP duality theory, we know that minimizing the primal problem (12) is, in a certain sense, equivalent to maximizing the dual function. In particular, if (12) has a solution, then the dual function $\mathcal{L}$ has a maximizer and $\max_{\lambda} \mathcal{L}(\boldsymbol{\lambda})$ is equal to the minimum in (12). In applying the DASA to the dual function, we need to take into account the fact that the minimizers in (13) are typically not unique, and the dual function $\mathcal{L}$ is typically not differentiable. Instead, $\mathcal{L}$ has a subgradient $\partial\mathcal{L}$ (see [11]) consisting of a set of vectors:

$$\partial\mathcal{L}(\boldsymbol{\lambda}) = \left\{\mathbf{b} - \mathbf{Ax} : \mathbf{x} \in \arg \min_{\mathbf{l} \le \mathbf{x} \le \mathbf{u}} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x})\right\}.$$

The dual variable $\boldsymbol{\lambda}$ maximizes $\mathcal{L}$ if and only if $\mathbf{0} \in \partial\mathcal{L}(\boldsymbol{\lambda})$, or equivalently, $\mathbf{Ax} = \mathbf{b}$ for some $\mathbf{x}$ that attains the minimum in (13).

The algorithmic steps of the LPDASA parallel the steps of the DASA. We start from an arbitrary $\boldsymbol{\lambda}_0$ and generate a finite sequence of iterates. If $\boldsymbol{\lambda}_k$ denotes the current iterate (initially $k = 0$), then either $\boldsymbol{\lambda}_k$ maximizes the dual function and we stop, or we move to the next iterate $\boldsymbol{\lambda}_{k+1}$ using a finite sequence of subiterates $\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots$ as follows:

LP Dual Active Set Algorithm

- *Convergence test*: Choose $\mathbf{y} \in \arg\ \min_{\mathbf{l} \le \mathbf{x} \le \mathbf{u}} \mathcal{L}(\boldsymbol{\lambda}_k, \mathbf{x})$, and let $\mathbf{z}$ denote any solution to the problem

$$\min \|\mathbf{Ax} - \mathbf{b}\| \quad \text{subject to } \mathbf{x}_B = \mathbf{y}_B,\ \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \tag{14}$$

  where $B = \{i : (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i \ne 0\}$. If $\mathbf{Az} = \mathbf{b}$, then $\mathbf{z}$ is a solution to (12) and $\boldsymbol{\lambda}_k$ maximizes the dual function. Otherwise, define $\boldsymbol{\mu}(t) = \boldsymbol{\lambda}_k + t(\mathbf{b} - \mathbf{Az})$, and let $\bar{t}$ be the smallest $t > 0$ for which $(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i = 0$ for some $i \in B$ if it exists; otherwise put $\bar{t} = +\infty$, and (12) is infeasible (no choice for $\mathbf{x}$ satisfies the constraints).
- *Dual initialization*: Set $j = 0$, $\boldsymbol{\nu}_0 = \boldsymbol{\mu}(\bar{t})$, and $B_0 = \{i : (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\nu}_0)_i \ne 0\}$.
- *Dual subiteration*: Let $\mathbf{p}$ be the orthogonal projection of $\mathbf{b} - \mathbf{A}_{B_j}\mathbf{z}_{B_j}$ into the null space of $\mathbf{A}_{F_j}^{\mathsf{T}}$, where F_j is the complement of B_j. Define $\boldsymbol{\mu}(t) = \boldsymbol{\nu}_j + t\mathbf{p}$, and let $\bar{t}$ be the smallest

$t > 0$ such that $(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i = 0$ for some $i \in B_j$, if it exists; otherwise put $\bar{t} = +\infty$, and (12) is infeasible.

- *Constraint deletion*: If $\bar{t} < \infty$, set

$$B_{j+1} = \{i \in B_j : (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(\bar{t}))_i \neq 0\}.$$

- *Stopping criterion*: If $\mathbf{p} = \mathbf{0}$, then increment k, set $\boldsymbol{\lambda}_k = \boldsymbol{\nu}_j$, and go to the convergence test. Otherwise, increment j, set $\boldsymbol{\nu}_j = \boldsymbol{\mu}(\bar{t})$, and continue the dual subiteration.

