An Infeasible Interior-Point Algorithm for Solving Primal and Dual Geometric Programs *

K. O. Kortanek
Dept. of Management Science
The University of Iowa, Iowa City, IA 52242, USA

Xiaojie Xu
Institute of Systems Science, Academia Sinica
Beijing 100080, China

Yinyu Ye
Dept. of Management Science
The University of Iowa, Iowa City, IA 52242, USA

February 8, 1996

Abstract

In this paper an algorithm is presented for solving the classical posynomial geometric programming dual pair of problems simultaneously. The approach is by means of a primal–dual infeasible algorithm developed simultaneously for (i), the dual geometric program after logarithmic transformation of its objective function and (ii), its Lagrangian dual program. Under rather general assumptions, the mechanism defines a primal–dual infeasible path from a specially constructed, perturbed Karush–Kuhn–Tucker system. *Subfeasible* solutions, Duffin[1956], are generated for each program whose primal and dual objective function values converge to the respective primal and dual program values.

The basic technique is one of a predictor–corrector type involving Newton's method applied to the perturbed KKT system, coupled with effective techniques for choosing iterate directions and step lengths. We also discuss implementation issues and some sparse matrix factorizations that take advantage of the very special structure of the Hessian matrix of the logarithmically transformed dual objective function.

Our computational results on 19 of the most challenging *GP* problems found in the literature are encouraging. The performance indicates that the algorithm is effective regardless of the *degree of difficulty*, which is a generally accepted measure in geometric programming.

Key words: Geometric programming, convex analysis, infeasible primal-dual algorithm, duality, subfeasible solutions, numerical implementation.

*Research supported in part by the University of Iowa Obermann Fellowship and by NSF Grant DDM-9207347.

1 Introduction

Posynomial geometric programming GP is a very broad class of mathematical problems which is useful in the study of a variety of optimization problems. Its great impact has been in the areas of (1) **engineering design** : [11],[5], [17], and [18]; (2) **economics & statistics** : [8], [36], [7], [23],[3], and [6]; (3) **manufacturing**: [24], [13],[38], and (4) **chemical equilibrium** : [11], [5], [34]). The familiar *primal signomial* GP is :

$$
\begin{array}{llll}
(GP) \quad V_{GP} := & \text{minimize} & g_0(t) & \\
 & \text{subject to} & g_k(t) \leq 1, & k = 1, 2, ..., p \\
 & & t_i > 0, & i = 1, 2, ..., m
\end{array} \tag{1}
$$

where

$$
g_0(t) = \sum_{i=1}^{n_0} c_i t_1^{a_{i1}} ... t_m^{a_{im}} \tag{2}
$$

$$
g_k(t) = \sum_{i=n_{k-1}+1}^{n_k} c_i t_1^{a_{i1}} ... t_m^{a_{im}}, \quad k = 1, 2, ..., p. \tag{3}
$$

Exponents a_{ij} are arbitrary real constants and coefficients c_i are positive. In this paper we restrict attention to *posynomial* GP , and so we shall delete *posynomial* with this understanding. We shall use the classical notation [11], $\sum_{[0]}$ for the summation in $g_0(t)$ and $\sum_{[k]}$ for the summation in $g_k(t)$ whenever possible.

It is well known that GP is transformably convex via $z_i = log(t_i)$, and hence convex programming algorithms apply to this formulation, including cutting planes, penalty methods, and more. As a convex program, there is, of course, the classical Lagrangian dual, and primal–dual algorithms are also applicable under certain regularity assumptions. In general, there are a large number of duality states between a convex program and its Lagrangian dual, and these are all realizeable in finite dimensions, see [26]. However, in GP the number of duality states is reduced and yet there are more than the classical 4 duality states of ordinary linear programming, LP. These duality states are best characterized with the classical, arithmetic/geometric inequality–based *dual geometric programming problem*, see [20]. However, unlike ordinary LP we know of no algorithmic developments designed to characterize which of the possible duality states prevails for a given dual pair of geometric programming problems.

It has been also known that GP is sensitive to perturbation procedures that may for example, arise in solution methods. To cite an important reference, [15],

> A careless perturbation may eliminate dual optimal solutions and make it impossible to solve the original problem.

A successful attempt at developing a stable perturbation method appears in [43].

Actually, this paper also develops a class of perturbations, but ones that are not in general feasible. Specifically we propose an infeasible interior–point algorithm for linear constrained convex programming which when applied to geometric programming solves both primal and dual GP problems simultaneously. Using the perturbed KKT system of the dual GP, we define an "infeasible" central– path which converges to a solution of both the primal and dual GP programs. The algorithm does not require the existence of an interior point for either program. Moreover, the algorithm has the feature of generating *subfeasible solutions*, [12, 11], when the primal geometric program is inconsistent but subconsistent. Computing functional values of subfeasible solutions gives rise to more conceivable duality states for the primal-dual pair. In this paper we limit these possibilities by considering "boundedly subfeasible" sequences for which our algorithm delivers a path whose primal-dual gap can be made arbitrarily small, even for the case where one of the pair of dual problem is inconsistent. In addition, the objective functions need not be differentiable at an optimal solution of a given consistent program.

Each iteration of the algorithm computes the Newton direction from a perturbed KKT system involving two parameters, γ and η. The affine direction (corresponding to a particular setting $\gamma = 0, \eta = 1$) is used to predict reductions in both the feasibility and complementarity residuals. Then proper parameters γ, η are chosen for the KKT system which is solved for the direction again. The indefinite reduced KKT matrix is factorized by performing diagonal pivots whose orders are chosen in terms of minimum degree to maintain the sparsity of the factors. A static data structure associated with the chosen pivoting order is allocated by a single symbolic factorization and used in subsequent iterations. We also develop tactics for taking large steps and reset dual slack variables according to certain rules.

Our computational results indicate that the algorithm, coupled with these new techniques, leads to an efficient and stable implementation for solving primal and dual geometric programs simultaneously. Competitive numerical results are presented for some of the most challenging posynomial geometric programming problems in the literature. The results indicate that the standard geometric programming measure of difficulty is no barrier to the methods we propose.

The paper is organized as follows. In Section 2 we state the dual GP formulation and review the special block structure of the Hessian of the negative of the logarithm of the dual GP objective function, which is not differentiable at any vector having zero coordinate. In Section 3 we review the duality of general linear constrained convex programming and the Wolfe dual, together with the concept of subfeasible solutions for each program. In this section we also introduce a perturbed KKT system in a constructive way, specifically for defining an "infeasible" primal–dual central path. We introduce the concept of boundedly subfeasible solutions and study their duality gap behavior. We show how to recover the primal GP solution in Section 4. In Section 5, we state our infeasible path–following algorithm as a predictor–corrector algorithm and present techniques for choosing directions and step lengths. In Section 6 we discuss some sparse matrix techniques for the reduced KKT system and provide specifications for initial points and the setting of various parameters employed in our implementation. Computational results on 19 GP problems in the literature are reported in Section 7. Finally, Section 8 contains our concluding remarks.

2 The dual geometric programming formulation

A specific procedure is used to construct the dual GP, [11]:

$$(GD) \quad V_{GD} := \quad \begin{array}{ll} \text{maximize} & \prod_{i=1}^{n_p} (c_i/x_i)^{x_i} \prod_{k=1}^{p} \lambda_k^{\lambda_k} \\ \text{subject to} & \sum_{[0]} x_i = 1, \\ & \sum_{i=1}^{n_p} x_i a_{ij} = 0, \quad j = 1, 2, ..., m \\ & x_i \geq 0, \quad i = 1, 2, ..., n_p \end{array} \tag{4}$$

where

$$\lambda_k = \sum_{[k]} x_i, \ \ k = 1, 2, ..., p. \tag{5}$$

For a GP having m x_i variables, p constraints and n_p posynomial terms we see that GD has n_p non–negative variables in $m+1$ linear equations. In the literature the *degree of difficulty* of a GP is defined by :

$$\text{degree of difficulty} = n_p - m - 1. \tag{6}$$

Let $F(x)$ denote the negative of the logarithm of the objective function of (GD), i.e.,

$$F(x) = \sum_{[0]} x_i \ln(\frac{x_i}{c_i}) + \sum_{k=1}^{p} \sum_{[k]} x_i \ln(\frac{x_i}{c_i \lambda_k}). \tag{7}$$

We see that (GD) has the structure of a linearly constrained convex programming problem,

$$\begin{array}{ll} \text{minimize} & F(x) \\ \text{subject to} & Ax = b, \quad x \geq 0, \end{array}$$

where the coefficient matrix is given by

$$A = \begin{pmatrix} 1 & ... & 1 & 0 & ... & 0 & ... & 0 & ... & 0 \\ a_{1,1} & ... & a_{n_0,1} & a_{n_0+1,1} & ... & a_{n_1,1} & ... & a_{n_{p-1}+1,1} & ... & a_{n_p,1} \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ a_{1,m} & ... & a_{n_0,m} & a_{n_0+1,m} & ... & a_{n_1,m} & ... & a_{n_{p-1}+1,m} & ... & a_{n_p,m} \end{pmatrix}, \tag{8}$$

and right side hand

$$b^T = (1, 0, ..., 0) \in R^{m+1}. \tag{9}$$

The gradient vector $\nabla F(x)$ is :

$$\frac{\partial F(x)}{\partial x_i} = \left\{ \begin{array}{ll} \ln(\frac{x_i}{c_i}) + 1, & \text{for } i \in [0] \\ & \\ \ln(\frac{x_i}{c_i \lambda_k}), & \text{for } i = n_0 + 1, ..., n_p. \end{array} \right\}, \tag{10}$$

and so one can verify that $F(x)$ is infinitely differentiable on $R^n_{++} := \{x \in R^n | \ x > 0\}$ and convex continuous on $R^n_{+} := \{x \in R^n | \ x \geq 0\}$, the non-negative orthant of R^n, where $(c_i/0)^0$ is defined by continuity, i.e.,

$$\lim_{x_i \to 0} (c_i/x_i)^{x_i} = 1.$$

An elementary equation will be useful.