The version of the LPDASA given above takes the largest step in the search direction $\mathbf{p}$ for which $\mathcal{L}^z_{B_j}$ matches $\mathcal{L}_{B_j}$ where

$$\mathcal{L}_B(\boldsymbol{\lambda}) = \min_{\mathbf{l}_B \le \mathbf{x}_B \le \mathbf{u}_B} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}) \quad \text{and} \quad \mathcal{L}^z_B(\boldsymbol{\lambda}) = \min_{\mathbf{x}_B = \mathbf{z}_B} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}).$$

This is the same stepsize discussed in Remark 3 for the DASA. The line search version of the LPDASA, analogous to the line search version of the DASA in Section 2, is given later.

The LP analogue of Theorem 1 is the following:

Theorem 2. *In a finite number of iterations, the LPDASA either determines that* (12) *is infeasible, or it obtains an optimal solution.*

Theorem 2 is obtained from the union of Lemmas 1–3 which follow.

Lemma 1. *If* $\mathbf{Az} = \mathbf{b}$ *where* $\mathbf{z}$ *is a solution to* (14), *then* $\mathbf{z}$ *is an optimal solution to the linear program* (12) *and* $\boldsymbol{\lambda}_k$ *maximizes the dual function* $\mathcal{L}$.

Proof: By the first-order optimality conditions associated with $\mathbf{y}$, we have

$$\begin{aligned}
(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i &= 0 \quad \text{if } l_i < y_i < u_i,\\
(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i &\ge 0 \quad \text{if } y_i = l_i,\\
(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i &\le 0 \quad \text{if } y_i = u_i.
\end{aligned}$$

Thus we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\lambda}_k) &= \mathbf{c}^{\mathsf{T}}\mathbf{y} + \boldsymbol{\lambda}_k^{\mathsf{T}}(\mathbf{b} - \mathbf{Ay})\\
&= (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)^{\mathsf{T}}\mathbf{y} + \mathbf{b}^{\mathsf{T}}\boldsymbol{\lambda}_k\\
&= \left(\mathbf{c}_B - \mathbf{A}_B^{\mathsf{T}}\boldsymbol{\lambda}_k\right)^{\mathsf{T}}\mathbf{y}_B + \mathbf{b}^{\mathsf{T}}\boldsymbol{\lambda}_k\\
&= \left(\mathbf{c}_B - \mathbf{A}_B^{\mathsf{T}}\boldsymbol{\lambda}_k\right)^{\mathsf{T}}\mathbf{z}_B + \mathbf{b}^{\mathsf{T}}\boldsymbol{\lambda}_k\\
&= (\mathbf{c}_B - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)^{\mathsf{T}}\mathbf{z} + \mathbf{b}^{\mathsf{T}}\boldsymbol{\lambda}_k\\
&= \mathbf{c}^{\mathsf{T}}z + \boldsymbol{\lambda}_k^{\mathsf{T}}(\mathbf{b} - \mathbf{Az})\\
&= \mathbf{c}^{\mathsf{T}}\mathbf{z}.
\end{aligned}$$

Since $\mathbf{z}$ is feasible for the primal problem (12), and the value of the primal equals the value of the dual, we conclude that $\mathbf{z}$ is optimal in the primal and $\boldsymbol{\lambda}_k$ maximizes the dual function. □

Lemma 2. *If $\bar{t}$ is infinite*, *then* (12) *is infeasible.*

Proof: In the LPDASA, there are two places where $\bar{t}$ is computed, in the convergence test and in the subiteration.

Case 1 (Convergence Test). Let $\mathbf{A}_i$ denote column i of $\mathbf{A}$ and let F denote the complement of B. By the first-order necessary conditions for the solution $\mathbf{z}$ of (14), the following relations hold:

$$\left.\begin{array}{ll} (\mathbf{Az}-\mathbf{b})^{\mathsf{T}}\mathbf{A}_i = 0 & \text{if } i \in F,\ l_i < z_i < u_i, \\ (\mathbf{Az}-\mathbf{b})^{\mathsf{T}}\mathbf{A}_i \geq 0 & \text{if } i \in F,\ z_i = l_i, \\ (\mathbf{Az}-\mathbf{b})^{\mathsf{T}}\mathbf{A}_i \leq 0 & \text{if } i \in F,\ z_i = u_i. \end{array}\right\} \tag{15}$$

By the definition of B,

$$(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i = 0 \quad \text{for each } i \in F. \tag{16}$$

Combining (15) and (16), the following relations hold for each $t \geq 0$:

$$\left.\begin{array}{ll} (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i = 0 & \text{if } i \in F,\ l_i < z_i < u_i, \\ (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i \geq 0 & \text{if } i \in F,\ z_i = l_i, \\ (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i \leq 0 & \text{if } i \in F,\ z_i = u_i. \end{array}\right\} \tag{17}$$

Since $\mathbf{z}_B = \mathbf{y}_B$ where

$$\mathbf{y} \in \arg\min_{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}),$$

the first-order necessary conditions and the definition of B imply that

$$\left.\begin{array}{ll} (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i > 0 & \text{if } i \in B,\ z_i = l_i, \\ (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\lambda}_k)_i < 0 & \text{if } i \in B,\ z_i = u_i. \end{array}\right\} \tag{18}$$

When $\bar{t}$ is infinite, the inequalities in (18) also hold when $\boldsymbol{\lambda}_k$ is replaced by $\boldsymbol{\mu}(t)$ for any $t \geq 0$:

$$\left.\begin{array}{ll} (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i > 0 & \text{if } i \in B,\ z_i = l_i, \\ (\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i < 0 & \text{if } i \in B,\ z_i = u_i. \end{array}\right\} \tag{19}$$

Combining (17) and (19) yields for any $t \geq 0$:

$$(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i = 0 \quad \text{if } l_i < z_i < u_i,$$
$$(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i \geq 0 \quad \text{if } z_i = l_i,$$
$$(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\mu}(t))_i \leq 0 \quad \text{if } z_i = u_i.$$

Consequently, we have

$$\mathbf{z} \in \arg\min\{\mathbf{c}^{\mathsf{T}}\mathbf{x} + \boldsymbol{\mu}(t)^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{x}) : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \tag{20}$$

for each $t \geq 0$. Putting $\mathbf{x} = \mathbf{z}$ in the extremand of (20) gives us:

$$\mathcal{L}(\boldsymbol{\mu}(t)) = \mathbf{c}^{\mathsf{T}}\mathbf{z} + (\boldsymbol{\lambda}_k + t(\mathbf{b} - \mathbf{A}\mathbf{z}))^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{z}) = \mathcal{L}(\boldsymbol{\lambda}_k) + t\|\mathbf{b} - \mathbf{A}\mathbf{z}\|^2. \tag{21}$$

If $\mathbf{A}\mathbf{z} \neq \mathbf{b}$, then $\mathcal{L}(\boldsymbol{\mu}(t))$ tends to infinity as t increases. Since any feasible point in the primal problem yields an upper bound for the dual function, we conclude that the primal problem is infeasible.

Case 2 (Subiteration). Since the signs of the components of the vector $(\mathbf{c} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\nu}_j)$ do not change during the subiteration, $\mathbf{z}$ satisfies (20) during the subiteration. Since $\mathbf{p}$ is the projection of $\mathbf{b} - \mathbf{A}_{B_j}\mathbf{z}_{B_j}$ into the null space of $\mathbf{A}_{F_j}^{\mathsf{T}}$, we conclude that

$$\mathbf{p}^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{z}) = \mathbf{p}^{\mathsf{T}}(\mathbf{b} - \mathbf{A}_{B_j}\mathbf{z}_{B_j}) = \mathbf{p}^{\mathsf{T}}\mathbf{p}.$$