$$x^T \nabla F(x) = F(x) + \sum_{[0]} x_i. \tag{11}$$

But one can also see that $F(x)$ is not differentiable at any $x \geq 0$ having a zero coordinate, which distinguishes (GD) from ordinary linearly constrained convex programming problems, and this is not a superficial distinction. The dual GP to the well–known Dembo example has a unique optimal solution occurring at $\hat{x}^*$ having two zero coordinates, and so ∇F does not exist at the optimal solution. For easy reference we give Dembo's primal GP next, [10] :

$$(DP) \qquad \min \{t_1 t_2 + t_1^{-1} t_2^{-1} |\ (1/4) t_1^{1/2} + t_2 \leq 1,\ t_1 > 0,\ t_2 > 0\}.$$

For convenience to the reader, we give the GP dual to this, while observing that it has a unique optimal solution, $x = (0.5, 0.5, 0, 0)$.

$$\begin{array}{llll} (DD) & \max & (1/x_1)^{x_1}(1/x_2)^{x_2}(0.25/x_3)^{x_3}(1/x_4)^{x_4}(x_3+x_4)^{(x_3+x_4)} & \\ & \text{subject to} & x_1 + x_2 & = 1 \\ & & x_1 - x_2 + 0.5 x_3 & = 0 \\ & & x_1 - x_2 \qquad\qquad + x_4 & = 0 \\ & & x \geq 0 & \end{array}$$

Additional properties of F are useful for algorithmic development. As shown in [11], the Hessian $\nabla^2 F(x)$ possesses the following diagonal block structure:

$$\nabla^2 F(x) = \begin{pmatrix} H_0 & & & \\ & H_1 & & \\ & & \ldots & \\ & & & H_p \end{pmatrix} \tag{12}$$

where

$$H_0 = \begin{pmatrix} \frac{1}{x_1} & & & \\ & \frac{1}{x_2} & & \\ & & \ldots & \\ & & & \frac{1}{x_{n_0}} \end{pmatrix} \tag{13}$$

and

$$H_k = \begin{pmatrix} \frac{1}{x_{n_{k-1}+1}} - \frac{1}{\lambda_k} & -\frac{1}{\lambda_k} & \ldots & -\frac{1}{\lambda_k} \\ -\frac{1}{\lambda_k} & \frac{1}{x_{n_{k-1}+2}} - \frac{1}{\lambda_k} & \ldots & -\frac{1}{\lambda_k} \\ \ldots & \ldots & \ldots & \ldots \\ -\frac{1}{\lambda_k} & -\frac{1}{\lambda_k} & \ldots & \frac{1}{x_{n_k}} - \frac{1}{\lambda_k} \end{pmatrix} \text{ for } k = 1, 2, \ldots, p. \tag{14}$$

$\nabla^2 F(x)$ is positive semi-definite; it is singular when (GP) has at least one constraint, i.e., $k \geq 1$, see [27].

3 Linearly constrained interior–differentiable convex programming

Motivated by properties reviewed in the previous section, we consider in general the following linearly constrained interior–differentiable convex program.

$$(P) \quad V_P := \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b, \quad x \geq 0 \end{array}$$

under the following general assumptions :

(A1) $A \in R^{m \times n}$, $b \in R^m$, $x \in R^n$.

(A2) $f : R^n_+ \to R$ is a convex continuous function.

(A3) On R^n_{++} f has continuous partial derivatives.

(A4) $-\infty < V_P < \infty$, where we adopt the convention that $V_P = \infty$ if and only if (P) is inconsistent.

Now the classical Wolfe dual to (P) is a special case of the Lagrangian dual and is the following :

$$(D) \quad V_D := \begin{array}{lllll} \text{Maximize} & b^T y & -x^T \nabla f(x) & +f(x) & \\ \text{subject to} & A^T y & -\nabla f(x) & +z & = 0 \\ & y \text{ free}, & x > 0, & z \geq 0 & \end{array}$$

where $y \in R^m$, $z \in R^n$ and ^{T} denotes transpose.

Observe that there are two dual pairs of programs : $\{(GP) : (GD)\}$ and $\{(P) : (D)\}$. Fig. 1 reviews the relationship between these dual pairs, presenting the more general Lagrangian dual for (D), and how (D) "connects" to primal (GP).

Definition 1 *Let*

$$(x, y, z) \in R^n_{++} \times R^m \times R^n_+.$$

Define the primal and dual feasibility residuals respectively by:

$$\begin{aligned} r_P(x) &:= b - Ax, && (15) \\ r_D(x, y, z) &:= \nabla f(x) - A^T y - z, && (16) \end{aligned}$$

and the the complementarity residual by

$$\mu(x, z) := x^T z / n. \qquad (17)$$

The duality gap between (P) and (D) is

$$x^T \nabla f(x) - b^T y \qquad (18)$$

Infeasible solutions to $(P)[(D)]$ which are arbitrarily close to being feasible can generally yield a different objective function value than the one for $(P)[(D)]$. This led to the following classical definition, whose original terminology we continue to use, see [12] and [11].

Definition 2

- *(P) is **subconsistent** iff there exists $\{x^k\}_k$ in R^n_+, termed a **subfeasible solution** for (P), such that*

$$\lim_{k\to\infty} r_P(x^k) = 0$$

The set of subfeasible solutions is denoted $\mathrm{AF(P)}$, *and the **subvalue** of (P) is*

$$\bar{V}_{\mathrm{P}} := inf_{\{x^k\}_k\in \mathrm{AF(P)}} \liminf_k f(x^k)$$

For (P) not subconsistent, $\bar{V}_{\mathrm{P}} = \infty$. A subfeasible solution is optimal if it yields $\bar{V}_{\mathrm{P}}$.

- *Similarly the set of **subfeasible solutions** for (D), denoted $AF(D)$, are those*

$$\{x^k, y^k, z^k\}_k \in R^n_{++} \times R^m \times R^n_+$$

such that

$$\lim_{k\to\infty} \nabla r_D(x^k, y^k, z^k) = 0$$

*with **dual subvalue***

$$\bar{V}_{\mathrm{D}} := sup_{\{x^k,y^k,z^k\}_k\in \mathrm{AF(D)}} \limsup_k (b^T y - \nabla f(x^k)^T x^k - f(x^k))$$

For (D) not subconsistent, $\bar{V}_{\mathrm{D}} = -\infty$. A subfeasible solution is optimal if it yields $\bar{V}_{\mathrm{D}}$.

Remark 1 *Subconsistency also applies to (GP) and (GD), with subconsistency of the latter identical to (P) because its constraints are linear. For (GP)–subconsistency we refer to [11, page 166], and we shall denote the subvalue of (GP) analogously by $\bar{V}_{\mathrm{GP}}$.*

Theorem 1

$$\bar{V}_{\mathrm{P}} = V_{\mathrm{D}} \;\le\; V_{\mathrm{P}} = \bar{V}_{\mathrm{D}}$$

Rockafellar [37, Corollary 30.2.2, page 315], while for geometric programming,

$$\bar{V}_{\mathrm{GP}} = V_{\mathrm{GD}} \tag{19}$$

(Duffin, Peterson and Zener [11, Theorem 1 page 166]

It is immediate from our assumptions $(A1) - (A4)$ that this Theorem implies that (D) is subconsistent, simply because $\bar{V}_D < \infty$. We state this as a corollary.

Corollary 1 *Under assumptions (A1)-(A4) Program (D) is subconsistent.*

The theorem states that in general, the objective function value of one program equals the subvalue of its dual program. See Kortanek [26] for a classification of the duality states that can occur between all occasions of values and subvalues with respect to a general convex program and its Lagrangian dual.

Remark 2 . *Without loss of generality we could replace "min" with "inf" in (21), see also Fiacco [16]. A finitely generated convex cone is a closed set, [37, Theorem 19.1], and therefore (P) is consistent if and only if it is subconsistent. However, a simple geometric programming example appears in [19, pages 437–438], where for (P), it is the case that $V_P = 0$ and $\bar{V}_P = -\infty$. Additional convex programming examples where $V_P \neq \bar{V}_P$ can be found in Goldstein [21] and Rockafellar [37].*

The paper develops both primal and dual infeasible but subfeasible paths by examining a perturbed KKT system. We chose this structure in order to introduce subfeasible solutions in a constructive way for both primal and dual programs simultaneously. Our development begins with the KKT system associated with (P) and (D) using definition 1.

$$\begin{array}{lccl} KKT & r_P(x) & = & 0 \\ & r_D(x,y,z) & = & 0 \\ & Xz & = & 0 \\ x > 0, & y \text{ free}, & z \geq 0, & \end{array}$$

where $X = \mathrm{diag}(x)$. Note that the system contains a nonlinear term $\nabla f(x)$. We call the first two equations the feasibility conditions and the third equation the complementarity condition.