Hence, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mu}(t)) &= \mathbf{c}^{\mathsf{T}}\mathbf{z} + (\boldsymbol{\nu}_j + t\mathbf{p})^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{z}) \\ &= \mathbf{c}^{\mathsf{T}}\mathbf{z} + \boldsymbol{\nu}_j^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{z}) + t\mathbf{p}^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{z}) \\ &= \mathcal{L}(\boldsymbol{\nu}_j) + t\mathbf{p}^{\mathsf{T}}\mathbf{p}. \end{aligned} \tag{22}$$

As t tends to infinity, the dual function $\mathcal{L}$ tends to infinity; again, the primal problem is infeasible. □

Since B_{j+1} is strictly contained in B_j for each j, the subiterations in the LPDASA terminate in a finite number of steps. Let $\mathbf{z}_k$ denote the solution of (14) employed at iteration k, and let C_k denote the final set B_j generated in the subiteration. As a consequence of the next lemma, the LPDASA reaches an optimal solution of (11), when it exists, in a finite number of iterations.

Lemma 3. *$(C_i, (\mathbf{z}_k)_{C_i}) \neq (C_j, (\mathbf{z}_k)_{C_j})$ for each $i \neq j$.*

Proof: Let us consider the optimization problem

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{x}_B = \mathbf{z}_B} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}). \tag{23}$$

Since the minimum in (23) is $-\infty$ unless $\mathbf{A}_F^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{c}_F$, where F is the complement of B, this problem is equivalent to

$$\max_{\lambda} \mathcal{L}(\boldsymbol{\lambda}, \mathbf{z}) \quad \text{subject to } \mathbf{A}_F^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{c}_F. \tag{24}$$

Discarding the constant term $\mathbf{c}_B^{\mathsf{T}}\mathbf{z}_B$ from the extremand of (24) yields the equivalent problem

$$\max_{\lambda} \boldsymbol{\lambda}^{\mathsf{T}}(\mathbf{b} - \mathbf{A}_B\mathbf{z}_B) \quad \text{subject to } \mathbf{A}_F^{\mathsf{T}}\boldsymbol{\lambda} = \mathbf{c}_F. \tag{25}$$

Assuming (25) is feasible, the maximum is $+\infty$ unless $\mathbf{b} - \mathbf{A}_B\mathbf{z}_B$ is orthogonal to the null space of $\mathbf{A}_F$; and if $\mathbf{b} - \mathbf{A}_B\mathbf{z}_B$ is orthogonal to the null space of $\mathbf{A}_F$, then any $\boldsymbol{\lambda}$ satisfying the constraint of (25) is an optimal solution. Returning to the final LP dual subiteration, $\mathbf{b} - \mathbf{A}_{C_k}\mathbf{z}_{C_k}$ is orthogonal to the null space of $\mathbf{A}_{F_k}^{\mathsf{T}}$, where F_k is the complement of C_k, and $\mathbf{A}_{C_k}^{\mathsf{T}}\boldsymbol{\lambda}_{k+1} = \mathbf{c}_{F_k}$. Hence, $\boldsymbol{\lambda}_{k+1}$ achieves the maximum in (23) corresponding to $F = F_k$ and by the equivalence between (23) and (24), the maximum value is $\mathcal{L}(\boldsymbol{\lambda}_{k+1}, \mathbf{z})$. During the proof of Lemma 2, we saw that for each subiterate,

$$\mathbf{z} \in \arg\min \left\{\mathbf{c}^{\mathsf{T}}\mathbf{x} + \boldsymbol{\nu}_j^{\mathsf{T}}(\mathbf{b} - \mathbf{A}\mathbf{x}) : \mathbf{l} \le \mathbf{x} \le \mathbf{u}\right\}.$$

In particular, replacing $\boldsymbol{\nu}_j$ by $\boldsymbol{\lambda}_{k+1}$, it follows that $\mathcal{L}(\boldsymbol{\lambda}_{k+1}) = \mathcal{L}(\boldsymbol{\lambda}_{k+1}, \mathbf{z})$, the optimal value in (23). Since $\bar{t}$ is strictly positive, relations (21) and (22) imply that

$$\mathcal{L}(\boldsymbol{\lambda}_k) < \mathcal{L}(\boldsymbol{\nu}_0) < \mathcal{L}(\boldsymbol{\nu}_1) < \cdots < \mathcal{L}(\boldsymbol{\lambda}_{k+1}).$$

Hence, the optimal value in (23) corresponding to $B = C_k$ is strictly increasing as a function of k, which implies that the pair $(C_k, (\mathbf{z}_k)_{C_k})$ never repeats. □

Since $(\mathbf{z}_k)_i = l_i$ or u_i for each $i \in C_k$, Lemma 3 implies finite termination of the LPDASA. The line search version of the LPDASA is basically the same as the orginal LPDASA, except for the line search feature and the treatment of $\mathbf{z}$.