A primal–dual interior–point algorithm was developed to solve the KKT system in [28], where the authors assumed that an interior point $(x^0 > 0, y^0, z^0 > 0)$ exists which satisfies the feasibility conditions, $r_P(x^0) = 0$ and $r_D(x^0, y^0, z^0) = 0$.

In this paper, we analyze an infeasible interior–point algorithm which attains the feasibility and complementarity conditions simultaneously. Let $(x^0 > 0, y^0, z^0 > 0)$ with $X^0 z^0 = \mu^0 e$ be an initial point and denote the initial residuals by

$$\mu^0 = \mu(x^0, z^0), \quad r_P^0 = r_P(x^0), \quad r_D^0 = r_D(x^0, y^0, z^0).$$

In this paper we choose $y^0 = 0$, $x^0 = e$, and $z^0 = \mu^0 e$ such that $z^0 - \nabla f(x^0) > 0$. The main structure underlying this extension is the following perturbed KKT system

$$\begin{array}{lccl} PKKT & r_P(x) & = & \theta\, r_P^0 \\ & r_D(x,y,z) & = & \theta\, r_D^0 \\ & Xz & = & \mu e \\ x > 0, & y \text{ free}, & z \geq 0, & \end{array} \tag{20}$$

where $0 < \theta \leq 1$. Observe that solutions to (20) as $\theta \downarrow 0$ are subfeasible.

Actually, the system PKKT is the KKT system of the following logarithmic barrier subproblem:

$$\begin{array}{lll} P(r_D^0, r_P^0, \theta, \mu) & \min & f(x) - \theta (r_D^0)^T x - \mu \sum_{i=1}^n \ln x_i \\ & \text{s.t.} & Ax = b - \theta r_P^0, \quad \text{and } x \geq 0. \end{array} \tag{21}$$

Associated with $P(r_D^0, r_P^0, \theta, \mu)$ is the following level set:

$$\mathcal{L}_\tau = \{x \in R^n |\ Ax = b - \theta r_P^0, \quad x \geq 0 \text{ and } f(x) - \theta (r_D^0)^T x \leq \tau\} \tag{22}$$

It is known that PKKT system has a unique solution if and only if any non-empty level set $\mathcal{L}_\tau$ is bounded, or if and only if $\mathcal{L}_\tau$ has no nonzero direction of recession, see Kortanek and Zhu [30, Proposition 1.1 and 1.2] and Fiacco [16, Section 2]. See Rockafellar [37, pages 66–70] for a definition of the recession function $(f0^+)$ of f. $\mathcal{L}_\tau$ having no nonzero direction of recession is **equivalent** to

$$(f0^+)(\bar{x}) - \theta \bar{x}^T r_D^0 \leq 0, \quad A\bar{x} = 0, \quad \text{and } \bar{x} \geq 0 \quad \text{imply } \bar{x} = 0, \tag{23}$$

Proposition 1 . *Under Assumptions (A1) − (A4), system PKKT has a unique (x, z) for any $\theta \in (0, 1]$ and $\mu \in (0, \mu^0]$. Furthermore, if A has a full row rank, then y is also unique.*

Proof Let $\hat{x}$ be feasible for (P). It follows directly from setting

$$x(\theta) = \theta x^0 + (1-\theta)\hat{x}$$

that $P(r_D^0, r_P^0, \theta, \mu)$ in (21) is consistent for any $\theta \in (0, 1]$. It suffices to verify (23). Let

$$\bar{x} \geq 0, \quad A\bar{x} = 0, \quad \text{and } (f0^+)(\bar{x}) - \theta \bar{x}^T r_D^0 \leq 0.$$

We must show that $\bar{x} = 0$. By Corollary 1, (D) is subconsistent. Therefore, let $\{y^k, x^k, z^k\}_k$ be an subfeasible sequence for (D), and note $z^0 > \nabla f(x^0)$, i.e. $z^0 - \nabla f(x^0) \in R^n_{++}$.

As employed in [30], we use a simple inequality between the recession function $(f0^+)$ and the gradient, i.e.,

$$(f0^+)(\bar{x}) - \theta \bar{x}^T r_D^0 \geq \nabla f(x^k)^T \bar{x} - \theta \bar{x}^T r_D^0, \tag{24}$$

for each k, see [37, Theorems 8.5 and 23.1]. But using $A\bar{x} = 0$, (23) and (24) we have

$$0 \geq \nabla f(x^k)^T \bar{x} - \theta \bar{x}^T (\nabla f(x^0) - z^0) \quad \text{for } k = 1, 2, ... \tag{25}$$

Adding $-\bar{x}^T z^k \leq 0$ to both sides of (25), using $A\bar{x} = 0$, and taking limit inferiors yields

$$0 \geq \underline{\lim}_k -\bar{x}^T z^k \geq \underline{\lim}_k \bar{x}^T(-A^T y^k + \nabla f(x^k) - z^k) - \bar{x}^T(\nabla f(x^0) - z^0). \tag{26}$$

But the limit on the right side of (26) is zero by subfeasibility as well as $(z^0 - \nabla f(x^0)) \in R^n_{++}$. Hence it must be the case that $\bar{x} = 0$. Uniqueness follows from the strict convexity of the objective function of (21) together with the existence of solutions to (21).

Q.E.D.

Remark 3 *According to Proposition 1, we do not, in particular, require the existence of a strictly positive point for (P), which is often a requirement of interior point methods.*

An important implication of proposition 1 is that the perturbed KKT system (20) defines a two-parameter $(\theta > 0, \mu > 0)$ surface, $x(\theta, \mu), y(\theta, \mu), z(\theta, \mu)$, starting from $x^0 > 0, y^0, z^0 > 0$. Such a surface for linear programming has been studied by Mizuno et al. [31]. Now we set $\mu = \theta \mu^0$ and $0 < \theta \leq 1$ in (20) to obtain

$$\begin{array}{rcl} r_P(x(\theta)) & = & \theta\, r_P^0 \\ r_D(x(\theta), y(\theta), z(\theta)) & = & \theta\, r_D^0 \\ X(\theta) z(\theta) & = & \mu e \\ x(\theta) > 0, & y(\theta) \text{ free}, & z(\theta) \geq 0, \end{array} \tag{27}$$

Analogous to the duality gap of (P) and (D) given in (18), (27) determines the **subfeasible duality gap**

$$x^T \nabla f(x) - b^T y = \theta \eta(x, y, z) \quad \textit{where} \quad \eta(x, y, z) = (x^T r_D^0 - y^T r_P^0 + x^T z). \tag{28}$$

Related to work of Kojima et.al. [25] and Nazareth [32], (27) defines an "infeasible" central path. Thus, given any fixed positive $\epsilon > 0$, by reducing θ we can always generate an ϵ-KKT point, $(x > 0, y, z \geq 0)$, such that

$$\|r_P(x)\| \leq \epsilon \|r_P^0\|, \quad \|r_D(x, y, z)\| \leq \epsilon \|r_D^0\|, \quad \text{and} \quad x^T z \leq \epsilon |(x^0)^T z^0|.$$

The remaining question is whether or not the path converges to optimal solutions to (P) and (D), as $\theta \to 0$. First, we introduce another definition.

Definition 3 *Problems (P) and (D) are said to be* **boundedly subconsistent** *if there exists a* **bounded** *sequence* $(\hat{x}^k > 0, \hat{y}^k, \hat{z}^k \geq 0)$, *termed* ***boundedly subfeasible***, *such that*

$$r_P(\hat{x}^k) \to 0, \quad r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k) \to 0, \quad \textit{and} \quad \mu(\hat{x}^k, \hat{z}^k) \to 0.$$

An important implication of the existence of a boundedly subfeasible sequence $\{(\hat{x}^k, \hat{y}^k, \hat{z}^k)\}_k$ is the following result.

Proposition 2 . *Let* $\{(\hat{x}^k, \hat{y}^k, \hat{z}^k)\}_k$ *be boundedly subfeasible, according to Definition 3, and let* $\hat{x}^*$ *be any limit point of* $\{x^k\}_k$. *Then* $\hat{x}^*$ *is optimal for (P) and* $\{(\hat{x}^k, \hat{y}^k, \hat{z}^k)\}_k$ *is a subfeasible optimal solution for (D) with the same objective function value as* (P).