LP Dual Active Set Algorithm (with line search)

- *Convergence test*: Choose $\mathbf{y} \in \arg\min_{\mathbf{l}\le\mathbf{x}\le\mathbf{u}} \mathcal{L}(\boldsymbol{\lambda}_k, \mathbf{x})$, and let $\mathbf{z}$ denote any solution to the problem

  $$\min \ \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \quad \text{subject to } \mathbf{x}_B = \mathbf{y}_B, \quad 1 \le \mathbf{x} \le \mathbf{u},$$

  where $B = \{i : (\mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda}_k)_i \ne 0\}$. If $\mathbf{A}\mathbf{z} = \mathbf{b}$, then $\mathbf{z}$ is a solution to (12) and $\boldsymbol{\lambda}_k$ maximizes the dual function. Otherwise, define $\boldsymbol{\mu}(t) = \boldsymbol{\lambda}_k + t(\mathbf{b} - \mathbf{A}\mathbf{z})$, and let $\bar{t}$ be the first maximizer, if it exists, of $\mathcal{L}_B(\boldsymbol{\mu}(t))$ over $t > 0$; otherwise put $\bar{t} = +\infty$, and (12) is infeasible.

If $\bar{t} < \infty$, define

$$\bar{z}_i = \begin{Bmatrix} l_i & \text{if } (\mathbf{c} - \mathbf{A^T}\boldsymbol{\mu}(\bar{t}))_i > 0, \\ u_i & \text{if } (\mathbf{c} - \mathbf{A^T}\boldsymbol{\mu}(\bar{t}))_i < 0, \\ z_i & \text{otherwise.} \end{Bmatrix} \tag{26}$$

- *Dual initialization*: Set $j = 0$, $\boldsymbol{\nu}_0 = \boldsymbol{\mu}(\bar{t})$, $\mathbf{z} = \bar{\mathbf{z}}$, and

$$B_0 = \{i : (\mathbf{c} - \mathbf{A^T}\boldsymbol{\nu}_0)_i \neq 0\}$$

- *Dual subiteration*: Let $\mathbf{p}$ be the orthogonal projection of $\mathbf{b} - \mathbf{A}_{B_j}\mathbf{z}_{B_j}$ into the null space of $\mathbf{A}_j^T$, where F_j is the complement of B_j. Define $\boldsymbol{\mu}(t) = \boldsymbol{\nu}_j + t\mathbf{p}$, and let $\bar{t}$ be the first maximizer, if it exists, of $\mathcal{L}_B(\boldsymbol{\mu}(t))$; over $t > 0$; otherwise put $\bar{t} = +\infty$ and (12) is infeasible. If $\bar{t} < \infty$, set $\bar{\mathbf{z}}$ according to (26).
- *Constraint deletion*: If $\bar{t} < \infty$, set

$$B_{j+1} = \{i \in B_j : (\mathbf{c} - \mathbf{A}^T\boldsymbol{\mu}(\bar{t}))_i \neq 0\}$$

- *Stopping criterion*: If $\mathbf{p} = 0$, then increment k, set $\boldsymbol{\lambda}_k = \boldsymbol{\nu}_j$, and go to the convergence test. Otherwise, increment j, set $\boldsymbol{\nu}_j = \boldsymbol{\mu}(\bar{t})$, set $\mathbf{z} = \bar{\mathbf{z}}$, and continue the dual subiteration.