Proof Without loss of generality, we assume $\lim_k \ \hat{x}^k = \hat{x}^*$. From the convexity of f we have

$$f(\hat{x}^k) + \nabla f(\hat{x}^k)^T(\hat{x}^* - \hat{x}^k) \leq f(\hat{x}^*), \quad \text{for each} \quad k. \tag{29}$$

Using the convergence of $r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k)$ to zero, we observe $\lim_k (\hat{x}^*)^T r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k) = 0$. We include this fact with (29) to obtain

$$\limsup_k [f(\hat{x}^k) + \nabla f(\hat{x}^k)^T(\hat{x}^* - \hat{x}^k) + (\hat{x}^*)^T r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k)] \leq f(\hat{x}^*), \tag{30}$$

Upon simplification (30) becomes

$$\limsup_k [b^T \hat{y}^k - \nabla f(\hat{x}^k)^T \hat{x}^k + f(\hat{x}^k)] \leq f(\hat{x}^*), \tag{31}$$

using also $(\hat{x}^*)^T \hat{z}^k \geq 0$. On the other hand, note that

$$\begin{aligned} (\hat{x}^k)^T r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k) &= -(\hat{y}^k)^T A \hat{x}^k + \nabla f(\hat{x}^k)^T \hat{x}^k - (\hat{z}^k)^T \hat{x}^k \\ &= -(\hat{y}^k)^T b + (\hat{y}^k)^T \hat{r}_P^k + \nabla f(\hat{x}^k)^T \hat{x}^k - (\hat{z}^k)^T \hat{x}^k. \end{aligned}$$

Since $\hat{x}^k$ and $\hat{y}^k$ are bounded, we must have

$$\lim_k \, [b^T \hat{y}^k - \nabla f(\hat{x}^k)^T \hat{x}^k] = 0,$$

i.e., (31) holds as an equality. Hence, $\hat{x}^*$ solves (P) and $\{(\hat{x}^k, \hat{y}^k, \hat{z}^k)\}_k$ is an subfeasible optimal solution for (D).

Q.E.D.

Proposition 2 is a convergence result deduced from the existence of any boundedly subfeasible solution. We now use this result to show that the specific subfeasible duality gap of the central path, (28), converges to 0.

Proposition 3 *If* (P) *and* (D) *are boundedly subconsistent, then the central path points*

$$(x(\theta), z(\theta)) \quad \textit{and the product} \quad y(\theta)^T r_P^0$$

are bounded for all $0 < \theta \leq 1$. *Moreover, the subfeasible duality gap (28) converges to zero.*

Proof For simplicity, let $(x, y, z) = (x(\theta), y(\theta), z(\theta))$ for $0 < \theta \le 1$. First, from relation (27) we derive

$$(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) = (x^0 - x)^T(z^0 - z) + (1-\theta)(x^0 - x)^T r_D^0 - (1-\theta)(y^0 - y)^T r_P^0. \quad (32)$$

Second, let $(\hat{x}^k > 0, \hat{y}^k, \hat{z}^k \ge 0)$ be any bounded sequence according to Definition 3. Then, again from relation (27), and from the $\mathbf{R}^n_{++}-$ differentiability and convexity of f, it follows that

$$0 \le (x - \hat{x}^k)^T(\nabla f(x) - \nabla f(\hat{x}^k)) = (x - \hat{x}^k)^T(z - \hat{z}^k) + (x - \hat{x}^k)^T(\theta r_D^0 - \hat{r}_D^k) - (y - \hat{y}^k)^T(\theta r_P^0 - \hat{r}_P^k).$$

Let $(\hat{x}^*, \hat{y}^*, \hat{z}^*)$ be a limit point of $(\hat{x}^k > 0, \hat{y}^k, \hat{z}^k \ge 0)$. Note that $(\hat{x}^*)^T\hat{z}^* = 0$. Then taking limit over k on both sides, we have

$$0 \le (x - \hat{x}^*)^T(z - \hat{z}^*) + \theta(x - \hat{x}^*)^T r_D^0 - \theta(y - \hat{y}^*)^T r_P^0. \quad (33)$$

Thus, multiplying (32) by θ and (33) by $(1-\theta)$, and combining the two inequalities we have

$$\theta(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) \le$$
$$\theta(x^0 - x)^T(z^0 - z) + (1-\theta)(x - \hat{x}^*)^T(z - \hat{z}^*) + \theta(1-\theta)((x^0 - \hat{x}^*)^T r_D^0 - (y^0 - \hat{y}^*)^T r_P^0)$$

or

$$\begin{aligned} &\theta(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) + \theta(x^T z^0 + z^T x^0) + (1-\theta)(x^T\hat{z}^* + z^T\hat{x}^*) && (34)\\ \le\ & \theta(x^0)^T z^0 + x^T z + \theta(1-\theta)[(x^0 - \hat{x}^*)^T r_D^0 - (y^0 - \hat{y}^*)^T r_P^0]. && (35) \end{aligned}$$

Dividing (34) by θ and noting $x^T z = n\mu^0\theta$, and recognizing $(x^T\hat{z}^* + z^T\hat{x}^*) \ge 0$, we have

$$\begin{aligned} &(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) + (x^T z^0 + z^T x^0)\\ &\quad \le (x^0)^T z^0 + n\mu^0 + (1-\theta)((x^0 - \hat{x}^*)^T r_D^0 - (y^0 - \hat{y}^*)^T r_P^0). \end{aligned}$$

Since its right-hand side is bounded and clearly both

$$(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) \ge 0 \quad \text{and} \quad (x^T z^0 + z^T x^0) > 0,$$

we have

$$(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) \le M^0 \quad \text{and} \quad (x^T z^0 + z^T x^0) \le M^0 \quad (36)$$

for some M^0 independent of θ. Thus, (x, z) is bounded since $x^0 > 0$ and $z^0 > 0$. Furthermore, from equation (32)

$$(1-\theta)(-x^T r_D^0 + y^T r_P^0) =$$
$$(x^0 - x)^T(\nabla f(x^0) - \nabla f(x)) - (x^0 - x)^T(z^0 - z) - (1-\theta)((x^0)^T r_D^0 - (y^0)^T r_P^0).$$

Thus, for $\theta \le 1/2$, there exists M^1 independent of θ such that

$$-M^1 \le (-x^T r_D^0 + y^T r_P^0) \le M^1,$$

which implies $y^T r_P^0$ is bounded, since x is bounded. Moreover, from (27) we have

$$x\nabla f(x) - b^T y = \theta(x^T r_D^0 - y^T r_P^0 + x^T z).$$

Therefore, we must have

$$x\nabla f(x) - b^T y \to 0 \quad \text{as} \quad \theta \to 0.$$

Q.E.D.

In fact, from the proof we see that we need only $(\hat{y}^k)^T r_p^0$ being bounded rather than $\hat{y}^k$ itself being bounded. Thus, the proposition can be further strengthened to

Corollary 2 *If there is a sequence* $(\hat{x}^k > 0, \hat{y}^k, \hat{z}^k \geq 0)$ *such that*

$$r_P(\hat{x}^k) \to 0, \quad r_D(\hat{x}^k, \hat{y}^k, \hat{z}^k) \to 0, \quad \mu(\hat{x}^k, \hat{z}^k) \to 0,$$

and $(\hat{x}^k, \hat{z}^k)$ *and* $(\hat{y}^k)^T r_P^0$ *are bounded, then the central path points* $(x(\theta), z(\theta))$ *and* $y(\theta)^T r_P^0$ *are bounded for all* $0 < \theta \leq 1$. *Moreover, the subfeasible duality gap (28) converges to* 0 *as* $\theta \to 0$.

4 Recovering an approximate solution for the primal GP

The definition of the central path, (27), is a statement about the existence of special subfeasible solutions to the dual pair of convex programs (P) and (D), where we shall now let $s(\theta) = z(\theta) + \theta\, r_D^0$, noting that

$$\liminf_{\theta \downarrow 0} s_i(\theta) \geq 0 \tag{37}$$

Let us specialize (27) to geometric programming, where now we return to F in place of the general f. For notational simplicity we shall delete θ as the argument in path points and s above.

Because of the special structure of the constraint matrix it now follows that

$$\sum_{[0]} x_i = 1 - \theta\ (r_P^0)_1. \tag{38}$$

Likewise the gradient conditions specialize to

$$\ln x_i - \ln c_i + 1 \quad = \quad y_1 + u_i + s_i, \quad i \in [0] \tag{39}$$

$$\ln x_i - \ln c_i - ln\lambda_k \quad = \quad u_i + s_i, \quad i \in [k], \quad k = 1, ..., p \tag{40}$$

where

$$\lambda_k = \sum_{[0]} x_i \quad \text{and} \quad u_i = \sum_{j=2}^{m+1} a_{ij-1}\ y_j \quad \text{for } k = 1, ..., p \text{ and } i = 1, ..., n_p.$$

While ours is not solely a *dual–based* method, we address the issue of recovering a primal GP solution from the Lagrangian dual solution at termination, a characteristically challenging task for purely dual–based methods, see [10]. This is our last result.

Proposition 4 *If* (P) *and* (D) *are boundedly subconsistent, then the central path*

$$(x(\theta), y(\theta), z(\theta))$$

yields optimal subfeasible solutions for (GP) *and* (GD), *namely*

$$\{x(\theta)\}_\theta \; for \; (GD)$$

and

$$\{t(\theta)\}_\theta \; for \; (GP)$$

under the transformation

$$t_j = e^{y_{j+1}}, j = 1 \ldots m.$$

In particular, V_{GD} *is attained.*

Proof Suppressing θ in x, y, z, s, we begin with (39) by making a substitution for 1 via (11) and (38), but simultaneously using (28) for a substitution for $x^T\nabla F(x)$. We also use $b^T y = y_1$. This changes (39) for each i to

$$\ln x_i \quad = \quad -\theta\ (\eta(x,y,z) + (r_P^0)_1) + F(x) + \ln c_i + u_i + s_i$$

We next obtain the following upon exponentiating and summing over [0].

$$\sum_{[0]} x_i = e^{F(x)}\ e^{-\theta\ (\eta(x,y,z)+(r_P^0)_1)} \sum_{[0]} c_i e^{u_i+s_i},$$

and using (38) again, we further obtain

$$e^{-F(x)}[1-\theta\ (r_P^0)_1] = e^{-\theta\ (\eta(x,y,z)+(r_P^0)_1)} \sum_{[0]} c_i e^{u_i+s_i}. \tag{41}$$

Transforming (40) immediately and summing over $[k]$, $k = 1, ..., p$ yields

$$\sum_{[k]} x_i = \lambda_k \sum_{[k]} c_i e^{u_i+s_i}, \quad k = 1, ..., p. \tag{42}$$

But from the definition of λ_k, necessarily positive, we have

$$\sum_{[k]} c_i e^{u_i+s_i} = 1, \quad k = 1, ..., p. \tag{43}$$

Referring back to (2) and (3), it follows that

$$\sum_{[k]} c_i e^{u_i} = g_k(t), \quad k = 0, ..., p \tag{44}$$

where

$$t_j = e^{y_{j+1}}, \quad j = 1, ..., m.$$

Therefore, it follows from (37),(43), and (44) that $\{t(\theta)\}_\theta$ is (GP) subfeasible, [11, page 166].
Making a substitution based on for u_i into (41), it follows that

$$e^{-F(x)}[1-\theta\ (r_P^0)_1] = e^{-\theta\ (\eta(x,y,z)+(r_P^0)_1)} \sum_{[0]} \{c_i \prod_{j=1}^{m} t_j^{a_{ij}}\} e^{s_i}. \tag{45}$$

Now $e^{-F(x)}$ is the (GD) objective function value corresponding to $x \equiv x(\theta)$. By Proposition 2 $\{x(\theta)\}_\theta$ has a limit point $\hat{x}^*$ as $\theta \to 0$ for which $F(\hat{x}^*) = V_P$. Hence $e^{-F(\hat{x}^*)} = V_{GD}$, i.e., V_{GD} *is attained.* Therefore taking the lim inf of both sides of (45) yields

$$V_{GD} = e^{-F(\hat{x}^*)} = \liminf_{\theta} \{e^{-\theta\ (\eta(x,y,z)+(r_P^0)_1)} \sum_{[0]} \{c_i \prod_{j=1}^{m} t_j^{a_{ij}}\} e^{s_i}\} \geq \bar{V}_{GP} \tag{46}$$

because $\{t(\theta)\}_\theta$ is subfeasible, $\theta\eta(x,y,z) \to 0$ *with* θ, and (37) holds. Combining (46) with (19) yields the desired conclusion,

$$V_{GD} = e^{-F(\hat{x}^*)} = \lim_{\theta} \sum_{[0]} c_i \prod_{j=1}^{m} t_j(\theta)^{a_{ij}} = \bar{V}_{GP}$$

Q.E.D.

5 A practical algorithm

Our algorithm is motivated by the idea of following the infeasible central path described in the previous section. All iterates satisfy

$$\begin{aligned} Xz &\geq \sigma\mu & (47) \\ \|Xz - \mu e\| &\leq \beta\mu, & (48) \end{aligned}$$

where $\sigma \in (0, 1]$ and $\beta \in (0, +\infty)$, and

$$\theta_P \ x^T z \geq \|r_P\| \quad \text{and} \quad \theta_D \ x^T z \geq \|r_D\|, \tag{49}$$

where θ_P and θ_D are positive constants and from Definition 1,

$$r_P := r_P(x), \quad r_D := r_D(x, y, z) \quad \text{and} \quad \mu := \mu(x, z).$$

In section 6.2 we will discuss how to choose these constants. Generally speaking, conditions (47) and (48) define a neighborhood of the central path, and condition (49) balances the convergence such that the reduction in complementarity is not faster than the reduction in feasibility. The latter was proposed in [25] and [40].

We now formally state the algorithm.

ALG 1 Primal-dual path-following algorithm
Given $f(x), A, b$ and initial point $(y^0, x^0 > 0, z^0 > 0)$, set $(y, x, z) \leftarrow (y^0, x^0, z^0)$.
While *(stopping criterion not satisfied)* **do**

1. *Set the Hessian matrix of f: $H \leftarrow \nabla^2 f(x)$.*
2. *Call* **SUB 1** *to choose parameters $\gamma \in [0, 1]$ and $\eta \in [0, 1]$.*
3. *Solve for direction from*

$$\begin{pmatrix} 0 & A & 0 \\ A^T & -H & I \\ 0 & Z & X \end{pmatrix} \begin{pmatrix} \delta_y \\ \delta_x \\ \delta_z \end{pmatrix} = \begin{pmatrix} \eta r_P \\ \eta r_D \\ \gamma\mu e - Xz \end{pmatrix}. \tag{50}$$

4. *Call* **SUB 2** *to choose a step length α.*
5. *Update solution*

$$\begin{aligned} y &\leftarrow y(\alpha) := y + \alpha\delta_y \\ x &\leftarrow x(\alpha) := x + \alpha\delta_x \\ z &\leftarrow z(\alpha) := z + \alpha\delta_z. \end{aligned}$$

6. *Call* **SUB 3** *to reset z.*

End do

The stopping criterion in ALG 1 is specified as follows

$$\frac{\|r_P\|_1}{1 + \|x\|_1} < \epsilon_P, \quad \frac{\|r_D\|_1}{1 + \|z\|_1} < \epsilon_D, \quad \frac{x^T z}{1 + \|x\|_1 + \|z\|_1} < \epsilon_C \tag{51}$$

where ϵ_P, ϵ_D, and ϵ_C are relative primal feasibility, dual feasibility, and complementarity tolerances. These criteria are used by almost all interior-point implementation codes for solving linear programs.

We now describe the subroutines called by the main algorithm.

5.1 Choosing parameters γ and η

An important practical issue in interior-point algorithms for linear programming is to perform an affine step to predict reduction in the residuals. Then, a reasonable centering factor is used in computing the final direction. This technique seems always to result in a large step in practice for linear programming (see [41]). Similar to solving LP, we first calculate the affine direction by solving (50) with $\eta = 1$ and $\gamma = 0$, and then choose a proper parameter pair (η, γ) according to the improvement made along the affine direction. What we mainly consider in choosing this parameter pair is the improvement in the complementarity residual.

SUB 1 Choosing γ and η via a prediction step

- Let $\eta = 1$ and $\gamma = 0$, solve (50) for $(\delta_y^a, \delta_x^a, \delta_z^a)$.
- Perform non-negative ratio test to get the largest step length $\alpha^a = \min(\alpha_x^a, \alpha_z^a)$ where

$$\begin{aligned}\alpha_x^a &= \max\{-x_j/(\delta_x^a)_j : (\delta_x^a)_j < 0\} \\ \alpha_z^a &= \max\{-z_j/(\delta_z^a)_j : (\delta_z^a)_j < 0\}.\end{aligned}$$

- Calculate

$$\mu^a = (x + \alpha^a \delta_x^a)^T (z + \alpha^a \delta_z^a)/n$$

  and set

$$\gamma = 0.5 \times (\mu^a/\mu), \quad \eta = 1 - \gamma.$$

5.2 Choosing step length α

Since the primal residual is a linear function, its reduction rate simply depends on a step length α. However, the dual and complementarity residuals involve nonlinear terms. Moving along the direction $(\delta_x, \delta_y, \delta_z)$ with a step length α, the changes in r_D and complementarity are

$$\begin{aligned}& r_D((x,y,z) + \alpha(\delta_x, \delta_y, \delta_z)) \\ = \;& \nabla f(x + \alpha\delta_x) - A^T(y + \alpha\delta_y) - (z + \alpha\delta_z) \\ = \;& \nabla f(x + \alpha\delta_x) - \nabla f(x) + r_D - \alpha(A^T\delta_y + \delta_z) \\ = \;& \nabla f(x + \alpha\delta_x) - \nabla f(x) - \alpha H\delta_x + r_D - \alpha(A^T\delta_y - H\delta_x + \delta_z) \\ = \;& \nabla f(x + \alpha\delta_x) - \nabla f(x) - \alpha H\delta_x + (1 - \alpha\eta) r_D \end{aligned} \tag{52}$$

and

$$\begin{aligned}(X + \alpha\Delta_x)(z + \alpha\delta_z) &= Xz + \alpha(X\delta_z + Z\delta_x) + \alpha^2\Delta_x\delta_z \\ &= Xz + \alpha(\gamma\mu e - Xz) + \alpha^2\Delta_x\delta_z \\ &= (1 - \alpha)Xz + \alpha\gamma\mu e + \alpha^2\Delta_x\delta_z \end{aligned} \tag{53}$$

respectively. Unlike r_P, r_D here can be much harder to control. In choosing step length α, our main attention is paid to reducing complementarity while r_D might be allowed to increase. Since the algorithm ensures that r_P and r_D satisfying (49) in each iteration, their convergence to zero directly follows from $x^T z \to 0$. We usually specify large θ_D so that (49) is easily satisfied.

Our step length α is chosen by the following procedure.