Remark 7. The function $\mathcal{L}_{B_j}(\boldsymbol{\mu}(t))$ is piecewise linear in t, and the $\bar{t}$ given in the original statement of the LPDASA corresponds to the first kink in the graph. In the line search version of the LPDASA, we go beyond this first kink, and find the maximum of the piecewise linear function $\mathcal{L}_{B_j}(\boldsymbol{\mu}(\cdot))$. The convergence proof for the line search version is nearly the same as the convergence proof for the original version. Lemma 1 is unchanged. Lemma 2 becomes a triviality since $\mathcal{L}_B(\boldsymbol{\lambda}) \le \mathcal{L}(\boldsymbol{\lambda})$ for any choice of B. Hence, if $\mathcal{L}_B$ tends to infinity for some choice of B, then $\mathcal{L}$ is unbounded as well. Finally, consider Lemma 3 and the nonrepetition of the sets C_k. As in Section 2, the iterates satisfy relations (8) and (9); in fact, in the LP context, the second inequality in (9) is an equality:

$$\mathcal{L}(\boldsymbol{\lambda}_k) \le \mathcal{L}_{C_k}(\boldsymbol{\lambda}_{k+1}) = \mathcal{L}(\boldsymbol{\lambda}_{k+1}). \tag{27}$$

From the proof of Lemma 2, we know that $\bar{t} > 0$ and $\mathcal{L}(\boldsymbol{\nu}_0) = \mathcal{L}_{B_0}(\boldsymbol{\nu}_0) > \mathcal{L}(\boldsymbol{\lambda}_k)$ when optimality has not yet been achieved. As a result, the first inequality in (27) is strict. In the proof of Lemma 3, we show that $\boldsymbol{\lambda}_{k+1}$ maximizes $\mathcal{L}_{C_k}^{z_k}$, where $\mathbf{z}_k$ is the final $\bar{\mathbf{z}}$ at iteration k, and

$$\mathcal{L}(\boldsymbol{\lambda}_{k+1}) = \max_{\boldsymbol{\lambda}} \mathcal{L}_{C_k}^{z_k}(\boldsymbol{\lambda}).$$

For $i \in C_k$, $(\mathbf{z}_k)_i$ is equal to either l_i or u_i. Since C_k and $(\mathbf{z}_k)_i$ for $i \in C_k$ are chosen from a finite set, we again achieve convergence in a finite number of iterations.

References

1. M. Bergounioux and K. Kunisch, "Primal-dual strategy for state-constrained optimal control problems," Computational Optimization and Applications, to appear.
2. F.H. Clarke, "Generalized gradients and applications," Transactions of the American Mathematical Society, vol. 205, pp. 247–262, 1975.
3. T.A. Davis and W.W. Hager, "Modifying a sparse Cholesky factorization," SIAM Journal on Matrix Analysis and Applications, vol. 20, pp. 606–627, 1999.
4. T.A. Davis and W.W. Hager, "Multiple rank modifications of a sparse Cholesky factorization," SIAM Journal on Matrix Analysis and Applications, vol. 22, pp. 997–1013, 2001.
5. P.E. Gill and W. Murray, "A numerically stable form of the simplex algorithm," Linear Algebra and Its Applications, vol. 7, pp. 99–138, 1973.
6. W.W. Hager, "The dual active set algorithm," in Advances in Optimization and Parallel Computing, P.M. Pardalos (Ed.), North Holland: Amsterdam, 1992, pp. 137–142.
7. W.W. Hager, "The LP dual active set algorithm," in High Performance Algorithms and Software in Nonlinear Optimization, R. De Leone, A. Murli, P.M. Paradalos, and G. Toraldo (Eds.), Kluwer: Dordrecht, 1998, pp. 243–254.
8. W.W. Hager and D. Hearn, "The dual active set algorithm applied to quadratic networks," Computational Optimization and Applications, vol. 1, pp. 349–373, 1993.
9. W.W. Hager and G. Ianculescu, "Dual approximations in optimal control," SIAM J. Control Optim., vol. 22, pp. 423–465, 1984.
10. O.L. Mangasarian, Nonlinear Programming, McGraw-Hill: New York, 1969.
11. R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton: NJ, 1970.