SUB 2 Choosing step length

- Ratio test: choose $\alpha_R = \min(\alpha_x, \alpha_z)$ where

$$\begin{aligned} \alpha_x &= \max\{-x_j/(\delta_x)_j : (\delta_x)_j < 0\} \\ \alpha_z &= \max\{-z_j/(\delta_z)_j : (\delta_z)_j < 0\} \end{aligned}$$

- Neighborhood test: choose $\alpha_N > 0$ such that $(x(\alpha_N), y(\alpha_N), z(\alpha_N))$ satisfy conditions (47) and (48).

- Balance test: choose α_C such that $(x(\alpha_C), y(\alpha_C), z(\alpha_C))$ satisfy condition (49) and

$$x(\alpha_C)^T z(\alpha_C) \leq \theta_C \; x^T z$$

  for some constant θ_C.

- Choose the final step length

$$\alpha = \min\{factor \times \alpha_R, \ \alpha_N, \ \alpha_C\}$$

  where $factor \in (0,1)$ is a safety factor.

In both Neighborhood and Balance steps we use a simple backtracking line search technique. That is, we start $\alpha = factor \times \alpha_R$. If the condition is satisfied, then we stop; otherwise, we reduce α by a fraction and test it again. The selection of large θ_P and θ_D has made the backtracking search hardly ever necessary. We have also chosen $\theta_C = 2 > 1$ to make the step length α long. Thus, the complementarity gap may not monotonically decrease at all iterations.

5.3 Resetting dual slacks

The slack vector z in dual constraint may cause a convergence problem. A typical phenomenon is a sequence of iterates which possesses $r_D(x^k, y^k, z^k) > 0$. We have observed this fact in some problems. The reason probably is that the default value of θ_D in (49) has been set too large. And once μ is small, then it is very difficult to reduce r_D. Note that in this case, if we reset the slack variable by $z \leftarrow z + r_D$, then the dual residual vanishes in the current iteration. Thus, in each iteration we reset slack variables z to reduce r_D.

SUB 3 Resetting slacks

let $\sigma = \nabla f(x) - A^T y$

for $i = 1, n$ **do**
  if $\sigma_i \geq 0$ **then**

$$z_i = \begin{cases} \sigma_i & if \quad \sigma_i \in (z_i/fac, \ z_i \times fac) \\ z_i/fac & if \quad \sigma_i \leq z_i/fac \\ z_i \times fac & if \quad \sigma_i \geq z_i \times fac \end{cases}$$

  end if

end do

Each z_i is reset according to its own value and σ_i. We set parameter $fac = 10^2$ to maintain that the point z after resetting is not far from the central path. This heuristic strategy seems working well in practice.

6 An Implementation : XGP

The efficiency of the algorithm in practice depends on the linear algebra employed for solving (50) and the selection of various algorithm parameters.

6.1 Solving for a direction

By eliminating δ_z and interchanging the order of δ_y and δ_x in (50), we have the following reduced system

$$\begin{pmatrix} -X^{-1}Z - H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} \eta r_D - X^{-1}(\gamma \mu e - Xz) \\ \eta \ r_P \end{pmatrix} \tag{54}$$

Let

$$M = \begin{pmatrix} -X^{-1}Z - H & A^T \\ A & 0 \end{pmatrix}.$$

M is not quasi-definite (see Vanderbei [39]) since the lower-right block is zero. However, $X^{-1}Z + H$ is symmetric positive definite since H is semi-definite and $X^{-1}Z$ are positive diagonal matrix. If we choose diagonal pivots from the $X^{-1}Z + H$ block in order to eliminate δ_x, the remaining matrix is $A(X^{-1}Z + H)^{-1}A^T$. The same pivoting strategy is applied to this remaining matrix.

We can choose a pivoting order to exploit sparsity by perform single symbolic factorization without considering numerical values, and then use the static data structure associated with this pivoting order in subsequent iterations. The heuristic used to choose the pivoting order is *minimum degree*.

One may argue that, even H is a three-diagonal matrix, the matrix $A(X^{-1}Z + H)^{-1}A^T$ will be totally dense. However, (12),(13) and (14) show that the H arising from a dual GP is diagonal block structured, and each block is sparse. Since $(X^{-1}Z + H)^{-1}$ has the same diagonal block structure of H, the matrix $A(X^{-1}Z + H)^{-1}A^T$ can be written in the following form

$$\sum_{k=0}^{p} A_k \bar{H}_k A_k^T$$

where $\bar{H}_k$ is denoted as the k-th block of $(X^{-1}Z + H)^{-1}$ and A_k is denoted as the sub-matrix of A consisting of columns corresponding to the k-th block of H. The sparsity of $A(X^{-1}Z + H)^{-1}A^T$ is determined by each A_k, much like that of the normal matrix arising from LP where the sparsity is determined by each column of A.

Moreover, when the GP problem has a high degree of difficulty (A has many columns), the efficiency of factorizing M lies in reducing the fill-in in the off-diagonal block of M. In this case, the lower-right block $A(X^{-1}Z + H)^{-1}A^T$ becomes relatively a small issue. Therefore, the algorithm we present here will be effective even a dense $A(X^{-1}Z + H)^{-1}A^T$ may be generated.

Also note that we do not require that A has a full rank in order to solve (50).

Our detailed factorization techniques are described below.

- Determine the pivoting order of M:
  1. Limited to the first n diagonals, determine a pivoting order $\mathcal{P}$ by minimum degree or local minimum fill-in heuristic.
  2. Perform a symbolic factorization to create $A(X^{-1}Z + H)^{-1}A^T$, then determine a pivoting order $\mathcal{Q}$ for this remaining part by minimum degree or local minimum fill-in heuristic.

3. The total pivoting order is $\mathcal{P}\&\mathcal{Q}$.

- Symbolic factorization:
  Perform a symbolic factorization of M according to the pivoting order $\mathcal{P}\&\mathcal{Q}$.
- Numerical factorization:
  Perform numerical factorization according to the data structure allocated by symbolic factorization.

Note that in the subsequent iterations we only need to set M and perform numerical factorization.

6.2 Setting an initial point and relevant parameters

Initial point is simply chosen as

$$x^0 = e, \quad y^0 = 0, \quad \text{and} \quad z^0 = e. \tag{55}$$

In **SUB 2**, the neighborhood size is chosen as

$$\sigma = 1. \times 10^{-8} \quad \text{and} \quad \beta = 1. \times 10^3, \tag{56}$$

the safety fraction of step length is

$$factor = 0.9995, \tag{57}$$

and the balance parameters are specified as

$$\theta_C = 2, \quad \theta_P = 10^2 \times \frac{\|r_P^0\|}{(x^0)^T z^0}, \quad \theta_D = 10^4 \times \frac{\|r_D^0\|}{(x^0)^T z^0}. \tag{58}$$

We set parameter $fac = 10^2$ in **SUB 3**.

Tolerances for primal and dual feasibilities and complementarity are specified as

$$\epsilon_P = 10^{-8}, \quad \epsilon_D = 10^{-8}, \quad \text{and } \epsilon_C = 10^{-12}. \tag{59}$$

Extensive computational tests indicate that the requirement of $\epsilon_C = 10^{-12}$ is very strong. The solution stopping by this specification usually is very accurate. The same step length is taken for all the variables. In all the tests the total iteration number is limited by 40.

7 Computational results

The present collection includes 19 geometric programming problems whose summary is given in Table 1, including number of variables, constraints, posynomial terms and degree of difficulty. The degree of difficulty defined by (6) was introduced in GP to measure the degrees of a freedom in the dual GP problem, see [11, Chap.1]. For reader's convenience, a problem from a particular paper will be identified by author name and year of publication and the reported objective function value. Problems and their corresponding references are Beck75x : [4], Demb76x : [9], Demb78x : [10], Fang88 : [15], Fiac81a : [17], Fiac81b : [18], Jha88 : [24], Rijc78x : [35],

The tests were conducted on an HP 9000/720 (50Mhz) work-station. The FORTRAN codes are compiled with optimization option +O3. The reported time (seconds) includes total computational effort except for the data input and solution output. This includes the following main operations are: function, gradient, and Hessian evaluation; forming the reduced augmented system and solving for the directions.

Table 1 contains the computational results for solving 19 GP problems, including number of iterations (iter), time, and objective values of primal GP and dual GP. The solution for primal GP is recovered by Proposition 4. Table 2 contains an accuracy measure of the recovered solution with respect to the original GP problem: relative duality gap

$$relgap = \frac{|obj\_primal\_GP - obj\_dual\_GP|}{1 + |obj\_dual\_GP|}$$

primal and dual infeasibilities

$$inf_D = \frac{\|Ax - b\|_1}{1 + \|x\|_1}, \quad inf_P = \sum_{k=1}^{p} \max\{g_k(t) - 1, 0\}.$$

In Table 3 we have reported the following comparative computational results for 3 problems which are becoming generally regarded as among the most challenging GP problems in the literature. Under **Efficiency**, in the left– most column we report the number of iterations of our algorithm and the number of major or outer iterations for the two cutting plane methods, [2] and [22], followed by the total number of iterations for Vial's algorithm, [40]. In the remaining columns under Efficiency we report the number of inner iterations for both cutting plane algorithms in parenthesis and the CPU time in seconds when reported. Under **Accuracy**, we report either a relative duality gap measure (2) or the duality gap itself (2) depending on which was reported or published. The results of [22] were also carried out on an HP 9000/720 workstation whose model is identical to the HP workstation we used.

8 Concluding remarks

In this paper we have obtained theoretical results for a class of linearly constrained, interior–differentiable convex programming problems based on an asymptotic analysis of a perturbed KKT system, which itself is derived from a perturbed logarithmic barrier subproblem of the original problem. The proposed algorithm delivers solutions to a perturbed problem which are arbitrary close to an optimal solution of the original linearly constrained convex problem under the fairly general assumption of bounded subconsistency. Both primal and Lagrangian dual solutions are solved simultaneously in this sense.

Our algorithmic implementation is for the special case of Geometric Programming. Its particular sparse Hessian matrix structure was exploited in solving the reduced KKT matrix arising from the dual GP problem. Using predicting and correcting techniques, we have demonstrated that our implementation is stable on a representative collection of GP problems found in the literature. Moreover, its efficiency seems to compare favorably with other geometric programming numerical results reported in the literature.

No effort is made here to present a complexity or convergence result for our practical algorithm and implemented computer code. (Some results can be found in, e.g., [1][25][42].) This is true for almost all interior-point computer implementations. In general, theoretical results provide general framework and structure for implementation. However, in practice, we always tune some parameters, step-sizes, tolerances in order to improve efficiency and accuracy. Our main goal has been the development of an efficient and accurate method, which can be readily implemented with basically a general setting of parameters across a fairly wide class of problems. There is always a gap between theoretical and practical algorithms, but this gap is common.

Some of the GP problems tested for this paper have been recently solved using a PC 486 DX2/50, see [29]. In addition, XGP has been also applied on a class of large scale multi–resource project management problems, see [38]. Kortanek and Xu constructed an example, based upon a communication from Carlton Scott at UC–Irvine, having 25 nodes, 5 resources, and 240 paths, which generated a GP having (a) 126 primal vars, (b) 245 dual constraints, (c) 3621 primal terms, and so an unprecedented degree of difficulty of 3494. In particular, the constraint matrix A is 127 by 3621 with 21097 non–zero entries. As reported in [29] this problem was solved in 115.53 seconds on a 24MB 90MHz Pentium system running ATT's f2cf Fortran to C translator, the GNU Foundation's GCC C compiler, and the Linux operating system (which is slower than the HP machine used on the main problems.) Essentially the same primal and dual accuracy was achieved as reported for the other problems.

Our principle conclusions about our numerical implementation for geometric programming are the following ones.

- *High accuracy* can be achieved in a fewer number of iterations.
- *On problems having an interior feasible point*, particularly on relative large problems, the performance of our algorithm seems significant. A word of caution should be made about making comparisons based upon Table 3. We recognize that CPU time is machine dependent and is influenced by possible sparse matrix algebra, coding, and languages used, see [40, Section 5]. In addition, codes based on [40, 2, 22] are general purpose codes for solving convex programs while ours is a special purpose code specifically designed for solving geometric programming problems. From the viewpoint of comparing the number of iterations, we have reported the number of *inner iterations* for cutting plane methods because one such iteration is approximately equivalent to one iteration of a primal–dual method for solving these GP problems. Also, the codes were run on different machines. Thus, more experiments are needed to make conclusive comparisons.
- *Problems not having an interior feasible point* are also solved successfully.
- The algorithm appears to be a *first step towards determining* which duality state prevails for a given primal geometric program and its geometric dual.

Acknowledgment We would like to thank the referees for their careful reading and constructive comments of the paper.

Table 1: Solving GP by A Primal-Dual Infeasible Path-Following Algorithm

Problem Summary						Efficiency		Accuracy	
Problem		Number of			Degree of			Objective value of	
name		vars	cons	terms	difficulty	iter	time	dual GP	primal GP
1	Beck751	7	4	18	10	13	0.04	1.80976476557e+03	1.80976476557e+03
2	Beck752	7	4	18	10	13	0.03	9.11880571430e+02	9.11880571431e+02
3	Beck753	7	4	18	10	14	0.04	5.43667958466e+02	5.43667958466e+02
4	Demb761	11	3	31	19	20	0.09	4.89197944642e+09	4.89197944494e+09
5	Demb762	11	3	31	19	16	0.07	3.17245814116e+00	3.17245813419e+00
6	Demb763	11	3	31	19	14	0.06	3.18325098244e+00	3.18325098251e+00
7	Demb781	2	1	4	1	7	0.00	2.00000000000e+00	2.00000000000e+00
8	Demb782	2	1	3	0	5	0.02	2.00000000000e+00	2.00000000000e+00
9	Fang88	11	3	28	16	16	0.05	3.10459947218e+00	3.10459947221e+00
10	Fiac81a	22	36	73	50	14	0.09	1.83180661058e+03	1.83180661058e+03
11	Fiac81b	10	7	20	9	14	0.04	3.23732738533e+07	3.23732738533e+07
12	Jha88	30	40	305	274	17	0.27	3.25140617241e+04	3.25140617240e+04
13	Rijc781	4	2	6	1	8	0.02	1.21031862246e−02	1.21031862246e−02
14	Rijc782	3	1	9	5	11	0.01	6.29984242792e+03	6.29984242792e+03
15	Rijc783	4	1	12	7	9	0.01	1.26303177993e+05	1.26303177993e+05
16	Rijc784	4	3	8	3	12	0.02	6.23249876121e+05	6.23249876107e+05
17	Rijc785	8	7	12	3	10	0.03	2.92294839249e+01	2.92294839249e+01
18	Rijc786	8	7	12	3	7	0.02	2.92264512244e+01	2.92264512244e+01
19	Rijc787	7	7	48	40	12	0.04	1.78477919998e+02	1.78477919998e+02

References

[1] K. Anstreicher and J.-Ph. Vial, "On the convergence of an infeasible primal-dual interior-point method for convex programming", *Optimization Methods and Software* 4 (1994) 273-283.

[2] O. Bahn, J.L. Goffin, J.-Ph. Vial and O. Du Merle, "Experimental behavior of an interior point cutting plane algorithm for convex programming: an application to geometric programming," *Discrete Applied Mathematics* 49 (1994) 2-23.

[3] H. El Barmi, and R.L. Dykstra, "Restricted multinomial maximum likelihood estimation based upon Fenchel duality," *Statistics and Probability Letters* 21 (1994) 121-130.

[4] P.A. Beck and J.G. Ecker, "A modified concave simplex algorithm for geometric programming", *Journal of Optimization Theory and Applications* 15 (1975) 189-202

[5] C.S. Beightler and D.T. Phillips, *Applied Geometric Programming* (John Wiley & Sons, NY, 1976).

[6] D.L. Bricker, K.O. Kortanek and L. Xu, "Maximum likelihood estimates with order restrictions on probabilities and odds ratios: a geometric programming approach", Applied Mathematical and Computational Sciences, The University of IA, Iowa City, Iowa 52242, September, 1995

[7] A. Charnes, W.W. Cooper, B. Golany and J. Masters, "Optimal design modification by geometric programming and constrained stochastic network models," *International Journal on Systems Science* 19 (1988) 825-844.

[8] A. Charnes, W.W. Cooper and K.O. Kortanek, "Semi-infinite programming, differentiability and geometric Programming," *Journal of Mathematical Sciences* 6 (1971) 19-40.

Table 2: Accuracy statistics

Problem Summary						Efficiency		Accuracy		
Problem name		Number of vars	cons	terms	Degree of difficulty	iter	time	$relgap$	inf_D	inf_P
1	Beck751	7	4	18	10	13	0.04	5.5e-15	6.6e-15	0.0e+00
2	Beck752	7	4	18	10	13	0.03	4.1e-13	1.1e-13	1.2e-13
3	Beck753	7	4	18	10	14	0.04	0.0e+00	1.7e-15	0.0e+00
4	Demb761	11	3	31	19	20	0.09	3.0e-10	2.4e-12	4.0e-15
5	Demb762	11	3	31	19	16	0.07	1.7e-09	3.0e-10	0.0e+00
6	Demb763	11	3	31	19	14	0.06	1.6e-11	3.1e-12	0.0e+00
7	Demb781	2	1	4	1	7	0.00	3.3e-15	5.1e-17	0.0e+00
8	Demb782	2	1	3	0	5	0.02	0.0e+00	3.1e-17	0.0e+00
9	Fang88	11	3	28	16	16	0.05	7.3e-12	1.9e-13	0.0e+00
10	Fiac81a	22	36	73	50	14	0.09	1.6e-14	1.5e-15	7.0e-14
11	Fiac81b	10	7	20	9	14	0.04	1.3e-12	3.4e-14	1.5e-10
12	Jha88	30	40	305	274	17	0.27	2.7e-12	2.5e-13	2.8e-13
13	Rijc781	4	2	6	1	8	0.02	8.9e-16	2.0e-16	0.0e+00
14	Rijc782	3	1	9	5	11	0.01	4.6e-13	3.5e-14	0.0e+00
15	Rijc783	4	1	12	7	9	0.01	0.0e+00	3.5e-17	0.0e+00
16	Rijc784	4	3	8	3	12	0.02	2.2e-11	3.1e-13	1.0e-11
17	Rijc785	8	7	12	3	10	0.03	9.8e-13	1.7e-14	5.1e-13
18	Rijc786	8	7	12	3	7	0.02	1.5e-12	5.3e-16	0.0e+00
19	Rijc787	7	7	48	40	12	0.04	6.1e-14	3.0e-16	1.0e-14

Table 3: Result Comparisons

Problem Summary						Efficiency			Accuracy
Problem name		Number of vars	cons	terms	Degree of difficulty	iter	inner	time	(Relative) duality gap
11	Fiac81b	10	7	20	9	14		0.04	1.3e-12
	*					20	(207)	6.54	8.1e-07
	**					89	(207)		9.0e-07
	***					27		4.63	1.3e-10
10	Fiac81a	22	36	73	50	14		0.09	1.6e-14
	*					17	(327)	39.60	2.5e-11
	**					146	(203)		9.0e-07
	***					20		16.00	3.8e-13
12	Jha88	30	40	305	274	17		0.27	2.7e-12
	*					21	(557)	162.88	1.4e-09
	**								
	***					20		30.00	1.9e-14

* Logarithmic barrier cutting plane [22]. The code is written in FORTRAN 77. The tests are carried out on an HP 9000/720 workstation, with Unix version 8.05 Operating System. Major (outer) iterations are reported with Linesearch, Table 2a, with the number of inner iterations given in parenthesis. The duality gap is reported.

** Analytical center cutting plane [2]. The code is written in PRO-MATLAB, and the tests are carried out on a SPARCstation 1. Outer iterations are first reported for Version C followed by the number of inner iterations in parenthesis, see Table 2 there, where one factorization/inner iteration is reported, as well as the duality gap.

*** Primal-dual algorithm based on primal GP [40]. The code is written in MATLAB, and the tests are conducted on a SUN SPARCworkstation IPC with 36 Mb of random access memory. Citing [40, Section 5], "the total number of iterations is a valid indicator of total computational effort", see Table 2. The reported gap values are not relative gaps.

[9] R.S. Dembo, "A set of geometric programming test problems and their solutions," *Mathematical Programming* 10 (1976) 192-213.

[10] R.S. Dembo, "Dual to primal conversion in geometric programming," *Journal of Optimization Theory and Applications* 26 (1978) 243-252.

[11] R.J. Duffin, E.L. Peterson and C. Zener, *Geometric Programming–Theory and Application* (John Wiley & Sons, NY, 1967).

[12] R.J. Duffin, "Infinite Programs", in: H. W. Kuhn and A. W. Tucker, eds., *Linear Inequalities and Related Systems*, (Princeton University Press, Princeton, NJ, 1956) pp.157-170

[13] J. Dupacŏvá, P. Charamza and J. Mádl "On stochastic aspects of a metal cutting problem," Department of Statistics, Charles University, Sokolovská 83, 18600 Prague, Czech Republic, June, 1993.

[14] A.S. El-Bakry, R.A. Tapia, T. Tsuchiya and Y. Zhang, "On the formulation and theory of the primal-dual Newton interior-point method for nonlinear programming," Department of Computational & Applied Mathematics, Rice University, Houston, Texas, October 1993

[15] S.C. Fang, E.L. Peterson and J.R. Rajasekera, "Controlled dual perturbations for posynomial programs," *European Journal of Operational Research* 35 (1988) 111-117.

[16] A.V. Fiacco, "Objective function and logarithmic barrier function properties in convex programming: level sets, solution attainment and strict convexity," *Optimization* 34 (1995) 213-222.

[17] A.V. Fiacco and A. Ghaemi, "Sensitivity analysis of a nonlinear water pollution control model using an upper hudson river data base," The George Washington University, Washington, D.C., August, 1981.

[18] A.V. Fiacco and A. Ghaemi, "Sensitivity and parametric bound analysis of an electric power generator GP model: optimal steam turbine-exhaust annulus and condensor sizes, serial T-437," The George Washington University, Washington, D.C., August, 1981

[19] W. Gochet, K.O.Kortanek and Y. Smeers, "Using semi-infinite programming in geometric programming," in : E. Shlifer, ed., *Proceedings of the 20th International TIMS: On Management Sciences, Developing Countries and National Priorities 2*, (Jerusalem Academic Press, Jerusalem, and North-Holland, Amsterdam, 1973) pp. 430-438

[20] W. Gochet, K.O. Kortanek, and Y. Smeers, "On a classification scheme for geometric programming and complementarity theorems," *Applicable Analysis* 6 (1976) 47-59.

[21] E.G. Goldstein, *Theory of Convex Programming*, Translations of Mathematical Monographs, (American Mathematical Society, Providence, RI, 1972).

[22] D. den Hertog, J. Kaliski, C. Roos and T. Terlaky, "A logarithmic barrier cutting plane method for convex programming," *Annals of Operations Research* 58 (1995) 69-90.

[23] R. Jagannathan, "A stochastic geometric programming problem with multiplicative recourse," *Operations Research Letters* 9 (1990) 99-104.

[24] S. Jha, K.O. Kortanek and H. No, "Lotsizing and setup time reduction under stochastic demand: a geometric programming approach," Working paper series No. 88-12, College of Business Administration, The University of Iowa, Iowa City, June, 1988

[25] M. Kojima, T. Noma and A. Yoshise, "Global convergence in infeasible-interior-point algorithms," *Mathematical Programming* 65 (1994) 43-72.

[26] K.O. Kortanek, "Classifying convex extremum problems over linear topologies having separation properties," *Journal of Mathematical Analysis and Applications* 46 (1974) 725-755.

[27] K.O. Kortanek and H. No, "A second order affine scaling algorithm for the geometric programming dual with logarithmic barrier," *Optimization* 23 (1992) 303-322.

[28] K.O. Kortanek, F. Potra and Y. Ye, "On some efficient interior point methods for nonlinear convex programming," *Linear Algebra and its Applications* 152 (1991) 169-189.

[29] K.O. Kortanek and X. Xu, "Numerical Experiments with XGP–An Optimizer for Geometric Programming and Some Comparisons to Other Solvers' College of Business Administration, The University of Iowa, Iowa City, IA, 52242, October, 1995.

[30] K.O. Kortanek and J. Zhu, "On controlling the parameter in the logarithmic barrier term for convex programming problems," *Journal of Optimization Theory and Applications* 84 (1995) 117-143.

[31] S. Mizuno, M.J. Todd and Y. Ye "A surface of analytic centers and infeasible-interior-point algorithms for linear programming," *Mathematics of Operations Research* 20 (1995) 135-162.

[32] J. L. Nazareth, "A framework for interior methods of linear programming," Department of Pure and Applied Mathematics, Washington State University, 1994.

[33] Yu. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming: Theory and Application*, (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1993).

[34] M. J. Rijckaert and X. M. Martens, "Bibliographical note on geometric programming," *Journal of Optimization Theory and Applications* 26 (1978) 325-337

[35] M. J. Rijckaert and X. M. Martens, "Comparison of generalized geometric programming algorithm," *Journal of Optimization Theory and Applications* 26 (1978) 205-242.

[36] T. Robertson, F.T. Wright and R.L. Dykstra, *Order Restricted Statistical Inference* (John Wiley and Sons, NY, 1988).

[37] R.T. Rockafellar, *Convex Analysis*, (Princeton University Press, Princeton, NJ, 1970).

[38] C.H. Scott and T.R. Jefferson, "Allocation of resources in project management," *International Journal on Systems Science* 26 (1995) 413-420.

[39] R.J. Vanderbei, "Symmetric quasi-definite matrices," *SIAM J. Optimization* 5 (1995) 100-113.

[40] J.-P. Vial, "Computational experience with a primal-dual interior-point method for smooth convex programming," *Optimization Methods and Software* 3 (1994) 285-316.

[41] X. Xu, "On the implementation of a homogeneous and self-dual linear programming algorithm," College of Business Administration, The University of Iowa, Iowa City, IA 52242, June,1994.

[42] X. Xu, "An $\mathcal{O}(\sqrt{n}L)$-iteration large-step infeasible path-following algorithm for linear programming," College of Business Administration, The University of Iowa, Iowa City, IA 52242 (Aug. 1994).

[43] J. Zhu, K.O. Kortanek and S. Huang, "Controlled dual perturbations for central path trajectories in geometric programming," *European Journal of Operational Research* 73 (1992) 524-531.

$$Geometric\ Programming\ Dualities$$

$$\begin{array}{ccc} Primal(GP) & \xleftrightarrow[GP\ Duality]{\qquad\qquad} & Dual(GP) \end{array}$$

$$-\log(GPdualfunction)$$

$$min\, F(x)$$
$$Ax = b, \quad x \geq 0$$

$$t = \exp(y)$$

$$Lagrangian\ Duality$$

$$LagrangianDual$$
$$sup_y \quad W(y)$$
$$where \quad W(y) = inf_{x \geq 0}\{F(x) - y^T(Ax - b)\}$$
$$subject\ to \quad W(y) > -\infty$$

Figure 1: The classical geometric programming duality is studied by means of a primal–dual infeasible algorithm *applied* to the dual GP after logarithmic transformation of its objective function and its Lagrangian dual. Conversion to a primal GP solution begins with a Lagrangian dual solution, thereby solving both primal and dual GP problems simultaneously.